Chapter III

Resource Allocation in CCN with Poisson Arrivals
CHAPTER 3

RESOURCE ALLOCATION IN CCN WITH POISSON ARRIVALS

3.1 Introduction

In this chapter a cloud computing network model with the objective to study the dynamic resource allocation problem in cloud computing is analyzed. Every successful servicing of cloud paradigm need an optimal resource allocation in turn to handle the problem of Bag of Tasks (BoTs). It is assumed that the allocation of BoTs are done by two stages of process namely classification according to service level agreement (SLA) of BoTs and service providing (SaaS, PaaS and IaaS). After classification the request is routed to any one of the service providers with corresponding probability. Thus the Cloud Computing Networks (CCN) become a general Open Jackson Queuing Network System and performance measures are obtained to study the efficiency of the CCN.

Cloud Computing enables the massive scale resource sharing, which allows users to access technology enabled service without the knowledge of the system. It also refers to the provision of computational resources on demand via a computer network. In the modern competitive business environment, providing Quality of Service (QoS) is the prime study for any service provider. With a high exposition of technological innovation and developments the Cloud Computing is making evolutionary changes in the modern era. The dynamic allocation of resources has
emerged as a promising technology to provide cost effectiveness with high performance cloud computing system for solving many complex problems in commercial application [46].

Resource allocation in a cloud computing environment can be modeled as allocating the required amount of multiple types of resources simultaneously from a common resource pool for a certain period of time for each request [36]. Classification of the job request and route them to the right server is the foremost work in resource allocation. There are three kinds of cloud services model, namely, Software as a Service (SaaS), Platform as a Service (PaaS) and Cloud Infrastructure as a Service (IaaS) [44]. The above three functions are not exactly bifurcated but they are inter related in nature.

*Software-as-a-Service (SaaS)* is a software distribution model in which applications are accessible through a single interface, like a web browser over the Internet. Users do not have to consider the underlying cloud infrastructure including servers, storage, platforms, etc.

*Platform-as-a-Service (PaaS)* provides a high level of integrated applications that control of distributed applications and their hosting environment configurations. In general, developers accept all instructions on the type of software that can be written to change built-in scalability.
Infrastructure-as-a-Service (IaaS) provides users with computation processing, storage, networks and computing resources. Also, IaaS sends programs and related data, while the cloud provider does the computation processing and returns the result [46].

Resource allocation in cloud computing is still a challenging issue. Due to existence of different workload types with various requirements that should be supported by cloud computing, no any single hardware or software solution can allocate resources to all imaginable types efficiently. Also, each type has its specific nature properties and a single solution cannot deal with in that regard optimally. Thus, it is a need to provide specific solutions for different workload types such as bag of tasks (BoTs) and message passing applications, and provision resources in such a way that customers be able to just concentrate on demanded requests [33]. The main advantage of having multiple servers in data center is, the increment in performance by reducing the mean queue length and waiting time (response time) than compared to the traditional approach of having only single server [50].

In this chapter, the first model is the dynamic resource allocation of bag of task to cloud computing center with Poisson type arrival process and exponential service time of tasks. Each stations has an M/M/s_i queue system. In this novel model, it is solved analytically to obtain important performance measures, like mean number of tasks and mean number of servers in the system. In
the second model, it is studied $M/G/s_1$ queueing system in the classification node of the CCN. The arrival process is assumed to be Poisson and the service time has general distribution. The researcher has employed the resource allocation techniques to minimize the resource cost and minimize the service response time. The researcher has modeled the cloud center as $M/G/s$ queuing system with single task arrivals and a task buffer of infinite capacity. Its performance measures are evaluated using a combination of a transform-based analytical method and an approximate Markov Chain technique, which allows us to obtain a complete probability distribution of response time and a number of tasks in the system.
3.2 Model: A - Dynamic Resource Allocation in CCN with Poisson Arrivals

3.2.1 Model Description

Consider a cloud computing network (CCN) which provides resources ranges from computing infrastructure and applications. The inter-arrival time of job requests to the classifier node is assumed to follow a Poisson process with parameter $\lambda_1 > 0$ and the task service times are also exponentially distributed with parameter $\mu_1 > 0$. Depending on the type of clients’ request and their SLA, three types of services are provided, namely Software (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). The cloud computing network diagram is described in figure 3.1. The bag of tasks arriving at the first station namely ‘Classifier’, are processed based on Poisson process and are directed to any one of the three service stations. The BoTs are taken for classification in FCFS discipline. After classification the BoTs move to any one of the stations which provide SaaS, PaaS and IaaS. Each station i has $s_i$ independent servers.

