INFORMATION PROCESSING 68 - NORTH-HOLLAND PUBLISHING COMPANY - AMSTERDAM (1969)

CERTAIN ANALYTIC RESULTS FOR TIME-SHARED PROCESSORS*

LEONARD KLEINROCK
University of California at Los Angeles
Los Angeles, California, USA

A basic model for time-shared systems with M consoles is introduced and analyzed. Published measurements of existing computer systems demonstrate the accuracy of the model in describing the behavior of the normalized average response time, taken as the performance measure of these systems.

A definition for system saturation is given which is both intuitively pleasing and analytically significant. The original system of M consoles with processor capacity C is compared to a class of comparative systems, the Nth class consisting of N processors, each of capacity C/N serving M/N consoles each (for $N = 2, 3, 4, \ldots$). The priority problem is also considered for $M = 2$ and the effect of discriminatory behavior is solved for and graphed.

1. INTRODUCTION

The concept of a computer utility is fast becoming a reality. Numerous services are now available whereby one can purchase a user terminal (console) and can then rent the use of a remote computer on a time-shared basis. Time-sharing has become big business [1]!

As the supply and demand for readily accessible, inexpensive computing power grows, so grows the need for quantitative analysis of the performance of time-shared systems. This need is beginning to be met as may be evidenced by a survey of the literature [2]. In this paper, we present some recent results and interpretations which we feel are significant in predicting the performance of these systems.

2. ANALYSIS OF THE M-CONSOLE MODEL

The theoretical results divide into two classes: *infinite* input population and *finite* input population. The first class is illustrated in fig. 1 in which we see the basic structure wherein a new arrival (from an infinite population of possible customers) enters a system of queues, is treated according to the imposed queueing discipline, finally reaching the head of the queue, is allowed entry into the service facility for a given number of seconds (a quantum) and then either (a) departs if the quantum was enough to satisfy his requirement or (b) cycles back to the system of queues to wait for another turn in service. Results for a number of these systems are available in the literature [2].

Of interest to us in this paper are models for the *finite* input population where we assume that M consoles generate requests for use of the service facility. These requests impinge upon the system (whose internal structure is identical to that of the infinite population models shown in fig. 1); upon departure, these customers "return" to their original console to generate new requests as shown in fig. 2. We refer to the time required for a console to generate a new request as the "think time". The system response time is the elapsed time from when a request is made to when that request is satisfied completely;

* This work was supported by the Advanced Research Projects Agency (SD-184).

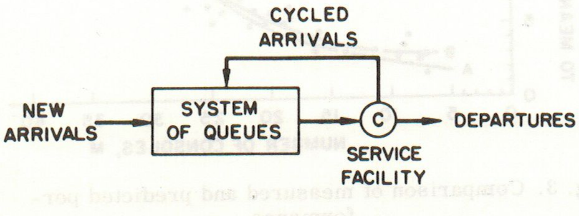

Fig. 1. Feedback queueing systems.

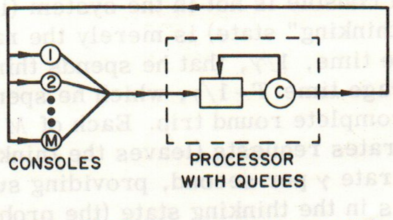

Fig. 2. Finite population model.

during this interval, the console, from which this request was made, is idle (nonthinking). The request is for a given number of "operations" in the service facility which can process at a rate of C operations/sec.

Below, we assume both that the think time for each console and that the size of each request are exponentially distributed with an average value of $1/\gamma$ sec for thinking and $1/\mu$ operations for each request, respectively. All quanta are assumed to be infinitesimal, and swap-time (the time lost in changing jobs) is assumed to be zero, thus leading to a processor-shared model [3]. In this case, then, when we find m consoles actively competing for use of the computer, we see that each console is being processed at a rate of C/m operations per second. The exponential assumptions along with the infinitesimal quanta produce a model for our time-shared system which is a continuous-time Markov process [4]. We let T be the average response time and take this as our performance measure.

