ISSN:0975-9646

Dynamic Resource Allocation to Maximize the Profits in Cloud Environment

[1]K.R Dilip
Department of Computer Science and Engineering,
JNTUA, Anantapur/ Annamacharya Institute of Technology and Sciences,Tirupathi, Chittor,
Andhra Pradesh, India

[2]J Bala Murali Krishna
Assistant Professor, Department of Computer Science Engineering
JNTUA, Anantapur/ Annamacharya Institute of Technology and Sciences,Tirupathi, Chittor,
Andhra Pradesh, India

Abstract: - service provider need to understand the service charges and business cost of application & configuration setting of multi-server in cloud. Analyze the character of amount of server usage, workload for application, configuration of multi-server, service level agreements, satisfaction of customer, quality of service, penalty of a low quality of server, cost of renting, cost of energy consumption & service provider margin and profit. Our approach is to treat a multi-server system as an M/M/m queuing model, we are optimizing the resource usage shared to multiple users' based on some formulas we are calculating the speed, power consumption, application running, idle speed and constant speed of server to obtain unit time performance..

Index Terms—Cloud computing, multi-server system, pricing model, profit, queuing model, response time, server configuration, service charge, service-level agreement, waiting time

1. INTRODUCTION

Cloud computing characteristics include a ubiquitous (network-based) access channel; resource pooling; multi tenancy automatic and elastic provisioning and release of computing capabilities; and metering of resource usage (typically on a pay-per-use basis).

Virtualization of resources such as processors, network, memory, and storage ensures scalability and high availability of computing capabilities. Clouds can dynamically provision these virtual resources to hosted applications or to clients that use them to develop their own applications or to store data. Rapid provisioning and dynamic reconfiguration of resources help cope with variable demand and ensure optimum resource utilization.

Cloud computing is quickly becoming an effective and efficient way of computing resources and computing services consolidation. 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 consumer's 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 turn's information technology into ordinary commodities and utilities by using the pay-per-use pricing model. However, cloud computing will never be free, and understanding the economics of cloud computing becomes critically important.

Cloud computing environment is a three-tier structure, which consists of 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 multi-server 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. A service provider can build different multi-server systems for different application domains, such that service requests of different nature are sent to different multi-server systems. Each multi-server system contains multiple servers, and such a multi-server 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 multi-server system is characterized by two basic features, i.e., the size of the multi-server system (the number of servers) and the speed of the multi-server 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 multi-server 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.

The service charge to a service request is determined by two factors, 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. 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 multi-server system, the speed of the servers (which may be faster or slower than the baseline speed), and the workload of the multi-server system.

Notice that the actual service time is a random variable, which is determined by the task waiting time once a multi-server system is established. There are many different service performance metrics in service-level agreements. Our performance metric in this paper is the task response time (or the turnaround 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. The cost of a service provider includes two components the renting cost and the utility cost. The renting cost is proportional to the size of a multi-server 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 multi-server 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 multi-server 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 multi-server system reduces the penalty of breaking a service-level agreement and increases the revenue. However, more servers (i.e., a larger multi-server 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 multi-server configuration (i.e., the size and the speed of a multi-server system), such that the expected profit is maximized.

In this paper, the problem of optimal multi-server configuration for profit maximization in a cloud computing environment is presented. Our approach is to treat a multi-server 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. The probability density function (pdf) of the waiting time of a newly arrived service request is derived.

This result is significant in its own right and is the base of our discussion. The expected service charge to a service request is calculated. 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. Here, P[e] is used 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 $\bar{x}$ to represent the expectation of x. A cloud computing service provider serves users' service requests by using a multi server system, which is constructed and maintained by an infrastructure vendor and rented by the service provider.

The architecture detail of the multi-server system can be quite flexible. Examples are blade servers and blade centers where each server is a server blade, clusters of traditional servers where each server is an ordinary processor, and multi-core server processors where each server is a single core. 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 server system. 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.

