

Table 1: Tempi esecuzione sperimentali (espessi in μs) ottenuti da vari test

$\lambda_{controller}(1)$	Total mean arrival rate into <i>controller</i> of class 1 tasks
$\lambda_{controller}(2)$	Total mean arrival rate into <i>controller</i> of class 2 tasks
$\mu_{cloudlet}(1)$	Mean service rate at <i>cloudlet</i> of class 1 tasks
$\mu_{cloudlet}(2)$	Mean service rate at <i>cloudlet</i> of class 2 tasks
$\mu_{cloud}(1)$	Mean service rate at <i>cloud</i> of class 1 tasks
$\mu_{cloud}(2)$	Mean service rate at <i>cloud</i> of class 2 tasks

1 Notation and Modeling for Classed Jackson Networks

2 Analytical solution

In this last section we will develop an analytical solution to validate the results obtained previously through our simulations.

2.1 System based on access control Algorithm 1

In order to compute all parameters and metrics of interest associated with above-mentioned system **based on access control Algorithm 1**, is crucial to determine the **fraction of jobs that are forwarded to the cloud**.

According to our model, the cloudlet has limited resources therefore he can accept jobs until their number does not exceed a given threshold N ; the key point is to compute the probability according to which the sum of job of each class in cloudlet system is equal to that threshold.

Formally, we can determine this probability by modelling the cloudlet system as **continuous-time Markov chain (CTMC)**, of which a graphical representation is shown in Figure 1, where the state, denoted with (n_1, n_2) , corresponds to the number of class 1 job n_1 and class 2 job n_2 present in system at a certain moment.

Obviously we can compute **limiting probabilities**, namely the probability according to which the chain is in a state j independently of the starting state, for each states of the Cloudlet CTMC by solving the **balance equations**, in which we can equate the rate at which the system leaves state j with the rate at which the system enters state j .¹