Figure 3.1 Cloud Computing Network
3.2.2 Analysis

The Cloud Computing Network, is modeled as an Open Jackson Queueing Network. Consider the general CCN with the following assumptions.

- The network has N single station with $s_i$ servers at each station.
- There is an unlimited waiting space at each station (the classification and service stations).
- The customers (BoTs request) arrive at station $i$ from outside according to a Poisson process with parameter $\lambda_i$ ($i = 1, 2, \ldots, N$) and $\lambda_i > 0$.
- All arrival processes are independent of each other.
- Service times for customers (service requests) at station $i$ are independent and identically distributed (iid) with exponential random variables with parameters $\mu_i$ ($i = 1, 2, \ldots, N$)
- Customers (service requests) finishing service at station $i$ join the queue at station $j$ with probability $p_{ij}$ or leave the network altogether with probability $r_i$, independently of each other.

The probabilities $p_{ij}$, $i, j \in S = \{1, 2, \ldots, N\}$ are called the routing probabilities and the matrix $P = (p_{ij})$ $i, j \in S$ is called the routing probability matrix. By our assumptions, the stochastic model of cloud computing network, becomes an Open Jackson Queuing Network with N stations and $s_i$ servers at each station and the system states are ergodic [29].
The routing matrix $P$ can be expressed as a transition probability matrix of the form

$$
P = \begin{bmatrix}
P_{11} & P_{12} & P_{13} & \cdots & P_{1N} \\
P_{21} & P_{22} & P_{23} & \cdots & P_{2N} \\
P_{31} & P_{32} & P_{33} & \cdots & P_{3N} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
P_{N1} & P_{N2} & P_{N3} & \cdots & P_{NN}
\end{bmatrix}
$$

with the condition $\sum_{j=1}^{N} P_{ij} + r_i = 1, \quad 1 \leq i \leq N$.

It is assumed in the CCN that each station has infinite capacity for waiting requests or jobs. This will lead to a new problem of stability analysis. Next we have to show that the CCN is stable in the long run. The next theorem proves the stability criteria for the proposed network.

**Theorem: 3.1**

The CCN, with external arrival rate vector $\overline{\lambda}$ and routing matrix $P$, is stable if the matrix $I - P$ is invertible and $a_i < s_i \mu_i$, for all $i = 1, 2, \ldots, N$,

$$
\text{where } \overline{a} = [a_1, a_2, a_3, \ldots, a_N], \text{ with } a_i = \lambda_i + \sum_{j=1}^{N} a_j p_{ji} \quad i = 1, 2, 3, \ldots, N.
$$
Proof:

Let $a_i$ denote the net arrival rate to the $i^{th}$ node in the CCN. Then

$$a_i = \lambda_i + \sum_{j=1}^{N} a_j p_{ji}, \quad i = 1, 2, 3, \ldots, N,$$

called the traffic equation. Since $I - P$ is invertible, the traffic equation has a unique solution $\bar{a} = \bar{\lambda} (I - P)^{-1}$. Here the $i^{th}$ service station has single queue having $s_i$ servers with arrival rate $a_i$ and mean service times $1/s_i \mu_i, \quad (i = 1, 2, \ldots, N)$.

By the stability theorem of M/M/s queues it follow that the CCN is stable if $a_i < s_i \mu_i, \quad i = 1, 2, \ldots, N$. Hence the theorem $\Box$

This proves that our CCN model for Cloud Computing becomes Open Jackson Network and is also stable in the long run.

This will induce us to compute the following performance measures:

1. Mean number of request waits in the network.
2. Probability that the network is busy.

3.2.3 Steady State Analysis

Consider the CCN with 4 stations namely classification, SaaS, PaaS and IaaS. The limiting behavior of the system in steady state can be studied as follows. Let $X_i(t)$ be the number of requests (BoTs) in the $i^{th}$ station $i = 1, 2, 3, 4$ at time $t$.

The state of the system at time $t$ be denoted as $X(t) = (X_1(t), X_2(t), X_3(t), X_4(t))$. 

