2011 17th Asia-Pacific Conference on Communications (APCC)
2nd – 5th October 2011 | Sutera Harbour Resort, Kota Kinabalu, Sabah, Malaysia

Modeling of Cloud System using Erlang Formulas

Mohamed Firdhous[1], Osman Ghazali[2], Suhaidi Hassan[3]
InterNetWorks Research Group
Universiti Utara Malaysia
Sintok, Kedah Darul Aman
Malaysia
mfirdhous@internetworks.my[1], osman@uum.edu.my[2], suhaidi@uum.edu.my[3]

Abstract – **Cloud computing has been considered as the 5th utility after electricity, water, gas and telephony. When the cloud computing matures, there will be multiple vendors offering different services at different Quality of Services and at different prices. This would necessitate new tools and mechanisms for analyzing the performance of the system for matching the offerings with requirements. In this paper the authors have modeled the cloud system using queuing theory, specifically Erlang formulas. Four different cloud utility models of various complexities have been presented and analyzed using simulations. The simulations results have also been presented along with an in depth discussion.**

Keywords – Cloud Computing; Erlang Formula; Queuing Theory; Quality of Service; Response Time

I. INTRODUCTION

Cloud computing has been considered the new computing paradigm that would change the way computing resources have been purchased and used. The cloud computing is now commonly known as the 5th utility based on its business model after electricity, water, gas and telephony [1]. Until now, the investments on computing resources were considered capital expenditure and these organizations had to spend that money upfront. With the advent of cloud computing, the expenditure on computing resources can be moved from capital to operational cost. Also they will be paying only for the services consumed rather than for the hardware or software resources.

Cloud computing involves several components such as network devices, computing resources, storage systems distributed over wide distances. Users have the flexibility to bring these distributed resources together to create a unique environment for themselves.

These components may be described using mathematical models along with causalities. Cloud systems would be receiving requests for different services and would in turn be evoking virtual devices for servicing them. It would be possible to model the incoming requests and the provisioning of services using statistics as all these operations are random processes.

This paper presents a mathematical model of a cloud computing system built using queuing theory and other mathematical tools. The proposed model was verified using simulations for their validity.

Mohamed Firdhous is a Senior Lecturer attached to the Faculty of Information Technology, University of Moratuwa, Sri Lanka. He is currently on leave pursuing his PhD at the Universiti Utara Malaysia.

II. CLOUD COMPUTING

Cloud services have been commonly identified under three main categories. They are namely Infrastructure as Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) [2]. Computer hardware resources such as processors, hard drive space are sold as services using virtualization techniques under IaaS. A virtual computer is similar to a real computer in every sense except it becomes alive and consumes real resources including processor cycles only when needed. The owner of a virtual computer can install the operating system of his or her choice and application software as if he or she owns a real computer. The hosting of these system and application software are independent of other software hosted in the same physical system but on different virtual computers, hence will not interfere with each other. Since the real physical resources are consumed only when the virtual system is active, the user will be charged only for the actual usage and not for the reservation of resources like in traditional data centers. PaaS provides a complete environment for application development and hosting including platform, tools and other resources on top of virtual computers. The PaaS owner can develop his application test and then provision that software across the Internet. SaaS is the new way of software provisioning as a service over the Internet rather than a commodity. Users can customize these applications to suit their requirements similar to an application hosted on a private computer [3].

Fig. 1 shows the layered architecture of a typical cloud computing system. This layered architecture includes five layers including the physical hardware layer and the virtualized

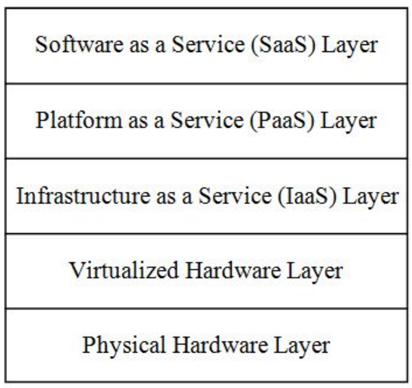

Figure 1. Layered architecture of cloud system

hardware layer in addition to the cloud service layers. The physical hardware layer provides the raw computing resources such as hard drive space, data store, computing power, networking etc. This layer is created using server class computers hosted at data centers, clusters, grids, storage networks or any other computing systems.

