PERFORMANCE FACTORS OF CLOUD COMPUTING DATA CENTERS USING [(M/G/1) : (∞/GDMODEL)] QUEUING SYSTEMS

N.Ani Brown Mary[1] and K.Saravanan[2]

[1]Department of Computer Science and Engineering,
Regional Centre of Anna University, Tirunelveli
anibrownvimal@gmail.com
[2]Assistant professor, Department of Computer Science and Engineering,
Regional Centre of Anna University, Tirunelveli
Saravanan.krishnann@gmail.com

ABSTRACT

The ever-increasing status of the cloud computing hypothesis and the budding concept of federated cloud computing have enthused research efforts towards intellectual cloud service selection aimed at developing techniques for enabling the cloud users to gain maximum benefit from cloud computing by selecting services which provide optimal performance at lowest possible cost. Cloud computing is a novel paradigm for the provision of computing infrastructure, which aims to shift the location of the computing infrastructure to the network in order to reduce the maintenance costs of hardware and software resources. Cloud computing systems vitally provide access to large pools of resources. Resources provided by cloud computing systems hide a great deal of services from the user through virtualization. In this paper, the cloud data center is modelled as [(M/G/1) : (∞/GD MODEL)] queuing system with a single task arrivals and a task request buffer of infinite capacity.

KEYWORDS

Cloud computing, performance analysis, response time, queuing theory, markov chain process

1. INTRODUCTION

Cloud computing is the Internet-based expansion and use of computer knowledge. It has become an IT buzzword for the past a few years. Cloud computing has been often used with synonymous terms such as software as a service (SaaS), grid computing, cluster computing, autonomic computing, and utility computing [1]. SaaS, Software as a Service, is a software delivery model in which software and related data are centrally hosted on the cloud. SaaS is typically accessed by users using a thin client via a web browser .Grid computing and cluster computing are two types of underlying computer technologies for the development of cloud computing. Autonomic computing is a computing system services that is capable of self-management, and utility computing is the packaging of computing resources such as computational and storage devices [2,3]. Cloud centers differ from conventional queuing systems in a number of important aspects. A Cloud center can have outsized number of capability (server) nodes, typically of the order of hundreds or thousands [4]; conventional queuing analysis rarely considers systems of this size. Task service times must be modeled by a general, rather than the more convenient exponential, probability distribution. The coefficient of variation of task service time may be high over the value of one. Due to the dynamic nature of cloud environments, diversity of users requests and

DOI: 10.5121/ijgca.2013.4101

time dependency of load, cloud centers must provide expected quality of service at widely varying loads [5,6].

Flourishing development of cloud computing paradigm necessitates accurate performance evaluation of cloud data centers. As exact modeling of cloud centers is not feasible due to the nature of cloud centers and diversity of user requests, we here describe a novel approximate analytical model for performance evaluation of cloud server farms and solve it to obtain accurate estimation of the complete probability distribution of the request response time and other important performance indicators. 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 obtain immediate service, on the other hand. The key benefit of having numerous servers in cloud computing is, the system performance increases efficiently 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, we modeled the cloud center as an [(M/G/1) : (∞/GD MODEL)] queuing system with single task arrivals and a task request buffer of infinite capacity. We evaluate the performance of queuing system using an analytical model and solve it to obtain important performance factors like mean number of tasks in the system. The residue of the paper is organized as follows. Section 2 describes the related work. Section 3 gives a brief overview on an assortment of queuing models and assumptions. Section 4 discusses our analytical model in detail. We present and discuss analytical results in section 5. Our findings are summarized in Section 6, where we also have outlined the directions for future effort.

2. RELATED WORK

Cloud computing provides user a complete software environment. It provides resources such as computing power, bandwidth and storage capacity. It has engrossed considerable investigate attention, but only a diminutive portion of the work done so far has addressed performance issues, and rigorous analytical approach has been adopted by only a handful among these. The response time is a major constraint in the queuing system as distribution of response time was obtained for a cloud center model as an M/M/m/m + r queuing system where both interarrival and service times were assumed to be exponentially distributed, and the system had a finite buffer of size m + r. The response time was broken down into waiting, service, and execution periods, assuming that all three periods are independent which is unrealistic, according to authors' own argument in [7].