This simple model has been carefully studied by queueing theorists [5], and corresponds to the finite-population single-server exponential queueing system. Below, we give the (easily obtained) results for the steady-state probability (denoted p_m) of finding m ($\leq M$) consoles actively competing for use of the computer facility (these consoles are said to be in the "system"). From any standard reference (such as p. 121 of [5]) we have that

$$p_m = p_0 \frac{M!}{(M-m)!}\left(\frac{\gamma}{\mu C}\right)^m \qquad m = 0,1,2,\ldots,M\,, \quad (1)$$

where

$$p_0 = \left[\sum_{m=0}^{M} \frac{M!}{(M-m)!}\left(\frac{\gamma}{\mu C}\right)^m\right]^{-1} . \qquad (2)$$

Let us now solve for T (average response time). Since the system is assumed to be in the steady state, we may use the fact that the rate at which customers enter the "system" (i.e., the dashed box in fig. 2) equals the rate at which they depart from the system. The fraction of time that a console is not in the system (i.e., he is in the "thinking" state) is merely the ratio of the average time, $1/\gamma$, that he spends thinking to the average time $T+1/\gamma$, which he spends in making a complete round trip. Each of M consoles generates requests (leaves the thinking state) at a rate γ per second, providing such a customer is in the thinking state (the probability that he is thinking is the fraction described above). Thus, the input rate of customers to the system is

$$M\gamma \frac{1/\gamma}{(1/\gamma)+T}$$

When m customers are in the system (probability p_m) then the rate at which each customer is being ejected is ‡ $\mu C/m$. Since there are m such, the average output rate of customers is

$$\sum_{m=1}^{M} (\mu C/m)m\, p_m = \mu C(1-p_0) .$$

Equating the input and output rates, we find that

$$T = \frac{M}{\mu C(1-p_0)} - \frac{1}{\gamma} . \qquad (3)$$

This result was first used by Scherr [6] for time-shared systems. Scherr also tested the worth of this model in the MIT time-sharing system. His principal finding is shown in fig. 3 where he has compared the results of measurement (shown as dotted data points and the least-squares fit, B-B, to these points) with the results of model analysis given by eq. (3) above (curve A-A). As can be

‡ The quantity $1/\mu C$ is the ratio of average number of operations per customer ($1/\mu$) to the number of operations per second (C), giving the average number of seconds of service per customer (when he is provided with a capacity of size C). The inverse, μC, is the rate at which he is completed. When provided with a capacity of C/m, his output rate is $\mu C/m$.

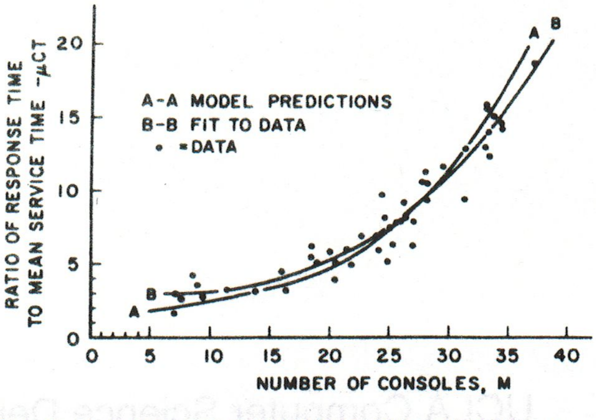

Fig. 3. Comparison of measured and predicted performance.

seen, the normalized ‡ response time, μCT, is accurately predicted by our model in spite of the fact that the MIT time-sharing system does not operate according to the assumptions of the model.

Due to the finite value of M, one questions whether it is possible to *saturate* the system. Indeed, if we define saturation as that point where the system goes unstable in some sense, such as average response time growing to infinity, then we see immediately that our system is never saturated (for $\gamma/\mu C < \infty$). (Such unstable behavior is possible in the infinite population case). Nevertheless there does exist an appropriate definition of saturation here as follows. If we replace each service time by its average $(1/\mu C)$, and if we schedule the arrivals to occur uniformly in time, each spending exactly $1/\gamma$ seconds thinking, then we see that the system can handle at most a number of consoles, M^* given by

$$M^* = \frac{1/\mu C + 1/\gamma}{1/\mu C} = \frac{\mu C + \gamma}{\gamma} \tag{4}$$