62
Suppose that the CCN is stable, with the unique solution $\mathbf{a}$ to the traffic equation,

$$a_j = \lambda_j + \sum_{i=1}^{4} a_i P_{ij}.$$  
We can define the joint probability distribution in the long run as follows:

Let $p(n_1, n_2, n_3, n_4) = \lim_{t \to \infty} p_r \left\{ X_1(t)=n_1, X_2(t)=n_2, X_3(t)=n_3, X_4(t)=n_4 \right\} = \prod_{i=1}^{4} p_i(n_i)$.

Since the stations are independent and $p_i(n_i)$ denotes the marginal probability that $X_i(t) = n_i$, that is there are $n_i$ request wait in the queue $M/M/s_i$, where $i^{\text{th}}$ station’s arrival rate $a_i$ and service rate $\mu_i$.

From limiting behavior of the queue $M/M/s_i$, we have

$$p_i(n) = p_i(0) \rho_i(n) \quad i = 1, 2, 3, 4.$$

where

$$p_i(0) = \sum_{n=0}^{s_i-1} \frac{1}{n!} \left( \frac{a_i}{\mu_i} \right)^n + \left( \frac{a_i}{\mu_i} \right)^{s_i} \left( \frac{1}{1 - \frac{a_i}{s_i \mu_i}} \right)^{-1}$$

and

$$\rho_i(n) = \begin{cases} 
\frac{1}{n!} \left( \frac{a_i}{\mu_i} \right)^n & \text{if } 0 \leq n \leq s_i - 1 \\
\frac{s_i^{-s_i}}{s_i!} \left( \frac{a_i}{s_i \mu_i} \right)^n & \text{if } n \geq s_i.
\end{cases}$$
Next we obtain the product form solution for the distribution of request in the CCN [48].

**Theorem: 3.2**

The limiting behavior (steady state solution) of the simple CCN is given by

\[ p(n_1, n_2, n_3, n_4) = p_1(n_1)p_2(n_2)p_3(n_3)p_4(n_4), \text{ for } n_i = 0, 1, 2, \ldots \text{ and } i = 1, 2, 3, 4. \]

**Proof:**

Since \( \{X(t); t \geq 0\} \) is a four dimensional Continuous Time Markov Chain (CTMC) with state space \( E = \mathbb{N}_0^x \mathbb{N}_0^y \mathbb{N}_0^z \mathbb{N}_0^w \), where \( \mathbb{N}_0 = \{0, 1, 2, 3\ldots\} \) and the limiting distribution obtained satisfy the balance equation, the product form solution of distribution of request in CCN is obtained easily \( \square \)

**3.2.4 System Performance Measures**

As we obtained the steady state probabilities for the number of customers in each of the stations as a product form solution, we are able to find the mean number of BoTs waiting. The following system performance measures are given by,

1. Mean number of requests waits in \( i^{th} \) station \( (L_i) \)

\[
L_i = \sum_{n=0}^{\infty} n p_i(n); i=1,2,3,4.
\]

\[
= \frac{\rho_i}{1-\rho_i}, i=1,2,3,4 \quad \text{where} \quad \rho_i = \frac{\lambda_i}{\mu_i},
\]

64
2. The probability that the CCN system is busy

\[ P = \prod_{i=1}^{4} (\rho_i), \quad i = 1, 2, 3, 4. \]

### 3.2.5 Cost Analysis

Let \( C_1 \) be the cost of waiting per unit time and \( C_2 \) be the service cost incurred per unit time. The expected total cost rate in the long run is given by

\[ TC = \sum_{i=1}^{4} \left( \frac{\rho_i}{1-\rho_i} \right) C_1 + \prod_{i=1}^{4} (\rho_i) C_2. \]

### 3.2.6 Numerical Examples

**Example 1:**

Consider a simple CCN with Poisson arrival rate \( \lambda_1 \) at station 1 and service rate \( \mu_i \) at station \( i \), \( i = 1, 2, 3, 4 \). The CCN diagram is shown in figure 3.2.

