Queueing Theory

My notes from the mini-course `https://www.cse.wustl.edu/~jain/queue`

1 Introduction

1.1 Notations

There are six components in a queue, which are the following (Kendall notations are in brackets).

1. *Arrival process (A)*: Let job j comes at time t_j. Then, the inter-arrival times are defined as $\tau_j \triangleq t_j - t_{j-1}$, which forms a sequence of i.i.d. random variables (in most problems). We are interested in the inter-arrival times not the arrival times. Kendall notation for some distributions are
   - M: Memoryless (Exponential). Expected time to the next arrival is a constant regardless of the time since the last arrival. Remembering the past history does not help.
   - E: Erlang. The time is the sum of several exponential random variables.
   - H: Hyper-exponential. The variable takes x_i with probability p_i where x_i is exponential distributed.
   - D: Deterministic.
   - G: General (Results valid for all distributions).
2. *Service time distribution (S)*: It is the time each job is in service. It is also i.i.d. with possible distributions: M, E, H, D, or G.
3. *Number of servers (m)*
4. *Buffering (B)*: Buffer size. Infinite by default.
5. *Population size (K)*: Finite or infinite. Infinite by default.
6. *Service discipline (SD)*: Some of the famous examples are the following.
   - *FCFS*: First-Come-First-Serve. (Default)
   - *LCFS*: Last-Come-First-Serve.
   - *LCFS-PR*: LCFS with Preemptive Resume (Newly arriving thread at ready list preempts the currently running thread. Preempted thread is appended to ready list. In case of no further arrivals, the ready list is processed without preemption).
   - *RR*: Round-Robin.

Remark 1. *The queue length is number of jobs waiting in the queue plus the number of jobs in service.*

Remark 2. *Group arrivals/service Several number of jobs come at the same time (also can be a train-like arrival). Then, we have to have the distribution for the group inter-arrival time and group size distribution. For example, $M^{[x]}$ where M indicates the group inter-arrival time and x can be a random variable which represents the group size.*

In the following, we define some key variables (by convention) for the analysis of queues.

- λ: Mean arrival rate ($\frac{1}{E[\tau]}$). It can be a function of the state of the system.
- μ: Mean service rate per server ($\frac{1}{E[s]}$). Total service rate for m servers is $m\mu$.
- n: Number of jobs in the system (queue length). $n = n_s + n_q$ where n_s is the number of jobs receiving service and n_q is the number of jobs waiting.
- r: Response time or the time in the system, which is equal to time waiting plus time receiving service.

1.2 Rules for All Queues

Note that the following rules apply to $G/G/m$ queues.

1. *Stability Condition*: Arrival rate must be less than service rate.

   $$\lambda < m\mu.$$

   Note that finite-population or finite-buffer systems are always stable. Instability means infinite queue.

   Remark 3. *It is a sufficient but not necessary condition. $D/D/1$ queue is stable at $\lambda = \mu$.*

2. *Number of jobs*: Note that n, n_q, and n_s are random variables.

   $$E[n] = E[n_q] + E[n_s].$$

   If the service rate is independent of the number in the queue, we have $Cov(n_s, n_q) = 0$. Hence,

   $$Var[n] = Var[n_q] + Var[n_s].$$

3. *Number versus Time*: If jobs are not lost due to insufficient buffers, we have: Mean number of jobs in the system = Arrival rate × Mean response time.
4. *Little's Law*: Mean number of jobs waiting = Arrival rate × Mean waiting time.
5. *Time in System* Note that r, w, and s are random variables.

   $$E[r] = E[w] + E[s].$$

   If service time is independent of the number of jobs in the queue, we have $Cov(w, s) = 0$. Hence,

   $$Var[r] = Var[w] + Var[s].$$

Little's Law : Little's law states that: Mean number of jobs waiting = Arrival rate × Mean waiting time. This relationship applies to all systems or parts of systems in which the number of jobs entering the system is equal to those completing service. It is based on a black-box view with only arrivals and departures of the system.

Proof. We consider a large enough time period T such that there are N arrivals and N departures within this time period. We define the total time spent inside the system by the all jobs as J. Then, the arrival rate can be expressed as N/T, for each job, the mean time in the system is J/N, and the mean number of jobs in the system is J/T. Hence, we have

$$\frac{J}{T} = \frac{N}{T} \times \frac{J}{N}.$$

□

1.3 Poisson Processes

This is a process with discrete state and continuous time.

