SOME RECENT RESULTS FOR TIME-SHARED PROCESSORS

Leonard Kleinrock

Associate Professor, Department of Engineering,
University of California at Los Angeles

ABSTRACT

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.

The performance measure is derived and interpreted, leading to a definition of system saturation which is a number of users, M^*, equal to (average think time plus average service time)/(average service time). This definition is both intuitively pleasing and analytically significant. Asymptotic expressions for the normalized average response time and for its inverse, the fraction of the computer available to each user on a personal basis, are given both for $M << M^*$ and $M >> M^*$. The system saturation is found to play a critical role for both asymptotic regions, as well as for the transition region.

The original system of M consoles with processor capacity C is compared to a class of comparative systems, the N^{th} class consisting of N processors, each of capacity C/N serving M/N consoles each (for N = 2, 3, 4, ...). These systems are all inferior to the original system, and this degradation in performance is discussed and graphed. For $M << M^*$, the degradation is considerable, whereas for $M >> M^*$, the effect becomes insignificant, approaching the performance of the original system. The conclusion drawn is that once the system is heavily saturated, it matters not whether the system is split into smaller systems.

SOME RECENT RESULTS FOR TIME-SHARED PROCESSORS*

Leonard Kleinrock

Associate Professor, Department of Engineering,
University of California at Los Angeles

I. INTRODUCTION

The last few years have seen the introduction and implementation of numerous operating time-shared computer systems; this activity is rapidly becoming big business [1]. More recently, the analysis of time-shared systems has begun to appear in the literature [2]. In this paper, we discuss some newly obtained results and interpretations and place them in relation to previously known results.

II. MODELS

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 is available in the literature [2].

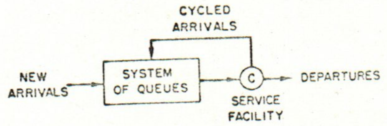

Fig. 1 Feedback Queueing Systems

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; 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.

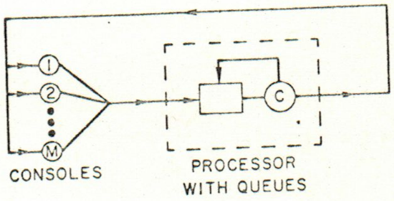

Fig. 2 Finite Population Model

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 per request, respectively. All quanta are

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

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]. We let T be the average response time and take this as our performance measure.

III. MEASUREMENTS

The analysis of the above model is summarized in Section IV. How good is our model? Scherr [4] reports on measurements carried out on the MIT time-sharing system. In Fig. 3 we show his comparison of the results of measuring this system (curve B-B fitted to the dotted data points) with the results of the model analysis (curve A-A). The figure shows that the normalized response time (see below) is accurately predicted by the model above.

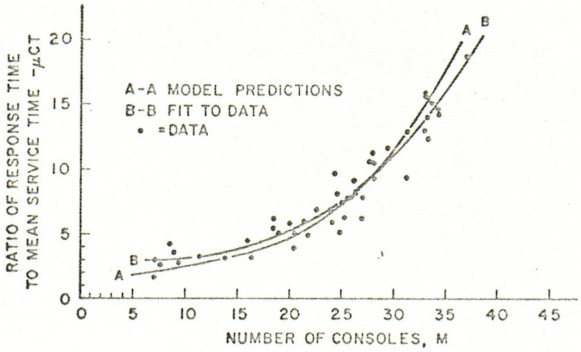

Fig. 3 Comparison of Measured and Predicted Performance

IV. ANALYSIS

To solve for T we need merely equate the customer input rate, $M\gamma[(1/\gamma)/(T+1/\gamma)]$ with the customer output rate, $\mu C(1-\pi_o)$, where π_o is the probability that all customers are in the thinking state. This yields

$$T = \frac{M/\mu C}{1-\pi_o} - \frac{1}{\gamma} \tag{1}$$

where

$$\pi_o = \left[\sum_{j=0}^{M} \frac{M!}{(M-j)!} (\gamma/\mu C)^j\right]^{-1} \tag{2}$$

