the Availability Digest

Configuring to Meet a Performance SLA – Part 3: Single Server with Exponential Response Time

February 2009

Many applications carry with them a performance Service Level Agreement (SLA) that specifies the response times that they must achieve. After all, if an application's response time is so slow that the application is not useful, the application is, in effect, down.

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, "When handling 50 transactions per second, 98% of all transactions must complete within 500 milliseconds."

In Part 1 of this series, we derived the basic average response-time expression for a single-server system. In Part 2, we extended that result to a multiserver system in which multiple servers work off a common work queue.

We now show how to size a system to meet a performance SLA. If service time is exponentially distributed, the solution to this question is straightforward. If service time is not exponentially distributed, the solution is more complex and involves the Gamma Distribution. In this part, we explore exponentially-distributed service times. In Part 4, we will extend this to servers with general service-time distributions.

First, we review the results of the first two parts of this series.

Reviewing the Average Response Time for a Single Server

In Part 1, we showed that the average response time for a single-server system was given by the Pollaczek-Khintchine equation:

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

where

- T_r is the average transaction-response time.
- T_s is the average service time of the server.
- L is the load on the server.
- k is the distribution coefficient of the server's service time.

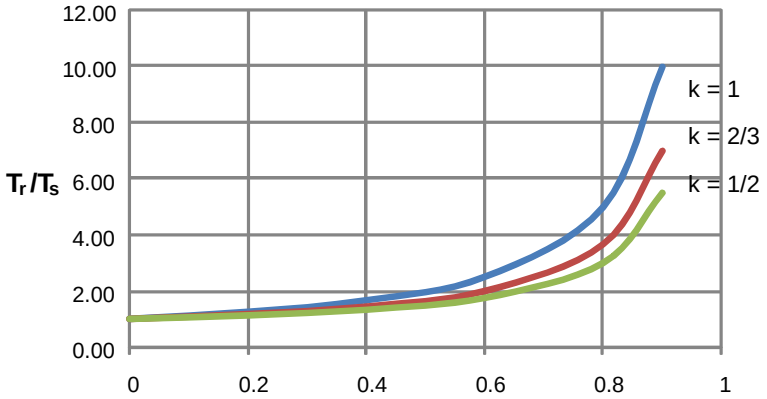
The distribution coefficient k depends upon the probability distribution of the server's service time. For instance, k = 1 applies to server distribution times that are random (that is, exponentially distributed), which is the common assumption.[1] In this case, Equation (1) reduces to

[1] More specifically, k is 1/2 the ratio of the service time's second moment to the square of its mean.

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

which for many of you is the well-recognized expression relating transaction-response time to server load.

As k becomes smaller, response time for a given load decreases. For uniform distribution times, *k* = 2/3. For constant distribution times, *k* = 1/2. The response-time/load relationship for different service-time distributions is shown in Figure 1, which plots transaction-response time normalized to service time (T_r / T_s) as a function of load.

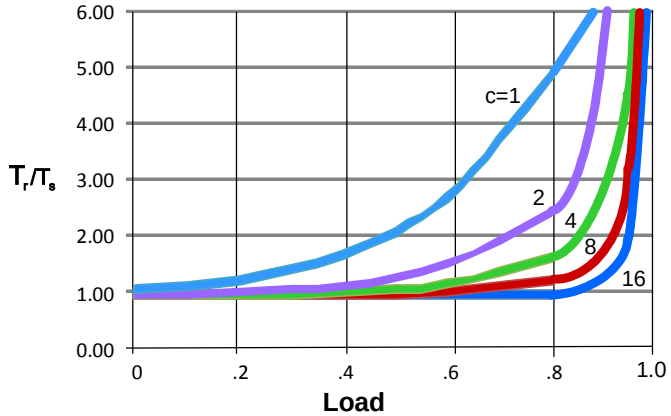


Single-Server Transaction Time
Figure 1

Reviewing the Average Response Time for a Multiserver System

The calculation of response time for a multiserver system is more complex. In a multiserver system, several like servers process transactions from a common work queue. Examples of such servers are Web farms and transaction-processing monitors, such as Tuxedo and NonStop Pathway, that distribute transactions to a pool of servers.

