the **Availability Digest**

Configuring to Meet a Performance SLA – Part 1:
Single Server Response Times
December 2008

If the performance of a data-processing system has degraded to the point that it is no longer useful to its users, the system is, for all practical purposes, down. It is, therefore, common to specify a performance requirement in the Service Level Agreement (SLA) for the system.

The performance requirement is often expressed as a probability that the system's transaction response time will be less than a given interval. For instance, "98% of all transactions must complete within 500 milliseconds."

The normal queuing relationship with which we are all probably familiar,

$$T_r = \frac{T_s}{(1-L)} \, , \tag{1}$$

gives only the average response time, T_r, to be expected if a system with a service time, T_s, is carrying a load of L. For instance, if the system under zero load gives a response time of 250 milliseconds (its service time), its average response time when occupied 75% of the time will be 1,000 milliseconds [250/(1 − 0.75)]. That is, 50% of all transactions will complete in less than one second.

This simple relationship tells us nothing about the distribution of response times, which is what we need to know in order to determine what percentage of response times will be in excess of some value, as specified by an SLA. If we knew this transaction-response time distribution, we could determine the transaction-processing power required of the system in order to meet an SLA specification. For instance, in the first example above, the question to be answered is what service time is required of our system in order that 98% of all transactions will complete in less than 500 milliseconds.

In this series of five articles, we will explore the solution to this question. The solution is not straightforward. We therefore will spend some time talking about performance calculations in the context of queuing theory. Our five sections will cover the following material:

- Part 1 reviews basic queuing theory. The simple queuing equation for a single server is derived so that the assumptions underlying this important relationship are understood. We will see that the well-known response-time formula as expressed by Equation (1) above is just one case of this relationship.

- Part 2 expands the simple queuing relationship from the case of a single server to that of multiple servers, such as the multiple processes in a server class managed by a transaction-monitoring facility such as Tuxedo or HP's NonStop Pathway.

- Part 3 answers the SLA configuration question for servers with exponential service times. It provides charts and an Excel spread sheet to determine the probability that the response will not exceed a specified value.

- Part 4 introduces the Gamma distribution and shows with charts and a spread sheet how to use it to answer the SLA configuration question for servers with arbitrary response times.

- Part 5 concludes with an example of determining the probability/maximum response time curve for a complex series of servers with differing service time distributions.

In all cases, we provide a simple Excel spreadsheet that can be used to make performance calculations.

The Basic Queuing Relationship

As a system becomes loaded, its response time gets longer because a new transaction must wait in line behind other transactions that arrived before it. Its response time is the time that it takes to process all earlier transactions waiting to be processed plus its own service time. As the server gets busier (i.e., as it becomes more loaded), the queue of waiting transactions grows longer; and transaction response times increase.

To understand this better, consider a server servicing transactions on a first-in, first-out basis. On the average, it takes a time of T_s to service a transaction. Some transactions take less time, others take more time; but the average service time is T_s. We will call T_s the *service time* of the server.
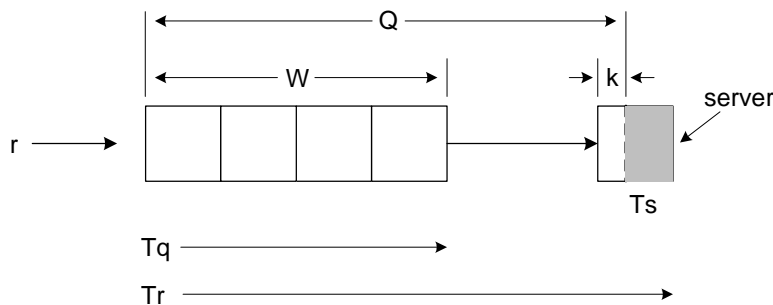
r = transaction arrival rate
W = number of transactions waiting for service
Q = number of transactions in the system
Ts = server service time
k = portion of current transaction service time remaining
Tq = time a transaction waits to be serviced (queuing delay)
Tr = total transaction service time (transaction-response time)

The average number of transactions in the system, which we will call the *queue length*, is denoted by *Q*. This comprises *W* transactions that are waiting for service plus that portion, *k*, of the current transaction that remains to be serviced.

Transactions are arriving at a rate of *r* transactions per second. Therefore, the server is carrying a load, *L*, of

$$L = rT_s \qquad\qquad\qquad (2)$$

2

© 2008 Sombers Associates, Inc., and W. H. Highleyman
www.availabilitydigest.com

That is, if transactions are arriving at an average rate of one per second, and if the average service time for a transaction is 0.5 seconds, the server is busy 50% of the time. It is carrying a load of 0.5.