![Figure 3.2 CCN - M/M/1: (M/M/3)](image-url)
This CCN is an open Jackson queuing with the following parameters:

\[ N = 4, s_1 = 1, s_2 = 3, s_3 = 3, s_4 = 3 \] and routing probability matrix,

\[
P = \begin{bmatrix}
0 & 0.2 & 0.3 & 0.5 \\
0.2 & 0 & 0.4 & 0.4 \\
0.1 & 0.2 & 0 & 0.7 \\
0.1 & 0.4 & 0.5 & 0
\end{bmatrix}
\]

For this CCN, \( r_1 = 0, r_2 = 1/3, r_3 = 1/3, r_4 = 1/3 \). The arrival rate of each station \( i \) is given by \( a_i \) where \( a = (a_1, a_2, a_3, a_4) \) and \( a = \lambda (I - P)^{-1} \). In steady state \( a_i < s_i \mu_i \) for \( i = 1, 2, 3, 4 \) and

\[
a_1 = \lambda_1 + \sum_{i=1}^{4} a_i p_{i1} \quad \text{and} \quad a_j = \lambda_1 + \sum_{i=1}^{4} a_i p_{ij} \quad \text{for} \quad j = 2, 3, 4.
\]

**Cost analysis:**

In this section we give a cost analysis for the CCN. We emphasis on the convexity of the total cost functions TC, which varies with the average throughput of the network

\[
\rho = \frac{1}{4} \left[ \sum_{i=1}^{4} \left( \frac{a_i}{s_i \mu_i} \right) \right]
\]

by imposing the cost structure:

\( C_1: \) cost of waiting per customer and \( C_2: \) cost of service per unit time.

As we are unable to prove the convexity of the expected total cost function TC analytically, we made numerical search by varying the parameters of the CCN system. We are able to get local optimum with convexity at specific intervals of the
parameter $\mu_i$. From table 3.1, we observe that whenever $\mu_1$ the service rate at cloud lies in the interval $(19, 20)$ and $\rho \in (1.59, 1.78)$ the optimal arrival rate $\lambda_1^* = 8$, with minimum total cost lies in $(23.94, 23.97)$.

Similarly whenever the service rate $\mu_1$ lies in the interval $(13, 18)$ and $\rho \in (1.59, 1.78)$, the optimal arrival rate is $\lambda_1^* = 7$, and whenever the service rate $\mu_1$, lies in the interval $(11, 12)$ and $\rho \in (1.53, 1.57)$, the optimal arrival rate is $\lambda_1^* = 6$.

Thus the overall optimality of the system is obtained at the level $\lambda_1^* = 7$, with $\mu_1 \in (13, 18)$ and $\rho \in (1.6, 1.8)$ in the figure 3.3. In this numerical example we assume that there is no feedback in the system and hence it becomes Tandem queue with Poisson arrival process. The average throughput of the system, $\bar{\rho} = \frac{1}{4} \sum_{i=1}^{4} \rho_i$, is the indicator for the variation in average total cost $\bar{C}$ of the system ($\bar{\rho}$ - be read as $\rho$ average).
Table 3.1 \( \bar{\rho} \) vs Average total cost