The virtualized hardware layer is created by installing virtualization software such as VMware, Virtual Machine Monitor (VMM), Xen, KVM etc. The virtualization software creates virtual machines by slicing the physical resources such as CPU, RAM, Storage, and Networking. The virtualization software would provide the necessary isolation and security to make the slices independent of each other. Fig. 2 shows a typical cloud system with multiple service providers, cloud intermediaries and clients.

From Fig. 2, it can be seen that a client requiring services may purchase different resources from different vendors to suit his requirements. Each vendor may have multiple physical devices each having multiple virtual devices. The intermediaries such as cloud coordinators, cloud brokers or cloud exchanges may also combine resources from multiple cloud providers and market them to customers as a single package [4].

When the requests for services are received by the service providers, the service providers have to identify the request and forward them to the appropriate system such as the server for providing the processor, hard drive, data store etc., providing the service. The service will be put on a queue first depending on the number of service requests arriving at the system per unit time. If the number of requests is short and the system is fast, the request will be serviced without delay. On the other hand, if the number of requests is large, the requests will have to wait for a long time, depending on the speed of the system and the availability of resources.

The requests will have to wait in multiple queues depending on the type of service and the number of intermediaries involved. If the client accesses the service from a single service provider directly and all the resources are located in a single physical computer, the request will go only through internal queues for processor time, disk access etc. If the request involves multiple service providers and also intermediaries, more queues will be involved. The number of queues involved and their performance will affect the overall performance of the client applications. The clients require certain level of quality of service agreed upon the Service Level Agreement (SLA) in terms of response times from service providers [5-7]. Hence it would be better to have a mechanism to model the performance of cloud services before entering into service agreements. Modeling cloud performance would help clients to select the right service provider who would meet their requirements. Cloud intermediaries would be able to match the client requirements with the right service offerings selected from multiple vendors. The service providers can also benefit from service modeling as they would be able to proactively scale their systems to meet the requirements of the intended market segment.

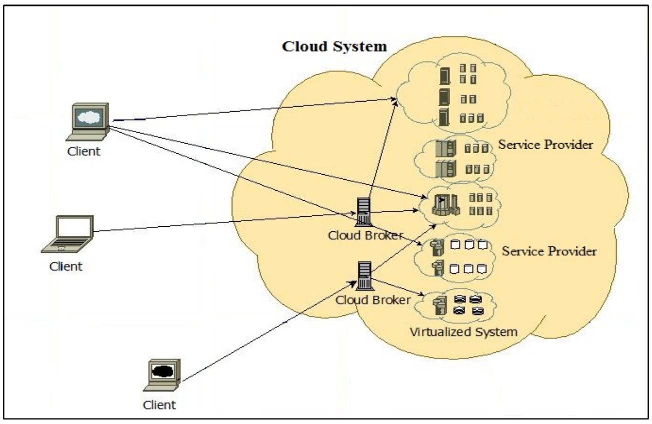

Figure 2. A cloud system

III. ERLANG FORMULAS

Erlang formulas have been used for studying the behavior of switching networks especially in traditional telephone exchanges and call centers [8]. There are two formals developed by Erlang in order to study the behavior of queuing networks namely Erlang B and Erlang C formulas [9]. There is a slight difference between the formulas as they study the behavior of queues under different conditions. Erlang B formula studies the blocking probability or the loss probability of system with limited capacity. The Erlang B formula considers the limited case of fixed number of servers and no waiting slots. Any client (or request) arriving when all the servers are busy will be turned away or lost and will have to try again as new arrivals. In contrast, Erlang C model assumes infinite waiting positions. That is, a customer will be served even after a long wait and hence will never be lost. The Erlang C formula computes the probability of all the servers being busy requiring the customers to be waiting in the queue [10].

