Performance Analysis of Cloud Using Queuing Model

[1]Sreemaan Vodela, [2]Mrs P. Akilandeswari(Assistant Professor)
Department of Computer Science & Engineering,
SRM University, Kattankulathur, Chennai, Tamilnadu

***Abstract* - Cloud computing is the emerging field of the computer science. It delivers on the emerging business essentials for quick, elastic and performance. Successful development of cloud computing paradigm necessitates accurate performance evaluation of cloud data centres. As exact modelling of cloud centres is not feasible due to the nature of cloud centres and diversity of user requests, in this project a novel approximate analytical model for performance evaluation of cloud server farms is described and solve it to obtain accurate estimation of the complete probability distribution of the request response time and other important performance factors. It allows cloud users to define the relationship between the number of servers and input buffer size, and the performance factors such as mean number of tasks, blocking probability, and probability that a task will get immediate service. Using the performance indicators, resource provisioning either under or over is avoided so that reliability is maintained to improve performance of the queuing model.**

***Index Terms* - Cloud computing, performance analysis, response time, queuing theory, blocking probability**

I. INTRODUCTION

Cloud Computing is a novel paradigm for the provision of computing base, which designs to change the location of the computing base to the network in order to reduce the costs of management and maintenance of hardware and software resources. Cloud computing has a service-oriented architecture in which services are broadly divided into three categories: Infrastructure-as-a-Service (IaaS), which includes equipment such as hardware, memory, hosts, and communicating components are made accessible over the Internet; Platform-as-a-Service (PaaS), which includes hardware and software computing platforms such as virtualized hosts, operating systems. Software-as-a-Service (SaaS), which includes software applications and other services. A cloud service is different from conventional hosting in three primary views. First, it is provided on request. Second, it is flexible since users can use the service have as much or as little as they want at any given period and third, the service is fully handled by the provider.
A cloud can have a large number of hosts, mostly of the order of hundreds or thousands. Conventional queuing analysis rarely takes systems of this size.

Cloud Computing Technology

Cloud computing is the delivery of computing as a service whereby resources and information are provided to computers and other devices as a utility (like the electricity grid) over a communication. Cloud computing provides services that do not require end-user knowledge of the physical location and configuration of the system that delivers services. Another concept can be given with the electricity grid, where users take power without needing to understand the component devices or infrastructure required to provide the service. Cloud computing is different from hosting services and assets at ISP data center. It is all about computing systems are logically at one place or virtual resources forming a Cloud and user community accessing with intranet or Internet. So, it means Cloud could reside in-premises or off-premises at service provider location. There are types of Cloud computing like 1.Private clouds 2. Public Clouds 3. Hybrid Clouds, say Mr. B.L.V. Rao - CIO and IT Leaders and expert in cloud computing. Cloud computing providers, hand over applications over the net, that can be accessed from browsers, PC's and mobile apps, while the business software and data are stored on servers at a remote location. In some cases, legacy applications (line of business applications that until now have been prevalent in thin client Windows computing) are delivered via a screen-sharing technology, while the resources are amalgamated at a remote data center location; in some cases, total business apps have been written using web-based technologies. Most cloud computing bases consists of services delivered through shared data-centers and appearing as a single point of access for user computing needs. Commercial extending may be required to meet service level agreements (SLAs), but specific terms are less often negotiated by smaller companies.

Cloud Working Progress

Cloud computing has been changing the way most people use the web and they stack their files. It's the system that runs sites like Amazon, Facebook and the core that allows us to take advantage of services like Google Docs and Gmail. The concept of the cloud has been around for a long time in many different forms in the organization world. It means so many computers serving as a service-oriented architecture to provide software and information. So many websites and host-based applications run on particular computers or servers. The way those are set up is that the cloud utilizes the resources from the computers as a *collective virtual*

computer, where the apps can run independently from peculiar computer configurations. They are essentially drifting around in a "cloud of resources", making hardware importance to the applications work to a lesser extent.

II. RELATED WORK

H. Khazaei et al [1] have done a research on Performance Analysis of Cloud Computing Centres. The model allows cloud operators to determine the relationship between the number of servers and input buffer size, on one side, whereas the performance indicators such as mean number of tasks in the system, blocking probability, and probability that a task will gain immediate service, on the other hand. The main profit of having numerous servers in cloud computing is, the system performance increases expeditiously by reducing the mean queue length and waiting time than compared to the conventional approach of having only single server so that the consumers need not wait for a long period of time and also queue length need not be bulky. In this paper, cloud centre is modelled as an [(M/G/1) : (∞/GD MODEL)] queuing system with single task arrivals and a task request buffer of infinite capacity. Performance of queuing system is evaluated using an analytical model and solved it to obtain important performance factors like mean number of tasks in the system.