The inter-arrival time and/or service time are not exponential is more complex. Most theoretical analyses have relied on extensive research in performance evaluation of M/G/m queueing systems. However, the probability distributions of response time and queue length in M/G/m cannot be obtained in closed form, which necessitated the search for a suitable approximation in [8]. They have measured the average response time of a service request in [14], but measurement techniques are hard to be used in computer service performance prediction. In order to compute a percentile of the response time one has to first find the probability distribution of the response time. This is not an easy task in a complex computing environment involving many computing nodes.

They have considered a state-dependent M/G/1 where the interarrival and the service time distributions depend on the amount of unfinished work in the system. They have applied perturbation methods to derive approximations for several measures pertaining to the unfinished work and the mean busy period in such a queue of [9]. The important observation is that the arrival of a consumer (the size of which depends on the state of the underlying Markov chain) can be viewed as the arrival of a super consumer, whose service time is distributed as the sum of the service requests of the consumers in the system of [11].A comparison is performed among all these approaches, mainly focusing on getting reliable estimates of prediction of performance of the various architectures depending on the workflows. However, no economic cost comparisons between the different platforms are shown. Some works have been presented to compare the performance achieved by means of the cloud with other approaches based on desktop workstations, local clusters, and HPC shared resources with reference to sample scientific workloads in [16].

2.1. Control Utilization Models

Control debauchery 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 Control debauchery; however, the dominant component in a well designed circuit is dynamic control utilization p (i.e., the switching component of power), which is approximately $P= bDW^2f$, where b is an activity factor, D is the loading capacitance, W is the supply voltage, and f is the clock frequency [12].

In the supreme case, the supply voltage and the clock frequency are related in such a way that $W \propto f^{y\varphi}$ for some constant $\varphi > 0$ [15]. The processor execution speed s is usually linearly proportional to the clock frequency, namely, $s \propto f$. For ease of discussion, we will assume that $W = cf^{\varphi}$ and $s = ef$, where c and e are some constants.

Hence, we know that Control Utilization is

$$P = bDW^2f$$

$$= bDc^2f^{2\varphi+1}$$

$$= (bDc^2/e^{2\varphi+1})/ s^{2\varphi+1}$$

$$P = \xi s^{\alpha}$$

Where $\xi = bDc^2/e^{2\varphi+1}$and $\alpha = 2\varphi + 1$. For illustration, by setting $c = 1.17$, $bD = 8.0$, $e = 2.0$, $\varphi = 0.6$, $\alpha = 2\varphi+1 = 3.0$, and $\xi = bDc^2/e^{\alpha} = 9.4192$, the value of P is calculated by the equation $P = bDW^2f = \xi s^{\alpha}$ is reasonably close to that in [13] for the Intel Pentium M processor.

2.2. Models and Assumptions

We can define ergodicity of a Markov chain as follows: A Markov chain is called ergodic if it is irreducible, recurrent non-null, and aperiodic. We define communicability as follows, State i communicates with j, written $i \rightarrow j$, if the chain may ever visit state j with positive probability, starting from i. That is, $i \rightarrow j$ if $pij(n) > 0$ for some $n \geq 0$. We say i and j intercommunicate if $i \rightarrow j$ and $j \rightarrow i$, in which case we write $i \leftrightarrow j$. It can be seen that $\leftrightarrow$ is an equivalence relation, hence the state space S can be partitioned into the equivalence classes of $\leftrightarrow$; within each equivalence class all states are of the same type.

A set C of states is called

(a) Closed, if $pij = 0$ for all $i \in C$, $j /\in C$.

(b) Irreducible, if $i \leftrightarrow j$ for all $i, j \in C$.