2. RELATED WORK

Cloud service differs from traditional hosting in three principal aspects. First, it is provided on demand; second, it is elastic since users that use the service have as much or as little as they want at any given time (typically by the minute or the hour); and third, the service is fully managed by the provider. Due to dynamic nature of cloud environments, diversity of user requests, and time dependency of load, providing agreed quality of service (QoS) while avoiding over provisioning is a difficult task. Since many of the large cloud centers employ virtualization to provide the required resources such as PMs, we consider PMs with a high degree of virtualization. Real cloud

providers offer complex requests for their users. For instance, in Amazon EC2, the user is allowed to run up to On-Demand or Reserved Instances, and up to 100 Spot Instances per region. We examined the effects of various parameters including ST arrival rate, task service time, the virtualization degree, and ST size on task rejection probability and total response delay. The stable, transient and unstable regimes of operation for given configurations have been identified so that capacity planning is going to be a less challenging task for cloud providers.

The cluster RMS supports four main functionalities: resource management; job queuing; job scheduling; and job execution. It manages and maintains status information of the resources such as processors and disk storage in the cluster system. Jobs submitted into the cluster system are initially placed into queues until there are available resources to execute the jobs. The cluster RMS then invokes a scheduler to determine how resources are assigned to jobs. After that, the cluster RMS dispatches the jobs to the assigned nodes and manages the job execution processes before returning the results to the users upon job completion.

Performance evaluation focuses on a producer by using a user-centric cost evaluation factor to determine the average yield or earning each cluster manager achieves. Simulation results show that considering and balancing the potential gain of accepting a task instantly with the risk of future loss provides better returns for competing cluster managers. To allow service providers to decide whether to accept work, a resource provider will offer predictions of how many resources it will have available when, using a set of tuples. The new algorithms are compared with previously proposed approaches, and evaluated across a range of operating conditions: load, resource price and quantity, utility function shape (client impatience), and resource uncertainty level. Our experiments show that the new algorithms can extract higher pro rates than the previous ones. In the future, we would like to experiment with nontechnical aspects of the economics-based approach, such as determining how to help customers express utility functions easily. We would also like to explore alternative ways to describe and manage risk. Resources to run the jobs are obtained from a separate resource provider, who offers predictions of the number of resources likely to be available at different times in the future, and their price.

3. PROPOSED SYSTEM

Multi server model:

A cloud computing service provider serves users' service requests by using a multi-server system, which is constructed and maintained by an infrastructure vendor and rented by the service provider. The architecture detail of the multi server system can be quite flexible. Examples are blade servers and blade centers where each server is a server blade, clusters of traditional servers where each server is an ordinary processor, and multi-core server processors where each server is a single core 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 multi-server system.

Here multi-server 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 inter-arrival times are independent and identically distributed exponential random variables with mean 1/ λ. Notice that although an M/G/m queuing system has been considered, the M/M/m queuing model is the only model that accommodates an analytical and closed form expression of the probability density function of the waiting time of a newly arrived service request. Let μ = 1 / $\bar{x}$ = s / $\bar{r}$ be the average service rate, i.e., the average number of service requests that can be finished by a server of S in one unit of time. The server utilization is ρ = λ/ mμ , which is the average percentage of time that a server of S is busy. Let P_k denote the probability that there are k service requests (waiting or being processed) in the M/M/m queuing system for S.

$$P_k = \begin{cases} P_0 \frac{(m\rho)^k}{k!}, & k \le m; \\ P_0 \frac{m^m \rho^k}{m!}, & k \ge m \end{cases}$$

Where,

$$P_0 = \left(\sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!} + \frac{(m\rho)^m}{m!} \; X \; \frac{1}{1-\rho} \right)^1$$

The probability of queuing (i.e., the probability that a newly submitted service request must wait because all servers are busy) is

$$P_q = \sum_{k=m}^{\infty} P_k = \frac{p_m}{1-\rho} = p_0 \frac{(m\rho)^m}{m!} \; X \; \frac{1}{1-\rho}$$

The average number of service requests (in waiting or in execution) in S is

$$\bar{N} = \sum_{k=0}^{\infty} k p_{k=m\rho + \frac{\rho}{1-\rho} P_q}$$

Applying little's result , we get the average task response time as,

$$\bar{N} = \sum_{k=0}^{\infty} k p_{k=m\rho + \frac{\rho}{1-\rho} P_q}$$

The average waiting time of a service request is

$$\bar{W} = \bar{T} - \bar{x} = \frac{p_m}{m(1-\rho)^2} \bar{x}$$

the waiting time is the source of customer dissatisfaction. A service provider should keep the waiting time to a low level by providing enough servers and/or increasing server speed, and be willing to pay back to a customer in case the waiting time exceeds certain limit.