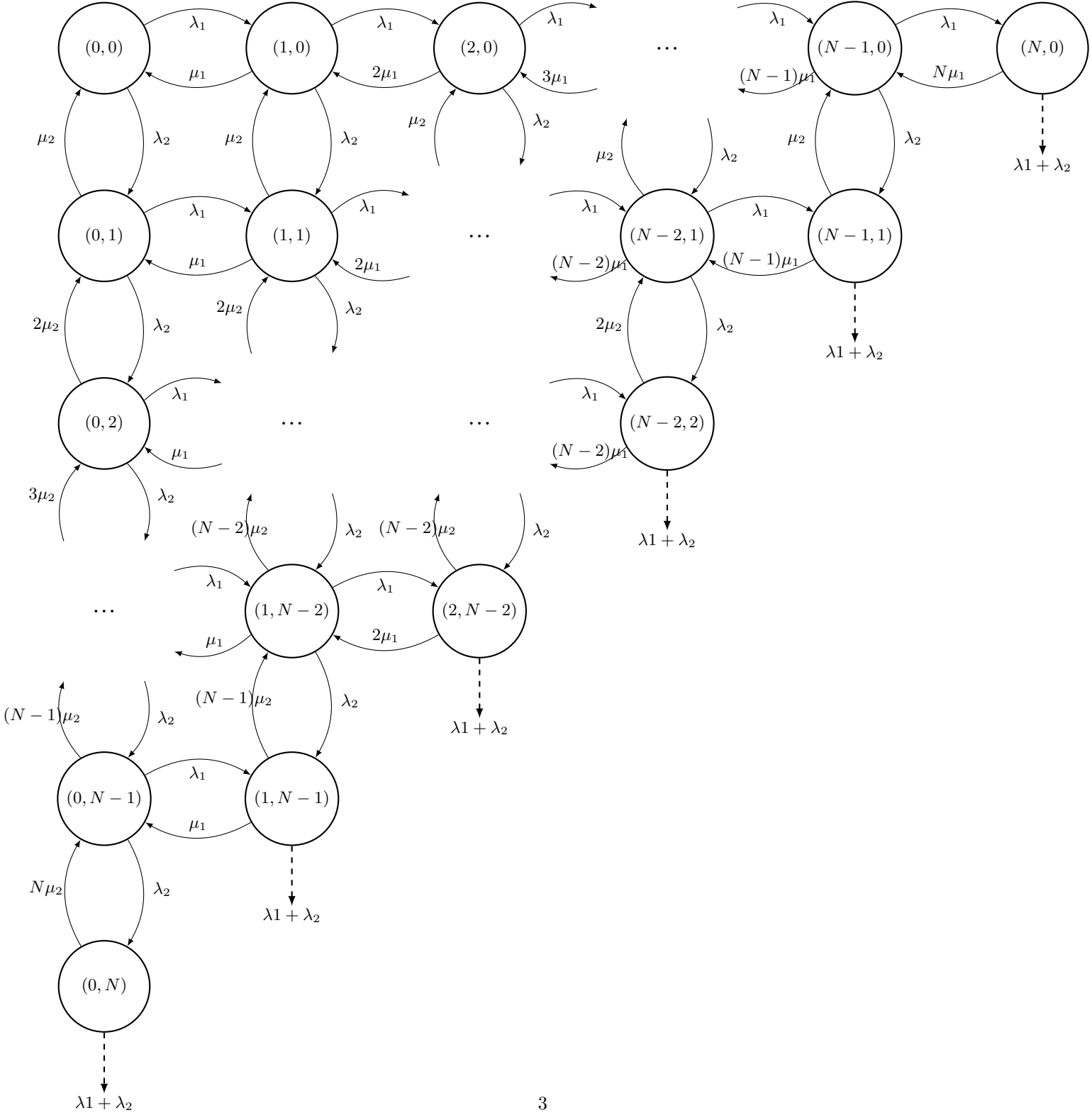
2.1.1 Balance equation computing

In order to compute $\pi_{(n_1, n_2)}$, we must resolve balance equations for given CTMC shown in Equation 2.2. Is important to remember that these equation, automatically generated through a very simple Java script fully described into `ALG-1-MATLAB-ScriptGenerator.java` file, have been resolved using a MATLABTMscript, called `ALG-1-MATLAT-Script.m`.

$$\left\{ \begin{array}{ll} (\lambda_1 + \lambda_2)\pi_{(0,0)} = \mu_1\pi_{(1,0)} + \mu_2\pi_{(0,1)} & \\ (\lambda_1 + \lambda_2 + i\mu_1)\pi_{(i,0)} = \lambda_1\pi_{(i-1,0)} + \mu_1(i+1)\pi_{(i+1,0)} + \mu_2\pi_{(i,1)} & \forall i \in \mathbb{N} \mid 1 \leq i \leq N-1 \\ (\lambda_1 + \lambda_2 + i\mu_2)\pi_{(0,i)} = \lambda_2\pi_{(0,i-1)} + \mu_1\pi_{(1,i)} + \mu_2(i+1)\pi_{(0,i+1)} & \forall i \in \mathbb{N} \mid 1 \leq i \leq N-1 \\ \mu_1 N \pi_{(N,0)} = \lambda_1 \pi_{(N-1,0)} & \\ \mu_2 N \pi_{(0,N)} = \lambda_2 \pi_{(0,N-1)} & \\ (i\mu_1 + j\mu_2)\pi_{(i,j)} = \lambda_1\pi_{(i-1,j)} + \lambda_2\pi_{(i,j-1)} & \forall i, j \in \mathbb{N} \mid \begin{array}{l} 1 \leq i \leq N-1 \\ 1 \leq j \leq N-1 \end{array} \mid i+j = N \\ (\lambda_1 + \lambda_2 + i\mu_1 + j\mu_2)\pi_{(i,j)} = \lambda_1\pi_{(i-1,j)} + \lambda_2\pi_{(i,j-1)} + \mu_1(i+1)\pi_{(i+1,j)} + \mu_2(j+1)\pi_{(i,j+1)} & \forall i, j \in \mathbb{N} \mid \begin{array}{l} 1 \leq i \leq N-1 \\ 1 \leq j \leq N-1 \end{array} \mid i+j < N \\ \sum \pi_{(i,j)} = 1 & \forall i, j \in \mathbb{N}_0 \mid \end{array} \right.$$