- The inter-arrival times τ are exponentially distributed (i.e., $f(\tau) = \lambda e^{-\lambda\tau}$ and $E[\tau] = \frac{1}{\lambda}$). Hence, number of arrivals n in any given interval are Poisson distributed (i.e., $n \sim Poisson$, $P(n\ arrivals\ in\ t) = (\lambda t)^n \frac{e^{-\lambda t}}{n!}$, and $E[n] = \lambda t$).
- Merging property $\lambda = \sum_{i=1}^{k} \lambda_i$ and Splitting property $\lambda = \sum_{i=1}^{n} p_i \lambda_i$.
- If the arrivals to a single server with exponential service time (with mean rate μ) are Poisson with mean rate λ, the departures are also Poisson with the same rate λ, provided $\lambda < \mu$. For multiple server, same conclusion with the condition being $\lambda < \sum_{i=1}^{n} \mu_i$. This tells us that the departure is also memoryless and the rate is also λ since the system is stable.
- *PASTA Property*: the probability of the state (can be number of jobs in waiting) as seen by an outside random observer is the same as the probability of the state seen by an arriving job.

2 Analysis of Single Queue

We first consider a Birth-and-Death process. When the system is in state n, it has n jobs in it. The new arrivals take place at a rate λ_n. The service rate is μ_n. We also assume that both the inter-arrival times and service times are exponentially distributed. Then, the steady-state probability p_n is

$$p_n = \frac{\lambda_0 \lambda_1 \cdots \lambda_{n-1}}{\mu_1 \mu_2 \cdots \mu_n} p_0 = p_0 \prod_{j=0}^{n-1} \frac{\lambda_j}{\mu_{j+1}}, \quad n = 1, 2, ..., \infty,$$

where p_0 is the probability of being at state 0. It can be calculated using the fact that all the probabilities must add up to one. More precisely,

$$p_0 = \frac{1}{\sum_{n=1}^{\infty} \prod_{j=0}^{n-1} \frac{\lambda_i}{\mu_{j+1}}}.$$

(The proof is based on considering a small enough time interval such that only one event will happen.)

M/M/1 Queue The state of the system will be the number of jobs in the system. Then, the process can be characterized by a Birth-and-Death process with

$$\lambda_n = \lambda, \quad n = 0, 1, 2, ..., \infty,$$

$$\mu_n = \mu, \quad n = 1, 2, ..., \infty.$$

Then, the probability of n jobs in the system is

$$p_n = \left(\frac{\lambda}{\mu}\right)^n p_0, \quad n = 1, 2, ..., \infty.$$

We define $\rho \triangleq \lambda/\mu$ as traffic intensity. For the stability $\rho < 1$. Then, we can get

$$p_n = (1-p)\rho^n, \quad n = 0, 1, ..., \infty.$$

Then, we can calculate several quantities that characterizes the system.

- *Utilization of the server*: Probability of having one or more jobs in the system is $U = \rho$.
- *Mean number of jobs in the system*:

$$E[n] = \sum_{n=0}^{\infty} np_n = \frac{\rho}{1-\rho}.$$

- *Variance of the number of jobs in the system*:

$$Var[n] = E[n^2] - (E[n])^2 = \frac{\rho}{(1-\rho)^2}.$$

- *Mean response time*:

$$E[r] = \frac{1}{\mu(1-\rho)}.$$

  This can be calculated using rule: $E[n] = \lambda \cdot E[r]$.

Remark 4. *We can use the Rules for All Queues to calculate other quantities such as $E[n_q]$, $E[r]$, $E[w]$, probability of finding n or more jobs in the system, and so on.*

M/M/m Queue : Similar to before, the number of jobs in the system can be cast into a Birth-and-Death process with birth rate $\lambda_i = \lambda$ for $i = 0, 1, 2, ..., \infty$ and death rate

$$\mu_i = \begin{cases} i\mu & i = 1, 2, ..., m \\ m\mu & otherwise. \end{cases}$$

Then, the quantities discussed for M/M/1 queue can be calculated similarly for M/M/m queue. For example

$$\rho = \frac{\lambda}{m\mu}, \quad U = \rho.$$

M/M/m/B Queue : Very similarly,

$$\lambda_i = \lambda, \quad i = 0, 1, 2, ..., B-1.$$

$$\mu_i = \begin{cases} i\mu & i = 1, 2, ..., m \\ m\mu & i = m+1, ..., B. \end{cases}$$

$$\rho = \frac{\lambda}{m\mu}, \quad U = \rho(1 - p_B).$$

3 Queueing Networks

- *Queueing Network*: It is the model in which jobs departing from one queue arrive at another queue (or possibly the same queue).
- *Open Queueing Network*: The arrivals and departures are external.
  - The number of jobs in the system is a random variable.
  - The throughput is the arrival rate.
  - The goal of analysis is to characterize the distribution of the number of jobs in the system.
- *Closed Queueing Network*: No external arrivals or departures (i.e., OUT is connected to IN). We need to identify which is IN and which is OUT.
  - Total number of jobs in the system is constant.
  - Throughput is the flow of jobs in the OUT-to-IN link.
  - The goal of analysis is to determine the throughput.
- *Mixed Queueing Network*: It allows both types of jobs (closed or open).
  - All jobs of a single class have the same service demands and transition probabilities.
  - Within each class, the jobs are indistinguishable.

