Queueing Models – Analytic Results

These lecture notes are concerned with an approach to obtain mathematical relationships between input parameters and performance metrics by considering their values over a finite observation interval. A link in a network (or a device in a computer system) is often modeled by a server in a queueing system. This server processes packets (or jobs) submitted by users. Common performance metrics are utilization, packet delay (or response time), and throughput. As to input parameters, popular choices are interarrival time and service time. For convenience, we will use the term job to represent an entity that receives service from a server in a queueing system.

1. Input parameters

Interarrival time is the time between successive job arrivals. Arrival rate (instead of interarrival time) may also be used as an input parameter. Their relationship is as follows:

arrival rate = 1 / mean interarrival time

Service time is the amount of time required to process a job, and is given by:

service time = service requirement / server capacity

Service requirement is measured in units of work. Server capacity, on the other hand, is the rate at which work is done and is measured in units or work per second. As an example, for a network link, a unit of work could be a bit; in this case, the server capacity is in bits per second. Similar to arrival rate, the relationship between service rate and service time is given by:

service rate = 1 / mean service time

A finite-population model is often used to study the performance of interactive systems or web based applications. See Figure 1.

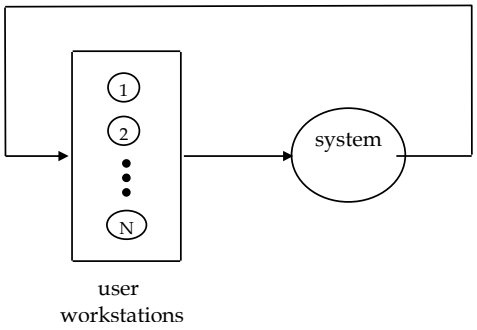

Figure 1

In this model, each of the *N* users behaves as follows. After a "think time", the user submits a job and then waits for the system response. When the response is received, the next think time of this user will begin, and the next job from this user will be submitted at the end of the think time. For this model, the input parameters in connection with job arrivals are the number of users *N* and the think time (and not the interarrival time).

2. Analytic Results – Single Server Case

Suppose the activities in a system are observed for time period of length L (or from 0 to L). Let

n = number of jobs arrived in (0, L) = number of jobs processed in (0, L)
x_j = service time of j^{th} job

2.1 Throughput

The arrival rate (denoted by λ) is given by $\lambda = n/L$. The throughput Y is the rate at which jobs are completed. Since n jobs are processed in (0, L), the throughput is also given by n/L. We thus have:

$$Y = \lambda = n/L$$

2.2 Utilization

The total time that the server is busy in (0, L) is $\sum_{j=1}^{n} x_j$. The utilization U, defined to be the percentage of time that the server is busy, is given by:

$$U = \frac{1}{L}\sum_{j=1}^{n} x_j = \lambda S \quad \text{or} \quad U = YS$$

where $S = \frac{1}{n}\sum_{j=1}^{n} x_j$ is the mean service time. Note that $0 \le U \le 1$ because the utilization cannot be negative and the server cannot be busy more than 100% of the time.

2.3 Mean Response Time – Little's Law

In Figure 2, we show, for each job processed by the server, the time at which the job arrives to the system and the time at which the job departs from the system. The number of jobs in the system as a function of time is also shown. It can be seen that the total area in Figure 2 is the sum of the response times of the n jobs that are processed in (0, L); n = 4 in this example. This total area is also equal to QL where Q is the mean number of jobs in the system. We thus have

$$\frac{1}{L}\sum_{j=1}^{n} r_j = Q$$

or $$\lambda R = Q \quad \text{or} \quad YR = Q \qquad (1)$$

where $R = \frac{1}{n}\sum_{j=1}^{n} r_j$ is the mean response time. Equation (1) is known as Little's Law. Note that Little's Law characterizes the relationship between one workload parameter λ and two performance metrics R and Q. This is often useful in the analysis of queueing models because if analytic result for one of the performance metrics (R or Q) is available, analytic result for the other metric can easily be obtained.

