ISSN: 2321-872X-Online 2321-8932-Print
International Journal of Advanced Research in Computer Science and Applications
Vol.2, Issue 2, February 2014

Dynamic Pricing Scheme for Service Providers under Multi Server Cloud Environment

A. Rajesh[1] C. Senthilkumar[2]

[1]Assistant Professor, Computer Applications, Erode Sengunthar Engineering College, Perundurai, Erode Tamilnadu, India.
[2]Assistant Professor, Computer Applications, Erode Sengunthar Engineering College, Perundurai, Erode Tamilnadu, India.

Abstract

Cloud computing is used to share resources under data sources and computational applications. Hardware, software and information are provided in cloud environment. Cloud resources and services are provided in two methods federated and commercial cloud models. Mutual resource and service sharing is performed under the federated cloud model. Pricing schemes are used in commercial clouds. Commercial clouds are constructed with infrastructure vendors, service providers, and consumers. An infrastructure vendor maintains basic hardware and software facilities. A service provider rents resources from the infrastructure vendors, builds appropriate multiserver systems and provides various services to users. A consumer submits a service request to a service provider, receives the desired result from the service provider with certain service-level agreement.

Pricing model of a service provider in cloud computing is based on two components, income and cost. In a service provider the income is the service charge to users and the cost is the renting cost plus the utility cost paid to infrastructure vendors. Service charge and business cost factors are used to maximize the profit for a service provider. Multiserver configuration, Service Level Agreement (SLA), service and application load properties are used to assign service costs. Consumer satisfaction, Quality of Service (QoS) and penalty parameters are used to decide the service costs. Renting cost, energy cost and service provider margin are also used for the service cost estimation process. Multiserver system is treated as M/M/m queuing model. Server speed and power consumption strategy is divided into two models such as idle-speed model and the constant-speed model. The weighting time of a service request is derived using the probability density function.

The service pricing model is improved to manage on demand, reservation, peak demand and peek supply situations. Data usage cost and communication cost metrics are added to the service charge functions. Dynamic service function selection model is integrated with the system. Service request and access levels are analyzed to estimate the profit level of the service provider.

Keywords: DVFS, Virtual Batching, Request Batching, Server Consolidation CPU resource allocation.

1. Introduction

Cloud computing is a recent trend in IT that moves computing and data away from desktop and portable PCs into large data centers. It refers to applications delivered as services over the Internet as well as to the actual cloud infrastructure — namely, the hardware and systems software in data centers that provide these services.

The key driving forces behind cloud computing are the ubiquity of broadband and wireless networking, falling storage costs, and progressive improvements in Internet computing software. Cloud-service clients will be able to add more capacity at peak demand, reduce costs, experiment with new services, and remove unneeded capacity, whereas service providers will increase utilization via multiplexing, and

allow for larger investments in software and hardware.

Currently, the main technical under-pinnings of cloud computing infrastructures and services include virtualization, service-oriented software, grid computing technologies, management of large facilities, and power efficiency. Consumers purchase such services in the form of infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), or software-as-a-service (SaaS) and sell value-added services to users. Within the cloud, the laws of probability give service providers great leverage through statistical multiplexing of varying workloads and easier management — a single software installation can cover many users' needs.

2. Related Work

There have been a number of studies exploiting market-based resource allocation to tackle this problem. Noticeable scheduling mechanisms include FirstPrice, FirstProfit and proportional-share. Most of them are limited to job scheduling in conventional supercomputing settings. Specifically, they are only applicable to scheduling batch jobs in systems with a fixed number of resources. User applications that require the processing of mashup services, which is common in the cloud are not considered by these mechanisms [5]. The scenario addressed in this study is different in terms of application type and the organization of the cloud. We consider a three-tier cloud structure, which consists of infrastructure vendors, service providers and consumers, even though the distinctions between them can be blurred; the latter two parties are of particular interest in this study.

Cloud computing has attracted considerable research attention, but only a small portion of the work done so far has addressed performance issues by rigorous analytical models. A general analytic model based approach for an end-to-end performance analysis of a cloud service is proposed. However the proposed model is limited to the single arrival of requests and the start up delay of cold PMs has not been captured. Also, the effect of virtualization was not reflected in their results. The cloud center was modeled as an M/M/m/m + r queuing system in which inter-arrival and service times are exponentially distributed, and the system has a finite buffer. The response time was partitioned into waiting, service, and execution periods, assuming that all three periods are independent. Our earlier work presents monolithic analytical models which are quite restrictive compared to this work in terms of extendability simplicity and computational cost [3]. Also, that work does not address the concept of virtualization as well as heterogeneous server pools and PMs. Authors applied classical Erlang loss formula and M/M/m/K queuing system for response time and outbound bandwidth modeling respectively.