without any mutual interference. For example, if each customer requires 35 sec for thinking and 1 sec for computation, then 36 such customers can be handled. This provides the basis on which we define M^* as the saturation point for our M-console system †. We plot eq. (3) again in fig. 4 where $M^* = 41$ $(1/\mu C = 0.88$, $1/\gamma = 35.2)$. We see that μCT begins to increase sharply in the vicinity $M \approx M^*$. For $M \ll M^*$, we see that μCT grows very slowly since customers tend to request computation during other customers' think-time; indeed, it can be shown that in this region, the results from infinite population queueing theory hold, giving (see below also)

$$\mu CT \simeq \frac{1}{1 - \frac{(M-1)\gamma}{\mu C}} \quad \text{for } M \ll M^* . \tag{5}$$

If we define

$$x = \gamma/\mu C \tag{6}$$

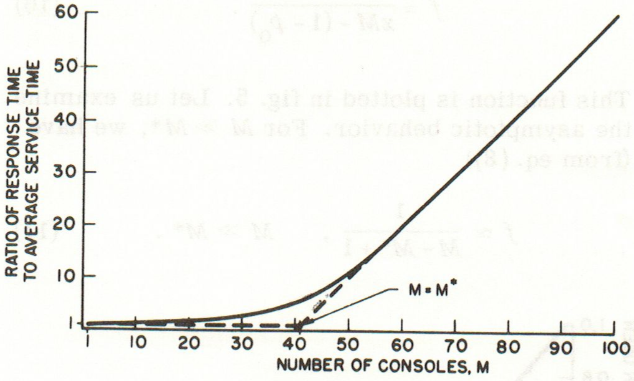

Fig. 4. Performance and saturation.

then

$$M^* = \frac{1+x}{x}. \tag{7}$$

For $M \gg M^*$, we see from eqs. (2) and (3) that

$$\mu CT \simeq M - \frac{1}{x} = M - M^* + 1 \qquad \text{for } M \gg \frac{1+x}{x}, \tag{8}$$

since $p_0 \to 0$. This asymptote is shown dashed in fig. 4, and we observe that it intersects the line $\mu CT = 1$ at $M = M^*$, since

$$M^* - \frac{1}{x} = 1 .$$

Since the slope of this asymptote is 1, it shows that each additional user "completely" interferes with all the other users, adding one more unit of normalized delay to μCT. The fact that the asymptote crosses $\mu CT = 1$ at precisely M^* shows, for $M \gg M^*$, that the system has "absorbed" M^* users and converted them into one user, and is now experiencing complete interference among the other $M - M^*$ users (i.e., the additional delay added to the response time for each user is $M - M^*$, since, from eq. (8), $\mu CT \simeq 1 + M - M^*$).

Let us now consider the function

$$f = 1/\mu CT \tag{9}$$

which represents the *fraction* of the processor which each user effectively sees as his own personal processor, for if a user spends μCT seconds in the system rather than 1 (normalized) second, it appears that he has been given $1/\mu CT$ of the processor. From eq. (3) we get

‡ If provided the full capacity, a customer will spend an average of $1/\mu C$ seconds in the system. We choose to normalize T with respect to this giving μCT which represents the factor by which a customer is delayed (due to his sharing the system) in relation to his time in system without sharing.

† This is similar to a definition given by Scherr [6].

$$f = \frac{x(1-p_0)}{xM - (1-p_0)} \; . \tag{10}$$

This function is plotted in fig. 5. Let us examine the asymptotic behavior. For $M \gg M^*$, we have (from eq. (8))

$$f \simeq \frac{1}{M - M^* + 1} \, , \qquad M \gg M^* \; . \tag{11}$$

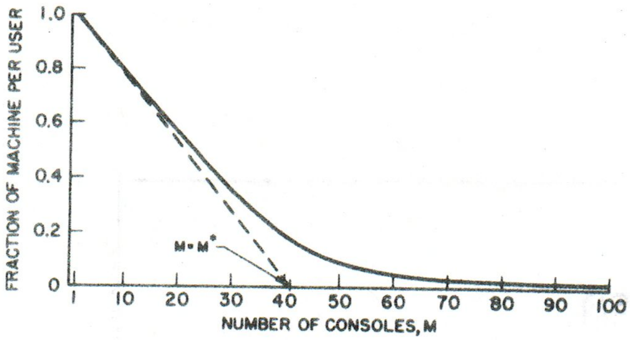

Fig. 5. Fractional machine availability.

