ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 2, pp: (462-471), Month: April-June 2014, Available at: www.researchpublish.com

OPTIMIZED PERFORMANCE EVALUATIONS OF CLOUD COMPUTING SERVERS

K. Sarathkumar

Computer Science Department, Saveetha School of Engineering
Saveetha University, Chennai

Abstract: **The Cloud computing is one of the emerging modern technologies that need to meet the emerging business needs for agility, flexibility, cost reduction and time-to-value. The developments in cloud computing paradigm necessitate faster and efficient performance evaluation of cloud computing servers. The advanced modeling of Cloud servers are not feasible one due to nature of cloud servers and diversity of user requests. This issue necessitates calculating performance of cloud servers and should be capable of handling several millions of requests in few seconds. It is so possible through the specialized computing facility called Virtualization and even with it, very hard to serve plenty of requests continuously or simultaneously without delay or collision. We proposed a new mechanism to overcome these performance problems and to aim for optimize their performance; we can achieve this goal with help of queuing theory. To tackle the problem of optimizing the performance of cloud servers, we model the cloud servers with multitenant architecture as an M/G/m/m + r queuing system with single task arrivals and a task buffer of finite capacity. In this model focusing on input buffer size, number of servers, mean number of request, response and etc. as metrics for performance. The proposed algorithm named approximate Analytical model, and is equivalent to the combination of Transformation based Analytical Model & an Approximate Markov Chain Model with supporting calculations works out to bring the optimized performance evaluation of cloud servers.**

Keywords: **Cloud Computing, Virtualization, and performance, Queuing theory.**

1. INTRODUCTION

Cloud computing is an emerging commercial infrastructure that promises to eliminate the need for maintaining expensive computing facilities. Cloud computing is the next natural step in the evolution of on-demand information technology services and products.

The cloud environment provides computing infrastructure which aims to shift the location of the computing infrastructure to the network in order to reduce the cost management and maintenance of hardware and software resources.

The need of performance evaluation of this novel paradigm leaves the path for the provision of virtualized resources on demand basis. Cloud computing has a service-oriented architecture in which services are broadly divided into three categories:

Infrastructure-as-a-Service (IaaS) is a computational service model widely applied in the cloud computing paradigm. In this model, virtualization technologies can be used to provide resources to cloud consumers**.[6]**Infrastructure-as-a-Service (IaaS), which includes equipment such as hardware, storage, servers, and networking components are made accessible over the Internet; Platform-as-a-Service (PaaS), which includes hardware and software computing platforms such as virtualized servers, operating systems, and the like; and Software-as-a-Service (SaaS), which includes software

applications and other hosted services[1],These major categories are have six sub categories as Storage as a service (STaaS), Security as a service (SECaaS), Data as a service (DaaS),Test environment as a service(TEaaS), Desktop as a service (DaaS), API as a service (APIaaS). **[1]** Clouds are virtual elements have the potential to provide services to their clients. The cloud computing is a large scale distributed computing paradigm.

The major topics in this area are:

1) Virtualization
2) Resource Time sharing
3) Clouds Consumers (Clients)

The cloud environment gives virtualization effect to its owners and its clients. The benefits include:

- ✓ Economy of scale
- ✓ Assists to supports other resources such as Clusters, Grids, and parallel production environments.

A well versed resource provisioning mechanism that is required for providing computing resources to cloud consumers. A well versed reservation mechanism is required to hold resources on demand. A well versed Pricing mechanism on pay-per-use basis. [2]

1.1 The need of Performance Evaluation:

i) To the successful development of cloud computing paradigm.

ii) To make full study of cloud Servers.

iii) To study the diversity of user requests.

iv) To insist on performance and availability across the entire cloud service delivery chain.

1.1.1 Objectives:

- ❖ To compute performance of cloud servers on Cloud Computing in a cloud environment.
- ❖ To link source and target domains for improving performance of Cloud Servers.
- ❖ To aim that the cloud works its best when source and target domains share same feature space.
- ❖ To measure the impact of cloud infrastructure and configuration on end-user experience under normal and peak conditions.
- ❖ Finally, optimize Cloud application the performance under Cloud servers aiming with reduced cost for its better performance achievement.