When a transaction arrives to be serviced, it will find W transactions waiting in front of it. In addition, with probability L, there will be a transaction in the process of being serviced (remember that L is the probability that the server is busy – that is, it is currently servicing a transaction). If a transaction is currently being serviced, then only kT_s time is left to complete its servicing.

The newly arrived transaction must wait while the current transaction is completed (kT_s seconds L of the time) and then for the W transactions in front of it to be serviced (WT_s seconds). Thus, it must wait for a time T_q (the *queuing delay*) of

$$T_q = WT + LkT_s \tag{3}$$

before its servicing can be started.

From the time that a transaction arrives in line to the time that its service begins (a time of T_q), an average of W other transactions must have arrived at the server in order to maintain the average queue length. Since transactions are arriving at a rate of r transactions per second, then

$$W = T_q r$$

or

$$T_q = W / r \tag{4}$$

Using Equations (2), (3) and (4) to solve for the waiting line length, W, gives

$$W = \frac{kLrT_s}{1 - rT_s} = \frac{kL^2}{1 - L} \tag{5}$$

The total length of the queue, Q, as seen by an arriving transaction is the waiting time, W, plus a transaction being serviced L of the time:

$$Q = W + L = \frac{kL^2}{1 - L} + L = \frac{L}{1 - L}[1 - (1 - k)L] \tag{6}$$

The response time, T_r, is determined in a similar manner. It is the total amount of time that the transaction must wait in order to complete its servicing – its waiting time plus its service time. During this time, Q new transactions must arrive to maintain the steady state:

$$Q = T_r r = T_r L / T_s \tag{7}$$

Setting Equations (6) and (7) equal and solving for T_r, we have

$$T_r = \frac{T_s}{1 - L}[1 - (1 - k)L] \tag{8}$$

Equation (8) is the result we are looking for. It is known as the Khintchine-Pollaczek formula and relates response time, T_r, to the server's service time and the server load, T_s *and* L. It is a very general result and depends only upon some simple characteristics of the system:

- There is only a single server.

- No limit is imposed on the length of the queue.
- Transactions to be serviced arrive independently.
- Service order is first-in, first-out.
- The queue is in a steady-state condition.

In particular, Equation (8) makes no assumption about the distribution of the server's service time. Rather, Equation (8) incorporates an additional factor of *k*. *k* is a function of the service-time distribution, since it defines the portion of the current transaction service time remaining when a transaction first enters the queue. We call *k the service-time distribution coefficient*.

Note that if *k* = 1, Equation (8) reduces to our familiar Equation (1). We now take a closer look at *k*.

The Service-Time Distribution Coefficient

The parameter *k* is a function of the service-time distribution. It is the average amount of service time that is left to be expended on the current transaction being serviced when a new transaction arrives. Let us look at *k* for certain important cases of service-time distributions.

Exponential Service Times

Exponential distributions describe processes that are random. That is, the probability that an event will happen in the next very small time interval[1] is always the same and is not affected by the previous history of event arrivals. It therefore has no memory. Many events in data-processing systems are characterized fairly closely by the exponential distribution. In particular, the arrival of transactions at the server has been assumed to be random. Their interarrival times are exponentially distributed.

In our case, the events in which we are interested are service completions. It can be shown that for random service-time completions, the probability that the remaining service time will be greater than a given amount, *t*, is given by (e^{-at}). The exponential distribution has the characteristic that the remaining service time after any arbitrary delay, assuming that servicing is still in progress, is still exponential. That is, it has no memory.[2]

Thus, one has the following interesting situation. If the average service time is T_s, and if the server is currently busy processing a transaction (no matter for how long), one is still going to have to wait an average time of T_s for the servicing of that transaction to complete.

Since *k* is the proportion of servicing time remaining when a new transaction enters the queue, then *k* = 1 in this case. Equations (6) and (8) become

$$Q = \frac{L}{1-L}$$

$$T_r = \frac{T_s}{1-L}$$

for exponentially-distributed (i.e., random) service times.

---

[1] A time interval small enough so that exactly one or zero events can happen in that interval.
[2] Random distributions, exponential distributions, and Poisson distributions all describe the same phenomena in different ways. The Poisson distribution gives the probability that *n* events will occur in some time *t*. All are related, and one can be derived from another. See Chapter 4, <u>Basic Performance Concepts: Queues – An Introduction</u>, *Performance Analysis of Transaction Processing Systems*, by W. H. Highleyman, published by Prentice-Hall in 1989.