<table>
<thead>
<tr>
<th>( \lambda ) ( \mu )</th>
<th>20</th>
<th>19</th>
<th>18</th>
<th>17</th>
<th>16</th>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \rho )</td>
<td>0.2222</td>
<td>0.2249</td>
<td>0.2278</td>
<td>0.231</td>
<td>0.2347</td>
<td>0.2389</td>
<td>0.2437</td>
<td>0.2491</td>
<td>0.2556</td>
<td>0.2631</td>
</tr>
<tr>
<td>( T_c )</td>
<td>65.6161</td>
<td>65.4522</td>
<td>65.2702</td>
<td>65.0669</td>
<td>64.8384</td>
<td>64.5796</td>
<td>64.284</td>
<td>63.9434</td>
<td>63.5465</td>
<td>63.0781</td>
<td>64.56754</td>
</tr>
<tr>
<td>2</td>
<td>( \rho )</td>
<td>0.4444</td>
<td>0.4497</td>
<td>0.4556</td>
<td>0.4621</td>
<td>0.4694</td>
<td>0.4778</td>
<td>0.4873</td>
<td>0.4983</td>
<td>0.5111</td>
<td>0.5263</td>
</tr>
<tr>
<td>( T_c )</td>
<td>53.0493</td>
<td>52.7864</td>
<td>52.4948</td>
<td>52.1694</td>
<td>51.8041</td>
<td>51.391</td>
<td>50.9203</td>
<td>50.3789</td>
<td>49.7498</td>
<td>49.01</td>
<td>51.3754</td>
</tr>
<tr>
<td>3</td>
<td>( \rho )</td>
<td>0.6667</td>
<td>0.6746</td>
<td>0.6833</td>
<td>0.6931</td>
<td>0.7042</td>
<td>0.7167</td>
<td>0.731</td>
<td>0.7474</td>
<td>0.7667</td>
<td>0.7894</td>
</tr>
<tr>
<td>( T_c )</td>
<td>43.0297</td>
<td>42.7225</td>
<td>42.3824</td>
<td>42.0038</td>
<td>41.5799</td>
<td>41.1024</td>
<td>40.5603</td>
<td>39.94</td>
<td>39.224</td>
<td>38.3892</td>
<td>41.09342</td>
</tr>
<tr>
<td>4</td>
<td>( \rho )</td>
<td>0.8889</td>
<td>0.8994</td>
<td>0.9111</td>
<td>0.9242</td>
<td>0.9389</td>
<td>0.9556</td>
<td>0.9746</td>
<td>0.9966</td>
<td>1.0222</td>
<td>1.0525</td>
</tr>
<tr>
<td>( T_c )</td>
<td>35.3148</td>
<td>35.009</td>
<td>34.672</td>
<td>34.2987</td>
<td>33.8834</td>
<td>33.4189</td>
<td>32.8968</td>
<td>32.3067</td>
<td>31.6368</td>
<td>30.8739</td>
<td>33.4311</td>
</tr>
<tr>
<td>5</td>
<td>( \rho )</td>
<td>1.1111</td>
<td>1.1243</td>
<td>1.1389</td>
<td>1.1552</td>
<td>1.1736</td>
<td>1.1944</td>
<td>1.2183</td>
<td>1.2457</td>
<td>1.2778</td>
<td>1.3157</td>
</tr>
<tr>
<td>6</td>
<td>( \rho )</td>
<td>1.3333</td>
<td>1.3491</td>
<td>1.3667</td>
<td>1.3863</td>
<td>1.4083</td>
<td>1.4333</td>
<td>1.4619</td>
<td>1.4949</td>
<td>1.5333</td>
<td>1.5788</td>
</tr>
<tr>
<td>7</td>
<td>( \rho )</td>
<td>1.5556</td>
<td>1.574</td>
<td>1.5944</td>
<td>1.6173</td>
<td>1.6431</td>
<td>1.6722</td>
<td>1.7056</td>
<td>1.744</td>
<td>1.7889</td>
<td>1.8419</td>
</tr>
<tr>
<td>8</td>
<td>( \rho )</td>
<td>1.7778</td>
<td>1.7988</td>
<td>1.8222</td>
<td>1.8484</td>
<td>1.8787</td>
<td>1.9111</td>
<td>1.9492</td>
<td>1.9932</td>
<td>2.0444</td>
<td>2.1051</td>
</tr>
<tr>
<td>9</td>
<td>( \rho )</td>
<td>2</td>
<td>2.0237</td>
<td>2.05</td>
<td>2.0794</td>
<td>2.1125</td>
<td>2.15</td>
<td>2.1929</td>
<td>2.2423</td>
<td>2.2962</td>
<td>2.3682</td>
</tr>
<tr>
<td>10</td>
<td>( \rho )</td>
<td>2.2222</td>
<td>2.2485</td>
<td>2.2778</td>
<td>2.3105</td>
<td>2.3472</td>
<td>2.3889</td>
<td>2.4365</td>
<td>2.4915</td>
<td>2.5556</td>
<td>2.6313</td>
</tr>
<tr>
<td>( T_c )</td>
<td>29.25</td>
<td>29.6477</td>
<td>30.1667</td>
<td>30.8634</td>
<td>31.8333</td>
<td>33.25</td>
<td>35.4663</td>
<td>39.3013</td>
<td>47.25</td>
<td>71.7955</td>
<td>37.88222</td>
</tr>
</tbody>
</table>

Figure 3.3 \( \bar{\rho} \) vs Average total cost
3.3 Model: B - Resource Allocation in CCN with General – Service Time

3.3.1 Introduction

Cloud computing is a new trend to solve cost intensive computing / processing problems. Resource allocation is a tedious task in cloud computing. Successful development of cloud computing paradigm necessitates accurate performance evaluation of cloud data centers. The computing resource allocation and performance maintenance have been one of the most important aspects of cloud computing problems. In this model, a resource allocation system has been modeled where the jobs request forms queue and the virtual machines are considered as service centers. It is considered a M/G/s queue as tool to regulate task’s arrivals, and general service time for requests (classification) with single server and infinite waiting space. This model is used to evaluate the performance analysis of cloud server farms and it is solved to obtain accurate estimation of the complete probability distribution of the request response time and other important performance indicators.