In April 2007, Gartner estimated that the Information and Communication Technologies (ICT) industry generates about 2% of the total global CO_2 emissions, which is equal to the aviation industry. As governments impose carbon emissions limits on the ICT industry like in the automobile industry, Cloud providers must reduce energy usage to meet the permissible restrictions. Thus, Cloud providers must ensure that data centers are utilized in a carbon-efficient manner to meet scaling demand. Otherwise, building more data centers without any carbon consideration is not viable since it is not environmentally sustainable and will ultimately violate the imposed carbon emissions limits [4]. This will in turn affect the future widespread adoption of Cloud computing, especially for the HPC community which demands scalable infrastructure to be delivered by Cloud providers. Companies like Alpiron already offer software for cost-efficient server management and promise to reduce energy cost by analyzing, via advanced algorithms, which server to shutdown or turn on during the runtime.

Motivated by this practice, this paper enhances the idea of cost-effective management by taking both the aspects of economic and environmental sustainability into account. In particular, we aim to examine how a Cloud provider can achieve optimal energy

sustainability of running HPC workloads across its entire Cloud infrastructure by harnessing the heterogeneity of multiple data centers geographically distributed in different locations worldwide.

3. Commercial Cloud Services

Cloud computing is quickly becoming an effective and efficient way of computing resources and computing services consolidation [10]. By centralized management of resources and services, cloud computing delivers hosted services over the Internet, such that accesses to shared hardware, software, databases, information, and all resources are provided to consumers on-demand. Cloud computing is able to provide the most cost-effective and energy-efficient way of computing resources management and computing services provision. Cloud computing turns information technology into ordinary commodities and utilities by using the pay-per-use pricing model. However, cloud computing will never be free [8] and understanding the economics of cloud computing becomes critically important.

One attractive cloud computing environment is a threetier structure [5], which consists of infrastructure vendors, service providers, and consumers. The three parties are also called cluster nodes, cluster managers, and consumers in cluster computing systems and resource providers, service providers, and clients in grid computing systems. An infrastructure vendor maintains basic hardware and software facilities. A service provider rents resources from the infrastructure vendors, builds appropriate multiserver systems, and provides various services to users. A consumer submits a service request to a service provider, receives the desired result from the service provider with certain service-level agreement, and pays for the service based on the amount of the service and the quality of the service [9]. A service provider can build different multiserver systems for different application domains, such that service requests of different nature are sent to different multiserver systems. Each multiserver system contains multiple servers, and such a multiserver system can be devoted to serve one type of service requests and applications. An application domain is characterized by two basic features, i.e., the workload of an application environment and the expected amount of a service. The configuration of a multiserver system is characterized by two basic features, i.e., the size of the multiserver system (the number of servers) and the speed of the multiserver system (execution speed of the servers).

Like all business, the pricing model of a service provider in cloud computing is based on two components, namely, the income and the cost. For a service provider, the income (i.e., the revenue) is the service charge to users, and the cost
is the renting cost plus the utility cost paid to infrastructure vendors. A pricing model in cloud computing includes many considerations, such as the amount of a service (the requirement of a service), the workload of an application environment, the configuration (the size and the speed) of a multiserver system, the service-level agreement, the satisfaction of a consumer (the expected service time), the quality of a service (the task waiting time and the task response time), the penalty of a low-quality service, the cost of renting, the cost of energy consumption, and a service provider's margin and profit. The profit (i.e., the net business gain) is the income minus the cost. To maximize the profit, a service provider should understand both service charges and business costs, and in particular, how they are determined by the characteristics of the applications and the configuration of a multiserver system.

The service charge to a service request is determined by two factors, i.e., the expected length of the service and the actual length of the service. The expected length of a service (i.e., the expected service time) is the execution time of an application on a standard server with a baseline or reference speed. Once the baseline speed is set, the expected length of a service is determined by a service request itself, i.e., the service requirement (amount of service)

measured by the number of instructions to be executed. The longer (shorter, respectively) the expected length of a service is, the more (less, respectively) the service charge is. The actual length of a service (i.e., the actual service time) is the actual execution time of an application. The actual length of a service depends on the size of a multiserver system, the speed of the servers (which may be faster or slower than the baseline speed), and the workload of the multiserver system. Notice that the actual service time is a random variable, which is determined by the task waiting time once a multiserver system is established.