Erlang B model is represented in Kendall notation as *M/M/n/n*. From the Kendall's notation, it can be seen that the last two parameters namely the number of servers and the number of clients are same indicating no waiting. The arrival of customers is assumed to be Poisson distributed with an inter arrival rate of λ and service time is exponentially distributed with a rate of μ. Erlang B function is given by;

$$P_{Br}(n,\rho) = \frac{\left({\rho^n}/{n!}\right)}{(1+\rho+ \rho^2/2! + \rho^3/3! \ldots + \rho^n/n!)} \quad (1)$$

where P_{Br} – probability of blocking

n – no. of servers

ρ – utility defined as $\rho = \frac{\lambda}{\mu}$

λ – arrival rate

μ – service rate

This model is inappropriate for a practical system as it would result in poor quality of service.

Erlang C formula with infinite waiting positions is represented in Kendall's notation as *M/M/n/∞*. The Erlang C formula is given by;

$$P_{cr}(n,\rho) = \frac{n*P_{Br}}{(n-\rho*(1-P_{Br}))} \quad (2)$$

where P_{cr} – probability of all servers being busy

P_{Br} – Erlang B function as defined in (1)

Combining (1) and (2) the Erlang C function can be defined in absolute terms as given in Equation (3).

$$P_{cr} = \frac{\frac{\rho^n n}{n!(n-\rho)}}{\sum_{i=0}^{n-1}\frac{\rho^i}{i!} + \frac{\rho^n n}{n!(n-\rho)}} \quad (3)$$

Evaluating Erlang formulas directly by computers is inefficient and may produce overflow as n! grows exponentially with an increase in n. Researchers have developed efficient algorithms to evaluate the Erlang formulas exploiting certain special properties and evaluating iteratively [11].

IV. MODELING CLOUD SYSTEMS

Cloud systems can be modeled connecting multiple *M/M/n* queues based on the configuration selected. In this paper, a few simple configurations will be evaluated for the purpose of computing the performance of these systems. These configurations have been selected for evaluation as they can be combined to build more complex systems. The performance of complex systems can be predicted based on the characteristics of basic systems.

A. *Model 1: Client Accesses Multiple Service Providers*

Fig. 3 shows the configuration where the client accesses the service providers directly for services. The client accesses different service providers for different services such as computer power from one service provider, drive space from a different service provider, data from another service provider

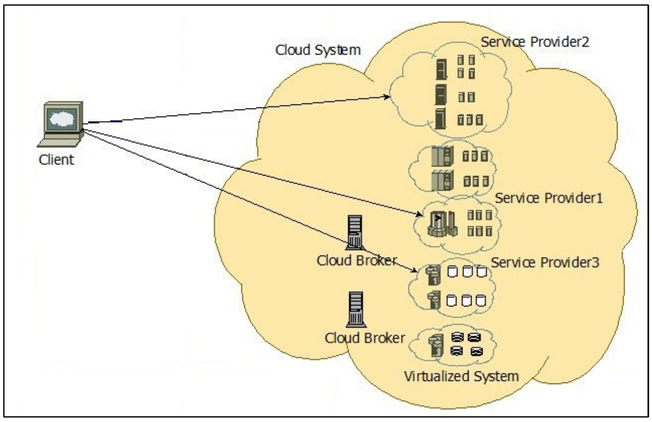

Figure 3. Client accessing multiple service providers in parallel

etc. This configuration can be modeled using parallel queues as shown in Fig. 4. The work has been assigned to different service providers depending on the service required. The feedback path is only the completion of operation message and hence assumed to consume no time.