The kendall's classification of queuing systems exists in several modifications. Queuing models are generally constructed to represent the steady state of a queuing system, that is, the typical, long run or average state of the system. As a consequence, these are stochastic models that represent the probability that a queuing system will be found in a particular configuration or state. A general procedure for constructing and analysing such queuing models is:

1. Identify the parameters of the system, such as the arrival rate, service time, queue capacity, and perhaps draw a diagram of the system, **2.** Identify the system states. (A state will generally represent the integer number of customers, people, jobs, calls, messages, etc. in the system and may or may not be limited), **3.** Draw a state transition diagram that represents the possible system states and identify the rates to enter and leave each state. This diagram is a representation of a Markov chain, **4.** Because the state transition diagram represents the steady state situation between states there is a balanced flow between states so the probabilities of being in adjacent states can be related mathematically in terms of the arrival and service rates and state probabilities, **5.** Express all the state probabilities in terms of the empty state probability, using the inter-state transition relationships, **6.** Determine the empty state probability by using the fact that all state probabilities always sum to 1.

M/M/1 represents a single server that has unlimited queue capacity and infinite calling population, both arrivals and service are Poisson (or random) processes, meaning the statistical distribution of both the inter-arrival times and the service times follow the exponential distribution. Because of the mathematical nature of the exponential distribution, a number of quite simple relationships can be derived for several performance measures based on knowing the arrival rate and service rate. M/G/1 represents a single server that has unlimited queue capacity and infinite calling population, while the arrival is still Poisson process, meaning the statistical distribution of the inter-arrival times still follow the exponential distribution, the distribution of the service time does not. The distribution of the service time may follow any general statistical distribution, not just exponential. Relationships can still be derived for a (limited) number of performance measures if one knows the arrival rate and the mean and variance of the service rate. However the derivations are generally more complex and difficult. As most of these results rely on some approximation(s) to obtain a closed-form solution, they are not universally applicable.

1. Approximations are reasonably accurate only when the number of servers is comparatively small, typically below 10 or so, which makes them unsuitable for performance analysis of cloud computing data centers.
2. Approximations are very sensitive to the probability distribution of task service times, and they become increasingly inaccurate when the coefficient of variation of the service time, CoV, increases toward and above the value of one.
3. Finally, approximation errors are particularly pronounced when the traffic intensity ρ is small, and/or when both the number of servers m and the CoV of the service time, are large. As a result, the results mentioned above are not directly applicable to performance analysis of cloud computing server farms where one or more of the following holds: the number of servers is huge; the distribution of service times is unknown and does not, in general, follow any of the well-behaved probability distributions such as exponential distribution; finally, the traffic intensity can vary in an extremely wide range.

2.3. The standard quantity of consumers in the system

The inter arrival and inter provision times were assumed already in the other queuing models so that exponential distributions are obtained with parameters λ and μ. Suppose if the arrivals and departures do not follow Poisson distribution then study of other models becomes difficult. But we can derive the formulas of a particular Non-Markovian model **[(M/G/1) : (∞/GD Model)]** where M indicates the number of arrivals in time t which follows a Poisson process , G indicates the General Output Distribution , ∞ indicates waiting space capacity is Infinite , GD indicates General Infinite Descriptive.

Let us assume that the arrivals follow a Poisson process with rate of arrival λ. We also assume that provision times are independently and identically distributed random variables with an arbitrary probability distribution. Let b(t) be the probability density function of provision time T between 2 departures. Let N(t) be the number of consumers in the system at time $t \geq 0$. Let t_n be the time instant at which the nth consumer completes service and departs. Let X_n represents the number of consumers in the system when the nth customer departs. Also, the sequence of random variables { X_n : n=1,2,3,.......} is a Markov chain. Hence we have,

$$X_{n+1} = \begin{cases} X_n - 1 + A, \text{ if } X_n > 0 \text{ i.e. } X_n \geq 1 \\ A \text{ if } X_n = 0 \end{cases}$$

where A is the number of customers arriving during the provision time "T" of the (n+1)th customer.

We know that, if $U(X_n)$ denotes the unit step function , then we can write,

$$U(X_n) = \begin{cases} 1, \text{ if } X_n > 0 \text{ or } X_n \geq 1 \\ 0, \text{ if } X_n = 0 \end{cases}$$

Therefore X_{n+1} can be written as

$$X_{n+1} = X_n - U(X_n) + A \quad \text{...............(1)}$$

Suppose the system is in steady state, then the probability of the number of consumers in the system is independent of time and hence is a constant.

