Distributed Software Architectures
07: Queuing Theory

Stefan Huber <shuber.lba@fh-salzburg.ac.at>

June 14, 2019

Contents

1 Introduction

Many real-world and artificial processes deal with waiting queues of all kinds. In this lecture on distributed software architectures we had contact with several examples, for instance:

- Message queuing systems are just made to deal with message-oriented communication and message queuing.
- Queues are used for Inter-Process Communication.
- In a client-server architecture connection requests from TCP clients queue up until they are accepted by means of the Berkeley socket API.
- In general, for all kind of I/O access we typically have to serialize access from multiple parties by organizing them in queues.

There is a plethora of further examples in all kind of domains in which queuing systems play an essential role, for instance, traffic systems, telecommunication systems, industrial factory design and material flow analysis, shop design, hospital management, project management, and so forth.

The field of queuing theory [1] deals with the analysis of formal queuing models. In this lecture we concentrate on a simple model called *M/M/1*. It consists of a single queue in which clients queue up and a single server that processes one client after the other, see figure 1. Typical questions addressed by queuing theory are:

(Q1) What is the average number of clients in the queue?

(Q2) What is the average throughput of the queuing system?

(Q3) What is the average waiting time for a client?

Those questions are stochastic in its nature and hence for a specific queuing model, like M/M/1, we have to declare the stochastic models of clients arrival and service time in addition to the modalities on how clients are handled. The answers to the above questions then tell us something about the response times for a TCP connection accepted by a server, the memory consumption of a queue in a message queuing system or the number of patients handled in a hospital per day, assuming that the stochastic models correspond sufficiently well to the problem at hand.

2 The M/M/1 queue model

The most simple example for a queuing system is the so-called M/M/1 queue. The notation "M/M/1" follows the *Kendall's notation* scheme [5], which describes all various kind of queuing models. The general notation "A/B/k" specifies a stochastic model for the client arrival (A), a stochastic model for the service handling (B) and the number (k) of servers. In our case M/M/1 stands for clients that arrive according to a so-called Poisson process (M) at rate λ, service times[1] that follow an exponential distribution (M) at rate μ, and a single server (1), see figure 1.

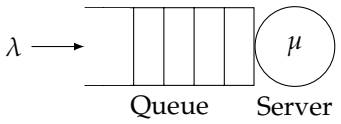

Figure 1: The M/M/1 queue model according to Kandall's notation.

One important result on M/M/1 queues is given by theorem 1, which gives us the probability distribution of the queue lengths. This theorem is a main ingredient in order to prove the questions Q1–Q3. In order to prove this theorem we need to discuss the arrival distribution and service distribution in further detail, which we skip until section 4.1.

Theorem 1. *The M/M/1 model follows a geometric distribution for the random variable X of the queue length. That is,*

$$p_k = \left(\frac{\lambda}{\mu}\right)^k \cdot \left(1 - \frac{\lambda}{\mu}\right), \tag{1}$$

with $p_k = P(X = k)$ denoting the probability that X has the value $k \in \mathbb{N}_0$.

Initially, of course, the queue is empty and does not follow the law given in theorem 1. If we run a M/M/1 for an indefinitely long time, however, we observe the probability distribution claimed in theorem 1. We then speak of a "stationary" distribution. More details are given in section 4.1.

[1] The time a server needs to handle a single client.

3 Performance analysis

For the following performance analysis we mostly follow van Steen and Tanenbaum [3], pages 16–17. We first define the *utilization* U of a service by the fraction of time where the server is active:

Definition 1. The utilization U is defined by

$$U = 1 - p_0 = \frac{\lambda}{\mu}. \tag{2}$$

Using this definition, we can rewrite eq. (1) as follows:

$$p_k = U^k(1 - U) \tag{3}$$

These $p_0, p_1, \ldots$ form the pdf[2] of a geometric distribution G_{1-U} with parameter $1 - U$, cf. [4]. In other words, the random variable X of the length of a M/M/1 queue follows a G_{1-U} distribution, written as $X \sim G_{1-U}$, for which the mean[3] is well known:

$$E(X) = \frac{U}{1 - U}. \tag{4}$$

This allows us to immediately answer our initial question Q1:

Lemma 2. *The (long-term) average number N of clients in a M/M/1 queue is*

$$N = \frac{U}{1 - U} = \frac{\lambda}{\mu - \lambda}. \tag{5}$$

Let us pause for a moment and discuss lemma 2 in more detail. If $\lambda > \mu$ then $N < 0$, which is obviously flawed. This case, however, is already invalid as U would be greater than 1 by eq. (2) and then the p_k in eq. (3) do not form a pdf in the first place. Nonetheless, for a M/M/1 queue the case $\lambda > \mu$ would mean that the client arrival rate is larger than the service rate and therefore the queue simply fills up indefinitely and therefore forms a trivial-pathological case leading to $N = \infty$. The case $\mu = \lambda$ is interesting because eq. (5) says that N is infinite. In this case the arrival rate equals the service rate and, prima vista, it may not appear sound that the queue size actually becomes infinite. However, note that in a single experiment we may run a M/M/1 queue for a long time but do not "observe" that the size grows straight towards infinity. But what eq. (5) says is that if we consider the graph of the pdf p_k then its center of gravity is at infinity, and this just a different interpretation of $E(X) = N$. So if we run an experiment very often for very long time then the average queue length tends towards infinity. To sum up, the only legitimate case is $\lambda < \mu$, in which case $0 \leq U < 1$. So let us make the assumption explicit from now on that $\lambda < \mu$.

In the next step we address the *throughput* X, which we define as average number of clients that are handled by the service per time unit. Clients are only served when the service is active, in which case the service rate is μ:

Definition 2. The throughput X is defined by

$$X = \underbrace{U \cdot \mu}_{\text{Service active}} + \underbrace{(1 - U) \cdot 0}_{\text{Service inactive}} = U \cdot \mu \tag{6}$$

[2] Probability density function.

[3] $E(X) = (1 - U)\sum_{k=0}^{\infty} kU^k = \sum_{k=0}^{\infty} kU^k - \sum_{k=0}^{\infty} kU^{k+1} = \sum_{k=1}^{\infty} kU^k - \sum_{k=1}^{\infty}(k-1)U^k = \sum_{k=0}^{\infty} U^k - 1 = \frac{U}{1-U}$.

As an immediate consequence of the definition we see that the throughput equals the arrival rate of clients in lemma 3, which answers question Q2:

Lemma 3. *The throughput X of an M/M/1 queue is*

$$X = U \cdot \mu = \lambda. \tag{7}$$

Although the lemma itself is trivial from a mathematical point of view, it is still interesting that the throughput is not degraded by larger queue lengths, or dependent on μ for that matter. Note that we require that $\mu > \lambda$ and from time to time the queue is becoming empty in the sense that $p_0 > 0$. That means, in an infinitely long time interval we handle all clients except an indefinitely small fraction and therefore the throughput equals the arrival rate.[4]

For our next step we first require the following result from queuing theory by Little [2]:

Theorem 4 (Little's law)**.** *Denoting by R the average time for a client until being served (response time), it holds that*

$$N = \lambda \cdot R. \tag{8}$$

This theorem is actually not limited to M/M/1 queues but to all kind of queuing system that reach a so-called stationary state. In particular, note that the arrival and service distribution plays no role in this formula. This theorem now answers Q3, the last of our questions:

Corollary 5. *The average response time R for a client is*

$$R = \frac{N}{\lambda} = \frac{1}{\mu - \lambda}. \tag{9}$$

This corollary says that when the arrival rate λ approaches the service rate μ then the average response time tends to infinity. If the service rate μ is much larger then the arrival rate λ then the response time is low.

4 Markov chains

In order to prove theorem 1 we use so-called Markov chains to model certain types of stochastic processes. First, however, let us quickly recap the arrival distribution and service distribution of a M/M/1 queue.

4.1 The M/M/1 queue as a stochastic process

Exponential distribution. In the M/M/1 model, the service time for a client follows an exponential distribution with rate μ. The pdf of the exponential distribution Exp_μ is given by

$$[0, \infty) \to [0, 1] \colon x \mapsto \mu e^{-\mu x}. \tag{10}$$

and the mean is $1/\mu$. It is the continuous counterpart of the discrete geometric distribution and like its counterpart it is *memoryless*.[5]

[4] If $\lambda \geq \mu$ then the utilization U is saturated at 1 and X does not grow beyond μ, no matter how large λ gets. This is what we sometimes observe in the waiting room of a doctor.

[5] Actually, these are the only two examples of memoryless distributions and it has to do with the fact that the only functions f that satisfies $f(a+b) = f(a) \cdot f(b)$ and $f(0) = 1$ are the exponential functions $x \mapsto e^{cx}$.