2. RELATED WORK

In cloud existing systems are so far Only small portions of performance issues are addressed , Only rigorous analytical approach has been adopted ,The distribution of response time was obtained for a cloud center, This is modeled as classical open network by assuming that both inter arrival times and service times are exponential[2].

By modeling cloud center as M / M / m / m+r queuing system:

- Distribution of response time was obtained.
- Both inter arrival time & service times are assumed to be exponentially distributed.
- The system has finite buffer size that is m + r.

The response time is broken into three periods:

- Waiting time
- Service time
- Execution time

The relationship among,

i) The maximum no. of tasks
ii) The minimum no. of resources
iii) The highest level of services

2.1 Reasoning the results of Existing system

Approximates are accurate when the no. of servers are comparatively small, Approximates are very sensitive and inaccurate when the Co-efficient of Variation (CoV) increases towards & above 1. Approximation errors are seen only when traffic intensity p is small and When both the no. of servers (m) and CoV of service time are large.

2.2 Draw backs using Existing model

- Approximates are accurate when the no. of servers are comparatively small.
- Approximates are very sensitive and inaccurate when the Co-efficient of Variation (CoV) increases towards & above 1.
- Approximation errors are seen only when traffic intensity ρ is small and / or when both the no. of servers (m) and (CoV) of service time are large.
- The System is not useful [2]:
- When no. of servers is huge.
- When the distribution of service time is unknown and does not.
- When the traffic intensity can vary in an extremely wide range.

  On small no. of servers, the result is exact for M/G/m/m+r queuing model when r=0, the result is reasonably accurate in general when r≠ 0 where r is the buffer rate.

The rate of lost tasks remains below the predefined level or specific threshold.

2.3 Problem Statement

- Major problem of classification is lack of persistency of cloud servers due to no optimized performance measurement analysis.
- Expensive amount of time used on connecting cloud server computing and possibility of reduction using Algorithms to gain optimal performance measurements.
- To solve above problem M/G/m/m + r Queuing Systems technique is to be used.
- A Semi supervised learning method is proposed to overcome the above problem by getting knowledge from auxiliary domain that is online (internet).

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 2, pp: (462-471), Month: April-June 2014, Available at: www.researchpublish.com

3. PROPOSED WORK

The proposed model insists on the cloud application that assists on performance evaluation. The proposed cloud computing model is comprised of a **front end** and a **back end**. These two elements are connected through a network, in most cases the Internet. The front end is the vehicle by which the user interacts with the system; the back end is the cloud itself, which contains one or more cloud servers. The front end is composed of a client computer, or the computer network of an enterprise, and the applications used to access the cloud. The back end provides will provide the required services in a required manner.

Cloud computing architecture can able to diverse in implementation both for public or private cloud computing. The commonly accepted cloud style by the users of public cloud computing will spread in different places geographically **[9].**

The proposed system architecture uses the multitenant architecture is essentially a shared utility, giving it massive economies of scale to optimize computing resources. The small number of servers with the following performance goals:

- Energy efficient servers
- Optimized runtime processing
- Optimized storage
- Predictable load balancing
- Continual analysis and energy improvement
- Micro-energy management
- Optimized power consumption
- Standardized architecture

will provide means for optimizing the cloud servers. However, the performance of cloud servers are based the number of metrics taken into account and its evaluated results under steadystate.