¹Cfr. Mor Harchol-Balter - *Performance Modeling and Design of Computer Systems* - Carnegie Mellon University, Pennsylvania, pag. 237

Figure 1: Cloudlet based on access control algorithm 1 modelled using a CTMC



2.1.2 Probabilities computing

Having found the stationary probabilities, we now find the **probability that an arriving job on controller has to be forwarded to cloud**, Π_A . Observe that:

- The class to which an arrival job belongs is not important for system based on access control Algorithm 1, that is we make no distinction between a class 1 or class 2 arrival job.
- Π_A is the probability that an arrival job find that the number of jobs present in cloudlet has exceeded given threshold N ; that is an arriving job on controller will be forwarded to cloud when $n_1 + n_2 = N$.

Formally:

$$\begin{aligned}\Pi_A &= P\{\text{An arrival sees } N \text{ jobs in cloudlet}\} \\ &= \text{Limiting probability that there are } N \text{ jobs in system} \\ &= \sum_{\substack{n_1+n_2=N \\ n_1, n_2 \in \mathbb{N}_0}} \pi(n_1, n_2)\end{aligned}\tag{1}$$

At this point we can easily compute the **probability that an arriving job on controller has to be accepted on cloudlet**, Π_B . Indeed:

$$\begin{aligned}\Pi_B &= 1 - P\{\text{An arrival sees } N \text{ jobs in cloudlet}\} \\ &= 1 - \Pi_A\end{aligned}\tag{2}$$

Is $\lambda_i(c)$ the **total arrival rate into a component i of class c job**. Applying the result obtained in 1, we can say for cloud that:

$$\begin{aligned}\lambda_{\text{cloud}}(1) &= \lambda_1(1) \cdot P\{\text{An arrival job on controller is sent to cloud}\} \\ &= \lambda_1(1) \cdot P\{\text{An arrival job on controller sees exactly } N \text{ jobs in cloudlet}\} \\ &= \lambda_1(1) \cdot \Pi_A\end{aligned}\tag{3}$$

At this point, knowing the value of Π_A e Π_B , equal to **0.4139** and **0.5861** respectively, we are finally able to calculate the values of following parameters:

$$\begin{aligned}\lambda_{\text{cloud}}(1) &= \lambda_1(1) \cdot \Pi_A = \mathbf{1.655600 \text{ jobs/sec}} \\ \lambda_{\text{cloud}}(2) &= \lambda_2(2) \cdot \Pi_A = \mathbf{2.586875 \text{ jobs/sec}} \\ \lambda_{\text{cloudlet}}(1) &= \lambda_1(1) \cdot \Pi_B = \mathbf{2.344400 \text{ jobs/sec}} \\ \lambda_{\text{cloudlet}}(2) &= \lambda_2(2) \cdot \Pi_B = \mathbf{3.663125 \text{ jobs/sec}}\end{aligned}\tag{4}$$

2.1.3 Average population

Is $E[N_i(c)]$ the **average number of class c jobs into a component i** . Compute average population for cloudlet is very easy because it's enough sum for each state of CTMC the corresponding limiting probability multiplied by the number of job. Formally:

$$\begin{aligned}
E[N_{\text{cloudlet}}(1)] &= \sum_{(n_1, n_2) \in M} n_1 \cdot \pi(n_1, n_2) \\
&= \mathbf{5.2040 \text{ jobs}} \\
E[N_{\text{cloudlet}}(2)] &= \sum_{(n_1, n_2) \in M} n_2 \cdot \pi(n_1, n_2) \\
&= \mathbf{13.5521 \text{ jobs}} \\
E[N_{\text{cloudlet}}] &= E[N_{\text{cloudlet}}(1)] + E[N_{\text{cloudlet}}(2)] \\
&= \sum_{(n_1, n_2) \in M} (n_1 + n_2) \cdot \pi(n_1, n_2) \\
&= \mathbf{18.7561 \text{ jobs}}
\end{aligned} \tag{5}$$