Using c to reflect the number of servers in the multiserver system, Figure 2 shows transaction-response time normalized to service time (T_r / T_s) as a function of load for multiserver systems using 1, 2, 4, 8, and 16 servers. The improvement in response time as servers are added is clearly shown.



Multiserver Transaction Time
Figure 2

Note that response time flattens out as more servers are added. Therefore, a multiserver system can be loaded more heavily than a single-server system. However, care must be taken as this means that the response-time/load curve breaks much more rapidly as the number of servers is increased.

Reference is made to Part 2 for the expressions governing multiserver response times.

Determining the Average Response Time Needed to Meet an SLA

We now turn our attention to determining what average response time we need in order to meet an SLA requirement such as "At 50 transactions per second, 98% of all transactions will complete within 500 msec." Knowing the average response time, we can calculate the load that can be carried by the server and still meet the SLA.

There are two distinct cases – the case of exponentially-distributed service times and the more general case of general service-time distributions. In this part, we consider servers with exponential service times. In Part 4, we will discuss servers with general service times, of which exponential service times are a special (and very important) case.

Exponential Service Times – Single Server

The cumulative probability distribution for the response time of a server with exponential service time is[2]

$$P(T_r < t) = 1 - e^{-t/T_r} \qquad (3)$$

where

T_r	is the response time.
t	is a time variable.
$P(T_r < t)$	is the probability that $T_r < t$.

We are interested in the probability that the response time, T_r, is less than some maximum response time T_m. Thus, we can rewrite Equation (3) as

$$P = P(T_r < T_m) = 1 - e^{-T_m/T_r} \qquad (4)$$

where T_m is the maximum allowable response time.

That is, the probability that the response time, T_r, will be less than T_m is P, where we use the term P to represent $P(T_r < T_m)$.

We can solve Equation (4) for the response time:

$$T_m / T_r = -\ln(1-P) \qquad (5)$$

As shown in Equation (2), we know that for a server with exponential response time,

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

[2] The exponential distribution and the Poisson distribution are both properties of a random distribution in which events are independent of each other. See Highleyman, Wilbur H., Chapter 4, Basic Performance Concepts, *Performance Analysis of Transaction Processing Systems*, Prentice-Hall; 1989.

where

T_s is the average service time of the server.
L is the load on the server.

Thus,

$$m = T_m / T_s = -\frac{\ln(1-P)}{(1-L)} \qquad (6)$$

where m is the maximum response time, T_m, normalized by T_s. P can now be written as

