Communication

W. Chang

Bulk Queue Model for Computer System Analysis

Abstract: A bulk queue model was developed for analyzing a multiprogrammed computer system. It can be used in conjunction with closed queuing models to study message queuing in a teleprocessing system. The model is based on an imbedded Markov chain analysis.

Introduction

Computer queues can be classified theoretically in two kinds. Messages or transactions which are waiting to be processed at the terminals or at the disks can be classified as external queues. When messages or transactions are processed by the computer, queues inside the computer can be formed because these messages or transactions contend for various resources in the computer system. Such internal queues can be studied by a closed computer network. The external queue is the subject of this discussion.

A number of tasks, i.e., messages, transactions, or job steps, etc., are processed concurrently within a computer system. Each of these tasks demands the services of the CPU, the channels, and the I/O devices. Each task waits for service if the facility is busy servicing other tasks. Clearly, the processing time of a task depends on several factors: the number of simultaneously presented tasks, the service times of I/O devices and the CPU, the number and the sequences of the I/O accesses, and the waiting times for the resources. Hence, the processing time of a task within a computer system is a function of the number of concurrent tasks in the system. The processing times can be analyzed by using closed queuing models [1-3]. In what follows, we describe the behavior of an external queue, assuming that we can determine the processing times from other models.

Multi-server model

We can formulate the problem as a multi-server system with c servers. We assume that the arrival distribution of messages or transactions at a computer installation for processing is Poisson with input density λ. The maximum number of messages or transactions which the computer system can process simultaneously is c. A queue is formed if all the servers are engaged in processing messages.

Let P_n be the probability that there are n items in the system, including those being served. Since the processing time depends on the state of the system, i.e., the number of simultaneously processing tasks in the system, the processing of each transaction (or the service time of each server) is assumed to depend on i, where $i = 1, 2, \cdots, c$. Let $H_i(x)$ be the service time distribution of each server, when i servers are engaged in services; let $\psi_i(s)$ be its Laplace transform. If we assume that $H_i(x)$ is exponentially distributed, the state probability P_n can be easily determined from the birth and death process; i.e., if

$$H_i(x) = 1 - e^{-\mu_i x}, \tag{1}$$

then

$$\lambda P_0 = \mu_1 P_1 \text{ and} \tag{2}$$

$$(\lambda + \mu_n)P_n = \lambda P_{n-1} + \mu_{n+1} P_{n+1}. \tag{3}$$

If n is larger than c, then $\mu_n = \mu_c$.

However, such an approach is not always satisfactory because $H_i(x)$ is not exponential. The processing time is the sum of many small time increments which include the CPU times, the I/O times, and the internal waiting times. In other words, the processing time distribution is a convolution of many service time and waiting time distributions from the closed queuing network. It is better approximated by an Erlang distribution. If the processing time is a constant, the solution of the multi-server queuing system is given in Riordan's book. [4] Our formulation follows this method closely.

- *Markov chain*

Assume that the processing time is closer to being constant than exponentially distributed. We can approximately form a Markov chain as follows.

Let the transition probability be

$$p_i(j) = \int_0^\infty e^{-\lambda x}[(\lambda x)^j/j!]\, dH_i(x), \tag{4}$$

where j is the number of new arrivals during a service time x. We can form the following set of linear equations:

$$\begin{aligned}
P_0 &= p_1(0)\, P_0 + p_1(0)\, P_1 + p_2(0)\, P_2 + \cdots + p_c(0)\, P_c, \\
P_1 &= p_1(1)\, P_0 + p_1(1)\, P_1 + p_2(1)\, P_2 + \cdots + p_c(1)\, P_c \\
&\quad + p_c(0)\, P_{c+1}, \\
&\vdots \\
P_n &= p_1(n)(P_0 + P_1) + \sum_{i=2}^{c-1} p_i(n)\, P_i + \sum_{j=0}^{n} p_c(n-j)\, P_{c+j}.
\end{aligned} \tag{5}$$

This formulation is similar to a single server system with bulk services. That is, during each service period, the system is capable of servicing up to c customers at a time. The system is assumed to be idle when an item arrives at time 0. The system serves this item with a service time distribution $H_1(x)$. When this item departs, the system is at whatever state corresponds to the number of arrivals during the service time. If there was no arrival, the system at the departure point is idle. If there was one arrival, the system is in state 1. Suppose that the system is in state c and there was no arrival; the system will return to the idle state after all c items depart. The system will be in state n if the previous state is less than c and during the service period exactly n new items arrive [or if the previous state is larger than or equal to c (at $c+j$ state) and exactly $n-j$ new items arrive].