This asymptote has a pole at $M = M^* - 1$; however, the fraction f goes to 1 at $M = M^*$. Thus the system behaves as if the number of completely interfering users was $M - M^* + 1$ instead of M (thus indicating that the system had transformed the first M^* users into one user). For $M \ll M^*$, we have from eq. (5) that

$$f \simeq 1 - (M-1)x \qquad M \ll M^* \; . \tag{12}$$

Let us derive this last equation from first principles. From eq. (2) we have

$$1 - p_0 = \frac{\sum_{m=1}^{M} \left[\frac{M!}{(M-m)!}\right] x^m}{\sum_{m=0}^{M} \left[\frac{M!}{(M-m)!}\right] x^m} \; .$$

From this last and eq. (10) we get

$$f = \frac{x \sum_{m=1}^{M} \left[\frac{M!}{(M-m)!}\right] x^m}{xM \sum_{m=0}^{M} \left[\frac{M!}{(M-m)!}\right] x^m - \sum_{m=1}^{M} \left[\frac{M!}{(M-m)!}\right] x^m}$$

thus

$$f = \frac{\sum_{m=1}^{M} \left[\frac{M!}{(M-m)!}\right] x^m}{\sum_{m=0}^{M} \left[\frac{M!}{(M-m)!}\right] m x^m} \; . \tag{13}$$

For $M \ll M^*$ (which implies $x(M-1) \ll 1$) we get

$$f \simeq \frac{1+(M-1)x}{1+2(M-1)x} = 1 + (M-1)x - 2(M-1)x[1+(M-1)x] + \ldots$$

thus

$$f \simeq 1 - (M-1)x$$

establishing eq. (12).

Eq. (12) shows that the slope of f as $M \to 1$ is merely $-x$. Thus, the tangent to f at $M = 1$ (shown as a dashed line in fig. 5) must intersect the horizontal axis at precisely $M = M^*$, the saturation load again!

The expected number E of active consoles is easily obtained from Little's result [7] which says, for any ergodic system, that the average number of people in that system is equal to the product of their average arrival rate to that system and their average time in that system. In our case, we know that the average arrival rate is $\mu C(1 - p_0)$ and so from eq. (3) we find that

$$E = M - \frac{1 - p_0}{x} \; . \tag{14}$$

3. COMPARATIVE SYSTEMS

It is interesting to observe the degradation in performance when we split the system of M consoles and a processor of capacity C referred to as an (M, C) system into two $(M/2, C/2)$ systems (see fig. 6). More generally, we consider

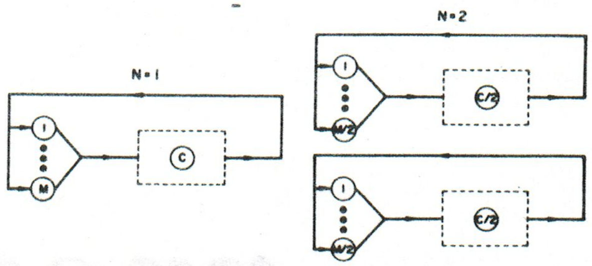

Fig. 6. Comparative systems ($N = 1, 2$).

$N(M/N, C/N)$ systems (N a positive integer). The behavior of this class is shown in fig. 7 where we plot μCT_N as a function of M/N (where T_N is the behavior of an $(M/N, C/N)$ system).