That is, the average size of the system at departure is

$$E(X_{n+1}) = E(X_n)$$

Taking expectation on both sides of (1), we get

$$E(X_{n+1}) = E(X_n - U(X_n) + A)$$

$$E(X_{n+1}) = E(X_n) - E(U(X_n)) + E(A) \quad \text{.............(2)}$$

Since $E(X_{n+1}) = E(X_n)$, we get

$$E(X_n) = E(X_n) - E(U(X_n)) + E(A)$$

$$E(U(X_n)) = E(A) \quad \text{................(3)}$$

Squaring equation(1), we have

$$X^2_{n+1} = (X_n - U(X_n) + A)^2$$

$$= X^2_n + U^2(X_n) + A^2 - 2X_n U(X_n) + 2 A X_n - 2 A U(X_n) \quad \text{.............(4)}$$

But

$$U^2(X_n) = \begin{cases} 1 \text{ if } X^2_n > 0 \\ 0 \text{ if } X^2_n = 0 \end{cases}$$

$$= \begin{cases} 1 \text{ if } X_n > 0 \\ 0 \text{ if } X_n = 0 \end{cases}$$

Therefore X_n denotes the number of consumers and hence X_n cannot be negative.

$U^2(X_n) = U(X_n) \quad [U(X_n) = 1 \text{ or } 0]$

Also,

$X_n U(X_n) = X_n$

Hence (4) becomes

$X^2_{n+1} = X^2_n + U(X_n) + A^2 - 2X_n + 2AX_n - 2AU(X_n)$

i.e.,

$2X_n - 2AX_n = X^2_n - X^2_{n+1} + U(X_n) + A^2 - 2AU(X_n)$

$2X_n(1 - A) = X^2_n - X^2_{n+1} + U(X_n) + A^2 - 2AU(X_n)$

Taking expectation on both sides, we get

$2[E(X_n) - E(AX_n)] = E(X^2_n) - E(X^2_{n+1}) + E(U(X_n)) + E(A^2) - 2E(AU(X_n))$

$2[E(X_n) - E(A)E(X_n)] = E(X^2_n) - E(X^2_{n+1}) + E(U(X_n)) + E(A^2) - 2E(AU(X_n))$

Therefore A and X_n are independent

$2E(X_n)[1 - E(A)] = E(A^2) - E(A^2) + E(A) + E(A^2) - 2E(A)E(A)$

$2E(X_n)[1 - E(A)] = E(A^2) + E(A) - 2[E(A)]^2$

$$E(X_n) = \frac{E(A) - 2[E(A)]^2 + E(A^2)}{2(1 - E(A))} \quad \text{.............(5)}$$

Since the arrivals during "T" is a Poison process with rate λ,

$E(A / T) = \lambda T$

$E(A^2 / T) = \lambda^2 T^2 + \lambda T \qquad \text{...............(6)}$

This is obtained by mean and variance of the poison process,

i.e.,

$E[X(t)] = \lambda t$

$$E[X^2 (t)] = \lambda^2 t^2 + \lambda t$$

Also,

$$E (A) = E (E (A/T))$$

$$= E (\lambda T)$$

$$E(A) = \lambda\, E(T) \qquad ....(7)$$

Similarly,

$$E (A^2) = E (E (A^2 / T))$$

$$= E (\lambda^2 T^2 + \lambda T)$$

$$E (A^2) = \lambda^2 E(T^2) + \lambda\, E(T) \qquad ....(8)$$

Now equation (5) becomes,

$$E (X_n) = \frac{\lambda^2 E(T^2) + \lambda\, E(T) + \lambda\, E(T) - 2[\lambda\, E(T)]^2}{2(1 - \lambda\, E(T))}$$

$$= \frac{\lambda^2 E(T^2) + 2\, \lambda\, E(T) - 2\, \lambda^2\, [E(T)]^2}{2(1 - \lambda\, E(T))}$$

$$= \frac{2\, \lambda\, E(T)\, [\, 1 - \lambda\, E(T)] + \lambda^2 E(T^2)}{2(1 - \lambda\, E(T))}$$

$$E (X_n) = \frac{2\, \lambda\, E(T)\, [\, 1 - \lambda\, E(T)]}{2(1 - \lambda\, E(T))} + \frac{\lambda^2 E(T^2)}{2(1 - \lambda\, E(T))} \qquad ....(9)$$

