PERFORMANCE ANALYSIS OF CLOUD COMPUTING CENTERS BY BREAKING-DOWN RESPONSE TIME

1PREETI KAMBLE, 2HEMLATA CHANNE

1,2Department of Computer Engineering, Pune Institute of Computer Technology, Pune, Maharashtra (India)
Email: preetikamble29@gmail.com, hemlata.channe@gmail.com

Abstract— Cloud computing is the emerging field of the computer science and it is occupying the most of the areas in computers. It delivers on the emerging business essentials for agile, flexible and time-to-value performance. The performance and availability of cloud applications has a noticeable impact on user adoption and revenue of the cloud. The work on the performance analysis of the cloud using M/G/m/m+r queuing system till date gives the novel and approximate analytical model. It gives the relationship between the input buffer size and number of servers available. It also gives the performance indicators like mean number of tasks in the system, task blocking probability and immediate service probability. In existing model, the performance can be improved by breaking down the response time in the setup, execution, return and clean up time. In this paper we are proposing a solution for this constraint. This approach also will give the all performance indicators like above model.

Keywords— cloud computing, performance analysis, response time, queuing theory, embedded Markov chain, semi-Markov process

I. INTRODUCTION

Cloud computing has taken the world of enterprise applications of information technology by storm. Almost every enterprise of any size now relies on cloud computing in some or the other way for meeting its targeted revenue and synchronization of work across all its work locations. It is the novel paradigm for provisioning computing infrastructure which aims to shift the location of the computing infrastructure to the network for reducing the maintenance and management cost of software and hardware resources [1].

According to National Institute of Standards and Technology (NIST), the cloud computing is defined as “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models [2].

Cloud computing is the service-oriented architecture and services are divided into main three categories. Software-as-a-Service (SaaS) is a software delivery model in which software and associated data are centrally hosted on the cloud. Platform-as-a Service (PaaS) provides a computing platform like execution runtime, database, web servers, development tools etc. In infrastructure-as-a-Service (IaaS), cloud providers offer computers, as physical or more often as virtual machines, servers, storage, load balancers, networks etc. [3].

We assume that in the cloud center, there is are facility nodes like web servers, database servers, directory servers and others, and after finishing the service, the task leaves the center. A Service Level Agreement (SLA) gives all the aspects of usage of cloud service and the obligations of service providers and clients. It also includes various descriptors known as Quality of Service (QoS) which includes availability, reliability, throughput, security, performance indicators like task blocking probability, immediate service probability, mean number of tasks in the system and may more [4].

A cloud center can have large number of facility (server) nodes [5]. In cloud, coefficient of variation of task service time may be high and cloud centers can have diversity of user’s requests [6]. These aspects are addressed in existing research, there is virtually no work that addresses all of them simultaneously. To fill this gap, there is a modeling of cloud center as an M/G/m/m+r queuing system [1]. It is with a single task arrivals and a task buffer of finite capacity. If we want to extend the above defined model then we have to break down the response time in the setup, execution, return and clean up time. In this paper, we are going to give the idea about breaking down the response time in the mentioned parts. We are using cloud simulation software CloudSim [25] for validating our proposed approach.

The remainder of the paper is organized as follows. Section 2 gives an overview of existing work on performance evaluation of cloud and characterization of M/G/m/m+r queuing system. Section 3 discusses basics of the M/G/m/m+r queuing system and our proposed approach over existing approach. In section 4, we have shown the cloud simulation done using CloudSim. We are using this simulation for obtaining our results. For mathematical graphs and results presentation, we are using Maple 13 software by MapleSoft, Inc.[7]. In section 5, we
are presenting the future work which can be stated as the remaining work about this approach.

II. RELATED WORK

Many researchers are attracted in the field of cloud computing, but there is relatively less focus on performance analysis of the cloud computing. This performance analysis field can give many smooth computing solutions in future. In [8], assuming interarrival and service times as exponential, the cloud is modeled as the classical open network, and its response time distribution was obtained. Using this distribution, the relationship between maximum number of tasks, highest level of service and minimum number of service resources was found.

The scenario where the inter arrival time and/or service time are not exponential, the analysis is more complex. On extensive research of M/G/m queuing system, most of the theoretical analyses are relied [9], [10], [11]. As probability distribution of queue length and response time in M/G/m and M/G/m/m+r queuing system can’t be obtained in closed form, this necessitated the search for a suitable approximation.

An approximate solution for steady state distribution of queue length in M/G/m queueing system assuming finite waiting space was described in [12]. As the approximations were given in explicit form, its numerical computation is easier than that of given in [13], [14].

A similar approach with respect to M/G/m queues was explained in [15]. In this approach, the extension of the analysis is done to approximate the task blocking probability and, determines the smallest buffer size i.e. the rate of lost tasks remains below a predefined level. The results of this show that the optimal buffer size is directly proportional to the order of convexity of service time.

In [16], another approximation technique was explained for blocking probability. The approach is based on the exact solution for the finite capacity M/G/m/m+r queueing system. Estimation of the task blocking probability helps to guide the buffer allocation. In [17], it was explained that average delay in queueing in M/G/m/m+r queueing system was based on the relationship of joint distribution of remaining service time to the equilibrium service distribution.

Most of these results are not universally applicable because they rely on some approximations. As approximations are reasonably accurate when there is small number of servers, it can’t give more accurate results in performance analysis of clouds. Approximations are very sensitive and dependant for coefficient of variance, CoV, and errors are pronounced when the traffic intensity ρ is small and/or number of servers m, CoV of service time are large [18], [19], [20].