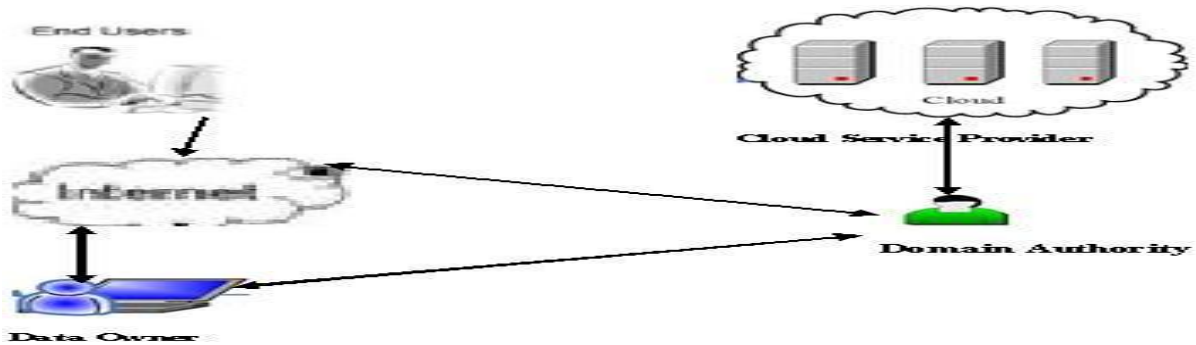

Fig 1 : Cloud's Multitenant architecture

3.1 Major objectives of proposed model are:

- To obtain a complete probability distribution of response time & the number of tasks in the system
- To do comparative study of them with the aim that performance of
- To calculate probability of immediate service (no waiting in the input buffer).
- To calculate task blocking probability.
- To determine the size of the buffer (needed that the blocking probability should remain below a predefined level).

The proposed system will indicate that the interarrival time of requests is exponentially distributed, while task service times are independent and identically distributed random variables that follow a general distribution with mean value of μ, where μ is service time (1 / μ is the mean service time). In a Cloud-specific evaluation, an attractive promise of clouds is that they can always provide resources on demand, without additional waiting time **[7].**

3.2 Resolving Methods

- To compute performance of cloud servers on Cloud Computing in a cloud environment.
- To link source and target domains for improving performance of Cloud Servers.
- To aim that the cloud works its best when source and target domains share same feature space.
- Finally, optimize the performance of Cloud servers for its better performance achievement.

3.3 Proposed Method

- To tackle the problem of optimizing the performance of cloud servers, we model the cloud center as an M/G/m/m + r queuing system with single task arrivals and a task buffer of finite capacity.**[8]**
- To bridge the gap using a well supervised learning algorithm called M/G/m/m + r Queuing Systems used for optimizing the performance of clod servers **[2].**
- The proposed system is to be modeled as M/G/m/m+r / n queuing system with single task arrivals & a task buffer of finite capacity **[2].**

  The dynamic nature of Cloud environment needs virtual sharing of services from its origin. That is, cloud service provider and used by low level authorities. It is illustrated as in the figure 2:

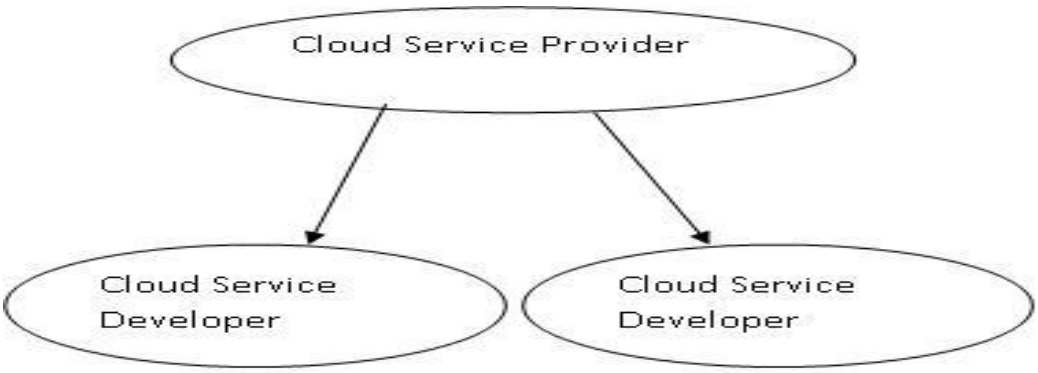

Fig. 2: Basic Relationship of Cloud Services

4. WORKING MODEL

The performance evaluation makes use of the probability distribution of major performance metrics such as no. of jobs, waiting time, response time. The major combination of algorithms such as Transformation based Analytical Model and an Approximate Markov Chain Model brings the optimizing solutions under the steady state of respective cloud computing resource provisioning mechanisms. The performance evaluation is performed which can reveal the importance of optimal computing resource provisioning **[6].**

