

# Performance Modelling (B1)

## PEPA Specification

The model is an M/M/1/3 queue (Markov arrivals, Markov service, 1 server and a limited buffer size of 3). Therefore, the Markov chain is finite and bounded at 0 and 3. It is also both irreducible and all states are recurrent.

The below model will have a stationary distribution as  $\lambda < \mu$ . If arrivals of jobs happened faster than we could serve them then the queue would grow indefinitely.

```
lambda = 2.0;
mu = 4.0;

B0 = (arrive, lambda).B1;
B1 = (arrive, lambda).B2 + (service, mu).B0;
B2 = (arrive, lambda).B3 + (service, mu).B1;
B3 = (service, mu).B2;

B0[1]
```

## Steady State Probabilities

The steady state is the proportion of time spent in each state. For this M/M/1/3 queue, the number of states is 4. The lower the state number the higher the probability of being in that state, this makes sense as the service rate is higher than the arrival rate.

### Tabular representation of the state space

State #	State	Steady State Probability
0	B0	0.5333...
1	B1	0.2666...
2	B2	0.1333...
3	B3	0.0666...

## Correctness of Model

We can prove that the steady state probabilities listed above are correct using the principal of global balance. First we calculate the steady state probability for the state 0. Below, variable  $\mathbf{r}$  is the utilization of the buffer; for the queue to be stable then  $\mathbf{r} < 1$  should hold.

With:

$$r = \frac{\lambda}{\mu} = \frac{2}{4} = 0.5$$

Then:

$$\pi_0 = \frac{1-r}{1-r^{N+1}} = \frac{1-0.5}{1-0.5^{3+1}} = 0.5\overline{3}$$

The other state probabilities can be calculated / proved using the following formula:

$$\pi_k = r^k \pi_0$$

Where **k** is the state number:

$$\pi_1 = 0.5^1 0.5\overline{3} = 0.2\overline{6}$$

$$\pi_2 = 0.5^2 0.5\overline{3} = 0.1\overline{3}$$

$$\pi_3 = 0.5^3 0.5\overline{3} = 0.0\overline{6}$$

### Average Queue Size / Average Response Time

The *average queue size* can be calculated using the following formula:

$$\frac{r}{1-r} = \frac{0.5}{1-0.5} = 1$$

A jobs response time is the elapsed time from when it enters and leaves the system. We can use Little's law to calculate the *average response time* (**W**) below. With **L** being the average number of jobs in the queue and **tilde lambda** being the effective arrival rate:

$$L = \tilde{\lambda} W$$

The effective arrival rate is less than lambda and calculated as:

$$\lambda(1 - \pi_N) = 2(1 - 0.0\overline{6}) = 1.8\overline{6}$$

Therefore, the *average response time* is:

$$W = \frac{0.5}{1.8\overline{6}} = 0.268\overline{3dp}$$

### Screenshots

## Script

```
mm13.pepa
1 lambda = 2.0;
2 mu = 4.0;
3
4 B0 = (arrive,lambda).B1;
5 B1 = (arrive,lambda).B2 + (service,mu).B0;
6 B2 = (arrive,lambda).B3 + (service,mu).B1;
7 B3 = (service,mu).B2;
8
9 B0[1]
10 |
```