Memoryless. This is the key property here and can be explained by means of the following example: Typical use cases for the exponential distribution waiting times for events, such the waiting time X for the radioactive decay of an atom that happens at rate μ. Assume we waited for the event for time b without success —we have the information $X \geq b$ therefore—then this does not give as any information about the waiting time from know on.[6] Hence, there is no memory—the past does not matter—in the sense that the probability for $X \geq a + b$ if we already know $X \geq b$ is just like we would start over again and consider the probability $X \geq a$:

$$P(X \geq a + b | X \geq b) = P(X \geq a).$$

Poisson process. The arrivals of clients of the M/M/1 queue are given by a so-called Poisson process. Roughly speaking, a stochastic process describes the random evolution of a random variable over time, e.g., Brownian motion of molecules in space (Wiener process) or the arrival events of clients at a M/M/1 queue over time (Poisson process). In a Poisson process the time between two consecutive events follows an exponential distribution and is therefore again memoryless. In the context of stochastic processes, however, this property also called *Markovian* and means that the faith of the process does not depend on the history but only the current state.

Another example of a stochastic process is the (length of a) M/M/1 queue that evolves over time in a random fashion. Since the arrival time and service time are both memoryless the M/M/1 process is consequently memoryless as well.

4.2 The M/M/1 queue as a Markov chain

In general, a Markov chain is a Markovian stochastic process. Very often, however, it is also assumed that the process evolves in a countable (or even finite) state space, and so do we. Then we can distinguish between two types of Markov chains:

- Discrete-time Markov chains (DTMC), where the time evolves in discrete steps. The so-called Bernoulli process is an example: Initialize X with zero and keep flipping a coin: Add 1 for heads and 0 for tails.
- Continuous-time Markov chains (CTMC), where the time evolves continuously. Since client arrival and service time is continuous, the M/M/1 queue is an example.

A DTMC is typically illustrated like a state machine (automaton) with a finite or a countable infinite number of states and the state transition probabilities as weights on edges. A certain state is the initial state and then in discrete steps a state transition happens with the given probabilities. Hence, for each state the sum of the probabilities of the outgoing edges has to be one. For the previously mentioned Bernoulli process, for instance, we get the DTMC in figure 2.

The above example is also *time-homogeneous*, which means that the transition probabilities are independent of time. The generality of Markov chains make them a versatile tool for many applications, for instance:

- Modeling the behavior of actors in games or real-world scenarios like traffic models.
- Modeling the behavior of software actors for automatic software testing.

[6] If we roll a dice and wait for the first occurrence of a six then knowing that we did not get six for the last k tries does not tell anything about the waiting time for the first six from now on. This example is modeled by the discrete geometric distribution, which is also memoryless.

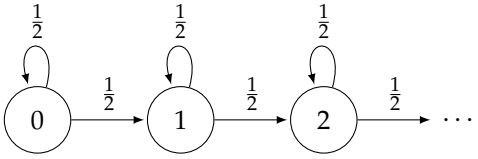

Figure 2: A discrete-time Markov chain of the Bernoulli process for a fair coin. It has a countably infinite number of states.

- Modeling the states of communication protocols and asking for the likelihood to end up in certain states.
- Google's page rank algorithm is based on Markov chains.[7]

The M/M/1 queue, however, is a continuous-time Markov chain. We again use an illustration similar to state machines, but instead of probabilities of state transitions we talk about transition *rates*. Figure 3 depicts an illustration of the M/M/1 queue Markov chain, where we have a transition rate of λ from state k to $k+1$ and a transition rate of μ from state $k+1$ to k, for each $k \in \mathbb{N}_0$.

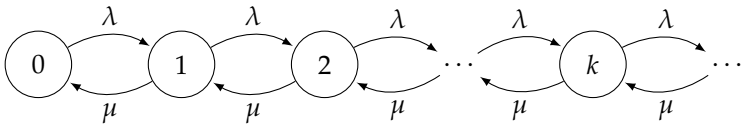

Figure 3: A continuous-time Markov chain of the M/M/1 queue.