The proposed working model requires the concrete Queuing Model with possibly applying the proposed algorithm called an approximate Analytical model with single task arrivals & a task buffer of finite capacity **[2].** The figure 3 illustrates the proposed system model:

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 2, pp: (462-471), Month: April-June 2014, Available at: www.researchpublish.com

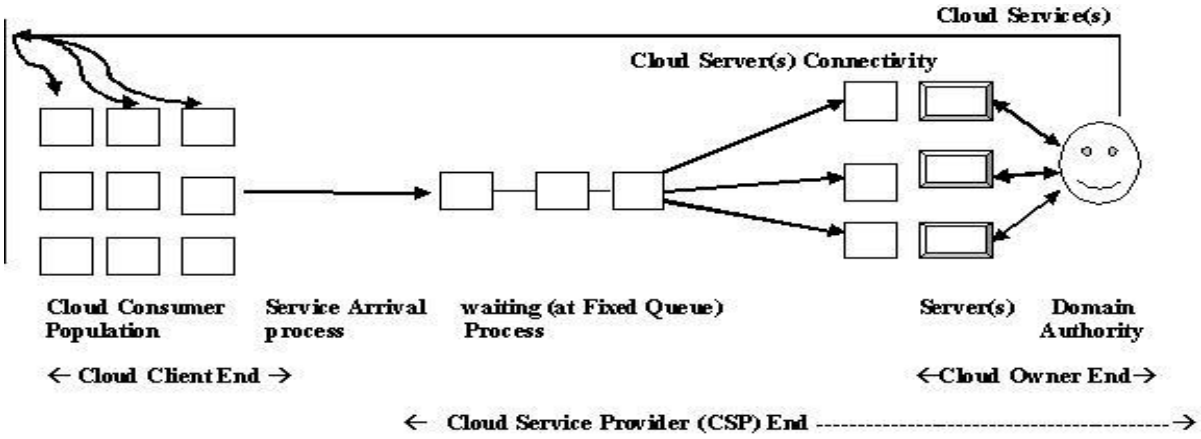

Figure 3: M/G/m/m+r/n/d Queuing Model based Cloud System Model

4.1 Transformation based Analytical Model

The transform based analytical model does the following major calculations:

No. of jobs in the system:

P {no. of jobs in the system} $p_0 = 1 – p$

An approximate Analytical model, and is equivalent to

The combination of Transformation based Analytical Model &an Approximate Markov Chain Model

P {n no. of jobs in the system} $p_n = p^n (1-p)$

P {No. of jobs >= n} p^n

Mean Queue length $N = E(n) = p /(1 – p)$

Variance of Queue length $Var (n) = p / (1-p)^2$

No. of waiting jobs in the queue:

P {No. of waiting jobs is k} $p (n_q = k) = 1 – p$

$$P\ k+1\ (1 – p)$$

Where $k > 0$

Mean no. of waiting jobs ($Nq = N –p$) $=E (nq) = p2 / (1-p)$

Response Time

Mean response Time R is $E (r) (1 / \mu) (1 - p) = 1 (\mu - \lambda)$

Variance of response Time r is $Var (r)= E(r)^2$

Where λ is the arrival time, μ is the service time

(1 /λ - mean arrival time , 1 / μ - mean service time)

P {response Time < = t} r(t) = 1 - p $e^{-(\mu - \lambda) t}$

Waiting Time

P {waiting time < = t} w (t) = 1 - p $e^{-(\mu - \lambda) t}$

Mean waiting Time E (w) = p [(1/μ) / (1 - p)

4.2 An Approximate Markov Chain Model

It is for the moments of task request arrivals and is selected as Markov points.