There are many different service performance metrics in service-level agreements [2]. Our performance metric in this paper is the task response time (or the turn around time), i.e., the time taken to complete a task, which includes task waiting time and task execution time. The service-level agreement is the promised time to complete a service, which is a constant times the expected length of a service. If the actual length of a service is (or, a service request is completed) within the service-level agreement, the service will be fully charged. However, if the actual length of a service exceeds the service-level agreement, the service charge will be reduced. The longer (shorter, respectively) the actual length of a service is, the more (less, respectively) the reduction of the service charge is. In other words, there is penalty for a service provider to break a service-level agreement. If the actual service time exceeds certain limit (which is service request dependent), a service will be entirely free with no charge. Notice that the service charge of a service request is a random variable, and we are interested in its expectation.

The cost of a service provider includes two components, i.e., the renting cost and the utility cost. The renting cost is proportional to the size of a multiserver system, i.e., the number of servers. The utility cost is essentially the cost of energy consumption and is determined by both the size and the speed of a multiserver system. The faster (slower, respectively) the speed is, the more (less, respectively) the utility cost is. To calculate the cost of energy consumption, we need to establish certain server speed and power consumption models.

To increase the revenue of business, a service provider can construct and configure a multiserver system with many servers of high speed. Since the actual service time (i.e., the task response time) contains task waiting time and task execution time, more servers reduce the waiting time and faster servers reduce both waiting time and execution time. Hence, a powerful multiserver system reduces the penalty of breaking a service-level agreement and increases the revenue. However, more servers (i.e., a larger multiserver system) increase the cost of facility renting from the infrastructure vendors and the cost of base power consumption. Furthermore, faster servers increase the cost of energy consumption. Such increased cost may counterweight the gain from penalty reduction. Therefore, for an application environment with specific workload which includes the task arrival rate and the average task execution requirement, a service provider needs to decide an optimal multiserver configuration (i.e., the size and the speed of a multiserver system), such that the expected profit is maximized.

In this paper, we study the problem of optimal multiserver configuration for profit maximization in a cloud computing environment. Our approach is to treat a multiserver system as an M/M/m queuing model, such that our optimization problem can be formulated and solved analytically. We consider two server speed and power consumption models, namely, the idle-speed model and the constant-speed model. Our main contributions are as follows. We derive the probability density function (pdf) of the waiting time of a newly arrived service request. This result is significant in its own right and is the base of our discussion. We calculate the expected service charge to a service request. Based on these results, we get the expected net business gain in one unit of time, and obtain the optimal server size and the optimal server speed numerically. To the best of our knowledge, there

has been no similar investigation in the literature, although the method of optimal multicore server processor configuration has been employed for other purposes, such as managing the power and performance tradeoff [7].

One related research is user-centric and market-based and utility-driven resource management and task scheduling, which have been considered for cluster computing systems and grid computing systems. To compete and bid for shared computing resources through the use of economic mechanisms such as auctions, a user can specify the value (utility, yield) of a task, i.e., the reward (price, profit) of completing the task. A utility function, which measures the value and importance of a task as well as a user's tolerance to delay and sensitivity to quality of service, supports market-based bidding, negotiation, and admission control. By taking an economic approach to providing service-oriented and utility computing, a service provider allocates resources and schedules tasks in such a way that the total profit earned is maximized. Instead of traditional system-centric performance optimization such as minimizing the average task response time, the main concern in such computational economy is user-centric performance optimization, i.e., maximizing the total utility delivered to the users (i.e., the total user-perceived value).

4. Problem Statement

Pricing model of a service provider in cloud computing is based on two components, income and cost. In a service provider the income is the service charge to users and the cost is the renting cost plus the utility cost paid to infrastructure vendors. Service charge and business cost factors are used to maximize the profit for a service provider. Multiserver configuration, Service Level Agreement (SLA), service and application load properties are used to assign service costs. Consumer satisfaction, Quality of Service (QoS) and penalty parameters are used to decide the service costs. Renting cost, energy cost and service provider margin are also used for the service cost estimation process. Multiserver system is treated as M/M/m queuing model. Server speed and power consumption strategy is divided into two models such as idle-speed model and the constant-speed model. The weighting time of a service request is derived using the probability density function. The following drawbacks are identified in the existing system.