Power consumption model:

Power dissipation and circuit delay in digital CMOS circuits can be accurately modelled 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 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 the average amount of energy consumed by a server in one unit of time, the speed of a server is zero when it is idle. 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 multi-server system S.

If all the servers have a fixed speed s, the execution time of a service request with execution requirement r is known as x = r/s. The response time to the service request is T = W + x = W + r/s. The response time T is related to the service charge to a customer of a service provider in cloud computing. To study the expected service charge to a customer, we need a complete specification of a service charge based on the amount of a service, the service-level agreement, the satisfaction of a consumer, the quality of a service, the penalty of a low-quality service, and a service provider's margin and profit.

Net business gain:

Since the number of service requests processed in one unit of time is λ in a stable M/M/m queuing system, the expected service charge in one unit of time is λC, which is actually the expected revenue of a service provider. Assume that the rental cost of one server for unit of time is β. Also, assume that the cost of energy is γ per Watt. The cost of a service provider is the sum of the cost of infrastructure renting and the cost of energy consumption, i.e., βm+γP. Then, the expected net business gain (i.e., the net profit) of a service provider in one unit of time is G= λC – (βm + γP), which is defined as the revenue minus the cost.

The more service requests bring more revenue and net business gain; however, after the number of service requests per unit of time reaches certain point, the excessive waiting time causes increased lateness penalty, so that there is no revenue and negative business gain. There are two situations that cause negative business gain. In the first case, there is no enough business (i.e., service requests). In this case, a service provider should consider reducing the number of servers m and/or server speed s, so that the cost of infrastructure renting and the cost of energy consumption can be reduced. In the second case, there is too much business (i.e., service requests). In this case, a service provider should consider increasing the number of servers and/or server speed, so that the waiting time can be reduced and the revenue can be increased. However, increasing the number of servers and/or server speed also increases the cost of infrastructure renting and the cost of energy consumption. Therefore, we have the problem of selecting the optimal server size and/or server speed so that the profit is maximized.

4. PROFIT MAXIMIZATION

A closed -form expression of C is used to formulate and solve our optimization problems analytically. The closed form approximation $\frac{(m\rho)^k}{k!}$ $\sum_{k=0}^{m-1} \approx e^{m\rho}$ which is accurate when m is not too small and ρ is not too large. We also use Stirling's approximation of m! i.e. $m! \approx \sqrt{2\pi m}\left(\frac{m}{e}\right)^m$ the following closed form approximation of p_m

$$p_m \approx \frac{1-\rho}{\sqrt{2\pi m}\ (1-\rho)(e^{\rho}/e\rho)m+1}$$

And the closed form approximation of p_q

$$p_q \approx \frac{1-\rho}{\sqrt{2\pi m}\ (1-\rho)(e^{\rho}/e\rho)m+1}$$

The expected service charge to a service request of closed form approximatuon is

$$C = a\bar{r}\left(1 - \frac{\bar{1}}{D_1 D_2 D_3}\right)$$

$$D_1 = \sqrt{2\pi m}(1-\rho)\left(e^{\rho}/e\rho\right) + 1$$
$$D_2 = (ms - \lambda\bar{r})(c/s_0 - 1/s) + 1$$
$$D_3 = (ms - \lambda\bar{r})(a/d + c/s_o - 1/s) + 1$$

We find m to maximize G , we use the optimum size Such server size optimization has clear physical interpretation. When m is small such that _ is close to 1, the waiting times of service requests are excessively long, and the service charges and the net business gain are low. As m increases, the waiting times are significantly reduced, and the service charges and the net business gain are increased. However, as m further increases, there will be no more increase in the expected services charge which has an upper bound $a\bar{r}$; on the other hand, the cost of a service provider (i.e., the rental cost and base power consumption) increases, so that the net business gain is actually reduced.