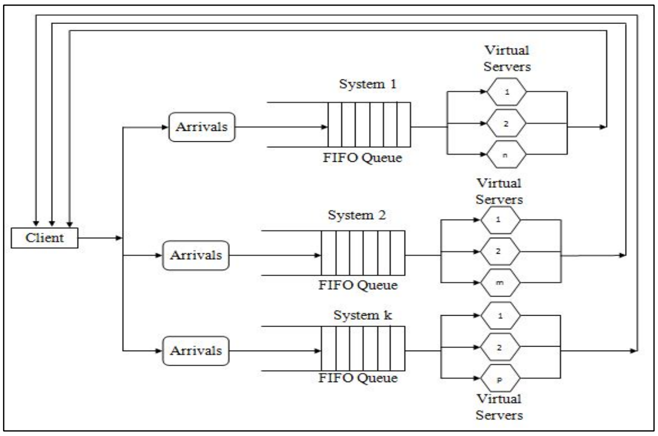

Figure 4. Parallel queues representing multiple service providers

If the arrival rates and service rates of the systems are assumed to be $\lambda_1, \lambda_2, \ldots \lambda_k$ and $\mu_1, \mu_2, \ldots \mu_k$ respectively.

The performance of each queue can be represented using parallel queues as follows:

$$P_{Br}(n_1,\rho_1) = \frac{\left({\rho_1}^{n_1}/_{n_1!}\right)}{(1+\rho_1+{\rho_1}^2/2!+{\rho_1}^3/3! \ldots.+{\rho_1}^{n_1}/n_1!)}$$

$$P_{Br}(n_2,\rho_2) = \frac{\left({\rho_2}^{n_2}/_{n_2!}\right)}{(1+\rho_2+{\rho_2}^2/2!+{\rho_2}^3/3! \ldots.+{\rho_2}^{n_2}/n_2!)}$$

$$P_{Br}(n_k,\rho_k) = \frac{\left({\rho_1}^{n_k}/_{n_k!}\right)}{(1+\rho_k+{\rho_k}^2/2!+k^3/3! \ldots.+{\rho_k}^{k}/n_k!)}$$

The work assigned to each service provider will be carried out in parallel and independent of each other. Hence the servers and the queues can be assumed to be isolated and independent of each other. The service providers may take different durations to complete the tasks assigned depending on the server performance and the type of service requested. Hence the completion of the slowest process will conclude the entire operation. Hence the system can be simplified to a single queue with the longest processing delay.

B. *Model 2: Client Accesses Multiple Service Providers through Cloud Broker*

Fig. 5 and Fig. 6 show the case where a client requests for services through a cloud intermediary known as cloud broker.

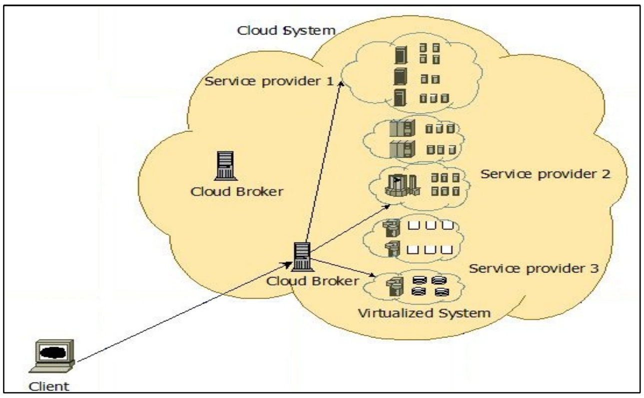

Figure 5. Accessing multiple service providers through cloud broker

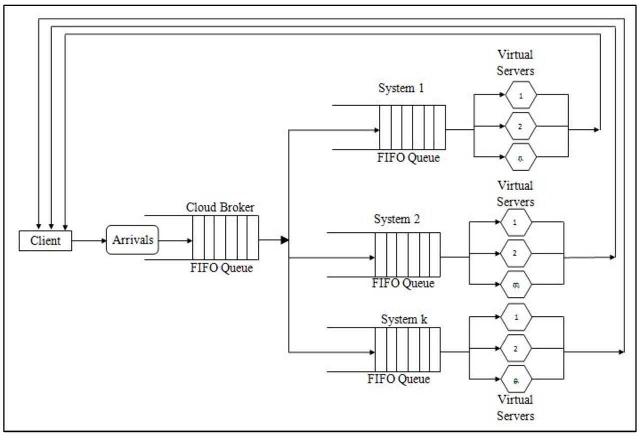

Figure 6. Queues representing access of multiple service providers through a cloud broker