Multiplying both sides of (4) by z^n, and noting that z^n can be written as $z^{n-j}z^{j+c}z^{-c}$, we obtain a generating function by summing up terms on both sides:

$$\begin{aligned}
U(z) &\equiv \sum_{n=0}^{\infty} P_n z^n = \sum_{j=0}^{\infty} z^j \int_0^\infty e^{-\lambda x}[(\lambda x)^j/j!]\, dH_i(x) \\
&= \int_0^\infty e^{-\lambda(1-z)x}\, dH_i(x) = \psi_i[\lambda(1-z)].
\end{aligned} \tag{6}$$

Using (5) we also have

$$\begin{aligned}
U(z) &= P_0\, \psi_1[\lambda(1-z)] + P_1\, \psi_1[\lambda(1-z)] + \cdots \\
&\quad + P_{c-1}\, \psi_{c-1}[\lambda(1-z)] \\
&\quad + \psi_c[\lambda(1-z)]\, [U(z) - \sum_{j=0}^{c-1} P_j\, z^j]/z^c,
\end{aligned} \tag{7}$$

or

$$U(z) = \frac{\sum_{j=0}^{c-1} P_j\{z^c\psi_j[\lambda(1-z)] - z^j\psi_c[\lambda(1-z)]\}}{z^c - \psi_c[\lambda(1-z)]}, \tag{8}$$

where $\psi_0[\lambda(1-z)] = \psi_1[\lambda(1-z)]$.

Since $z^c - \psi_c[\lambda(1-z)]$ has exactly c roots within the unit circle $z = 1$, and since $U(1) = 1$, we can use these roots to determine $P_0, P_1, \cdots, P_{c-1}$ (the numerator must vanish at these roots).

• *Queuing time distribution*

Let $\theta(s)$ be the Laplace transform of the queuing time distribution (queuing time = waiting time + service time). Since the number of new arrivals during the queuing time is equal to the queue size, i.e., $\theta[\lambda(1-z)] = U(z)$, we have

$$\begin{aligned}
\theta(s) &= U(1 - s/\lambda) \\
&= \frac{\sum_{j=0}^{c-1} P_j[(\lambda - s)^c\psi_j(s) - \lambda^{c-j}(\lambda - s)^j\psi_c(s)]}{(\lambda - s)^c - \lambda^c\psi_c(s)},
\end{aligned} \tag{9}$$

• *Waiting time distribution*

Let $W(x)$ be the waiting time distribution and let $\Omega(s)$ be its Laplace transform. Since the queuing time is the sum of the waiting time and the service time, we have

$$\begin{aligned}
\theta(s) &= (P_0 + P_1)\psi_1(s) + \sum_{j=2}^{c-1} P_j\psi_j(s) \\
&\quad + [\Omega(s) - \sum_{j=0}^{c-1} P_j]\psi_c(s),
\end{aligned} \tag{10}$$

from which $\Omega(s)$ can be obtained as

$$\begin{aligned}
\Omega(s) &= \sum_{j=0}^{c-1} P_j\, [(\lambda - s)^c - 1]\, \psi_j(s) \\
&\quad + \sum_{j=0}^{c-1} P_j\, [1 - \lambda^{c-j}\, (\lambda - s)^j]\, \psi_c(s) \\
&\quad \div [(\lambda - s)^c - \lambda^c\, \psi_c(s)]\, \psi_c(s),
\end{aligned} \tag{11}$$

where $\psi_0(s) = \psi_1(s)$.

Example

Let $c = 2$ and $\lambda = 0.1$. Let the service time be Erlang-2 distributed:

$$\psi_0(s) = \psi_1(s) = [0.25/(s + 0.25)]^2 \qquad \text{Mean} = 8;$$

$$\psi_2(s) = [0.2/(s + 0.2)]^2 \qquad \text{Mean} = 10.$$

The denominator of $\theta(s)$ can be factored as

$$s\,(s - 0.155)\,(s^2 + 0.355s + 0.025) = 0;$$

thus $s_1 = 0.155$. Substituting this value of s into the numerator of $\theta(s)$, we find $P_1 = 0.698\, P_0$. With this relationship, and after some simplifying, we have

$$\begin{aligned}
\theta(s) &= P_0\, [0.25/(s + 0.25)]^2\, (1.7s^2 + 0.646s \\
&\quad + 0.06)/(s^2 + 0.355s + 0.025).
\end{aligned}$$

Since $\theta(0) = 1$, we find $P_0 = 0.25$, $P_1 = 0.17$, and $-\theta'(0) =$

10.1. Finally, with these values and the mean serving times the mean waiting time W can be determined from

$10.1 = (P_0 + P_1)8 + (1 - P_0 - P_1)(W + 10)$, or

$W = 1.72$.

This model can be used with other closed queuing models to study the queuing behavior of a computer system. That problem requires the determination of a number of roots in the denominator of an equation. If c is large, one must use a computer program to determine the roots, which may also be a time-consuming task.

References

1. M. Reiser and H. Kobayashi, "Recursive Algorithms for General Queuing Networks with Exponential Servers," Research Report RC 4254, IBM Thomas J. Watson Research Center, Yorktown Heights, N.Y., March 1973.
2. M. Chandy, U. Herzog and L. Woo, "Parametric Analysis of Queuing Network Models," Research Report RC 4730, IBM Thomas J. Watson Research Center, Yorktown Heights, N.Y., March 1974.
3. D. P. Gaver and G. S. Shedler, "Processor Utilization in Multiprogramming Systems via Diffusion Approximations," *Operations Research* **21,** 569 (1973).
4. J. Riordan, *Stochastic Service Systems*, John Wiley & Sons, Inc., New York, 1962, p. 115.

Received January 29, 1974

The author's current address is the IBM Data Processing Division Headquarters, White Plains, New York 10601; this work was done at the IBM Education Center in Poughkeepsie, New York.