ISSN: 2321-7782 (Online)
Volume 2, Issue 5, May 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

A Study of Profit Maximization Based On Multiserver in Cloud Computing

A. Sahaya Princy[1]
II M.E(CSE)
Shivani Engineering College
Trichy – India

Dr. V. Sampath Kumar[2]
Head of the Department
Dept. of computer science & engg
Shivani Engineering College
Trichy – India

Abstract: ***Economics of cloud becomes an effective and efficient way of computing resources. Increasing the profit of service provider includes service charge and business cost. Amount of service, workload, service level agreement, quality of service, energy consumption are considered for the profit maximization. Server speed and power consumption of two systems consider the idle speed model and constant speed model. Probability density function of the waiting time of the newly arrived service is described. Service charge and gain for the expected service request is calculated. A consumer submits a service request to a service provider. Consumer receives the result from the service provider with certain SLA and pays for the service based on the amount of the service and the quality of the service. Service provider allocates resources and schedules tasks in such a way that the total profit earned is maximized. The best possible solution is allocating limited resources to achieve maximum profit. It is applicable only where all relationships are linear, and can accommodate only a limited class of cost functions. The profit maximization calculation is done using biased sampling and linear problem solving technique.***

Keywords: Cloud computing, collective server, revenue escalation, service charge, net gain, scheduling.

I. INTRODUCTION

Cloud computing is quickly becoming an effective and efficient way of computing resources. By centralized management of resources and services, cloud computing delivers hosted services over the Internet. Cloud computing is able to provide the most cost-effective and energy-efficient way of computing resources management. Cloud computing turn's information technology into ordinary commodities and utilities by using the pay-per-use pricing model. 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. Then 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.

Owing to redundancy of computer system networks and storage system cloud may not be reliable for data, the security score is concerned. In cloud computing security is tremendously improved because of a superior technology security system, which is now easily available and affordable. Applications no longer run on the desktop Personal Computer but run in the cloud. This means that the PC does not need the processing power or hard disk space as demanded by traditional desktop software. Powerful servers and the like are no longer required. The computing power of the cloud can be used to replace or supplement internal computing resources. Organizations no longer have to purchase computing resources to handle the capacity peaks.

Peaks are easily handled by the cloud. The Payment for most cloud computing services is based on a pay as you go model. This means that customers only pay for what they use. Distributed systems are groups of networked computers, which have the same goal for their work. In parallel computing, all processors may have access to a shared memory to exchange information between processors. In distributed computing, each processor has its own private memory which is the distributed memory. Information is exchanged by passing messages between the processors. A process knows its own state and it knows what state other processes were in progress. Distributed computing also refers to the use of distributed systems to solve computational problems. In distributed computing a problem is divided into many tasks. Each of which is solved by one or more computers which communicate with each other by message passing method.

Parallel and Distributed computing techniques have proved to be effective in tackling the problem with high computational complexity in a wide range of domains. Distributed computing is the process of aggregating the power of several computing entities which are logically distributed and may even be geologically distributed to collaboratively run a single computational task in a transparent and coherent way. Cloud computing is emerging at the convergence of three major trends of service orientation, virtualization and standardization of computing through Internet. Most cloud computing infrastructures consist of services delivered through shared data centers and appear as a single point of access for consumers' computing needs.

II. Related Work

A. *Multi server configuration for profit maximization*

The service charge and the business costs are determined in the optimal multi server configuration. The pricing model is taken into consideration which depends on certain factors such as amount of service, configuration of the multi server system, the service level agreement, the quality of service, workload environment and the satisfaction of the customer. Two server speed models and power consumption models are examined to prove the profit maximization. Server size is the major factor which enhances the server speed. The cost of a service provider includes two components namely the renting cost and the utility cost. Penalty can be applied to the low quality service where the profit earned can be maximized in the user centric systems. Two server speed and power consumption models namely the idle speed model and the constant speed model are considered. Power consumption at idle and constant speed model is calculated.

The waiting time of the request in the queue for getting response is then calculated. And the service charge to a customer based on the amount of a service, SLA, satisfaction, quality of service, penalty of a low quality service and server speed are calculated. The actual length of a service exceeds the service level agreement, and then service charge will be reduced. The longer the actual length of a service is, more the reduction of the service charge. The task response time or the turnaround time is 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 time for executing the expected length of a service. The power consumption and waiting time only are considered for the optimal server configuration. The accuracy of the server optimization is low. Server size and server speed is not taken into consideration.