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 costs of management and maintenance of hardware and software resources. This cloud concept emphasizes the transfer of management, maintenance and investment from the customer to the provider. This is 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. Users get the computing resources and services by means of customized service level agreement (SLA); they only pay the fee according to the using time, using manner or the amount of data transferred [29]. The main focus of the SLA is the QoS, and it includes availability, throughput, reliability, security, and many other parameters. Performance indicators such as response time, task blocking probability, probability of immediate service, and mean number of tasks in the system, all of which may be determined by using the tools of queuing theory.

Cloud Computing has become one of the most talked about technologies in recent times and has got lots of attention from media as well as analysts because of the opportunities it is offering. It encompasses different types of services. The cloud has a service-oriented architecture, and there are three classes of technology capabilities that are being offered as a service: Infrastructure-as-a-Service (IaaS), where equipment such as hardware, storage, servers and network components are accessible via the internet, the platform-as-a-Service (PaaS), which is a central component of the Cloud: the PaaS is responsible for developing applications for the cloud. It includes hardware with operating systems, virtualized
servers, etc; and finally the Software-as-a-Service (SaaS) (resources software), which includes applications and other hosted services [16].

Queuing theory is a collection of Mathematical models of various queuing systems. Queues or waiting lines arise when demand for a service facility exceeds the capacity of that facility i.e. the customers do not get service immediately upon request but must wait. The basic queuing process consists of customers arriving at a queuing system to receive some service. If the servers are busy, they join the queue in a waiting room (i.e., wait in line). They are then served according to a prescribed system discipline.

However, cloud centers differ from traditional queuing systems in a number of important aspects:

- A cloud center can have a large number of facility (server) nodes, typically of the order of hundreds or thousands; traditional 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. Moreover, the coefficient of variation of task service time may be high (well over the value of 1).
- Due to the dynamic nature of cloud environments, diversity of users’ requests and time dependency of load, cloud centers must provide expected quality of service at widely varying loads [23].
The authors already developed a CCN model, which has M/M/s type service stations [37]. In this paper, the authors study the resource allocation techniques to minimize the resource cost and minimize the service response time for cloud service providers. The cloud center is modeled as M/G/s queuing system with single task arrivals and a task buffer of infinite capacity. The performance measures are evaluated by using a combination of a transform-based analytical model and an approximate Markov Chain model, which allows us to obtain a complete probability distribution of response time and a number of tasks in the system.

### 3.3.2 Model Description

Consider a cloud computing network (CCN) which provides resources ranging from computing infrastructure and applications. The inter-arrival time of requests to the classifier node is exponentially distributed with parameter \( \lambda_1 > 0 \) and the task service times are iid random variable with mean \( \tau_1 > 0 \). Generally there are three kinds of requests. Depending on the type of clients’ request with reference to SLA, three types of services are provided, namely Software (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). The bag of tasks arrive at the first station namely ‘Classifier’, according to a Poisson process with rate \( \lambda_1 > 0 \). The bag of tasks (BoTs) are taken for classification in FCFS discipline. After classification of BoTs according to SLA it moves to any one of the stations which provides SaaS, PaaS and IaaS. Each station \( i \) has \( s_i \) independent servers, and the queueing model at station \( i \) is M/G/s\(_{i}\) type. The cloud computing network diagram is described in the figure 3.4.
3.3.3 Analysis

The Cloud Computing Network, is modeled as an Open Jackson Queueing Network. Consider a general CCN with the following assumptions.

- The network has $N$ single station with $s_i$ servers at each station $i$.
- There is an unlimited waiting space at each station (the classification and service stations).
- The customers (BoTs request) arrive at station $i$ from outside the network according to a Poisson process with parameters $\lambda_i (i = 1, 2, \ldots, N)$ and $\lambda_i > 0$.
- All arrival processes are independent of each other.
- Service times for customers (service requests) of station $i$ are independent and identically distributed (iid) random variables with mean $\tau_i$. 

![Figure 3.4 CCN with M/G/s_i](image)
Customers (service requests) finishing service at station \( i \) proceed to join the queue at station \( j \) with probability \( p_{ij} \) or leave the network altogether with probability \( r_i \) independently of each other [37].