We can compute cloud's average population using Little's Law. In general we have:

$$\begin{aligned}
E[N] &= \lambda \cdot E[T] \\
&= \lambda \cdot (E[T_Q] + E[S]) = \lambda \cdot (E[T_Q] + \frac{1}{\mu})
\end{aligned} \tag{6}$$

Since we have modelled system's cloud component as a $M/M/\infty$ queueing system, there is no job's waiting time because this kind of system is made up of an infinite number of server. So we can say that:

$$\begin{aligned}
E[N_{\text{cloud}}(1)] &= \frac{\lambda_{\text{cloud}}(1)}{\mu_{\text{cloud}}(1)} = \mathbf{6.6224 \text{ jobs}} \\
E[N_{\text{cloud}}(2)] &= \frac{\lambda_{\text{cloud}}(2)}{\mu_{\text{cloud}}(2)} = \mathbf{11.75852273 \text{ jobs}}
\end{aligned} \tag{7}$$

2.1.4 Response Time

There are two ways to compute mean response time of both class of job that they experienced both on cloud and cloudlet:

- Since we know average population and average arrival rate for both class of job, we can easily compute it using Little's Law.
- Since our system haven't a queue, there is no waiting time experienced by jobs therefore $E[T]$ is simply equal to average service time $E[S]$.

So, is $E[T_i(c)]$ the **mean response time experienced by a class c jobs into a component i** , is true that:

$$\begin{aligned}
E[T_{\text{cloudlet}}(1)] &= \frac{E[N_{\text{cloudlet}}(1)]}{\lambda_{\text{cloudlet}}(1)} = \mathbf{2.219757721 \text{ s}} \\
&= E[S_{\text{cloudlet}}(1)] \\
&= \frac{1}{\mu_{\text{cloudlet}}(1)} \\
E[T_{\text{cloudlet}}(2)] &= \frac{E[N_{\text{cloudlet}}(2)]}{\lambda_{\text{cloudlet}}(2)} = \mathbf{3.699600751 \text{ s}} \\
&= E[S_{\text{cloudlet}}(2)] \\
&= \frac{1}{\mu_{\text{cloudlet}}(2)} \\
E[T_{\text{cloud}}(1)] &= \frac{E[N_{\text{cloud}}(1)]}{\lambda_{\text{cloud}}(1)} = \mathbf{4 \text{ s}} \\
&= E[S_{\text{cloud}}(1)] \\
&= \frac{1}{\mu_{\text{cloud}}(1)} \\
E[T_{\text{cloud}}(2)] &= \frac{E[N_{\text{cloud}}(2)]}{\lambda_{\text{cloud}}(2)} = \mathbf{4.545454547 \text{ s}} \\
&= E[S_{\text{cloud}}(2)] \\
&= \frac{1}{\mu_{\text{cloud}}(2)}
\end{aligned} \tag{8}$$

Instead if we want to calculate global mean response time experienced by a job of class 1 $E[T](1)$:

$$\begin{aligned}
E[T](1) &= E[T_{\text{cloudlet}}(1)] \cdot P\{\text{An arrival class 1 job is accepted on controller}\} \\
&\quad + E[T_{\text{cloud}}(1)] \cdot P\{\text{An arrival class 1 job is accepted on cloud}\} \\
&= E[T_{\text{cloudlet}}(1)] \cdot \Pi_{\text{JobSendToCloudlet}} + E[T_{\text{cloud}}(1)] \cdot \Pi_{\text{JobSendToCloud}}
\end{aligned} \tag{9}$$

Similarly:

$$E[T](2) = E[T_{\text{cloudlet}}(2)] \cdot \Pi_{\text{JobSendToCloudlet}} + E[T_{\text{cloud}}(2)] \cdot \Pi_{\text{JobSendToCloud}} \tag{10}$$