Example 1. *Series Network We consider a tandem of M M/M/1 queues. Then, each queue can be analyzed independently because the arrivals for each queue are all Poisson with rate λ. Then, the joint probability of queue lengths is*

$$P(n_1, n_2, ..., n_M) = p_1(n_M)p_2(n_M) \cdots p_M(n_M).$$

This kind of network is called product form network.

More formally,

Definition 1. *Product Form Network we define the product form network as the network in which*

$$P(n_1, n_2, ..., n_M) = \frac{1}{G(N)} \prod_{i=1}^{M} f_i(n_i),$$

where $f_i(n_i)$ is some function of the number of jobs at the ith facility and $G(N)$ is a normalizing constant and is a function of the total number of jobs in the system.

Remark 5. *The product form network are not necessarily memoryless.*

Theorem 1. *Any arbitrary open network of m server queues (M/M/m queues) with exponentially distributed service times has a product form.*

Remark 6. *It shows that any arbitrary closed networks of m-server queues with exponentially distributed service times also have a product form solution. Note that there is no arrivals for closed network and the internal flows are not Poisson. Later on it showed that product form solutions exist for an even broader class of networks.*

4 Operational Laws

The laws in this section do not require any assumptions about the distribution of service times or inter-arrival times. The only assumption we made here is that the system will not loss anything while in operation.

We first define some operational quantities. These quantities are called operational quantity is because they can be directly measured during a finite observation period.

- *Observation interval*: T_i.
- *Number of arrivals*: A_i.
- *Number of completion*: C_i.
- *Busy time*: B_i.

For these quantities, we can calculate the following.

- *Arrival Rate*: $\lambda_i = \frac{A_i}{T}$.
- *Throughput*: $X_i = \frac{C_i}{T}$.
- *Utilization*: $U_i = \frac{B_i}{T}$.
- *Mean Service Time*: $S_i = \frac{B_i}{C_i}$.

Utilization Law :

$$U_i = X_i S_i.$$

Forced Flow Law : It considers a system with multiple devices and relates the throughput of the system to that of individual device. If observation period T is such that $A_i = C_i$ and each job makes V_i requests for ith device in the system, then, we have

$$C_i = C_0 V_i,$$

where V_i is called visit ratio (It can be larger than 1 when some jobs needs to pass the queue multiple times). We know that the system throughput is $X = C_0/T$. Then, we have

$$X_i = X V_i.$$

Combing the above two laws, we get

$$U_i = X_i S_i = X V_i S_i = X D_i,$$

where $D_i \triangleq V_i S_i$ is the total service demand on the device for all visits of a job.

Definition 2 (Bottleneck Device)**.** *The device with the highest D_i has the highest utilization and is the bottleneck device.*

Here, we investigate another quantity: transition probabilities p_{ij}, which is defined as the probability of a job moving to jth queue after service completion at ith queue. We define p_{i0} as the probability of a job existing the system after service completion at ith queue.

Remark 7. *Visit ratios and transition probabilities are equivalent in the sense that given one we can always find the other. To this end, we notice that*

$$C_j = \sum_{i=0}^{M} C_i p_{ij}.$$

Then, we divide both side by C_0, we get

$$V_j = \sum_{i=0}^{M} V_i p_{ij}.$$

This is called visit ratio equation. Note that $V_0 = 1$ since every complete job will exist the system.

Little's Law :

$$Q_i = \lambda_i R_i,$$

where Q_i is the mean number in the device and R_i is the mean time in the device. If the job flow is balanced, the arrival rate is equal to the throughput. Hence, $Q_i = X_i R_i$.

General Response Time Law : We consider s system where there is one terminal per user and the rest of the system (central subsystem) is shared by all the users. According to Little's Law, for the central subsystem, we have

$$Q = XR.$$

where Q is the total number of jobs in the system, X is the system throughput, and R is the system response time. Then, we have

$$Q = Q_1 + Q_2 + \cdots + Q_M,$$

$$XR = X_1 R_1 + X_2 R_2 + \cdots + X_M R_M.$$

Then, dividing both side by X and applying the forced flow law yield

$$R = \sum_{i=1}^{M} R_i V_i.$$

Interactive Response Time Law : We define Z as the think time, which is the time a typical user delays, or thinks, between submitting commands. Then, the total cycle time of request is $R+Z$ and each user generates about $T/(R+Z)$ requests in T. For a system with N users,

$$X = N(T/(R+Z))/T = N/(R+Z).$$

Or,

$$R = (N/X) - Z.$$

4.1 Bottleneck Analysis

- The bottleneck device is the key limiting factor in achieving higher throughput.
- Improving the bottleneck device will provide the highest payoff in terms of system throughput.
- Improving other devices will have little effect on the system performance.
- Identifying the bottleneck device should be the first step in any performance improvement project.