Constant Service Times

If the service time is a constant, on the average, half of the service time will remain when a new transaction enters the queue. Therefore, $k = 1/2$; and Equations (6) and (8) become

$$Q = \frac{L}{1-L}(1-L/2)$$

$$T_r = \frac{T_s}{1-L}(1-L/2)$$

Uniform Service Times

A service time that can range from zero to s seconds with equal probability is a uniform distribution. Disk-head seek time is a good example of this distribution. It can be shown that $k = 2/3$ for this case, and

$$Q = \frac{L}{1-L}(1-L/3)$$

$$T_r = \frac{T_s}{1-L}(1-L/3)$$

General Distributions

Khintchine and Pollaczek have shown in general that for arbitrary distributions, k is given by

$$k = \frac{1}{2}\frac{E(T_s^2)}{T_s^2} \tag{9}$$

where

$E(T_s^2)$ is the second moment of the service-time distribution.

Discrete Service Times

Often, the service time will be one of a set of constant times, each with a duration of T_{si} and a probability of p_i. Equation (9) can be extended to cover this case:

$$k = \frac{1}{2}\frac{\sum_i p_i T_{si}^2}{\left(\sum_i p_i T_{si}\right)^2}$$

Response Time Chart

The response times for exponential service-time distributions ($k = 1$), uniform service-time distributions ($k = 2/3$), and constant service-time distributions are shown in the following chart. This chart shows the response time, T_r, relative to the service time, T_s, (that is, T_r/T_s) as a function of server load.

Note that the response time curve is flatter for those distributions with lower values of k. As the service time becomes less random, the server is able to carry additional load.
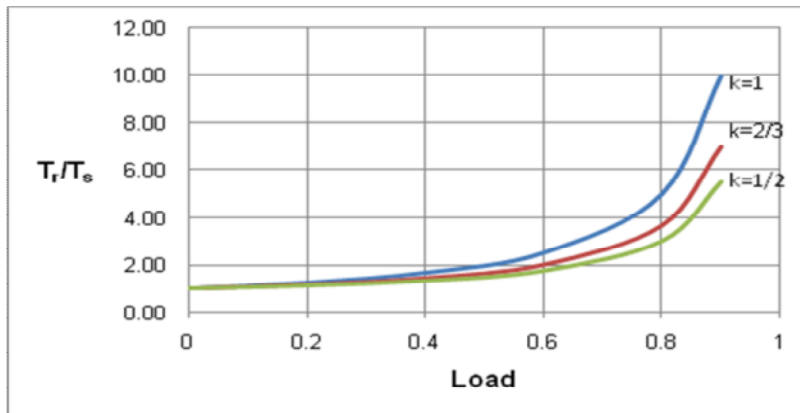
Little's Law

Equation (7) gives an additional insight into queuing systems. Known as Little's Law, it states that

$$Q = T_r r \qquad (10)$$

That is, the number of transactions in a system is equal to the product of the transaction rate, r, and the transaction response time, T_r. T_r can be interpreted as the *lifetime* of the transaction in the system.

For instance, if the transaction rate is 100 transactions per second, and the transaction response time is 50 milliseconds (0.05 second), then on the average there will be five transactions (100 x 0.05) in the system at any one time.

This can be very important if there are system limits that constrain the number of transactions that can be currently processed simultaneously.

Summary

The Khintchine-Pollaczek Equation (8) characterizes transaction-response times in a very general manner. It is valid across many more queuing disciplines than the simple first-in, first-out discipline discussed above. It can be shown that the processing order of transactions is not important so long as a transaction is not selected for service based on its characteristics. For instance, this relationship applies equally well to a round-robin or polling-service algorithm but not to a service procedure that gives priority to short messages over long messages.

An Excel spread sheet that is useful in making many of these response time calculations can be found at http://www.availabilitydigest.com/public_articles/0312/performance_sla_1.xls.

Equation (8) can also be extended to multiserver systems in which multiple servers cooperate to service a common queue. This architecture models many real-world systems, from Web server farms to server classes managed by transaction-management facilities such as Tuxedo and HP's NonStop Pathway. This is the subject of our next article, Configuring to Meet a Performance SLA – Part 2: Multiserver Response Times.