In [1], Hamzeh Khazaei, Jelena Misic and Vojislav B. Misic, proposed the analytical technique based on approximate Markov chain model for performance analysis of cloud computing center. The assumption of general service time for request and large number of servers made this model flexible and scalable. This model can be extended by breaking down response time in setup, execution, return and response time. And this approach we are elaborating in our paper.

III. A PROPOSED SYSTEM

As stated earlier we are providing the new approach for performance analysis of the cloud computing centers by modeling the cloud as M/G/m/m+r queueing system.

- The queuing systems constitute a central tool for modeling 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 [21].

- An M/G/m queueing system model is a queue model where arrivals are Markovian (modulated by a Poisson process), service times have a General distribution and there is m number of servers. Poisson process means a stochastic process which counts the number of events and the time that these events occur in a given time interval. The time between each couple of successive events has an exponential distribution with parameter λ and each of these inter-arrival times is assumed to be independent of other inter-arrival times [22]. A stochastic process (random process) is a collection of random variables and it is often used to represent the evolution of some random value or system, over time [23].

- An M/G/m/m+r queueing system is the extension of the M/G/m queueing system. It adds the r as the finite buffer size in the system. Hence the capacity of the system is m+r [1].

In proposed approach we are modeling the cloud server farm as an M/G/m/m+r queueing system, hence the inter arrival time of requests is exponentially distributed and task service times are independent. The identically distributed random variables follow the general distribution with mean value of μ. As the population size of a typical cloud center is inversely proportional to the probability that a given user will request service, the arrival process is modeled as a Markovian Process [24]. But in this paper we have done simulations according to simple queuing system, and in future work we are planning to do precise implementation of M/G/m/m+r queueing system by breaking down response time.
Fig. 1. Embedded Markov Chains (Source:[1])

The basic structure of the embedded Markov chain is shown in Fig. 1. The semi-Markov process records the state at time and embedded Markov chain only observes the state at which the system receives an arrival [1], [3]. This different behaviors can be observed in Fig. 2 and Fig. 3.

Fig. 2. Semi-Markov Process (Point represents the observation of the state of the system)

Fig. 3. Embedded Markov Chains (Point represents the observation of the state of the system)

Mathematical Model:

This system can be modeled mathematically by the following way.

\[ S = \{ I_p, F_n, O_p, S_c, F_l \} \]

Where

\[ I_p = \text{Input to the system} \]
\[ = \text{Requests from the clients or tasks for the servers. We are assuming 1 client for 1 request.} \]
\[ F_n = \text{Operations for serving the request in cloud center.} \]

\[ = (c_1, c_2, c_3, ..., c_n) \]

\[ O_p = \text{Operation for serving the request in cloud center. The performance parameters like task blocking probability, immediate service probability and departure probability are decided and calculated in this phase.} \]

\[ = \{ \text{initialize\_cloud(), start\_datacenter(), start\_broker(), create\_vm\_in\_datacenter(), create\_host(), send\_cloudlet\_to\_vm(), execute\_cloudlet\_req(), recv\_cloudlet\_after\_exec(), destroy\_vm(), shutdown\_broker(), shutdown\_datacenter(), print\_output() } \]

Mapping of datacenter and host:

Many-to-many mapping

Mapping of virtual machine and datacenter:

Many-to-many mapping
Op = Output of the server
= It consists of leaving user after serving it completely. It can be stated as the various parameters like arrival time of the request, end time of the request i.e. when that request get fully served and that client leaves the system. It also gives that how much debit that client should pay for using cloud according to type and energy required to process the request.
= (start_time, end_time, debt, user_id, cloudlet_id, status, datacenter_id, VM_id, execution_time),
Sc = Success of the system in serving request. It shows the status value in the output as ‘SUCCESS’
Fl = Failure of the system in serving the request. It may be caused due to performance parameters stated in operations. It shows the status value in the output as ‘FAILURE’

Algorithm:
The following algorithm is the algorithm for simulating cloud and analyzing performance of the cloud by existing formulae.

Abbreviations:
• Data Centers : DC
• Brokers : B
• Virtual Machines : VM
• System : S
• Cloud Simulation Packages and Libraries: CSPL
• Host : H
• Cloudlets : C_let
• Cloudlet Specifications : C_let_spec
• Task Blocking Probability: TSB
• Departure Probability : DP
• Immediate Service Probability : ISP

Algorithm: Cloud Simulation and Breaking-down Response Time
Input : No. of DC, No. of H, No. of C_let, C_let_spec.
Output: debt w. r. t datacenter and user_id, start_time, end_time, status, TBP, ISP, DP

Initialize CSPL
Create DC & B with all characteristics.
Define properties of VM and C_let.
Create VM in DC with H
while(error) do
    Create VM in another DC with H
end while
Model S -> M/G/m+m+queuing model
while( No. of C_let!=0) do
    Send C_let to VM
    Do setup for C_let processing
    Execute C_let
    Return the results for C_let
    Clean up data of C_let
    C_let--;
end while

Calculate debt w. r. t DC and user_id
Calculate performance parameters from results

IV. CONCLUSION AND FUTURE WORK

From this we can state that the complex cloud can be formed and analysis of cloud can be done by analyzing the results on the cloud. In this paper, we have studied the related work in the field of the performance analysis of the cloud computing centers. We have modeled the problem mathematically and studied the mapping of the functions (here, can be called as entities) in the system. We have proposed the algorithm to accomplish the desired solution.

In future work, we are planning to make more complex cloud simulation in which there will be many arrivals of requests and by observing the request serving method of the cloud and the results, we will analyze cloud performance parameters like departure probabilities, immediate service probability, and task blocking probability. We are also planning to extend this simulation by modeling the cloud system in M/G/m/m+r queuing system.

REFERENCES

[2] NIST document for definition of cloud computing
Performance Analysis Of Cloud Computing Centers By Breaking-Down Response Time