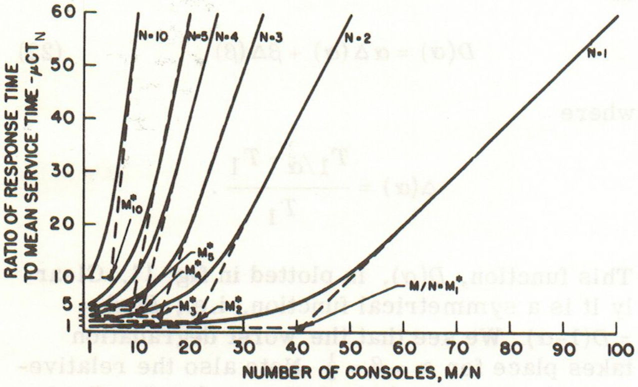

Fig. 7. Performance of comparative systems.

From eq. (3) we have

$$\mu CT_N = \frac{N(M/N)}{1 - p_0(N)} - \frac{1}{x} \tag{15}$$

where

$$[p_0(N)]^{-1} = \sum_{m=0}^{M/N} \frac{(M/N)!}{[(M/N) - m]!}\left(\frac{\gamma N}{\mu C}\right)^m . \tag{16}$$

From these last we see that for $M/N = 1$,

$$\mu CT_N = \frac{N}{1 - \frac{1}{1 + Nx}} - \frac{1}{x}$$

or

$$\mu CT_N = N \quad \text{for } M/N = 1 .$$

For such systems, the analogous saturation load, M_N^* is defined as

$$M_N^* = \frac{N/\mu C + 1/\gamma}{N/\mu C} = (Nx + 1)/Nx .$$

For $M/N \gg M_N^*$, $p_0(N) \to 0$ and so eq. (15) gives

$$\mu CT_N \simeq N(M/N) - \frac{1}{x} = N\left[\frac{M}{N} - M_N^* + 1\right] \quad \text{for } M \gg M_N^* \tag{17}$$

This asymptote (shown dashed in fig. 7) intersects the line $\mu CT_N = N$ at precisely $M/N = M_N^*$.

The inverse, $f_N = 1/\mu CT_N$ is again the fraction of the original machine (capacity C) seen by a user in an $(M/N, C/N)$ system and this is plotted in fig. 8.

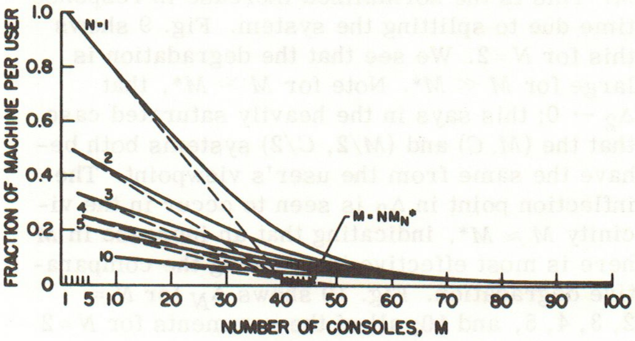

Fig. 8. Fractional use of comparative systems.

From eq. (15) we have

$$f_N = \frac{x(1 - p_0(N))}{N(M/N)x - (1 - p_0(N))} . \tag{18}$$

For $M/N \gg M_N^*$ we have from eq. (17) that

$$f_N \simeq \frac{1}{N(\frac{M}{N} - M_N^* + 1)} \quad \text{for } M/N \gg M_N^* . \tag{19}$$

This shows that M_N^* users are absorbed into one user. In a fashion similar to the derivation of eq. (13) we have that

$$Nf_N = \frac{\sum_{m=1}^{M/N} \frac{(M/N)!}{[(M/N) - m]!}(Nx)^m}{\sum_{m=0}^{M/N} \frac{(M/N)!}{[(M/N) - m]!} m(Nx)^m} . \tag{20}$$

For $M/N \ll M_N^*$ (which implies that $Nx[(M/N) - 1] \ll 1$) we get

$$Nf_N \simeq \frac{1 + (\frac{M}{N} - 1)Nx}{1 + 2(\frac{M}{N} - 1)Nx} .$$

Thus

$$Nf_N \simeq 1 - (\frac{M}{N} - 1)Nx \quad \text{for } \frac{M}{N} \ll M_N^* . \tag{21}$$

Eq. (21) shows that the slope of Nf_N as $M \to N$ is $-x$. Thus the tangent to f_N at $M = N$ (shown as dashed lines in fig. 8) intersects the horizontal axis at $M = N M_N^*$, again showing the significance of the saturation load.