## State Space View

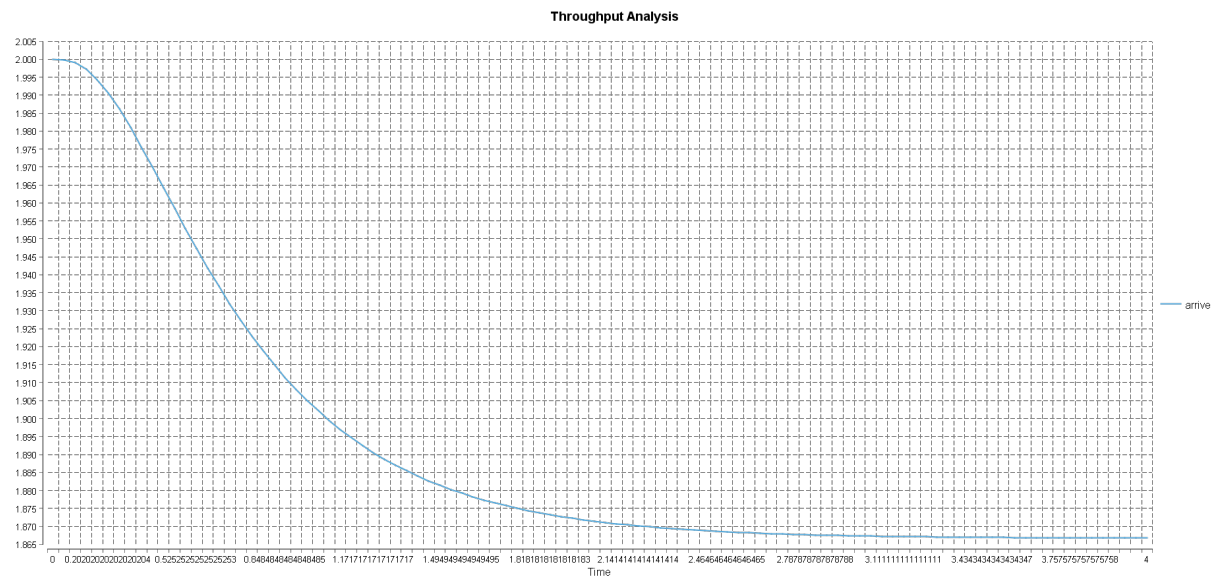
State Space View			Problems
4 states			
1	B0	0.5333333333333333	
2	B1	0.2666666666666666	
3	B2	0.1333333333333333	
4	B3	0.0666666666666665	

## Throughput

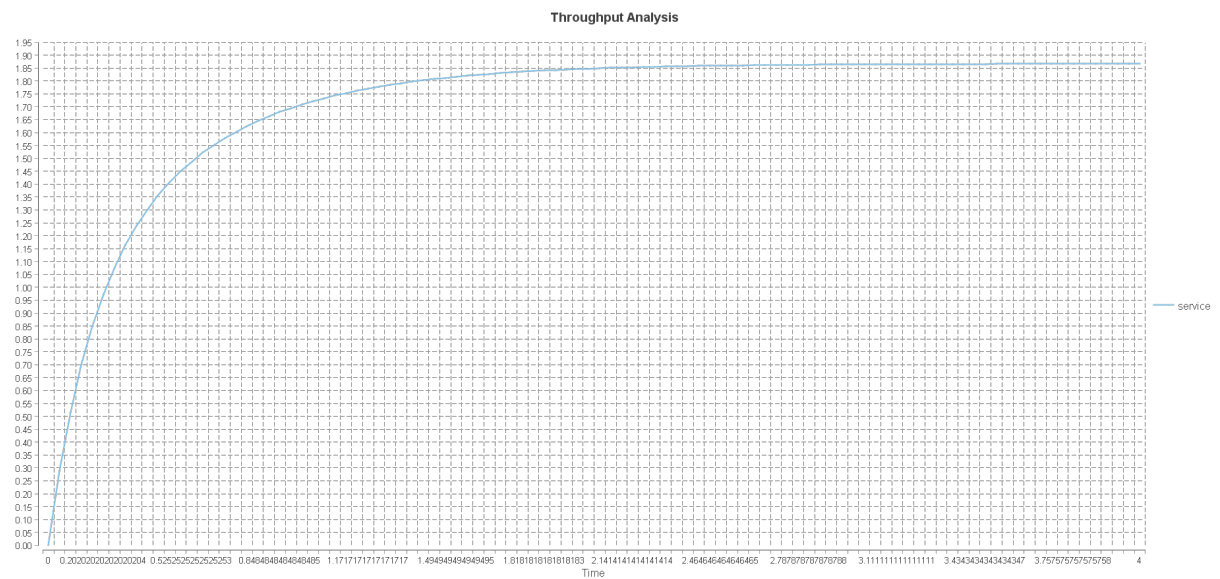
Utilisation	Throughput	Population
Action	Throughput	
arrive	1.8666666666666667	
service	1.8666666666666667	