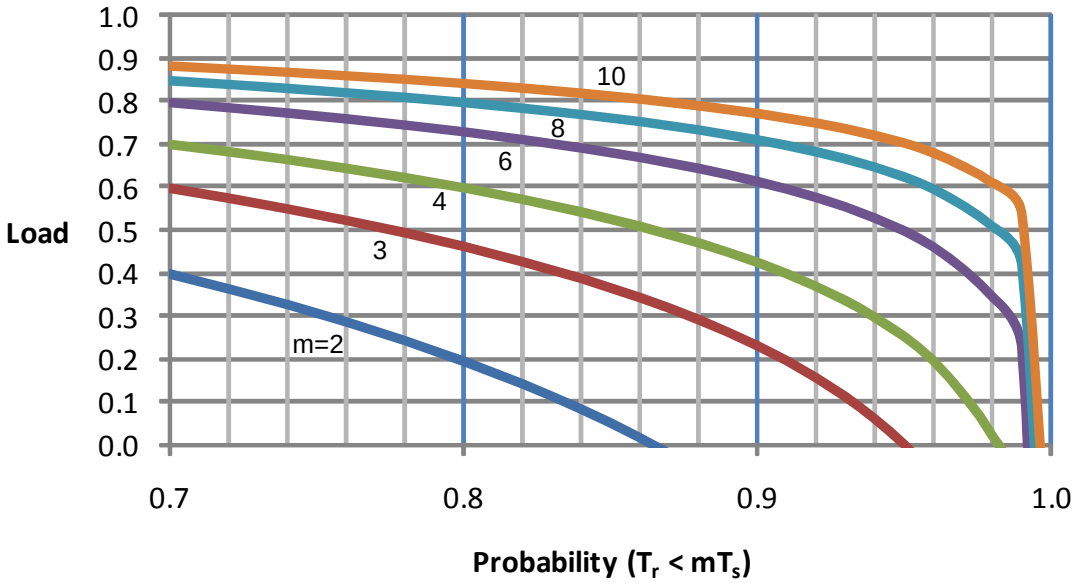
$$P = P(T_r < mT_s)$$

That is, P is the probability that the response time will less than m times the service time.

Given the specification that we would like P of all transactions to be completed in less than m service times, we can calculate from Equation (6) the server load that will allow us to meet this specification:

$$L = 1 + \frac{\ln(1-P)}{m} \qquad (7)$$

This relationship is shown in Figure 3 for various values of m. For instance, if 95% of all transactions (P) are to complete in less than six service times (m), we can load the server up to 50%. That is, if the service time is 10 msec., the average response time will be 20 msec. [Equation (2)]; and 95% of all transactions will complete in less than 60 msec. [Equation (7)].



Allowable Load To Meet Performance SLA
Figure 3

Figure 3 is useful if the server is already chosen, and one must determine how much load it can carry and still meet the performance SLA. In the above example, with a service time of 10 msec. and a load of 50%, the server can handle 50 transactions per second and meet the SLA.

However, if one is instead trying to decide how fast a server must be, the required service time is of more interest. Noting that

$$L = rT_s \tag{8}$$

where *r* is the transaction rate, we can rewrite Equation (6) as

$$T_m / T_s = -\frac{\ln(1-P)}{(1-rT_s)}$$

Solving for the service time, T_s, we have

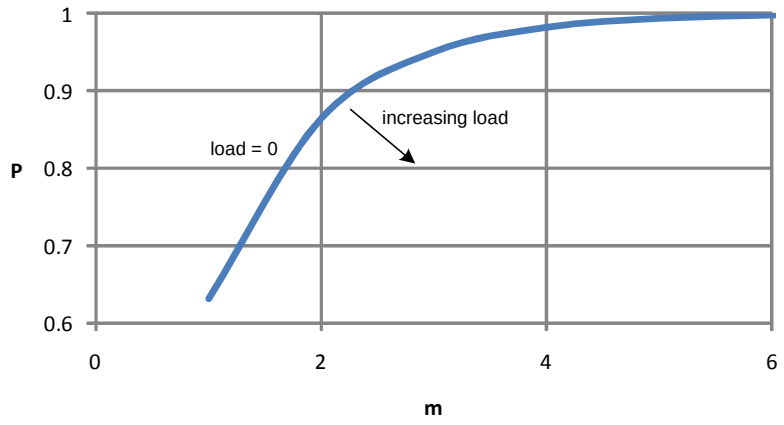
$$T_s = \frac{T_m}{rT_m - \ln(1-P)} \tag{9}$$

Given a transaction rate, *r*, a probability, *P*, and a maximum allowable response time, T_m, we can determine how powerful a server we need in terms of what its average service time, T_s, must be. For instance, in our previous example, if at 50 transactions per second (*r), 95% (P*) of all transactions must complete within 0.06 seconds (T_m), the average service time of the server must be 10 milliseconds.

There is one important observation to be made, and that is that there are certain SLA specifications that simply cannot be met if service times are exponential. This can be shown by noting that the server will perform its fastest under zero load. Under zero load, Equation (6) becomes

$$m = -\ln(1-P) \tag{10}$$

This is plotted in Figure 4. For any given value of *m*, the probability *P* that the response time will be less than mT_s cannot exceed the value above the load line.



Best Performance at Zero Load
Figure 4

For instance, the best performance that one can get out of a server with exponential service time is

- 63% of all transactions will complete in less than one service time.
- 86% of all transactions will complete in less than two service times.
- 99.3% of all transactions will complete in less than five service times.