Let us consider this degradation, as N increases, by plotting $\Delta_N = (T_N - T_1)/T_1$ versus M. This is the normalized increase in response time due to splitting the system. Fig. 9 shows this for $N=2$. We see that the degradation is large for $M \ll M^*$. Note for $M \gg M^*$, that $\Delta_2 \to 0$; this says in the heavily saturated case that the (M, C) and $(M/2, C/2)$ systems both behave the same from the user's viewpoint. The inflection point in Δ_2 is seen to occur in the vicinity $M \approx M^*$, indicating that an increase in M here is most effective in reducing the comparative degradation. Fig. 10 shows Δ_N for $N =$ 2, 3, 4, 5, and 10; all of the comments for $N=2$ apply to this last also.

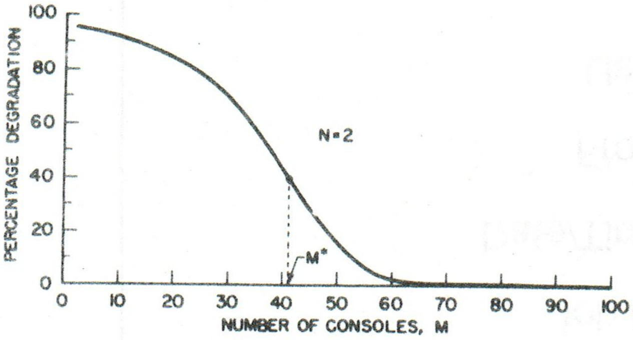

Fig. 9. Percentage degradation of comparative systems ($N = 2$).

For the $N=2$ case, we consider splitting the system into one $(\alpha M, \alpha C)$ system and one $(\beta M, \beta C)$ system where $\alpha + \beta = 1$. For such a split, the appropriate comparison is

$$D(\alpha) \equiv \frac{\alpha T_{1/\alpha} + \beta T_{1/\beta} - T_1}{T_1}$$

or

$$D(\alpha) = \alpha \Delta(\alpha) + \beta \Delta(\beta) \ , \qquad (22)$$

where

$$\Delta(\alpha) = \frac{T_{1/\alpha} - T_1}{T_1} .$$

This function, $D(\alpha)$, is plotted in fig. 11. Clearly it is a symmetrical function, i.e., $D(\alpha) =$ $= D(1-\alpha)$. We see that the worst degradation takes place for $\alpha = \beta = \frac{1}{2}$. Note also the relatively flat peak to the degradation, indicating the insensitivity of $D(\alpha)$ near $\alpha = \frac{1}{2}$. Thus we begin to effect an improvement only when α changes beyond the middle third of its range. As α decreases, this improvement is due to the $(\alpha M, \alpha C)$ system moving to a less saturated point while the $(\beta M, \beta C)$ system moves to a more saturated point. We show $D(\alpha)$ only for $(1/M) \leq \alpha \leq (M-1)/M$ since it is undefined beyond this range. It is easy to show that $D(1/M)$ $\simeq 1/\mu C T_1$ and this approximation improves as M approaches infinity.

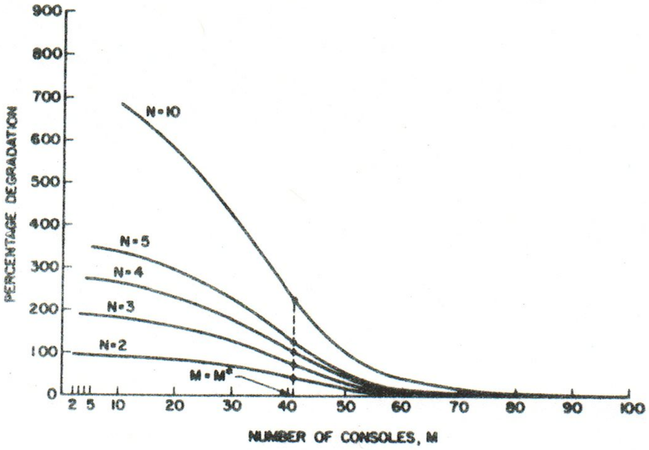

Fig. 10. Percentage degradation of comparative systems ($N = 2, 3, 4, 5, 10$).