Initially we start at state 0. Then a transition to state 1 happens at rate λ meaning that the time until the transition happens is exponentially distributed with rate λ. When we are in state $k > 1$ then two things can happen: a client arrives and we make a transition to $k+1$ or the server is done with handling the current client and we make a transition to $k-1$. The former happens at rate λ and the latter at rate μ. Both "event sources" together happen at rate $\lambda + \mu$ and with probability $\lambda/\lambda+\mu$ the client arrival happens first and we make a transition to $k+1$, and with probability $\mu/\lambda+\mu$ the server is done first and we make a transition to $k-1$.[8]

We can now consider for any time $t \geq 0$ the probability $p_k(t)$ that the Markov chain is in state k. So for any fixed $t \geq 0$ the $p_k(t)$ form a pdf. At time zero we are in state 0 and all the probability mass is at state 0:

$$p_k(0) = \begin{cases} 1 & \text{for } k = 0 \\ 0 & \text{for } k > 0 \end{cases}.$$

Roughly speaking, as time passes probability mass is moving to higher states. If time goes to infinity the Markov chain reaches a stationary distribution, if it actually possesses one. We can actually consider the derivative $p_k'(t)$ which is proportional to

$$\lambda p_{k-1}(t) + \mu p_{k+1}(t) - \lambda p_k(t) - \mu p_k(t).$$

[7]One could consider a website visitor as process, the websites are the states and links are edges. The transition probabilities could be uniform for each node. We could then ask which states are more likely visited and use this probability to rank search results. Also note that the so-called stationary distribution of a finite DTMC is related to spectral theory, namely the eigenvector to the eigenvalue 1.

[8]If X is exponentially distributed at rate μ and Y is exponentially distributed at rate λ then $P(Y < X) = \frac{\lambda}{\lambda+\mu}$.

This formula can bee seen as the "net in-flow" of "probability mass" for the state k. At $t = \infty$, when we reach a stationary distribution, we therefore have $p_k'(t) = 0$, which means that for all $k \geq 1$ we have

$$0 = \lambda p_{k-1} + \mu p_{k+1} - \lambda p_k - \mu p_k. \tag{11}$$

We drop the time in the above equation as time does not play a role anymore in the stationary situation. In some sense, we reached an equilibrium in the "flow of probability mass". This eq. (11) from the theory of Markov chains now allows us to prove theorem 1.

Proof of theorem 1. For $k \geq 1$ we can rewrite eq. (11) and repeatedly re-substitute as follows:

$$\begin{aligned}
\mu p_k &= \lambda p_{k-1} + \mu p_{k+1} - \lambda p_k \\
&= \lambda p_{k-1} + (\lambda p_k + \mu p_{k+2} - \lambda p_{k+1}) - \lambda p_k \\
&= \lambda p_{k-1} + \mu p_{k+2} - \lambda p_{k+1}
\end{aligned}$$

We can repeat this re-substitution arbitrarily often and obtain for arbitrary $n \geq 0$

$$\mu p_k = \lambda p_{k-1} + \mu p_{k+n+1} - \lambda p_{k+n}.$$

As the p_k form a pdf we know that $\sum_{k=0}^{\infty} p_k$ converges and therefore $\lim_{n\to\infty} p_n = 0$. We can use this fact in the above equation and obtain

$$p_k = \frac{\lambda}{\mu} p_{k-1} \tag{12}$$

and after resolving the recursion we obtain

$$p_k = \left(\frac{\lambda}{\mu}\right)^k \cdot p_0. \tag{13}$$

Again using the fact that the p_k form a pdf we obtain

$$\begin{aligned}
1 &= \sum_{k=0}^{\infty} p_k \\
&= p_0 \cdot \sum_{k=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^k \\
&= p_0 \frac{1}{1 - \frac{\lambda}{\mu}}
\end{aligned}$$

which finally yields

$$p_0 = 1 - \frac{\lambda}{\mu} \tag{14}$$

$$p_k = \left(\frac{\lambda}{\mu}\right)^k \cdot \left(1 - \frac{\lambda}{\mu}\right). \tag{15}$$

□

References

[1] Leonard Kleinrock. *Queueing Systems*. Vol. 1: *Theory*. New York, USA: Wiley-Interscience, 1975. ISBN: 0471491101.

[2] John D. C. Little. "A Proof for the Queuing Formula: $L = \lambda W$." In: *Operations Research* 9.3 (1961), pp. 383–387. DOI: 10.1287/opre.9.3.383.

[3] Maarten van Steen and Andrew S. Tanenbaum. *Distributed Sytems*. 3rd ed. CreateSpace Independent Publishing Platform, Feb. 2017. ISBN: 978-1543057386.

[4] *Wikipedia: Geometric Distribution*. June 8, 2019. URL: https://en.wikipedia.org/wiki/Geometric_Distribution.

[5] *Wikipedia: Kendall's notation*. June 8, 2019. URL: https://en.wikipedia.org/wiki/Kendall's_notation.