B. *Load distribution for multiple heterogeneous blade servers*

Optimal load distribution in a heterogeneous distributed computer system with both generic and dedicated applications in the problem is formulated as a multivariable optimization problem based on a queuing model. The development of algorithms is to find the numerical solution of an optimal load distribution and the minimum average response time of generic tasks. The two different situations of special tasks namely special tasks with and without higher priority are determined. The server sizes, server speeds, task execution requirement and the arrival rates of special tasks all have significant impact on the average response time of generic tasks especially when the total arrival rate of generic tasks is large.

The server size heterogeneity and the server speed heterogeneity do not have much impact on the average response time of generic tasks. The advantage of blade servers comes not only from the consolidation benefits of housing several servers in a

single chassis, but also from the consolidation of associated resources like storage and networking equipment into a smaller architecture that can be managed through a single interface. A server blade is a thin, modular electronic circuit board containing one, two or more microprocessors and memory. Optimal load distribution of generic tasks together with special tasks must be studied before for cluster and grid computing environments, where each server is modeled as a queuing system with a single server. The power and performance optimization is important for a cloud computing provider to efficiently utilize all the available resources.

Optimal load distribution of generic tasks without special tasks for a group of heterogeneous multi server queuing systems is analyzed, where each server is modeled as a queuing system with a single server. Multiple heterogeneous multi core server processors across clouds and data centers, aggregate the performance of the cloud. The clouds can be optimized by load distribution and balancing. Energy efficiency is one of the most important issues for large scale server systems in current and future data centers. The multi core processor technology needs to provide new levels of performance and energy efficiency.

C. Profit-driven service request scheduling

Scheduling takes into account not only the profit achievable from the current service, but also the profit from other services being processed on the same service instance. Specifically, the service is assigned only if the assignment of that service onto the service instance yields some additional profit. Each service is associated with a decay rate and therefore the assignment of a service to a service instance on which some other services of the same kind are being processed may result in a certain degree of profit loss. Three tier cloud architecture is utilized in this form of scheduling.

Cloud service scheduling is categorized at user level and system level. At user level the scheduling deals with problems raised by service provision between providers and customers. The system level scheduling handles resource management within datacenter. Datacenter consists of many physical machines. Millions of tasks from users are received. Assignment of these tasks to physical machine is done at datacenter. This assignment or scheduling significantly impacts the performance of datacenter. In addition to system utilization, other requirements like QoS, SLA, resource sharing, fault tolerance, reliability, real time satisfaction etc should be taken into consideration.

A scheduling event takes place either at the time that a new service request arrives or at the time that a scheduled service completes its processing and its successor service is ready to start by the completion. It maintains a service queue containing to be dispatched services according to the precedence constraints. When a new application arrives, its entry service is the only one to be processed and checks all service instances. Two sets of profit driven service request scheduling algorithms are devised incorporating a pricing model using PS and two allowable delay metrics. It is identified that those two allowable delay metrics enable effective exploitation of characteristics of precedence constrained applications.

D. Taxonomy of market-based resource management

The market based resource management has two major categories such as the computing platforms and the taxonomies. They possess the cluster, distributed databases, grids, parallel and distributed systems, peer to peer and world wide web along with the various models of taxonomy. The market based cluster consists of three aspects such as consumers, cluster manager and cluster nodes which performs the management and interfacing functions within the worker to the manager.

Neptune is the resource director of a policy driven fabric management system that dynamically reconfigures the resources in a computing utility cluster. It implements an online control mechanism subject to policy based performance and resource configuration objectives. Then it reassigns servers and bandwidth among a set of service domains based on the predefined policy in response to the workload changes.

The Neptune builds and executes a reconfiguration plan through a planning framework and breaks the reconfiguration objectives into individual tasks delegated to set of lower level resource managers. The multi server domain computing utility is

adopted. Consumers have different requirements and needs for various job, thus can assign value or utility to their job requests. During job submission to the cluster RMS, consumers can specify their requirements and preferences for each respective job using Quality of Service parameters.

The cluster RMS then considers these QoS parameters when making resource allocation decisions. This provides a user centric approach with better user personalization since consumers can potentially affect the resource allocation outcomes, based on their assigned utility. Thus, the objective of the cluster RMS is to maximize overall consumers' utility satisfaction.