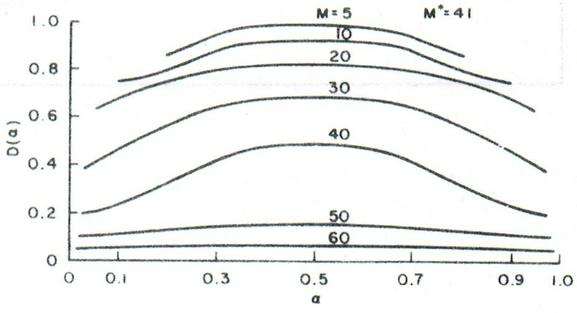

Fig. 11. Normalized degradation for the systems $(\alpha M, \alpha C)$ and $(\beta M, \beta C)$.

4. THE PRIORITY CASE FOR $M = 2$

Let us consider an $M=2$ system with priorities. We operate the priority by assuming that a fraction $g_1/(g_1+g_2) \equiv G_1$ of the capacity is given to console 1 and that a fraction $G_2 = 1 - G_1$ is given to console 2 whenever both are competing for service $(g_1, g_2) \geq 0$). Otherwise, the system operates in a manner identical to the non-priority system. We define the steady-state probability, for $i, j = 0, 1$

$$p_{ij} = P[i \text{ customers from console 1 and } j \text{ customers from console 2 in system}] .$$

Clearly

$$p_{00} = p_0 , \quad p_{01} + p_{10} = p_1 , \quad p_{11} = p_2 , \tag{23}$$

where p_m is as defined in section 2. Let

E_i = expected number of customers from console i in system.

Clearly

$$E_1 + E_2 = E ,$$

where E is given by eq. (14) for $M = 2$.

There are many ways to solve for p_{ij}, and all of them are trivial for this case $M = 2$. Indeed, from eq. (23) only p_{10} is unknown. We obtain it by writing the equilibrium equation (see [5] for this method)

$$p_{10}(\mu C + \gamma) = \gamma p_0 + \gamma C G_1 p_2$$

giving

$$p_{10} = \frac{x}{1+x}(p_0 + p_1 G_2) . \tag{24}$$

Since

$$E_1 = p_{10} + p_2$$

we find

$$E_1 = \frac{x}{1+x}(1 + p_1 G_2) = \frac{E}{2} + \frac{G_2 - G_1}{1+x}\left(\frac{p_2}{2}\right) . \tag{25}$$

Also

$$E_2 = p_{01} + p_2$$

and so

$$E_2 = \frac{x}{1+x}(1 + p_1 G_1) = \frac{E}{2} + \frac{G_1 - G_2}{1+x}\left(\frac{p_2}{2}\right) . \tag{26}$$

Let

T_i = Average response time for console i .

Using Little's [7] results again, and noting that the input rate for console i is $\gamma(1 - E_i)$, we obtain

$$T_i = \frac{E_i}{\gamma(1 - E_i)} . \tag{27}$$

Thus,

$$T_1 \; \frac{1}{\mu C} \, \frac{1 + p_1 G_2}{1 - x p_1 G_2} \tag{28}$$

and

$$T_2 = \frac{1}{\mu C} \, \frac{1 + p_1 G_1}{x p_1 G_1} . \tag{29}$$

In figs. 12 and 13 we plot $\mu C T_i$ for $g_1 = 8 g_2$, $g_1 = 2 g_2$, respectively, also showing $\mu C T$ for $g_1 = g_2$. We see the discrimination possible with such priority systems. Note that as $x \to \infty$, $\mu C T_i \to 1/1 - G_j \; (i \neq j)$.

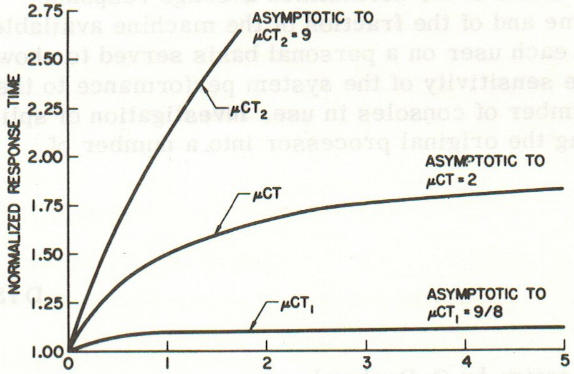

Fig. 12. Normalized response time for $G_1 = \frac{8}{9}$, $G_2 = \frac{1}{9}$.