In Fig. 4, we plot μCT which is the ratio of T to the average service time $1/\mu C$; this is curve A-A of Fig. 3. Note for large M, that $\mu CT \rightarrow M - \mu C/\gamma$ since π_o must approach zero. This asymptote is shown dashed in Fig. 4, and we observe that it crosses the line $\mu CT = 1$ at $M = (\mu C + \gamma)/\gamma$. We recognize that $(\mu C + \gamma)/\gamma = [(1/\mu C) + (1/\gamma)]/(1/\mu C)$ which is the ratio of average service time plus average think time to the average service time; this quantity is defined as the saturation number of users M^* in the system since in the deterministic case, at most exactly M^* users can receive $1/\mu C$ seconds of service without mutually interfering if each requires $1/\gamma$ sec. for thinking. We see that μCT begins to increase sharply in the region, $M \approx M^*$. The asymptotic form

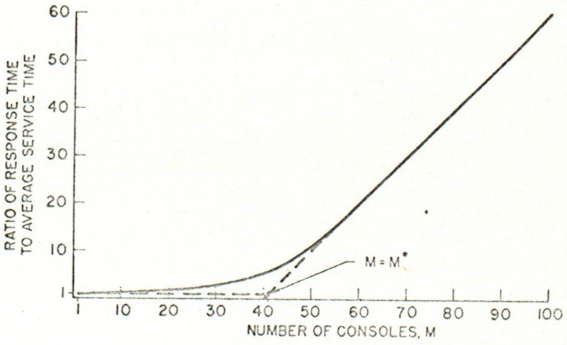

Fig. 4 Performance and Saturation

merely shows that each additional user "completely" interferes with all the other users, adding one more unit of normalized delay to the average response time.

We also consider the function $f = 1/\mu CT$ which represents the fraction of the processor which each user effectively sees as his personal processor; see Fig. 5. This shows the effect of adding additional consoles. It can be shown that the slope of f as $M \to 1$ is merely $-\mu C/\gamma$ and so the tangent shown in this figure crosses the horizontal axis at precisely $M = M^*$, the saturation load again! For $M >> M^*$, $f \to 1/(M-M^*+1)$.

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).

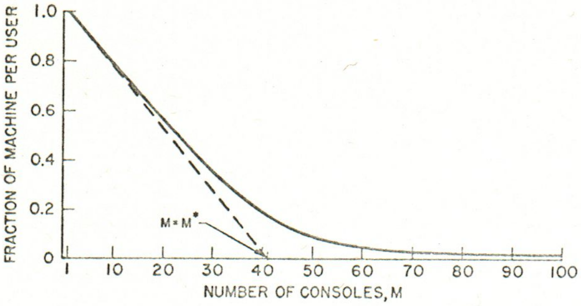

Fig. 5 Fractional Use

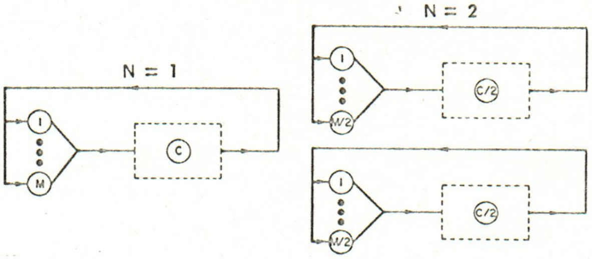

Fig. 6 Comparative Systems (N = 1, 2)

In general, we consider 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). At $M/N = 1$, the M/N, C/N system gives $\mu CT_N = N$. Note that the asymptote $\mu CT_N \to N(M/N) - \mu C/\gamma$ for $M/N >> (\mu C/\gamma N) + 1 \equiv M_N^*$ intersects the line $\mu CT_N = N$ at precisely $M/N = M_N^*$ (saturation point for the M/N, C/N system). 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.