The standard quantity of consumers in the system is obtained from the given equation. Notice that a multi server system with multiple identical servers has been configured to serve requests from certain application domain. Therefore, we will only focus on standard quantity of consumers in the system and do not consider other sources of delay, such as resource allocation and provision, virtual machine instantiation and deployment, and other overhead in a complex cloud computing environment.

2.4. Waiting Time Distribution

The waiting time of a consumer in the system is obtained with the help of the equation that has been already calculated as standard quantity of consumers in the system.

$$E (X_n) = \frac{2\, \lambda\, E(T)\, [\, 1 - \lambda\, E(T)]}{2(1 - \lambda\, E(T))} + \frac{\lambda^2 E(T^2)}{2(1 - \lambda\, E(T))}$$

$$= \frac{2\, \lambda\, E(T)\, [\, 1 - \lambda\, E(T)]}{2(1 - \lambda\, E(T))\mu} + \frac{\lambda^2 E(T^2)}{2(1 - \lambda\, E(T))\mu}$$

$$E (Xi_n) = \frac{2\, \rho\, E(T)\, [\, 1 - \lambda\, E(T)]}{2(1 - \lambda\, E(T))} + \rho\frac{\lambda\, E(T^2)}{2(1 - \lambda\, E(T))} \qquad ....(10)$$

$$E\ (Xii_n) = \left\{ \frac{2\ \lambda\ E(T)\ [\ 1 - \lambda\ E(T)]}{2(1 - \lambda\ E(T))} + \frac{\lambda^2 E(T^2)}{2(1 - \lambda\ E(T))} - \right\} \rho \quad ....(11)$$

$$E\ (Xiii_n) = \left\{ \frac{2\ \ E(T)\ [\ 1 - \lambda\ E(T)]}{2(1 - \lambda\ E(T))} + \frac{\lambda\ E(T^2)}{2(1 - \lambda\ E(T))} \right\} - (1/\mu) \quad ....(12)$$

With the help of waiting time distribution the delay and the queuing values are obtained by the consumers in the queue those who are waiting for the resources to be provided by the providers.

2.5. Figures and Tables

Table 1. Utility and Delay

M/GD	Utility	Queue	Delay
1000	5.62037	0	0.00634
5000	21.16239	0	0.61171
10000	47.63793	1	5.96816
15000	86.85273	2	26.43108
20000	87.68589	2	29.94669
25000	88.572	2	48.10515
30000	89.55422	2	58.40535
35000	90.09729	3	51.01807
40000	113.27625	3	46.98262
45000	135.6864	4	44.77912
50000	157.6482	4	44.00284

Depending on the file sizes that is allotted in bytes the values are calculated for response time of user, and the users waiting in the queue and the waiting time is calculated. Here we can see clearly that the response time is more when compared with the waiting time.