It observes the state at which system has an arrival.

It has steady-state solution, that is, on the average case a single departure between every two successive arrivals.

Using the proposed system, it is possible to calculate the distribution of no. of tasks in the system as well as the mean responsible time.

Using the Embedded Markov chain, it is possible to calculate the transition probabilities associated with it.

It calculates the no of tasks that are serviced during the interval between two successive task arrivals.

Where M is the inter arrival time distribution and also called the Markovian (passion, exponential) distribution, G is the service time distribution or General distribution, m is the no. of servers, m+r is the total buffer size required or system capacity, n is the calling population

To do so, the recommended method called the transition probability matrix p with limited elements which is equivalent to the n, with it is to be derived.

4.3 Transition Probabilities for the Embedded Markov Chain

- Inter arrival times are IID according to A(t) with mean time being 1 / λ
- Service times are IID and is exponentially distributed with mean 1/μ
- Instead of keeping track how long since the past arrival occurs, look at the arrival instants, which form an imbedded Markov chain.
- Let q0 n be the number of customers in the system immediately prior to the arrival of customer Cn.
- let v0 n+1 be the number of customers served during the arrival of Cn and Cn+1. We have
- q0
- n+1 = q0
- n + 1 = v0
- n+1: (1)
- Now we need to find the transition probabilities of this embedded Markov chain.

Supporting calculations

The supporting calculations are need to optimize the cloud server's performance evaluation towards it's enhancements.

The other major calculations are:

- Distribution of no. of tasks in the system
- Uses Probability Generating Functions (PGFs) and it calculates the no. of tasks in the system at the time of task arrival.
- Distribution of Waiting time and Response time
- It measures the Waiting time W in the steady state and is equivalent to Q, the queue length.
- Probability of immediate service
- It measures the no. of ideal servers at the time of task arrival. If it is one, it is equivalent to service time
- Blocking Probability and Buffer size

To keep the blocking probability below the threshold value (€), it is recommended to use minimum buffer size (g).

4.4 Experimental Setup

The proposed system model identifies the Oracle system 10g as Database server, VM Ware as Virtualization Machine(s) set up and introduces Cloud servers (both can act together as Cloud Backend). Any client software such as PHP with Apache with web browser can act as Cloud Front end. The Oracle Performance & Tuning Tool called AWR (Automated Workload Report) is used as performance evaluation tool.

The performance metrics and its specifications used are as follows:

Performance Metrics Specification	**Queuing Method Formulae**
A = The number of arrivals	U = (S λ) / M
T = Specific Time period	λ = A/T
U = CPU Utilization	R=S+Q (or) R=S/(1-U)
S = Service time	Q=R-S trx/s
M =No. of CPUs	
□ □ λ = □ Arrival rate	
U = CPU Utilization	
R = Response time	
S = Service time	
Q = Wait time in the Queue	

The results as an AWR (Automated Workload Report) were recorded.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 2, pp: (462-471), Month: April-June 2014, Available at: www.researchpublish.com

4.5 Experimental Solution Validations

- The Optimized results obtained through the proposed system can enhance the performance of the cloud servers.
- The queuing model – M/G/m/m+r used in designed using FCFS (First Come First Serve) assists unlimited clients to receive the better service.
- The transform based model adopted accelerates the QoS (Quality of Service) and promises SLA (Service Level Agreements) factors to its clients.