The situation is similar to the case of branching queues where all the incoming traffic to the cloud broker queue is diverted to different queues based on certain probabilities. But in reality, the case is more complicated as there will be several cloud intermediaries who will direct traffic to various service providers. Hence a service provider may receive traffic from multiple different cloud intermediaries and clients directly. In such a case, it will be safe to assume each service provider queue to be independent of both other service providers and cloud intermediaries.

The analysis of this system can be simplified by considering different service providers independently and then consider the cloud broker queue to be in series with the queue of the slowest service provider.

C. *Model 3: Cloud System with Combined Parallel and Series Access*

The model shown in Fig. 7 and Fig. 8 represents the situation where the required resources such as data, program code etc., are brought to another computer for processing. The request will undergo multiple parallel queuing initially and then it will be queued at the final processing node. This case is similar to the situation discussed in Model 2 except that the order of different types of queues is reversed.

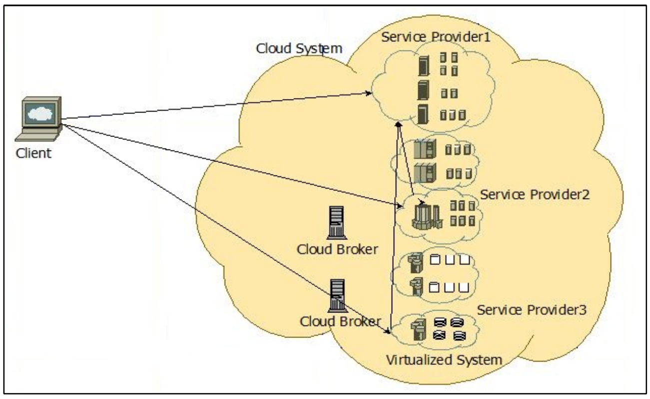

Figure 7. Accessing cloud system multiple service providers in parallel and series

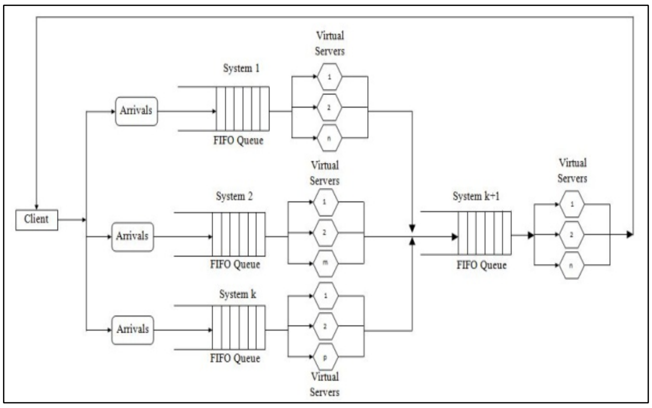

Figure 8. Queues representing multiple service providers

Model 3 can be analyzed similar to Model 2, except changing the order of the queue types.

D. *Model 4: Cloud System with Cloud Broker and Mixed Queuing*

The Model represented in Fig. 9 and Fig. 10 is a combination of Models 2 and 3. The client accesses services through a cloud intermediaries and the final processing is carried out by a single service provider.