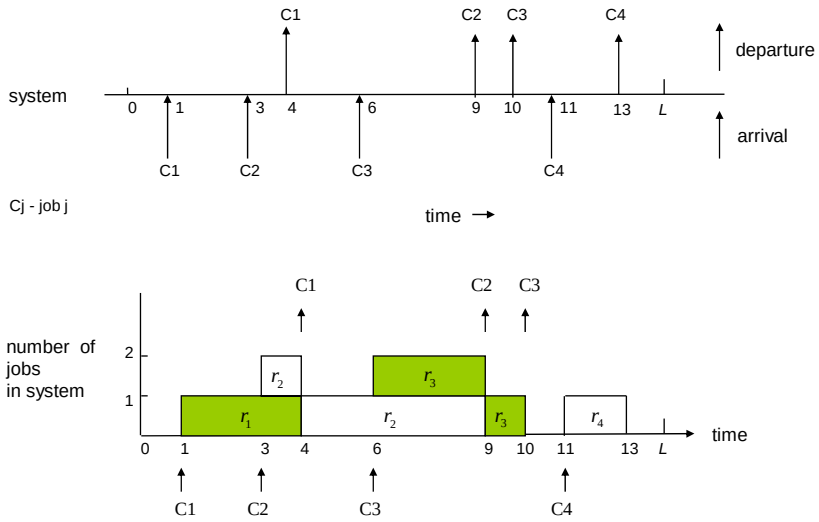

Figure 2

2.4 Finite population Model

Analytic results can also be obtained for the finite population model shown in Figure 1. The input parameters are (i) *N*, the number of users, (ii) *Z*, the mean think time, and (iii) *S*, the mean service time. We again consider an observation period *L* where the number of jobs arrived to the system in (0, *L*) = number of jobs processed by the system in (0, *L*).

For the finite population model, the throughput *Y* is the rate at which responses are returned to the user workstations. As discussed in Section 2.1, *Y* is the same as the arrival rate of jobs to the system. Let $E[N_t]$ and $E[N_s]$ be the mean number of users in the thinking state and the mean number of jobs in the system, respectively. Applying Little's Law to the user workstations, we get $YZ = E[N_t]$. Applying Little's Law to the system, we get $YR = E[N_s]$. Since $E[N_s] + E[N_t] = N$, we have

$$Y(Z + R) = N \text{ or } R = N/Y - Z \qquad (2)$$

This equation show the relationship between two input parameters *N* and *Z* and two performance metrics *R* and *Y*.

A lower bound for the mean response time can also be obtained. Since $U = YS \le 1$, we have

$$R \ge NS - Z$$

Another lower bound is $R \geq S$ since the mean response time cannot be smaller than the mean service time. N^*, the value of N at which the two bounds meet, is referred to as the system saturation point. We thus have $N^*S - Z = S$ or

$$N^* = (S + Z)/S$$

The lower bound mean response time can then be written as follows:

$$R \geq \begin{cases} S & 1 \leq N \leq N^* \\ NS - Z & N > N^* \end{cases}$$

As to the throughput, we note from Equation (2) that $Y = N/(R + Z)$. Applying the lower bound result for R, we get the following upper bound for throughput:

$$Y \leq \begin{cases} N/(S + Z) & 1 \leq N \leq N^* \\ 1/S & N > N^* \end{cases}$$

2.5 Remarks

The results in Sections 2.1 to 2.4, for throughput, utilization, and Little's Law, respectively, are not affected by the resource management scheme (e.g., scheduling discipline) used by the system. They are also not affected by the probability distribution of interarrival time, service time, or think time. Also, Little's Law can be applied to any types of queueing models, from single server queue to a network of queues. An important requirement, however, is that the system is stable. In the example above, this is indicated by the scenario considered where all jobs that arrived in (0, L) are processed in (0, L), or the system returns to the idle state at time L.