User Call (trx/s)	Elapsed time	Time Period (Sec)	Service Time trx/s (μ)	M (No of CPU)
	E	T =E *60	S = μ	NUM_CPUS
3.39	12.29	737.4	0.78	1
2.810	72.260	4335.6	0.73	1
0.650	124.390	7463.4	0.15	1
2.740	55.420	3325.2	0.58	1
2.680	60.550	3633	0.45	1
2.990	59.530	3571.8	0.51	1
2.980	60.070	3604.2	0.6	1
2.750	60.570	3634.2	0.51	1

Lambda	Utilization	CPU Utilization %
λ=Arrival Rate	U=(S*λ/M)	
0.00459723	0.0035858421	0.358584
0.00064812	0.0004731294	0.047313
0.00008709	0.0000130638	0.001306
0.00082401	0.0004779261	0.047793
0.00073768	0.0003319571	0.033196
0.00083711	0.0004269276	0.042693
0.00082681	0.0004960879	0.049609
0.00075670	0.0003859171	0.038592

Response Time	Queue Waiting Time
0.78280702	0.002807
0.73034555	0.000346
0.150002	0.000002
0.58027733	0.000277
0.45014943	0.000149
0.51021783	0.000218
0.6002978	0.000298
0.51019689	0.000197

Table 1: Experimental Query Evaluation Results (under Oracle 10g - Performance Tuning)

By referring the table, the cloud service provider in a real time cloud environment, holds the key go with request(s) and to denial of request(s) from cloud consumers.

The real time data collected and evaluated clearly shows that the optimized performance evaluation is basically based on the major influencing factors for performance evaluation are number of CPUs, Waiting Time, and Response Time.

5. CONCLUSION AND FUTURE WORK

The optimistic performance is the vital calculation required for modern cloud farm called cloud servers

In this paper, we compute and study optimized performance measurements of cloud servers:

Obtain an accurate estimation of the complete probability distribution of the request-response time & measure other important indicators.

Also studies the relationship between the number of servers and its input buffer sizes & performance indicators such as Mean no. of tasks in the system, Task Blocking Probability, The probability that a task will obtain immediate service.

As part of future work, the Tasks are to be divided into sub tasks. The performance measurements are updated. The response time is to be divided into several components such as set up time, execution time, return time, clean up time are to be combined and calculated to enhance the performance of cloud servers.

The service distribution is to be measured using the enhanced Queuing model M /G / m / m+r / n / d, where d is the service discipline assumed.

So, that the accuracy of the performance of cloud servers to be measured and optimized, for the parallel distinct homogeneous cloud servers.

REFERENCES

[1] B. Furht, "Cloud Computing Fundamentals," Handbook of Cloud Computing, pp. 3-19, Springer, 2010.

[2] HamzehKhazaei, Student Member, IEEE, JelenaMisic, Senior Member, IEEE, andVojislav B. Misic, Senior Member, IEEE "Performance Analysis of Cloud Computing Centers Using M/G/m/m + r Queuing Systems", Vol. 23, NO. 5, May 2012

[3] Paul Marshall, Kate Keahey and Tim Freeman , "Improving Utilization of Infrastructure Clouds" IEEE/ACM Cloud Computing May 2011

[4] Karim ABBAS and Djamil A¨ISSANI, " Approximation in an *M.G/*1 queuing system with breakdowns and repairs", Laboratory of Modelization and Optimization of Systems University of Bejaia, 06000(Algeria)

[5] P. Hokstad, "Approximations for the M/G/m Queues," Operations Research, vol. 26, pp. 510-523, 1978.

[6] SivadonChaisiri, Student Member, IEEE, Bu-Sung Lee, Member, IEEE, and DusitNiyato, Member, IEEE "Optimization of Resource Provisioning Cost in Cloud Computing" Vol. 5, No. 2, April-June 2012

[7] AlexandruIosup, Member, IEEE, Simon Ostermann, M. NezihYigitbasi, Member, IEEE,RaduProdan, Member, IEEE, Thomas Fahringer, Member, IEEE, and Dick H.J. Epema, Member, IEEE "Performance Analysis of Cloud ComputingServices for Many-Tasks Scientific Computing "

[8] J.M. Smith, "M/G/c/K Blocking Probability Models and System Performance," Performance Evaluation, vol. 52, pp. 237-267, May 2003.

[9] SinungSuakanto, Suhono H Supangkat, Suhardi and RoberdSaragih, "PERFORMANCE MEASUREMENT OF CLOUD COMPUTING SERVICES" IJCCSA,Vol.2, No.2, April 2012