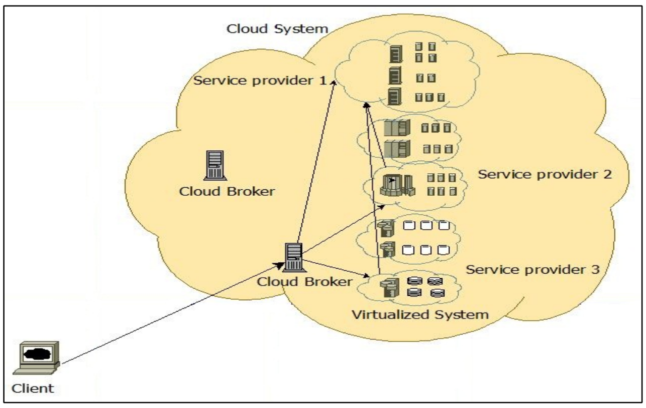

Figure 9. Accessing multiple service providers through cloud broker

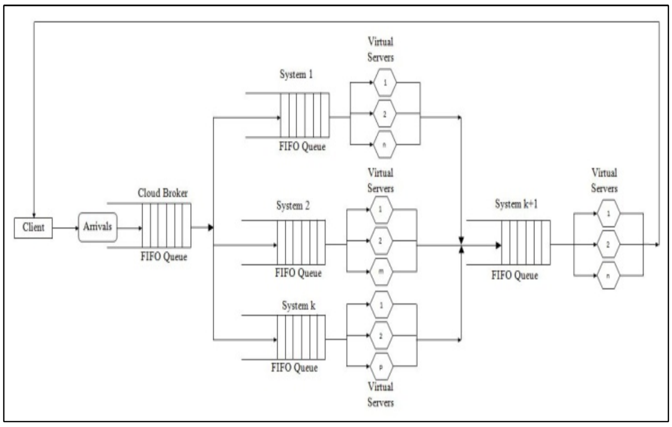

Figure 10. Parallel queues representing multiple service providers

The situation shown in Model 4 can be analyzed using a combination of queues connected as series-parallel-series fashion and considering each queue independent of each other.

V. SIMULATION

The performance of the models proposed has been analyzed through simulations. A simulation environment comprising of various queues have been developed using GNU Octave. The *M/M/n* queues were simulated using the qnetworks, the Queuing Networks analysis package for GNU Octave [12]. The simulation environment was set up with four service providers and a cloud broker with varying capacities and performance metrics. The parameters of the service provider are as shown in Table 1. An additional fixed traffic was added to the system at every service provider node to have an environment similar to the real world systems.

TABLE 1: PARAMETERS OF SERVICE PROVIDERS

	No. of Virtual Servers	Response Time (ms)
Service Provider 1	5	1.0
Service Provider 2	3	1.8
Service Provider 3	2	2.5
Service Provider 4	5	1.0
Cloud Broker	1	0.1

Fig. 11 shows the response time experienced traffic under increasing load. From the figure it can be seen that Model 4 undergoes the largest delay compared to other models due to more queues. As the load increases, the response times start to show an exponential increase in the response time. Model 1 shows a rapid increase compared to other models.

Fig. 12 shows the average response time of the models under increasing load. Initially both Models 3 and 4 have similar response time but later the average response time of model 4 starts to increase. This is due to the increasing load put on the service provider 4. The average response time of Model 1 shows a rapid increase compared to other models.