3. Analytic Results – Queueing Networks

3.1 Description of Queueing Networks

A queueing network consists of a network of service centres. Jobs move from one server centre to another according to transition probability (denoted by p_{ij}). p_{ij} is defined as follows:

p_{ij} = probability that a job, after completing service at service centre i, will next move to service centre j.

We consider three types of queueing networks:

(1) Open network model with a single server at each service centre – There are M service centres (1 to M). p_{ij} is defined for $i, j = 1, 2, \ldots, M$. In addition, we define p_{iM+1} to be the probability that a job will depart from the network after completing service at server i. M+1 is an artificial service centre used to represent departures from the network. We also require that $\sum_{j=1}^{M+1} p_{ij} = 1$ for all i.

(2) Closed network model with a single server at each service centre – There are M service centres (1 to M). p_{ij} is defined for $i, j = 1, 2, \ldots, M$. We require that $\sum_{j=1}^{M} p_{ij} = 1$ for all i.

(3) Closed network model for interactive systems – There are M+1 service centres (0 to M); service centre 0 has N user workstations (similar to the finite-population model); there is a single server at each of service centres 1 to M. p_{ij} is defined for $i, j = 0, 1, \ldots, M$. We require that $\sum_{j=0}^{M} p_{ij} = 1$ for all i.

3.2 Throughput, Utilization, and Mean Response Time

3.2.1 Open Network Model

We note from our results in Section 2.1 that for a single server, the throughput Y is equal to the arrival rate λ when the system is stable. Consider a server, say server i, in an open network. This server sees two types of arrivals:

(i) External – jobs arriving from outside the network, and

(ii) Internal – jobs arriving at server i immediately after completing service at some other server or completing service at server i itself.

This is shown in Figure 3 below. The total arrival rate is the sum of the external and internal arrival rates. Let γ_i and λ_i be the external arrival rate and total arrival rate to server i, respectively. We have:

$$\lambda_i = \gamma_i + \text{internal arrival rate}$$

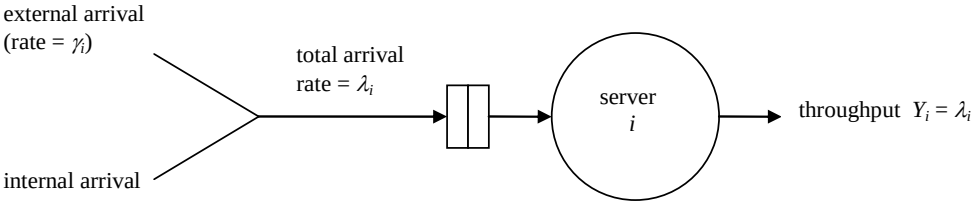

Figure 3

An internal arrival can come from any server in the network, including server i. Consider jobs completing service at server j (j = 1, 2, …, M). Each of these jobs has probability p_{ji} of visiting server i next. Since the departure rate of jobs from server j (or throughput of server j) is λ_j, we conclude that the internal arrival rate from server j to server i is $\lambda_j p_{ji}$. We can thus write:

$$\lambda_i = \gamma_i + \sum_{j=1}^{M} \lambda_j p_{ji} \qquad i = 1, 2, \ldots, M \qquad (3)$$

We have M equations and M unknowns. Exact values of the total arrival rate λ_i can be determined from Equation (3).

The throughput at server i (denoted by Y_i) is equal to the total arrival rate λ_i. We thus have

$$Y_i = \lambda_i \qquad i = 1, 2, \ldots, M$$

Let U_i be the utilization of server *i*. U_i is given by:

$$U_i = \lambda_i S_i$$

where S_i is the mean service time at server *i*. As to the mean response time, we apply Little's Law to server *i* and obtain

$$\lambda_i R_i = Q_i$$

where R_i and Q_i are the mean response time and mean number of jobs at server *i*, respectively.