The probabilities \( p_{ij} \), \( i, j \in S = \{1, 2, \ldots, N\} \) are called the routing probabilities and the matrix \( P = (p_{ij}) \) \( i, j \in S \) is called the routing probability matrix. By our assumption, the stochastic model of cloud computing network, becomes an Open Jackson Queuing Network with \( N \) stations and \( s_i \) server at each station [29].

The routing matrix \( P \) can be expressed as a transition probability matrix of the form

\[
P = \begin{bmatrix}
p_{11} & p_{12} & p_{13} & \cdots & p_{1N} \\
p_{21} & p_{22} & p_{23} & \cdots & p_{2N} \\
p_{31} & p_{32} & p_{33} & \cdots & p_{3N} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
p_{N1} & p_{N2} & p_{N3} & \cdots & p_{NN}
\end{bmatrix}
\]

It is assumed in the tandem CCN that each station has infinite capacity for customers (jobs), and is of only feed forward type. External arrival of bag of tasks is allowed to stations 1 (node 1) only. Transition of jobs among nodes is allowed. (\( P_{ij} \geq 0 \)). Next we have to show that the CCN is stable in the long run.

Our previous work [37] is based on M/M/s and it represents a single station that has unlimited queue capacity and infinite calling population, arrival process is Poisson and service time is exponentially distributed 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 the arrival rate and service rate. In this model the authors considered the nodes which have M/G/s_i system with servers that has unlimited queue capacity and infinite calling population, while the arrival is still Poisson process. The statistical distribution of the inter-arrival times still follow the exponential distribution and the distribution of the service time does not. The distribution of the service time may follow any general statistical distribution. 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 times [29].

Consider a CCN queueing system where customers arrive according to a PP (\( \lambda \)) and require iid service times with mean \( \tau_i \), variance \( \sigma_i^2 \) and second moment \( s_i^2 = \sigma_i^2 + \tau_i^2 \), \( i = 1, 2, 3, 4 \). The service times may not be exponentially distributed. The queue is serviced by a \( s_i \) servers at each station and has infinite waiting room. Such a system is called a network of M/G/s_i queues.

Let \( X(t) \) be the number of customers in the system at time \( t \). \( \{X(t), t \geq 0\} \) is a continuous-time stochastic process with state space \( E = \{0, 1, 2, \ldots\} \). Knowing the current state \( X(t) \) does not provide enough information about the remaining service time of the customer in service (unless the service times are exponentially distributed), and hence we cannot predict the future based solely on \( X(t) \). Hence
\{X(t), t \geq 0\} is not a CTMC. Hence we will not be able to study an M/G/s queue in as much detail as the M/M/1 queue. Instead we shall satisfy ourselves with results about the expected number of customers and expected waiting time of customers in the M/G/s queue in a steady state. Since the arrival process to our Open Jackson Network is Poisson, the whole networks (CCN) can be viewed as 4 independent M/G/s queues [29].

### 3.3.4 Stability Analysis

For the CCN, it is considered, that \( \rho_i = \lambda_i \tau_i \) be the traffic intensity.

**Theorem: 3.3** (Condition of Stability). Consider a CCN with 4 nodes and queues with \( s_i \) servers and infinite capacity. Suppose the customers enter at rate \( \lambda \) and the mean service time is \( \tau \). The queue is stable if \( \lambda \tau_i < s_i, i = 1, 2, 3, 4 \).

**Proof:**

We have, \( B = \text{expected number of busy servers} = \lambda \tau_i \), where, \( \lambda \) is the arrival rate of entering customers. However, the number of busy servers cannot exceed the total number of servers. Hence, for the argument to be valid, we must have

\[
\lambda \tau_i < s_i .
\]

It follows that the M/G/s queue is stable if
\[ \rho_i < 1, \text{ for each } i, \text{ and } \rho_i = \frac{\lambda \tau_i}{s_i}, \] where \( s_i \) is the number of servers at station \( i \).

Indeed, it is possible to show that the \( M/G/s \) queue is unstable if \( \rho_i \geq 1 \): Thus \( \rho_i < 1 \) for every \( i \) is a necessity and a sufficient condition of stability. We shall assume that the CCN is stable. \( \square \)

### 3.3.5 Steady State Analysis

Consider the CCN with 4 stations namely classification, SaaS, PaaS and IaaS. The limiting behavior of the system in steady state can be studied as follows. See figure 3.4 (page - 74).