- Data access information are not used in the cost functions
- Static pricing model
- Service charging function selection is not provided
- Profit level prediction is not provided

5. Multi Server Model for Cloud Services

Throughout the paper, we use P[e] to denote the probability of an event e. For a random variable x, we use f_x (t) to represent the probability density function of x, and f_x (t) to represent the cumulative distribution function (cdf) of x, and x to represent the expectation of x.

A cloud computing service provider serves users' service requests by using a multiserver system, which is constructed and maintained by an infrastructure vendor and rented by the service provider. The architecture detail of the multiserver system can be quite flexible. Examples are blade servers and blade centers where each server is a server blade [6], clusters of traditional servers where each server is an ordinary processor and multicore server processors where each server is a single core [7]. We will simply call these blades/processors/cores as servers. Users (i.e., customers of a service provider) submit service requests (i.e., applications and tasks) to a service provider, and the service provider serves the requests (i.e., run the applications and perform the tasks) on a multiserver system.

Assume that a multiserver system S has m identical servers. In this paper, a multiserver system is treated as an M/M/m queuing system which is elaborated as follows. There is a Poisson stream of service requests with arrival rate λ, i.e., the interarrival times are independent and identically distributed (i.i.d.) exponential

random variables with mean 1= λ. A multiserver system S maintains a queue with infinite capacity for waiting tasks when all the m servers are busy. The first-come-first-served (FCFS) queuing discipline is adopted. The task execution requirements (measured by the number of instructions to be executed) are i.i.d. exponential random variables r with mean r. The m servers (i.e., blades/processors/cores) of S have identical execution speed s (measured by the number of instructions that can be executed in one unit of time). Hence, the task execution times on the servers of S are i.i.d. exponential random variables x = r=/s with mean x = r/s.

6. Power Consumption Models

Power dissipation and circuit delay in digital CMOS circuits can be accurately modeled by simple equations, even for complex microprocessor circuits. CMOS circuits have dynamic, static, and short-circuit power dissipation; however, the dominant component in a well-designed circuit is dynamic power consumption P (i.e., the switching component of power), which is approximately $P = aCV^2f$, where a is an activity factor, C is the loading capacitance, V is the supply voltage, and f is the clock frequency. In the ideal case, the supply voltage and the clock frequency are related in such a way that $V \alpha f^{\phi}$ for some constant $\phi > 0$. The processor execution speed s is usually linearly proportional to the clock frequency, namely, $s \alpha f$. For ease of discussion, we will assume that $V = bf^{\phi}$ and s = cf, where b and c are some constants. Hence, we know that power consumption is $P = aCV^2f = ab^2Cf^{2\phi+1} = (ab^2C/c^{2\phi+1})s^{2\phi+1} = \xi s^{\alpha}$, where $\xi = ab^2C/c^{2\phi+1}$ and $\alpha = 2\phi+1$. For instance, by setting b = 1:16, aC = 7:0, c = 1:0, ϕ = 0:5, $\alpha = 2\phi+1 = 2:0$, and $\xi = ab^2C/c^{\alpha} = 9:4192$, the value of P calculated by the equation $P = aCV^2f = \xi s^{\alpha}$ is reasonably close to the Intel Pentium M processor.

We will consider two types of server speed and power consumption models. In the idle-speed model, a server runs at zero speed when there is no task to perform. Since the power for speed s is ξs^{α}, the average amount of energy consumed by a server in one unit of time is $p\xi s^{\alpha} = \lambda/m\ r\xi s^{\alpha-1}$, where we notice that the speed of a server is zero when it is idle. The average amount of energy consumed by an m-server system S in one unit of time, i.e., the power supply to the multiserver system S, is $P = mp\xi s^{\alpha} = \lambda r\xi s^{\alpha-1}$, where $mp = \lambda x$ is the average number of busy servers in S. Since a server still consumes some amount of power P* even when it is idle (assume that an idle server consumes certain base power P*, which includes static power dissipation, short-circuit power dissipation, and other leakage and wasted power [1]), we will include P* in P, i.e., $P = p\xi s^{\alpha} + P^*) = \lambda\ r\xi s^{\alpha-1} + mP^*$. Notice that when P* = 0, the above P is independent of m.