The system throughput, *Y*, defined to be the number of jobs departing from the network per second, can be determined as follows. A departure from the network may occur at any of the *M* servers. Consider server *i*. A job completing service at this server has probability p_{iM+1} of departing from the network immediately. The system throughput can therefore be written as:

$$Y = \sum_{i=1}^{M} \lambda_i p_{iM+1} \tag{4}$$

Since the network model is stable, *Y* is also the total external arrival rate to the network, i.e.,

$$Y = \sum_{i=1}^{M} \gamma_i \tag{5}$$

In Appendix A, we show that Equations (4) and (5) yield the same result.

Let *R* be the mean response time over all jobs processed by the servers in the network and *Q* be the mean number of jobs in the network. From Little's law, we have $R = Q/\gamma$. Since $Q = \sum_{i=1}^{M} Q_i$ and $\lambda_i R_i = Q_i$, *R* is given by:

$$R = \frac{1}{\gamma} \sum_{i=1}^{M} \lambda_i R_i$$

3.2.2 Closed Network Model with a Single Server at each Service Centre

A closed network has no external arrivals and no departures from the network. The total number of jobs in the network is a constant. With no external arrivals, γ_i is zero for all *i*, and Equation (3) becomes:

$$\lambda_i = \sum_{j=1}^{M} \lambda_j p_{ji} \qquad i = 1, 2, ..., M \tag{6}$$

The *M* equations in (6) are not linearly independent, e.g., the M^{th} equation can be obtained as the sum of the first *M*-1 equations. As a result, we can only determine the relative values of the total arrival rates at the various servers. Specifically, we can determine the values of λ_i/λ_j for all *i, j*. Since $Y_i = \lambda_i$, we can also determine the relative values of the throughputs at the various server, namely, the ratio Y_i/Y_j for all *i, j*. Specifically, $Y_i/Y_j = \lambda_i/\lambda_j$.
In addition, since $U_i = \lambda_i S_i$, we can determine the relative utilizations, i.e.,

$$U_i / U_j = \lambda_i S_i / \lambda_j S_j$$

Let *b* be the server that has the highest utilization, i.e., $U_b / U_i > 1$ for all $i \neq b$. Since $U_b \leq 1$, we have the following upper bound for U_i:

$$U_i \leq \lambda_i S_i / \lambda_b S_b$$

for all $i \neq b$.

An example of this type of closed network models is a cyclic queue which has been used to determine the throughput of window flow control.

3.2.3 Closed Network Model for Web Applications

Consider a web application model where service centre 0 represents *N* user workstations. The behavior of each user is similar to that for the finite-population model. Service centres 1 and 2 represent the web/application server and database server, respectively. Each user, after spending a think time at his/her workstation, submits a job. This job will first visit the web/application server ($p_{01} = 1$). After receiving the required service at this server, a job will visit the database server with probability p_{12}, or a response will be returned to the user workstation with probability p_{10}. After receiving service at the database server, a job will return to the web/application server with probability 1 ($p_{21} = 1$).

The relative arrival rates can be determined from:

$$\lambda_i = \sum_{j=0}^{M} \lambda_j p_{ji} \qquad i = 0, 1, 2 \qquad (7)$$

The system throughput *Y* is given by λ_0, the total arrival rate to the user workstations.

We define V_i, the visit ratio of server *i*, to be the mean number of visits by a job to server *i*, *i* = 1, 2. V_i can be written as:

$$V_i = \lambda_i / \lambda_0 \qquad (8)$$

V_i can be determined from Equation (7) because it is equal to the ratio of two total arrival rates. For the web application model, Equation (7) is given by:

$$\lambda_0 = \lambda_1 p_{10}$$
$$\lambda_1 = \lambda_0 + \lambda_2$$
$$\lambda_2 = \lambda_1 p_{12}$$

The solution to V_i is:

$$V_1 = \lambda_1 / \lambda_0 = 1 / p_{10}$$
$$V_2 = \lambda_2 / \lambda_0 = (\lambda_2 / \lambda_1)(\lambda_1 / \lambda_0) = p_{12} / p_{10}$$