5. CONCLUSION

We have proposed a pricing model for cloud computing which takes many factors into considerations, such as the requirement r of a service, the workload of an application environment, the configuration (m and s) of a multi-server system, the service level agreement c, the satisfaction (r and s) of a consumer, the quality (W and T) of a service, the penalty d of a low-quality service, the cost of renting, the cost of energy consumption, and a service provider's margin and profit a. By using an M/M/ m queuing model, we formulated and solved the problem of optimal multi-server configuration for profit maximization in a cloud computing environment. Our discussion can be easily extended to other service charge functions. Our methodology can be applied to other pricing models. At three-tier cloud structure, which consists of infrastructure vendors, service providers and consumers, the latter two

parties are particular interest to us. Clearly, scheduling strategies in this scenario should satisfy the objectives of both parties. Our contributions include the development of a pricing model using processor-sharing for clouds, the application of this pricing model to composite services with dependency consideration, and the development of two sets of profit-driven scheduling algorithms.

REFERENCES

[1] http://en.wikipedia.org/wiki/CMOS, 2012.
[2] http://en.wikipedia.org/wiki/Service_level_agreement, 2012.
[3] M. Armbrust et al., "Above the Clouds: A Berkeley View of Cloud Computing," Technical Report No. UCB/EECS-2009-28, Feb. 2009.
[4] R. Buyya, D. Abramson, J. Giddy, and H. Stockinger, "Economic Models for Resource Management and Scheduling in Grid Computing," Concurrency and Computation: Practice and Experience, vol. 14, pp. 1507-1542, 2007.
[5] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the Fifth Utility," Future Generation Computer Systems, vol. 25, no. 6, pp. 599-616, 2009.
[6] A.P. Chandrakasan, S. Sheng, and R.W. Brodersen, "Low-Power CMOS Digital Design," IEEE J. Solid-State Circuits, vol. 27, no. 4, pp. 473-484, Apr. 1992.
[7] B.N. Chun and D.E. Culler, "User-Centric Performance Analysis of Market-Based Cluster Batch Schedulers," Proc. Second IEEE/ ACM Int'l Symp. Cluster Computing and the Grid, 2002.
[8] D. Durkee, "Why Cloud Computing Will Never be Free," Comm. ACM, vol. 53, no. 5, pp. 62-69, 2010.
[9] R. Ghosh, K.S. Trivedi, V.K. Naik, and D.S. Kim, "End-to-End Performability Analysis for Infrastructure-as-a-Service Cloud: An Interacting Stochastic Models Approach," Proc. 16th IEEE Pacific Rim Int'l Symp. Dependable Computing, pp. 125-132, 2010.
[10] K. Hwang, G.C. Fox, and J.J. Dongarra, Distributed and Cloud Computing. Morgan Kaufmann, 2012.

K.R Dilip received the B.Tech Degree in Computer Science and Engineering from Chadalawada Ramanamma Engineering College, University of JNTUA in 2009. He is currently working towards the Master's Degree in Computer Science, in Annamacharya Institute of Technology and Sciences University of JNTUA. Her interest lies in the areas of Cloud Computing, Database, Web Technologies and networks.

J. Bala Murali Krishna Received B.E and M.E Degrees in Computer Science and Engineering from VMKV Engineering College, Salem, University of Madras and Vinayaga Mission University, Salem in 1994&2007 respectively. Currently He is a Assistant Professor in the Department of Computer Science and Engineering at AITS-Tirupati.
He has published a paper titled **"**Hierarchical attribute-based Access Control and scalable user revocation data sharing in cloud server**".** In Journals and refereed Conference Proceedings. His area of interest is Cloud Computing Computer Networks, Web Technologies and Databases, Data Mining.