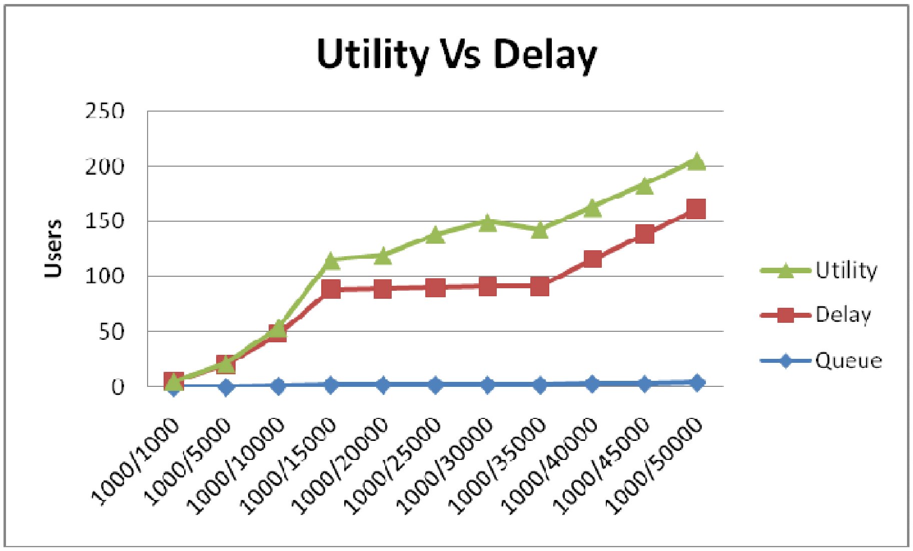

Figure 1. Utility and Delay

3. CONCLUSIONS

Performance assessment of server farms is an imperative aspect of cloud computing which is of decisive curiosity for both cloud providers and cloud customers. In this project we have proposed an analytical model for performance evaluation of a cloud computing data centre. In future, the results can be analysed using simulation. As mean is calculated as well as standard deviation can be computed. The blocking probability and probability of immediate service can be computed.

In future, this methodology can be used to improve the profit of service providers with the help of spot pricing technique. Spot pricing is a very important technique that is used to improve the profit of service providers by consumer satisfaction also.

ACKNOWLEDGEMENTS

None of this work would have been possible without the selfless assistance of a great number of people. I would like to gratefully thank all those members for their valued guidance, time, helpful discussion and contribution to this work.

REFERENCES

[1] W. Kim, "Cloud computing: Today and Tomorrow," Journal of Object Technology, 8, 2009.

[2] Wikipedia, "Cloud Computing," In http://en. wikipedia.org /wiki/Cloud_computing.

[3] L. Vaquero , L. Rodero-Merino , J. Caceres and m. Lindner, "A break in the clouds: towards a cloud definition," ACM SIGCOMM Computer Communication Review, vol. 39, no.1, 2009.

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

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

[6] J. Baker, C. Bond, J. Corbett, J.J. Furman, A. Khorlin, J. Larsonand, J.M. Leon, Y. Li, A. Lloyd, and V. Yushprakh, "Megastore: Providing Scalable, Highly Available Storage for Interactive Services," Proc. Conf. Innovative Data Systems Research (CIDR), pp. 223-234, Jan. 2011.

[7] B. Yang, F. Tan, Y. Dai, and S. Guo, "Performance Evaluation of Cloud Service Considering Fault Recovery," Proc. First Int'l Conf. Cloud Computing (CloudCom '09), pp. 571-576, Dec. 2009.

[8] D. D. Yao, "Refining the diffusion approximation for the M/G/m queue," Operations Research, vol. 33, pp. 1266–1277, 1985.

[9] C. Knessl, B. Matkowsky, Z. Schuss and C. Tier, "Asymptotic analysis of a state-dependent M/G/1 queueing system," SIAM J. Appl. Math. 46 (1986) 483–505.

[10] D.M. Lucantoni , "New results on the single-server queue with a batch Markovian arrival process," Stochastic Models 7, 1–46,1991.

[11] M.B. Comb´e and O.J. Boxma , "BMAP modelling of a correlated queue. In: Network performance modeling and simulation", J. Walrand, K. Bagchi and G.W. Zobrist (eds.) 177–196, 1998.

[12] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Lowpower CMOS digital design," IEEE Journal on Solid-State Circuits, vol. 27, no. 4, pp. 473-484, 1992.

[13] Intel, "Enhanced Intel SpeedStep Technology for the Intel Pentium M Processor" – White Paper, March 2004.

[14] J. Martin, and A. Nilsson, "On service level agreements for IP networks," In Proceedings of the IEEE INFOCOM, June 2002.

[15] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "Theoretical and practical limits of dynamic voltage scaling," Proceedings of the 41st Design Automation Conference, pp. 868-873, 2004.

[16] Y. Simmhan and L. Ramakrishnan, "Comparison of resource platform selection approaches for scientific workflows," 19th ACM Intl. Symp. on High Performance Distributed Computing, HPDC, Chicago, IL, 2010, pp. 445–450.