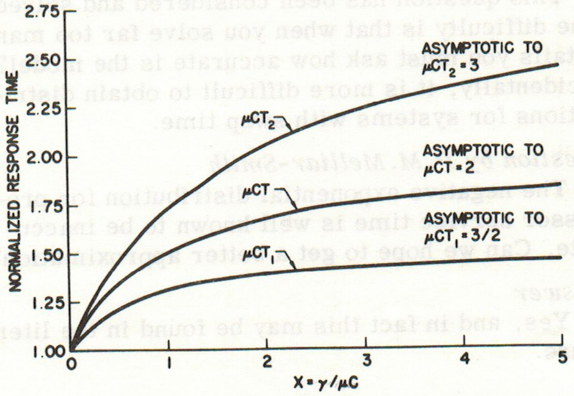

Fig. 13. Normalized response time for $G_1 = \frac{2}{3}$, $G_2 = \frac{1}{3}$.

The case for $M > 2$ is more difficult. It is possible to obtain the following bounds on E_i in the general case,

$$\frac{x}{1+x} \leqslant E_i \leqslant \frac{x}{x+G_i}$$

but these bounds are not especially tight.

5. CONCLUSION

We feel that the simple processor-sharing model herein described gives an accurate description of the behavior of the normalized average response time for finite population time-shared systems. The saturation load, $M^*=$ (think time plus service time)/(service time) is a meaningful definition for saturation, which is both intuitively pleasing and analytically significant.

Plots of the normalized average response time and of the fraction of the machine available to each user on a personal basis served to show the sensitivity of the system performance to the number of consoles in use. Investigation of splitting the original processor into a number of smaller machines, each with proportionally fewer consoles showed for $M \ll M^*$ that the degradation was large, whereas for $M \gg M^*$, the degradation was almost unnoticable (the heavily saturated case). The priority case showed the effective discrimination possible between consoles for $M = 2$.

REFERENCES

[1] H. Hyman. The time-sharing business. Datamation, Vol. 13, no. 2, February 1967, pp. 49-57.

[2] G. Estrin and L. Kleinrock. Measures, models and measurements for time-shared computer utilities. Proc. of 22nd Nat. Conf. of the ACM. August 1967, pp. 85-96.

[3] L. Kleinrock. Time-shared systems - A theoretical treatment. Journal of the ACM. April 1967. pp. 242-261.

[4] R. A. Howard. Dynamic programming and Markov processes. MIT Press (1960).

[5] T. L. Saaty. Elements of queueing theory. McGraw-Hill (1961).

[6] A. L. Scherr. An analysis of time-shared computer systems. MIT Press (1967).

[7] J. D. C. Little. A proof for the queueing formula $L = \lambda W$, Operations Research. Vol. 9 (1961) pp. 383-387.

DISCUSSION

Question by O. Rechard

From the standpoint of a user I would be much more interested in the probability of a very long delay rather than the average delay. Have you studied this?

Answer

This question has been considered and solved. The difficulty is that when you solve far too many details you'must ask how accurate is the model? Incidentally, it is more difficult to obtain distributions for systems with swap time.

Question by P. M. Melliar-Smith

The negative exponential distribution for processor service time is well known to be inaccurate. Can we hope to get a better approximation?

Answer

Yes, and in fact this may be found in the literature.

Question by M. Jones

It appears that this work is very similar to that of Scheer, of MIT. Have you any results for the dependence of think time on processor time?

Answer

No. Queuing Theory falls down when you no longer have independent distributions for service and arrival time. The point of departure for my work is the work of Scheer, as I indicated; the extensions include: a good interpretation and understanding of saturation; an investigation of partitioned systems and their comparison to the original systems; and a study of the two-priority system and its performance.