E. Performance analysis using queuing systems

The moments of task request arrivals are selected as Markov points. Two successive task request arrivals and task departures occur between them. The number of departures may be anywhere between 0 and 1, but it is likely to be low in fact, when the system is in the steady state, there will be on the average a single departure between every two successive arrivals. The embedded Markov chain is homogeneous and ergodic, it has a steady-state solution. Therefore the distribution of number of tasks in the system as well as the mean response time is calculated.

All servers are busy throughout the inter arrival time. Considering the number of tasks that depart from the system between the two Markov points. While the number may be anywhere between 0 and 1, it is likely to be close to 1. Then even in cases where several such departures occur between successive arrivals, it is unlikely that there will be more than one departure from any given server. In order to minimize the error while keeping the model tractable, assume that there are no more than three task departures between two successive task arrivals.

The queuing system indicates inter arrival time of requests which is exponentially distributed. The task service times are independent and identically distributed random variable that follows a general distribution with mean value. Approximations are very sensitive to the probability distribution of the task service time. It becomes increasingly accurate when the coefficient of variation of the service time increases. Approximation errors are particularly pronounced when the traffic intensity is small and when both the number of servers has the large service time. Two successive task request arrivals and task departures occur when the system is in steady state.

III. CONCLUSION

Two types of power consumption models such as idle speed and constant speed model determines the power consumption range. In idle speed model speed of the server is zero at that time there is no task to perform in the server. In constant speed model all the servers run at certain speed limit. Waiting time of the request is reduced by fair scheduling and the average for waiting time of service request is calculated. The penalty is based on the performance of the speed of the server and the expected charge of the service request is calculated. The net business gain is the result of difference with the cost of the server to revenue of the server. When the server size increases, the waiting time will become decreased also the service charge and the gain are increased.

References

1. H. Khazaei, J. Misic, and V.B. Misic, "Performance Analysis of Cloud Computing Centers Using M/G/m/m+r Queuing Systems," IEEE Trans. Parallel and Distributed Systems, vol. 23, no. 5, pp. 936-943, May 2012.
2. 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.
3. 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.
4. 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.
5. F.I. Popovici and J. Wilkes, "Profitable Services in an Uncertain World," Proc. ACM/IEEE Conf. Supercomputing, 2005.
6. J. Sherwani, N. Ali, N. Lotia, Z. Hayat, and R. Buyya, "Libra: A Computational Economy-Based Job Scheduling System for Clusters," Software - Practice and Experience, vol. 34, pp. 573-590, 2004.

7. C.S. Yeo and R. Buyya, "A Taxonomy of Market-Based Resource Management Systems for Utility-Driven Cluster Computing," Software - Practice and Experience, vol. 36, pp. 1381-1419, 2006.
8. B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "Theoretical and Practical Limits of Dynamic Voltage Scaling," Proc. 41st Design Automation Conf., pp. 868-873, 2004.
9. M. Armbrust et al., "Above the Clouds: A Berkeley View of Cloud Computing," Technical Report No. UCB/EECS-2009-28, Feb. 2009.
10. 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.
11. 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.
12. 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.
13. B.N. Chun and D.E. Culler, "User-Centric Performance Analysis of Market-Based Cluster Batch Schedulers," Proc. Second IEEE/ ACM
14. Int'l Symp. Cluster Computing and the Grid, 2002.
15. D. Durkee, "Why Cloud Computing Will Never be Free," Comm. ACM, vol. 53, no. 5, pp. 62-69, 2010.
16. 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.
17. K. Hwang, G.C. Fox, and J.J. Dongarra, Distributed and Cloud Computing. Morgan Kaufmann, 2012.
18. "Enhanced Intel SpeedStep Technology for the Intel Pentium M Processor,"White Paper, Intel, Mar. 2004.
19. D.E. Irwin, L.E. Grit, and J.S. Chase, "Balancing Risk and Reward in a Market-Based Task Service," Proc. 13th IEEE Int'l Symp. High Performance Distributed Computing, pp. 160-169, 2004

AUTHOR(S) PROFILE

A.SAHAYA PRINCY received the B.Tech. degree in Information Technology from Kurinji College Of Engineering And Technology, Anna Univercity, Chennai, Tamil, India and the M.E. Computer Science and Engineering from Shivani Engineering College, Anna Univercity, Chennai, Tamilnadu, India in 2012 and 2014.