## Graphs

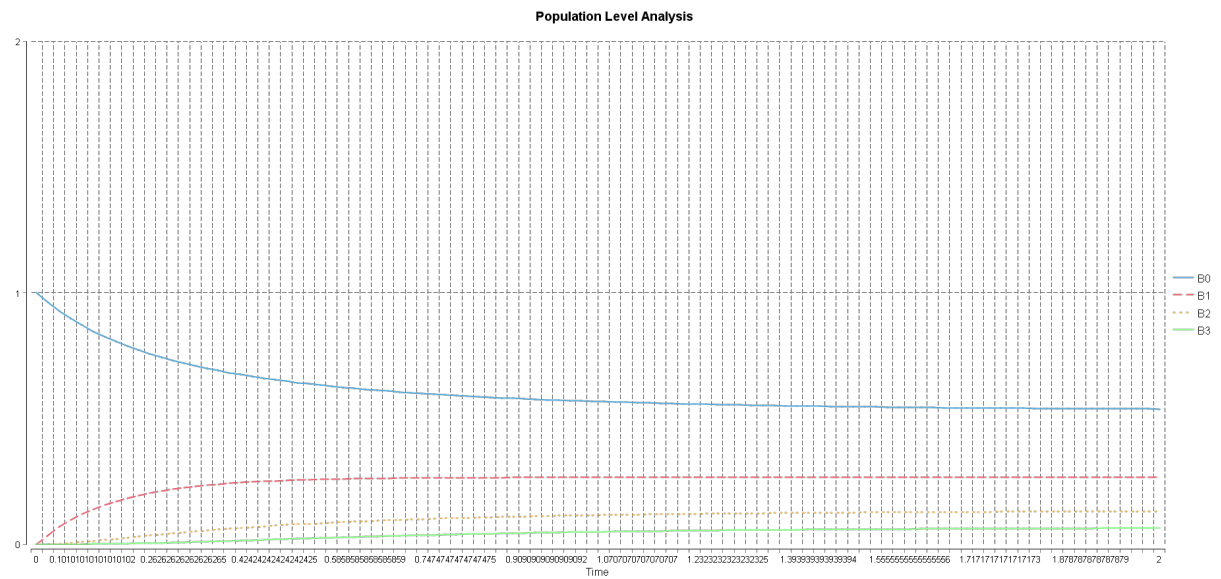
### Throughput analysis arrive



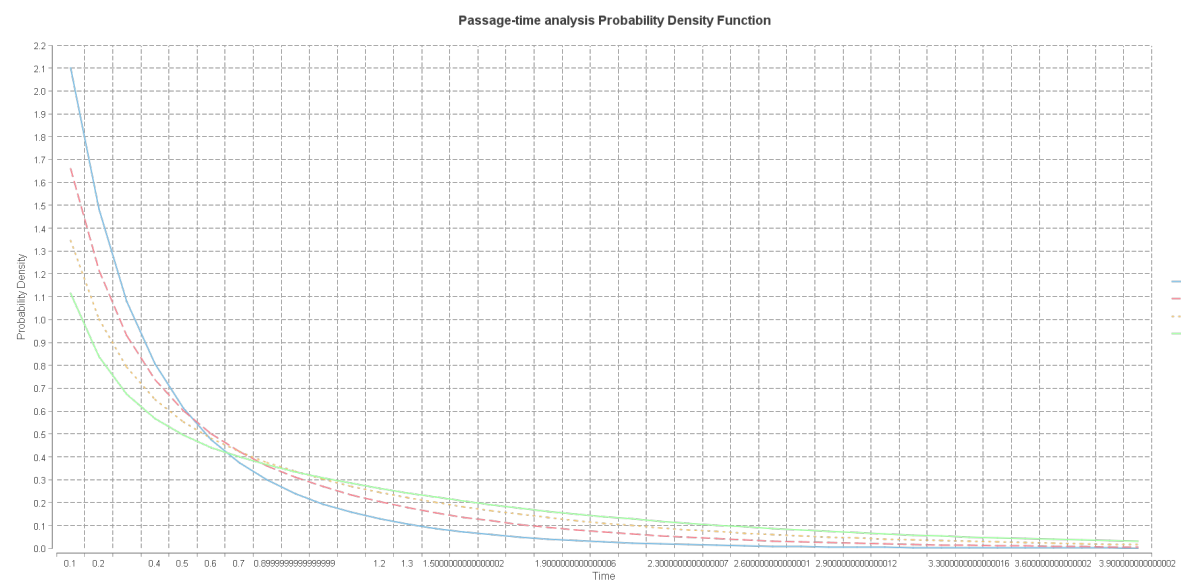
## Throughput analysis service



## Population level analysis



## Probability density function (lambda)



## Probability density function (mu)

Passage-time analysis Probability Density Function