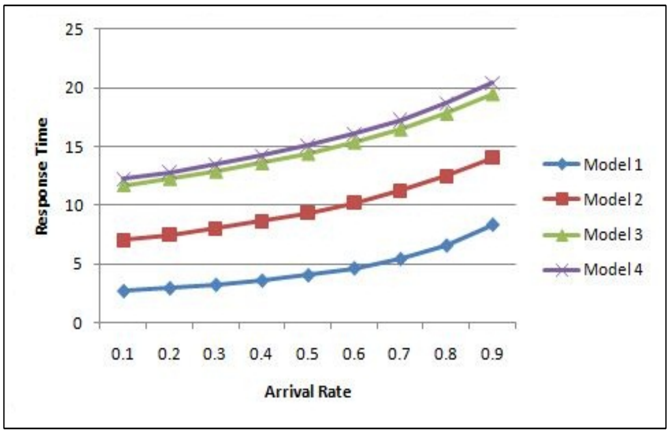

Figure 11. Response time of different models under increasing load

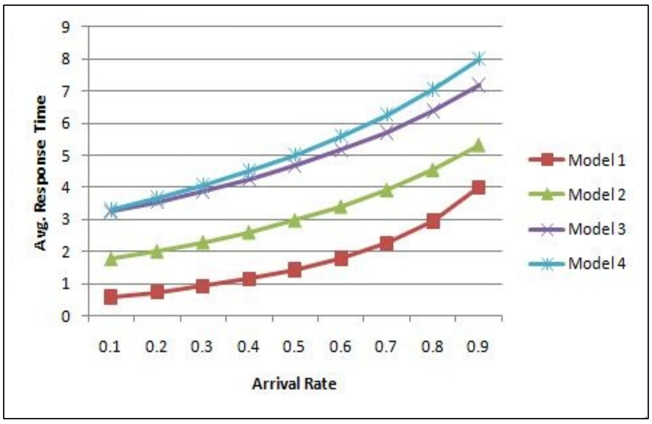

Figure 12. Avegare response time of different models under increasing load

Fig. 13 shows comparison between the response time underwent by simulation traffic and the average response time of the model. Form the figure, it can be seen that traffic undergoes rapid change in response time under increasing load compared to the average response time of the model. From the customers' point of view, this is very important as they would like to have predictable delay rather than varying delays.

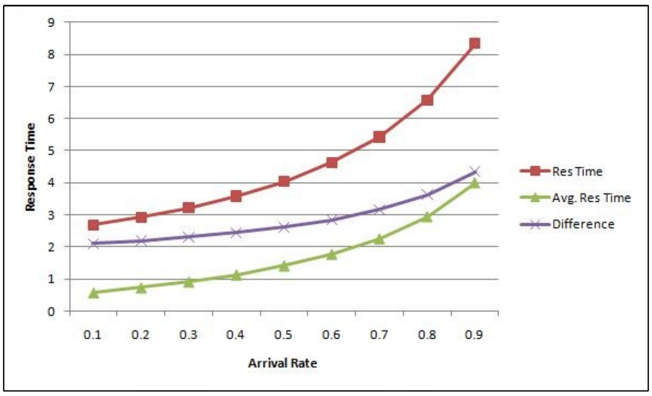

Figure 13. Comparison of response time against average response time for model 1

Fig. 14 shows the comparison of response times of the simulated traffic against the average response time of the model. Though the response time of the simulated traffic increases more rapidly compared to the average traffic, the ratio of the increase is more in Model 1 compared to Model 4. The difference between the response times under Model 1 has more than doubled when the arrival rate was increased from 0.1 to 0.9 whereas under Model 4 shows only around 50 percent increase.