Since the system throughput $Y = \lambda_0$, we have $\lambda_i = YV_i$ (see Equation (8)). The utilization of server *i* is then given by:

$$U_i = \lambda_i S_i = YV_i S_i = YD_i \qquad (9)$$

where

$$D_i = V_i S_i$$

is the total mean service time at server *i* per job. From Equation (9), we can write:

$$\frac{U_i}{U_j} = \frac{YD_i}{YD_j} = \frac{D_i}{D_j} \tag{10}$$

Equation (10) allows us to determine the relative values of the utilizations at the various servers. We can also determine an upper bound for the utilization of each server. Let *b* be the server that has the highest utilization, i.e., $U_b/U_i > 1$ for all $i \neq b$. Since $U_b \leq 1$, we have

$$U_i \leq D_i/D_b$$

for all $i \neq b$.

In addition, we can obtain results for the mean response time of jobs submitted by a user. Let *R*(*N*) and *Y*(*N*) be the mean response time and throughput when the number of users is *N*, respectively. By considering the web/application and database servers together as "system", the results in Section 2.4 for the finite population model are directly applicable. We thus have

$$R(N) = N/Y(N) - Z \tag{11}$$

where Z is the mean think time.

We can also obtain a lower bound for the mean response time. Recall that *b* is the server that has the highest utilization, i.e., $U_b/U_i > 1$ for all $i \neq b$. Since $U_b \leq 1$, we have, from Equation (9), $Y(N) \leq 1/D_b$. Substituting this inequality into Equation (11), we obtain the following lower bound:

$$R(N) \geq ND_b - Z$$

Let $D = \sum_i D_i$ be the sum of the total mean service times of a job at all the servers. Another lower bound is $R(N) \geq D$ because the mean response time cannot be smaller than *D*. The system saturation point N^* is now given by:

$$N^* = (D + Z)/D_b$$

The lower bound mean response time can then be written as follows

$$R(N) \geq \begin{cases} D & 1 \leq N \leq N^* \\ ND_b - Z & N > N^* \end{cases}$$

As to the throughput, we note from Equation (11) that $Y(N) = N/(R(N) + Z)$. Applying the lower bound result for $R(N)$, we get the following upper bound for throughput:

$$Y(N) \leq \begin{cases} N/(D + Z) & 1 \leq N \leq N^* \\ 1/D_b & N > N^* \end{cases}$$

Appendix A

Summing $\lambda_i = \gamma_i + \sum_{j=1}^{M} \lambda_j p_{ji}$ from i = 1 to M, we have:

$$\begin{aligned}\sum_{i=1}^{M} \lambda_i &= \sum_{i=1}^{M} \gamma_i + \sum_{i=1}^{M}\sum_{j=1}^{M} \lambda_j p_{ji} \\ &= \sum_{i=1}^{M} \gamma_i + \sum_{j=1}^{M}\sum_{i=1}^{M} \lambda_j p_{ji} \\ &= \sum_{i=1}^{M} \gamma_i + \sum_{j=1}^{M} \lambda_j (1 - p_{jM+1})\end{aligned}$$

Rearranging, we get:

$$\sum_{i=1}^{M} \gamma_i = \sum_{i=1}^{M} \lambda_i - \sum_{j=1}^{M} \lambda_j (1 - p_{jM+1})$$

The left hand side equals to the network throughput as obtained by Equation (5). On the right hand side, the variable j in the second summation can be changed to i without affecting the results of the summation. With such a change, the right hand side becomes:

$$\sum_{i=1}^{M} \lambda_i - \sum_{i=1}^{M} \lambda_i (1 - p_{iM+1}) = \sum_{i=1}^{M} \lambda_i p_{iM+1}$$

This is the network throughput as obtained by Equation (4). Equations (4) and (5) therefore yield the same results.

Reference: Raj Jain, “The Art of Computer Systems Performance Analysis”, Wiley, 1991