J. Baker et al [2] have done a research on Megastore: Providing Scalable, Highly Available Storage for Interactive Services. Megastore is a storage system developed to meet the storage requirements of today's interactive online services. It is new in that it mixes the scalability of a NoSQL data-store with the convenience of a conventional RDBMS. It uses synchronous reverberation to achieve high availability and a consistent view of the data. It provides fully serializable ACID semantics over distant replicas with low enough latencies to support interactive applications. It can be accomplished by taking a middle ground in the RDBMS vs. NoSQL design space: and partitioning the data store and replicate each partition, providing full ACID semantics within partitions, but only bounded consistency guarantees across them. A traditional database features are provided, such as secondary indexes, but only those features that can scale within user-tolerable latency limits, and only with the semantics that partitioning strategy can support. Data for most Internet services can be suitably partitioned (e.g., by user) to make this approach feasible, and that a small, but not Spartan, set of features can considerably ease the burden of developing cloud applications. While many systems use Paxos only for locking, master election, or replication of metadata, we believe that Megastore is the largest system deployed that uses Paxos to replicate primary user data across datacenters. Megastore has been widely distributed within Google for several years. It handles more than three billion write and 20 billion read transactions daily and stores nearly a petabyte of primary data across many global datacenters. The key contributions of this paper are: 1. The design of a data model and storage system that allows rapid development of interactive applications where high availability and scalability are built-in from the start; 2. An implementation of the Paxos replication and consensus algorithm optimized for low-latency operation across geographically distributed datacenters to provide high availability for the system; 3. A report on our experience with a large-scale deployment of Megastore at Google.

J.M. Smith et al [3] has done a research on M/G/c/K Blocking Probability Models and System Performance. In this paper, closed-form expressions are developed for the blocking probability of these general service systems and use of these expressions in the optimization of these systems is also demonstrated. There are essentially two problems of interest in this paper. The first is how to estimate the blocking probability *pK* and the second problem concerns the allocation of buffers so that the loss/delay blocking probability will be below a specific threshold. In one sense, the second problem will be handled initially, and then it is displayed after arriving at a closed-form expression for the optimal buffer size, an estimate of the blocking probability can be arrived. Heuristics based upon this approach have one computational advantage since the probabilities of the infinite model only have to be computed once. However, for some situations, computing these probabilities can be problematic when the buffer size gets large. In certain buffer design contexts, one may use an expression for the optimal size as an objective function. In this sense, the convex nature of the objective function is important to understand. If one examines the literature on the convexity of queuing systems, one will come across the work of Harel and Zipkin, and Harel, and more recently Liyanage and Shantikumar. Liyanage and Shantikumar compile comprehensive results for many systems including some new results on finite systems. Harel and Zipkin examined the convexity properties of queuing systems and generally focused on infinite queuing systems except for their examination of Erlang loss systems. Pacheco has generalized the analysis of Erlang loss systems and some of the same results. However, to the author's knowledge, no one has really examined the convexity properties of the blocking probabilities relaxing the integrality of *K* starting from the *M/M/*1*/K* system and then on to the *M/M/c/K* systems. If one relaxes this integrality as we shall see, then a closed-form expressions for *K* can be realized. Also, when examining this closed-form expression, the convexity properties will be examined and ultimately, with this closed-form expression, we will develop our approximation for *pK.*

B.N.W. Ma et al [4] have done a research on Approximation of the Mean Queue Length of an M=G=c Queuing System. In the almost all literature mentioned above, the approximation results are compared with those of the M/PH/c queue for the quality of approximation. The PH distributions used for comparisons are selected by the mean and the coefficient of variation (the ratio of the standard deviation to the mean) of the service time. There may be many PH distributions with the same mean and variance. So, unless the system performance measures are depend only on the first two moments of the service time, the approximations works well for some cases but not for other cases even though the service time distributions have the common mean and variance. Objective of this paper has two folds. One is to investigate numerically the sensitivity of the system characteristics such as the mean and the standard deviation of the queue length and the blocking probability with respect to the moments of the service time. The other is to propose an approximation of the steady state distribution of the number of customers in M/G/c queue. It is shown numerically that the mean and the standard deviation of the number of customers in queue are strongly affected by the third moment of the service time for some cases. Based on the sensitivity analysis, the service time distribution with PH distribution is approximated by matching the first three moments of the service times and use M/PH/c queue for an

approximation of M/G/c queue. Simulation models are developed with ARENA. Simulation run time is set to 80,000 unit times including 20,000 unit times of warm-up period, where the expected value of service time is one unit time. Different random number streams are used for the distributions of inter-arrival times, service times. Ten replications are conducted for each case and the average value and the half length of 95% confidence interval are obtained. An appropriate PH distribution is chosen by fitting the first three moments of the service time and then compute the performance characteristics of the approximating system.

III. PROPOSED ANALYTICAL MODEL

Model Analysis

This is a model of a cloud computing resource pool. The model comprises a queuing and waiting area and a resource pool, wherein a group of user requests enter the resource pool for a user to select and use after a service time in the queuing and waiting area. The relationship between the quantity of resources integrated by the resource pool and a mean waiting time and obtains a conclusion that the mean waiting time can be maintained in a user tolerable range by only integrating moderate quantity of the resources by the cloud computing resource pool. In this model a Virtual Machine is created and allocated some storage which can be used as the resource pool. Http Get and Post method are used in this proposed system. Http Get method is used to send the http URL link. User sends the Http link to the server. It will receive the link and send the acknowledgement to the client. Then, in post method one user and multiple servers. The user sends the link to all servers in similar time the last server sends the acknowledgement to its preceding server. Then that server sends the both acknowledgement to main server. That will be sent all acknowledgements to user.