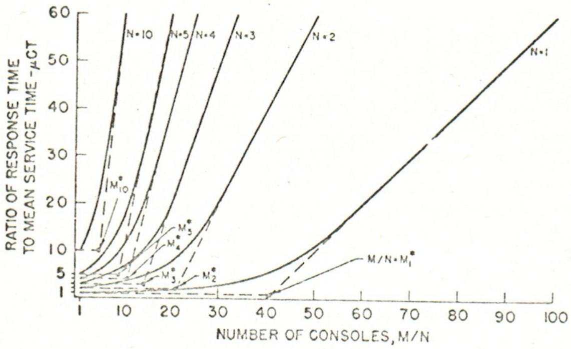

Fig. 7 Performance of Comparative Systems

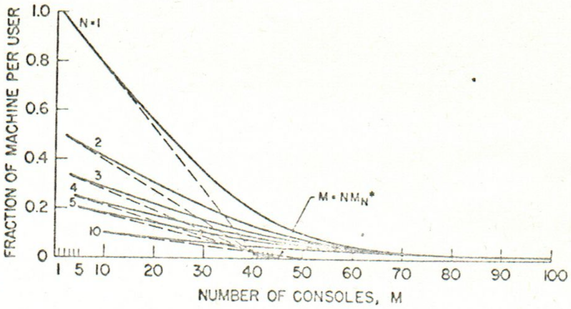

Fig. 8 Fractional Use of Comparative Systems

Note that the tangent to f_N at $M = N$ is a line intersecting the horizontal axis at $M = NM_N^*$.

Lastly, we 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. Figure 9 shows this for $N = 2$. We see that the degradation is large for $M << M^*$. Note for $M >> M^*$, that $\Delta_2 \rightarrow 0$; 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 is seen to occur in the vicinity $M \approx M^*$, indicating that the smallest rate of degradation occurs there. Figure 10 shows Δ_N for $N = 2, 3, 4, 5$, and 10; all of the comments for $N = 2$ apply to this last also.

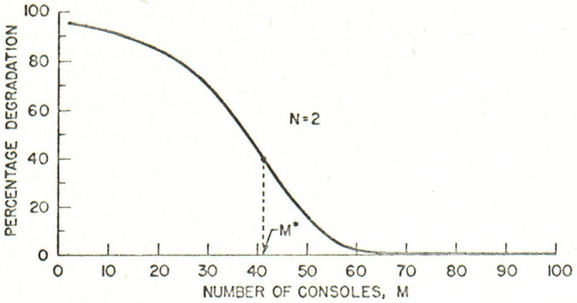

Fig. 9 Percentage Degradation of Comparative Systems (N = 2)

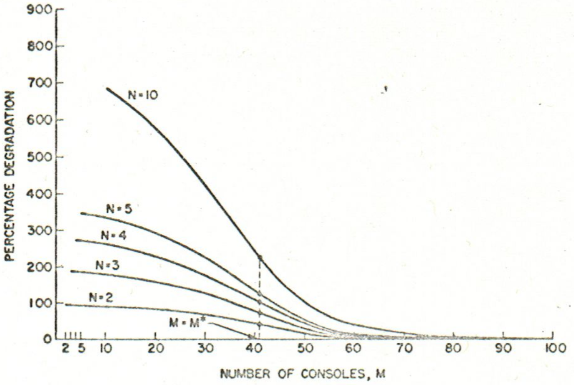

Fig. 10 Percentage Degradation of Comparative Systems (N = 2, 3, 4, 5, 10)

V. CONCLUSION

We feel that the simple processor-sharing model 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 $M << M^*$ that the degradation was large, whereas for $M >> M^*$, the degradation was almost unnoticable (the heavily saturated case).

REFERENCES

1. Hyman, H. "The Time-Sharing Business," *Datamation*, Vol. 13, No. 2, February 1967, pp. 49-57.
2. Estrin, G. and L. Kleinrock, "Measures, Models and Measurements for Time-Shared Computer Utilities," Proc. of 22nd National Conference of the ACM, August 1967, pp. 85-96.
3. Kleinrock, L. "Time-Shared Systems – A Theoretical Treatment,"*Journal of the A.C.M.*, April 1967, pp. 242-261.
4. Scherr, A.L., "Time-Sharing Measurements," *Datamation*, April 1966, pp. 22-26.

We acknowledge Jean-Loup E. Baer for computing the curves shown.