Let \( X_i(t) \) be the number of requests (BoTs) in the \( i^{th} \) station \( i = 1, 2, 3, 4 \) at time \( t \). The state of the system at time \( t \) be denoted as \( X(t) = (X_1(t), X_2(t), X_3(t), X_4(t)) \). Hence the CCN becomes an Open Jackson Queueing Network with \( M/G/s_i \) at each station, where \( \lambda_1 = \lambda, \lambda_i = 0, i = 2, 3, 4 \). Mean service time at station \( i \) is \( \tau_i \). The routing probability matrix is

\[
P = \begin{bmatrix}
0 & 1 & 1 & 1 \\
3 & 3 & 3 & 3 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{bmatrix},
\]

and \( r_1 = 0, r_2 = r_3 = r_4 = 1. \)
As each station \(i = 1, 2, 3, 4\) has a M/G/s\(_i\) queue, the expected number of customers in station \(i\) is given by

\[
L_i = \rho_i + \frac{\lambda^2 s_i^2}{2(1 - \rho_i)},
\]

where, \(s_i^2 = \sigma_i^2 + \tau_i^2\), and the expected waiting time in the queue is given by

\[
W_i = \tau_i + \frac{\lambda s_i^2}{2(1 - \rho_i)}, \text{ where } s_i^2 \text{ is the sample variance for service time at station } i.
\]

### 3.3.6 System Performance Measures

As we obtained the steady state probabilities for the number of customers in each of the station, we are able to find the mean number of BoTs waiting. The following system performance measures are crucial for our model [32].

The traffic intensity is given by

\[
\rho_i = \frac{\lambda \tau_i}{s_i}.
\]

1. Expected waiting time in the system

\[
W = \sum_{i=1}^{4} \left[ \tau_i + \frac{\lambda s_i^2}{2(1 - \rho_i)} \right], \text{ where } s_i^2 = \sigma_i^2 + \tau_i^2.
\]
2. The expected number of tasks waiting for transmission is given by

\[ L = \sum_{i=1}^{4} \left[ \rho_i + \frac{\lambda^2 s_i^2}{2(1 - \rho_i)} \right]. \]

### 3.3.7 Numerical Examples

In this section, we evaluate the performance of a cloud computing center as a network of M/G/s queues. The following numerical example illustrates the model.

Consider the CCN with 4 stations,

\[ \tau = (0.1, 0.2, 0.3, 0.4), \quad \lambda = 3, \quad \text{and} \quad \sigma = (0.1, 0.1, 0.1, 0.1). \]

<table>
<thead>
<tr>
<th>Number of servers</th>
<th>Server Average</th>
<th>Waiting time (W)</th>
<th>Length of queue (L)</th>
</tr>
</thead>
<tbody>
<tr>
<td>s_1 s_2 s_3 s_4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 2 3 2</td>
<td>2.5</td>
<td>1.9923</td>
<td>4.2768</td>
</tr>
<tr>
<td>3 3 3 3</td>
<td>3.0</td>
<td>1.7664</td>
<td>3.2991</td>
</tr>
<tr>
<td>3 4 3 4</td>
<td>3.5</td>
<td>1.7001</td>
<td>2.9504</td>
</tr>
<tr>
<td>3 5 3 5</td>
<td>4.0</td>
<td>1.6684</td>
<td>2.7654</td>
</tr>
<tr>
<td>3 6 3 6</td>
<td>4.5</td>
<td>1.6497</td>
<td>2.6491</td>
</tr>
<tr>
<td>3 7 3 7</td>
<td>5.0</td>
<td>1.6374</td>
<td>2.5694</td>
</tr>
<tr>
<td>3 8 3 8</td>
<td>5.5</td>
<td>1.6287</td>
<td>2.5111</td>
</tr>
<tr>
<td>3 9 3 9</td>
<td>6.0</td>
<td>1.6222</td>
<td>2.4666</td>
</tr>
<tr>
<td>3 10 3 10</td>
<td>6.5</td>
<td>1.6172</td>
<td>2.4315</td>
</tr>
</tbody>
</table>

Table 3.2 Average number of servers, Waiting time and Length of queue.
From Table 3.5 and 3.6, we observe that the length of queue in the system decreases with the average number of servers and the waiting time of a customer in the system also decreases with the average number of servers. In general, ‘efficiency of a server’ is measured by the mean service time $\tau_i$ of a customer and its variance $\sigma_i^2$. 