If a server with a service time of 50 msec. is being used, and if the SLA calls for 90% of all transactions to be completed within 100 msec. (m = 2), this specification cannot be met with this server. A faster server is required.

Exponential Service Times – Multiserver Systems

There is a very important property of multiservers with exponential service times: their response times are also exponentially distributed.[3] Specifically,

> If one or more servers with exponential service times are driven from a common queue with Poisson-type arrivals, the outputs from that queue are Poisson-distributed with the departure rate equal to the arrival rate.

Therefore, the general concepts discussed above apply to multiserver systems, except that the relationship between response time, T_r, and service time, T_s, is a function not only of the load on the servers but also of the number of servers in the multiserver system.

In Part 2, we noted that this relationship is given by

$$T_r = \frac{k(cL)^c}{c(c!)(1-L)^2} p_0 T_s + T_s \qquad (11)$$

where p_0, the probability that the queue is empty, is given by

$$\frac{1}{p_0} = \sum_{n=0}^{c-1} \frac{(cL)^n}{n!} + \frac{(cL)^c}{c!(1-L)}$$

and

c is the number of servers in the multiserver system.

Part 2 includes a spreadsheet to calculate this response time. This spreadsheet can be found at http://www.availabilitydigest.com/public_articles/0401/performance_sla.xls.

In terms of our analysis above, Equation (5) still holds and is repeated here for convenience:

$$T_m / T_r = -\ln(1-P) \qquad (5)$$

However, converting this to reflect service time, T_s, is not so straightforward since T_s is related to T_r by Equation (11) above. The procedure to use is as follows:

1) Given P and T_m from the SLA specification, find the average allowable response time from Equation (5):

$$T_r = -\frac{T_m}{\ln(1-P)} \qquad (12)$$

2) Knowing the service time, T_s, of the servers under consideration and the number of servers, c, in the multiserver system, use Equation (11) to determine the load that can be tolerated. This can be done as an iterative calculation using the Part 2 spreadsheet referenced above (which itself is an iterative calculation done by Excel).

Noting that Equation (11) reduces to Equation (2) for c = 1, this two-step process is what was combined into a single step for the single-server case, resulting in Equation (7).

[3] T. L. Saaty, pg. 12-3, *Elements of Queuing Theory*, McGraw-Hill; 1961.

Let us use our previous example as an illustration. Let us assume that we are using an 8-processor multiserver system with each processor having an average service time, T_s, of 10 msec. We want 95% of all transactions to complete within a time, T_m, of 60 msec. From Equation (12), we find that the allowable average response time, T_r, is 20 msec. Iterating Equation (11) via the Part 2 spreadsheet referenced above, we find that we can impose a load of 91% on the servers in the multiserver system and still achieve a 20 msec. response time. We can load the single server only 50%. Thus, each server in the multiserver system can process a transaction rate that is approximately 80% higher than a single server. With eight servers in the system, this system will have a capacity 14.4 times greater than that of a single server.

We must note one caution here. It might be tempting to think that we can get by with smaller servers in the multiserver system and still achieve a 20 msec. response time. If these eight servers each had a service time of 144 milliseconds instead of 10 msec., would that not give the multiserver system the same capacity as a 10-msec. single server system? The answer is yes, but the response time would be 14.4 times slower. 95% of all transactions would complete in less than 864 msec, not 60 msec. as called for by the SLA.

Summary

Using Figure 3, we can now answer the question as to how much we can load a single server and meet an SLA requirement that *P* of all transactions will complete in less than *m* service times. The spreadsheet found at

http://www.availabilitydigest.com/public_articles/0402/performance_sla_3.xls

can be used to make this calculation [Equation (7)]. Alternatively, this spreadsheet also provides a section to calculate the service time needed to handle a given transaction rate and still meet the SLA specification [Equation (9)].

For multiserver systems, the spreadsheet of Part 2 can be used iteratively to relate service time to response time in order to respond to the SLA specification.

In our next part, we will extend these techniques to the problem of servers with general service-time distributions. This is where we will meet the Gamma distribution.