In the constant-speed model, all servers run at the speed s even if there is no task to perform. Again, we use P to represent the power allocated to multiserver system S. Since the power for speed s is ξs^{α}, the power allocated to multiserver system S is $P = m(\xi s^{\alpha} + P^*)$.

7. Dynamic Pricing Scheme for Commercial Clouds

The cloud services are provided in federated model and commercial model. Mutual sharing is carried out under the federated model. Cloud services are charged under the commercial cloud environment. Service provider provides the cloud services to the users. Infrastructure vendors provide the infrastructure for the cloud service providers. Storage and computational infrastructures are accessed from the infrastructure vendors. Service cost is collected from the users. Different pricing functions are used to decide cost for the cloud services. Application and service load, renting cost, energy cost and Quality of Service (QoS) factors are used in the cloud service cost estimation process. Data access cost and bandwidth usage levels are also considered in the cost estimation process. The system is designed with the following objectives.

- To handle cloud service under commercial service provider environment

- To manage service provider, infrastructure vendors and user transactions for service provisioning process.
- To improve the profit level for the service providers
- To support multi pricing scheme for service providers
- To provide dynamic service charge function selection scheme
- To include data and storage cost for services
- To assist the user for service discovery under the cloud environment

The service pricing model is improved to manage on demand, reservation, peak demand and peek supply situations. Data usage cost and communication cost metrics are added to the service charge functions. Dynamic service function selection model is integrated with the system. Service request and access levels are analyzed to estimate the profit level of the service provider.

The commercial cloud service provisioning scheme is improved with dynamic pricing models. Charging function selection mechanism is used in the system. Profit prediction and analysis mechanism is integrated with the system. The system is divided into five major modules. They are infrastructure vendor, service provider, cloud consumer, pricing process and service usage analysis.

Cloud resources are provided under infrastructure vendor module. Service provider provides services for the consumers. Cloud service requests are submitted by the cloud consumers. Pricing process module is used to calculate resource prices. Service usage analysis module is designed to estimate the profit levels.

8. Conclusion

Cloud service providers provide services to the consumers based on demand model. Different charging parameters are used to estimate the service cost for a consumer. Supply / demand based pricing model is used to increase the profit level of the service providers. The system also supports dynamic service charge function insertion mechanism. The system uses optimal server size and optimal server speed. Cost and energy efficient system. The system achieves high profit level under the service provider. Supply demand based pricing mechanism increases the service provider income.

REFERENCES

[1] http://en.wikipedia.org/wiki/CMOS, 2012.
[2]http://en.wikipedia.org/wiki/Service_level_agreement, 2012.
[3] Hamzeh Khazaei, Jelena Misic, Vojislav B. Misic and Nasim Beigi-Mohammadi, "Availability Analysis of Cloud Computing Centers", 2012.
[4] Saurabh Kumar Garg, Chee Shin Yeo, Arun Anandasivam and Rajkumar Buyya, "Environment-conscious scheduling of HPC applications on distributed Cloud-oriented data centers", Elsevier, 2010.
[5] Y.C. Lee, C. Wang, A.Y. Zomaya, and B.B. Zhou, "Profit-Driven Service Request Scheduling in Clouds," Proc. 10th IEEE/ACM Int'l Conf. Cluster, Cloud and Grid Computing, pp. 15-24, 2010.
[6] K. Li, "Optimal Load Distribution for Multiple Heterogeneous Blade Servers in a Cloud Computing Environment," Proc. 25th IEEE Int'l Parallel and Distributed Processing Symp. Workshops, pp. 943-952, May 2011.
[7] K. Li, "Optimal Configuration of a Multicore Server Processor for Managing the Power and Performance Tradeoff," J. Supercomputing, vol. 61, no. 1, pp. 189-214, 2012.
[8] D. Durkee, "Why Cloud Computing Will Never be Free," Comm. ACM, vol. 53, no. 5, pp. 62-69, 2010.
[9] Junwei Cao, Kai Hwang, Keqin Li, and Albert Y. Zomaya, "Optimal Multiserver Configuration for Profit Maximization in Cloud Computing" IEEE Transactions On Parallel And Distributed Systems, Vol. 24, No. 6, June 2013
[10] K. Hwang, G.C. Fox, and J.J. Dongarra, Distributed and Cloud Computing. Morgan Kaufmann, 2012.