Throughput $X(N)$ and response times $R(N)$ of the system are bound as follows:

$$X(N) \leq \min\{\frac{1}{D_{max}}, \frac{N}{D+Z}\},$$

$$R(N) \geq \max\{D, ND_{max} - Z\},$$

where N is the number of users (jobs), D_{max} is the maximum total service demand and $D = \sum_i D_i$ is the sum of total service demands on all devices except terminals.

Proof.

- For the bottleneck device, we have $U_b = XD_{max} \leq 1$. Hence,

$$X \leq \frac{1}{D_{max}}$$

- With just one jobs in the system, there will be no queueing (waiting). Hence, the response time is the total of service demand.

$$R(1) = D_1 + D_2 + \cdots + D_M = D.$$

With more than one jobs, there will be queueing (waiting). Hence, the response time will be higher.

$$R(N) \geq D.$$

- Applying the interactive response time law to the bounds:

$$R = (N/X) - Z.$$

We can get

$$R(N) = \frac{N}{X(N)} - Z \geq ND_{max} - Z.$$

$$X(N) = \frac{N}{R(N)+Z} \leq \frac{N}{D+Z}.$$

□

We define the keen N^* as the N such that the two quantities in min and min are equal to the corresponding ones. More precisely, we have

$$N^* = \frac{D+Z}{D_{max}}.$$

N^* is called optimal operational point (it is optimal in the sense that it yields the largest power which is X/R. It is another way to find the knee). If the number of jobs is more than N^*, then we can say with certainty that there is queueing somewhere in the system.

Remark 8. *The queue starts as long as the number of jobs $N > 1$. However, when N is small, we will have many idle devices. When N is large, many jobs will wait for a long time. Hence, there is a optimal N which can be viewed as a balance point.*

5 Mean Value Analysis

Mean-value analysis (MVA) allows solving *closed* queueing networks. Meanwhile, it only gives mean values not other statistics like variance. The algorithm can be divided into four steps.

1. Given a closed queueing network with N jobs

   $$R_i(N) = S_i(1 + Q_i(N-1)),$$

   where $Q_i(N-1)$ is the mean queue length at ith device with $N-1$ jobs in the network.

   - We assume that the service time is exponential.
   - Since $N = 0$ is easy to analysis, we can calculate $R_i(N)$ for all N iteratively.

2. Given the response times at individual devices, the system response time using the general response time law is

   $$R(N) = \sum_{i=1}^{M} R_i(N).$$

3. The system throughput using the interactive response time law is

   $$X(N) = \frac{N}{R(N)+Z}.$$

4. The device queue lengths with N jobs in the network using Little's law are

   $$Q_i(N) = X_i(N)R_i(N) = X(N)V_iR_i(N).$$

The algorithm can be summarized as the following. MVA is applicable when the following assumptions are satisfies.

- The network is a product form network with exponentially distributed service times.
- Job flow balance. (IN = OUT)
- Only one job in or out at a time.

Algorithm 1 Mean Value Analysis

Require:

N = number of users.
Z = think time.
M = number of devices.
S_i = service time of ith device.
V_i = number of visits to ith device.

1: **procedure** MVA(N,Z,M,S_i,V_i)
2: Initialize $Q_i = 0 \quad 1 \leq i \leq M$
3: **for** $n = 1$ to N **do**
4: **for** $i = 1$ to M **do**
5: $R_i = S_i(1 + Q_i)$ for fixed capacity device.
6: $R_i = S_i$ for delay centers.
7: $R = \sum_{i=1}^{M} R_i V_i$
8: $X = \frac{N}{Z+R}$
9: **for** $i = 1$ to M **do**
10: $Q_i = X V_i R_i$
11: $X_i = X V_i$
12: $U_i = X S_i V_i$
return X, R, Q_i, R_i, U_i

- Only fixed-capacity service centers or delay centers. (Can be extended to load dependent servers)
- Device Homogeneity: A device's service rate for a particular class does not depend on the state of the system in any way except for the total device queue length and the designated class's queue length. It implies the following
  - A job may not be present (waiting for service or receiving service) at two or more devices at the same time.
  - A device renders service whenever jobs are present; its ability to render service is not controlled by any other device.
  - Interaction among jobs is limited to queueing for physical devices, for example, there should not be any synchronization requirements.
  - Interaction among jobs is limited to queueing for physical devices, for example, there should not be any synchronization requirements.
  - If service rates differ by class, the service rate for a class depends only on the queue length of that class at the device and not on the queue lengths of other classes. This means that the servers do not discriminate against jobs in a class depending on the queue lengths of other classes. (No priority)