Finally we can compute global response time using following expression:

$$E[T] = E[T](1) \cdot \frac{\lambda(1)}{\lambda(1) + \lambda(2)} + E[T](2) \cdot \frac{\lambda(2)}{\lambda(1) + \lambda(2)} \tag{11}$$

We can also compute global response time for cloudlet or cloud only:

$$E[T_{\text{cloudlet}}] = E[T_{\text{cloudlet}}(1)] \cdot \frac{\lambda_{\text{cloudlet}}(1)}{\lambda_{\text{cloudlet}}(1) + \lambda_{\text{cloudlet}}(2)} + E[T_{\text{cloudlet}}(2)] \cdot \frac{\lambda_{\text{cloudlet}}(2)}{\lambda_{\text{cloudlet}}(1) + \lambda_{\text{cloudlet}}(2)} \quad (12)$$

$$E[T_{\text{cloud}}] = E[T_{\text{cloud}}(1)] \cdot \frac{\lambda_{\text{cloud}}(1)}{\lambda_{\text{cloud}}(1) + \lambda_{\text{cloud}}(2)} + E[T_{\text{cloud}}(2)] \cdot \frac{\lambda_{\text{cloud}}(2)}{\lambda_{\text{cloud}}(1) + \lambda_{\text{cloud}}(2)} \quad (13)$$

2.1.5 Average

2.2 System based on access control Algorithm 2

$$\left\{ \begin{array}{ll} (\lambda_1 + \lambda_2)\pi_{(0,0)} = \mu_1\pi_{(1,0)} + \mu_2\pi_{(0,1)} & \\ (\lambda_1 + \lambda_2 + i\mu_1)\pi_{(i,0)} = \lambda_1\pi_{(i-1,0)} + \mu_1(i+1)\pi_{(i+1,0)} + \mu_2\pi_{(i,1)} & \forall i \in \mathbb{N} \mid 1 \leq i \leq S-1 \\ (\lambda_1 + \lambda_2 + i\mu_2)\pi_{(0,i)} = \lambda_2\pi_{(0,i-1)} + \mu_1\pi_{(1,i)} + \mu_2(i+1)\pi_{(0,i+1)} & \forall i \in \mathbb{N} \mid 1 \leq i \leq S-1 \\ (S\mu_1 + \lambda_1)\pi_{(S,0)} = \lambda_1\pi_{(S-1,0)} + \lambda_1\pi_{(S-1,1)} + (S+1)\mu_1\pi_{(S+1,0)} & \\ (\lambda_1 + S\mu_2)\pi_{(0,S)} = \lambda_2\pi_{(0,S-1)} & \\ (i\mu_1 + j\mu_2 + \lambda_1)\pi_{(i,j)} = \lambda_1\pi_{(i-1,j)} + \lambda_2\pi_{(i,j-1)} & \forall i, j \in \mathbb{N} \mid \begin{array}{l} 1 \leq i \leq S-1 \\ 1 \leq j \leq S-1 \end{array} \mid i+j = S \\ (\lambda_1 + \lambda_2 + i\mu_1 + j\mu_2)\pi_{(i,j)} = \lambda_1\pi_{(i-1,j)} + \lambda_2\pi_{(i,j-1)} + \mu_1(i+1)\pi_{(i+1,j)} + \mu_2(j+1)\pi_{(i,j+1)} & \forall i, j \in \mathbb{N} \mid \begin{array}{l} 1 \leq i \leq N-1 \\ 1 \leq j \leq N-1 \end{array} \mid i+j < S \\ (i\mu_1 + \lambda_1)\pi_{(i,0)} = \mu_1(i+1)\pi_{(i+1,0)} + \lambda_1\pi_{(i-1,0)} & \forall i \in \mathbb{N} \mid S+1 \leq i \leq N-1 \\ N\mu_1\pi_{(N,0)} = \lambda_1\pi_{(N-1,0)} & \\ \sum \pi_{(i,j)} = 1 & \forall i, j \in \mathbb{N}_0 \end{array} \right.$$

Figure 2: Cloudlet system component modeled using a CTMC

