5 The $M/M/1$ system with subcontracting

One of the ways to deal with temporary overload is subcontracting. When the total number of jobs in the system becomes too high, the throughput times will be too high as well. In order to stay away from this situation one might use subcontracting. Some time before the actual overload occurs, some of the jobs are sent to subcontracters. Also order rejection might occur. Practically this means that under some load conditions, some of the jobs will not enter this production system, but will be executed elsewhere. For us it is important to know how this influences the mean throughput times and how many jobs are not accepted.

Let us look at a simple form of subcontracting for the $M/M/1$ system. Jobs arrive according to a Poisson process with rate λ. The arrival stream consists of two classes of jobs, 1 and 2. The jobs from both classes have exponential processing times with the same mean $1/\mu$. With α we denote the probability that an arriving job is of class 1.

Class 1 jobs have to be accepted, whereas class 2 jobs can be sent to a subcontractor. We assume that the decision whether or not to accept a job is only taken upon arrival. The natural subcontracting rule is: accept all jobs as long as the number of jobs in the system is less than N and do not accept class 2 jobs whenever the system contains N or more jobs. Accepted jobs are served in order of arrival, so FCFS.

5.1 The equilibrium distribution

Due to the exponential interarrival and processing times this is once again an easy to analyse Markov process. The state can be characterized by the total number of jobs in the system. This neglects the information about the class of the jobs, but for an arriving job it is irrelevant which classes the jobs in the system belong to. Remark that it is essential here that both type of jobs have the same processing time.

Let p_k be the equilibrium probability of having k jobs in the system. Then, equating the mean number of transitions per time unit out of the set $\{0, \ldots, k\}$ to the number into this set, we get

$$\begin{aligned} \lambda p_k &= \mu p_{k+1}, \qquad 0 \le k < N \\ \alpha\lambda p_k &= \mu p_{k+1}, \qquad k \ge N. \end{aligned}$$

So, with $\rho = \lambda/\mu$,

$$p_k = \rho^k p_0, \qquad 0 \le k < N,$$

and

$$p_{N+k} = \rho^N (\alpha\rho)^k p_0, \qquad k \ge 0.$$

From the normalization, $\sum_0^\infty p_k = 1$, we get

$$\frac{1}{p_0} = \sum_{k=0}^{N-1} \rho^k + \frac{\rho^N}{1-\alpha\rho} = \frac{1-\rho^N}{1-\rho} + \frac{\rho^N}{1-\alpha\rho}.$$

5.2 Performance measures

The important performance measures for this model are P_{rej}, the fraction of the jobs from class 2 that is not accepted, and $E(S_1)$ and $E(S_2)$, the mean throughput times for the jobs of the two classes. Using PASTA we have

$$
\begin{aligned}
P_{rej} &= \sum_{k=0}^{\infty} p_{N+k} = p_0\rho^N/(1-\alpha\rho), \\
E(S_2) &= p_0 \sum_{k=0}^{N-1} \rho^k (k+1)\frac{1}{\mu} \\
&= p_0 \frac{1-\rho^{N+1}-(N+1)\rho^N(1-\rho)}{(1-\rho)^2}\frac{1}{\mu},
\end{aligned}
$$

and

$$
\begin{aligned}
E(S_1) &= p_0 \sum_{k=0}^{N-1} \rho^k (k+1)\frac{1}{\mu} + p_0 \sum_{k=0}^{\infty} (\alpha\rho)^k \rho^N (k+N+1)\frac{1}{\mu} \\
&= p_0 \left(\frac{1-\rho^{N+1}-(N+1)\rho^N(1-\rho)}{(1-\rho)^2} + \frac{\rho^N N}{1-\alpha\rho} + \frac{\rho^N}{(1-\alpha\rho)^2} \right) \frac{1}{\mu}.
\end{aligned}
$$

Note that for class 2 jobs we have to be careful. The mean throughput time for an arbitrary class 2 job is not the same as the mean throughput time for an accepted class 2 job. If we denote the latter by $E(S_2|\text{accepted})$, then the relation between the two quantities is given by

$$E(S_2) = P_{rej} \cdot 0 + (1 - P_{rej}) \cdot E(S_2|\text{accepted}).$$

Furthermore, it is not difficult to compute the distribution of the throughput time. An arriving job that enters the system when there are already k jobs present has an Erlang distributed thoughput time with parameters $k+1$ and μ.

5.3 Numerical results

In the following two tables we consider the results for two cases, one with a basic load of 0.95 and the other with a load of 1.05. The latter system would explode without a subcontracting rule. Both classes contribute 50 percent to the load, so by not accepting class 2 jobs the load drops to 0.475 and 0.525, respectively.

From the results we see that by sending a limited amount of jobs to subcontractors, the performance improves considerably. For $N = 20$ and $\rho = 0.95$ only 5 percent of the jobs from stream 2 are sent to subcontractors while the throughput time is reduced by more than 50 percent. Note that 5 percent of stream 2 amounts to about 1 job a month. What we further see is that the system that would explode without a subcontracting or rejection option behaves quite well for $N = 20$ or $N = 10$.

ρ	N	$E(S_1)$	$E(S_2\|\text{accepted})$	P_{rej}
0.95	∞	80	80	0.000
	40	57.9	56.4	0.014
	20	37.9	35.3	0.051
	10	23.7	20.3	0.124
1.05	∞	∞	∞	0.000
	40	113.3	106.5	0.109
	20	54.2	48.4	0.145
	10	28.9	23.6	0.214

Table 1: Results for $\alpha = 0.5$, $1/\mu = 4$ hours and $\rho = 0.95$ and $\rho = 1.05$, respectively.

Remark 5.1 Another possibility to deal with temporary overload is to speed up the machine. The natural rule is: speed up the machine from μ to $(1+\alpha)\mu$ as soon as the number of jobs in the system is greater than N, and slow down to μ again as soon as this number drops below N. The Markov process describing this system is very similar to the $M/M/1$ system with subcontracting.

Remark 5.2 In the $M/M/1$ model with subcontracting the transition rates from state i depend on i; i.e., the arrival rate in state i is λ for $i < N$, and this rate is $\alpha\lambda$ for $i \geq N$. This model is a special case of a *birth-death process*, where the rate from state i to $i+1$ is λ_i (the birth rate) and the rate from state i to $i-1$ is μ_i (the death rate). It is easily verified that the equilibrium probabilities of a birth-death process are given by

$$p_i = \frac{\rho_i}{\sum_{j=0}^{\infty} \rho_j}, \qquad i \geq 0,$$

where $\rho_0 = 1$ and

$$\rho_i = \frac{\lambda_0 \lambda_1 \cdots \lambda_{i-1}}{\mu_1 \mu_2 \cdots \mu_i}, \qquad i \geq 1.$$