Queuing Theory

The queuing systems constitute a central tool for Modelling and analysis of performance of telecommunication systems, computer systems and many others. There are some differentiating factors like arrival process, service process, number of servers, number of queues, number of waiting places, server discipline (i.e. FIFO, LIFO etc.), scheduling, information available, etc. In cloud computing, there are a lot of users who access the service. We design cloud computing as in Figure 1.This model consists of cloud architecture which can be a service centre. The service centre is a single point of access for all kinds of customers all over the world. The service centre is a collection of service resources which is provided by the provider to host all applications for users. Each user can employ to use the service according to the different kinds of requesting and pays some money to the provider of the service.

Cloud computing provider builds the service centre to be used by users, such as Amazon which provides different types of manners. In this paper, we use the on-demand examples. On-demand examples let you pay for compute capacity by the hour with no long-term consignment. This frees you from the costs and complications of designing, buying, and maintaining hardware and transforms what are commonly large fixed costs into much smaller variable costs. The cloud computing service model displayed in Figure 1 can be mapped as a queuing model in Figure 2.

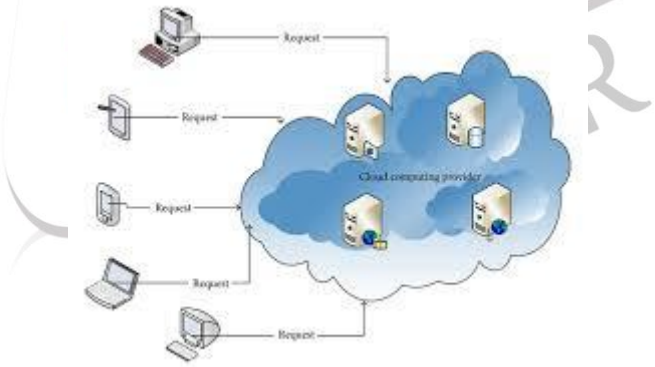

Figure 1: An example of request for cloud computing service model.

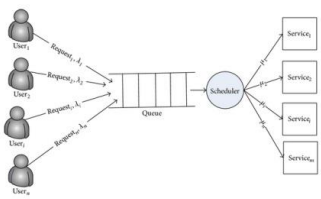

Figure 2: A queuing performance mode for computer services in cloud computing.

Waiting and Response Time Analysis

A waiting time is the period of time between when an action is requested or mandated and when it occurs. This time for which a request will wait in the queue called as waiting time should be analyzed. In technology, response time is the time a system or functional unit takes to react to a given input. Response time is the total amount of time it takes a request for service. That service

can be anything from a memory fetch, to a disk IO, to a database query, or loading a full web page.After getting served the response time for a particular service should also be analysed

IV. CONCLUSION

In the context of the evolution of cloud computing network, to develop realistic models which represent such a network appears as a great challenge for researchers. In this paper we have proposed an analytical model for performance evaluation of a cloud using the queuing model. In this model we calculated analytically the performance indicators such as the average number of tasks in the system, blocking probability, probability of immediate service and the average of response time.

REFERENCES

[1] H. Khazaei, J. Misic, and V.B. Misic, "Performance Analysis of Cloud Computing Centers," Proc. Seventh Int'l ICST Conf. Heterogeneous Networking for Quality, Reliability, Security and Robustness (Q Shine), Nov. 2010.

[2] Jason Baker, Chris Bond, James C. Corbett, JJ Furman, Andrey Khorlin, James Larson, Jean Michel L´eon, Yawei Li, Alexander Lloyd, Vadim Yushprakh Google, Inc "Megastore: Providing Scalable, Highly Available Storage for Interactive Services".

[3] J. MacGregor Smith, Department of mechanical and industrial engineering, University of Massachusetts, Marston Hall, Amherst, MA "M/G/c/K Blocking Probability Models and System Performance".

[4] Bobby N.W.MA Ryerson Polytechnical Institute, Toronto, Ontario, Canada, John W.Mark University of Waterloo, Waterloo, Ontario, Canada, "Approximation of the Mean Queue Length of an M=G=c Queuing System"

[5] L.M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A Break in the Clouds: Towards a Cloud Definition," ACM SIGCOMM Computer Comm. Rev., vol. 39, pp. 50-55, Dec. 2008.

[6] L. Wang, G. von Laszewski, A. Younge, X. He, M. Kunze, J. Tao, and C. Fu, "Cloud Computing: A Perspective Study," New Generation Computing, vol. 28, pp. 137-146, 2010.

[7] L. Kleinrock, Queuing Systems: Theory, vol. 1, Wiley-Interscience, 1975.

[8] Amazon Elastic Compute Cloud, User Guide, API Version ed., Amazon Web Service LLC or Its Affiliate, http://aws.amazon.com/documentation/ec2, Aug. 2010.

[9] K. Xiong and H. Perros, "Service Performance and Analysis in Cloud Computing," Proc. IEEE World Conf. Services, pp. 693-700, 2009.