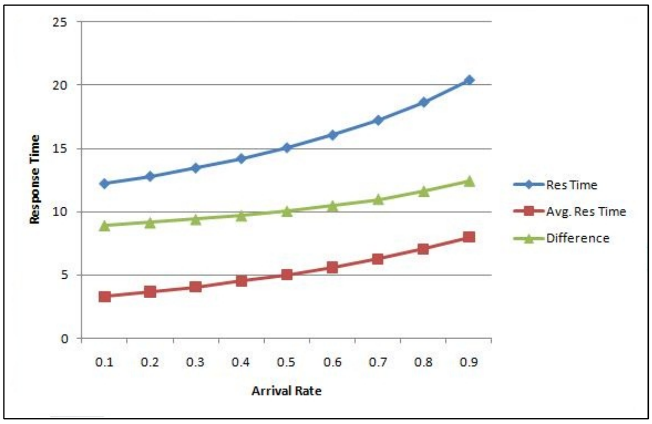

Figure 14. Comparison of response time against average response time for model 4

VI. CONCLUSIONS

In this paper, the authors have modeled the cloud system using queuing theory specifically Erlang formulas. The cloud system has been modeled using Erlang C formula and four different cloud utility models have been presented. Finally the presented models have been simulated in order to characterize the performance of the models and the results have been presented. The performance of the systems has been analyzed from the customers' perspective rather than the providers' perspective. So only the response times of different configurations have been studied in this work. From the simulation results it could be seen that Model 1 undergoes less delay compared to all other models. This is mainly due to the fewer bottlenecks in the system. But the performance of Model 1 degrades faster compared to other models as the arrival rate increases due to lack of coordination between the service providers as all the service providers under Model 1 are independent. The rapid increase in the response time compared to the average response time of the system is important from the customers' point of view. Customers would like to have guaranteed performance within a certain confidence level rather than an average performance guarantee. This analysis and results would help both customers as well as service providers. The customers can realize the limitation of Model 1 in terms of performance as it degrades faster under increased load compared to other models. Service providers can use this results to size their systems for a given traffic load and guaranteed performance rather than depending on the average performance of the system.

The study has considered only the performance of the service providers. The performances of the intermediate networks connecting the client to the service providers and in between the service providers were not considered. This is a major limitation of the study as the network latency plays a major role in the performance of interconnected systems. The authors propose study the performance of a complete system comprising of clients, service providers and intermediate networks in a future work.

REFERENCES

[1] 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 5th utility," Future Generation Computer Systems, vol. 26, no. 6, pp. 599-616, June 2009.

[2] C. Vecchiola, S. Pandey, and R. Buyya, "High-performance cloud computing: A view of scientific applications," in 10th International Symposium on Pervasive Systems, Algorithms, and Networks, Kaohsiung, Taiwan, pp. 4-16, 2009.

[3] R. Prodan, and S. Ostermann, "A survey and taxonomy of Infrastructure as a Service and web hosting cloud providers," in 10th IEEE/ACM International Conference on Grid Computing, Banff, AB, Canada, pp. 17-25, 2009.

[4] R. Buyya, R. Ranjan, and R. N. Calheiros, "InterCloud: Utility-oriented federation of cloud computing environments for scaling of application services," in Algorithms and Architectures for Parallel Processing. Berlin / Heidelberg: Springer, pp. 13-31, 2010.

[5] H. Ludwig, "Web services QoS: External SLAs and internal policies or: How do we deliver what we promise?," in Fourth International Conference on Web Information Systems Engineering Workshops (WISEW'03), Rome, Italy, pp. 115-120, 2003.

[6] H. J. Moon, Y. Chi, and H. Hacıgümüş, "SLA-aware profit optimization in cloud services via resource scheduling," in 6th World Congress on Services (SERVICES-1), Miami, FL, USA, pp. 152-153, 2010.

[7] Y. C. Lee, C. Wang, A. Y. Zomaya, and B. B. Zhou, "Profit-driven service request scheduling in clouds," in 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, Melbourne, Australia, pp. 15-24, 2010.

[8] K. Hisashi, and B. L. Mark, "Generalized loss models and queueing-loss models," International Transactions in Operational Research, vol. 9, pp. 97-112, 2002.

[9] L. A. G. Franzese, M. M. Fioroni, R. C. Botter, and P. J. de F. Filho, "Comparison of call center models," in Winter Simulation Conference, Austin, TX, USA, pp. 2963-2970, 2009.

[10] I. Angus, "An introduction to Erlang B and Erlang C," Telemanagement - the Angus Report on Business Telecommunications in Canada, pp. 6-8, July-August 2001.

[11] S. Qiao and L. Qiao, "A robust and efficient algorithm for evaluating Erlang B formula," Dept. of Comuting and Software, McMaster University, Hamilton, ON, Canada, Technical Report, 1998.

[12] M. Marzolla, "The qnetworks toolbox: a software package for queueing networks analysis," in 17th International Conference on Analytical and Stochastic Modeling Techniques and Applications (ASMTA 2010), Cardiff, UK, pp. 102-116, 2010.