INFORMATION PROCESSING 74 – NORTH-HOLLAND PUBLISHING COMPANY (1974)

# RESOURCE ALLOCATION IN COMPUTER SYSTEMS AND COMPUTER-COMMUNICATION NETWORKS*

Leonard KLEINROCK

*Computer Science Department, University of California, Los Angeles, California, USA*

(INVITED PAPER)

Resource allocation is basically a problem in conflict resolution. Due to the bursty demands by users, resource sharing has developed as an efficient means of utilizing resources, for example, time-sharing, multiplexing, multiprogramming and packet switching. In this paper, we discuss the resource sharing tradeoffs among performance, throughput, efficiency, resource capacity and the number of resources. We show that the average response time of the system may be reduced significantly as one scales up both the throughput and the capacity. Also, a single resource system of capacity C is often superior to a system of m resources each of capacity C/m. Thus, based only on average response time, we find that a large shared single resource is to be preferred.

## 1. INTRODUCTION

The information processing industry is one of the fastest growing, most dynamic and most glamorous industries on the scientific scene today. An enormous need for access to information processing already exists in industries such as banking, insurance, medicine, education, government, retail organizations, transportation and large corporations as well as the consumer himself. It is estimated that at the end of 1971 almost a quarter of a million terminals were in use and approximately four to five million terminals are projected for 1980 [1]. In the recent Eurodata report [2,3], it was estimated that the data traffic in 17 west European countries would grow by a factor of six from 1972 to 1980 at which time the demand from terminals is estimated at $10^{12}$ bits per day; in these same countries the number of terminals will grow from eighty thousand in 1972 to an estimate of just under one-half million in 1980 and to eight hundred thousand by 1985.

Just what is it that these terminals are really accessing? The most general answer must be that they are accessing resources of various kinds in the form of processing capacity, storage capacity and communications capacity. As the capacity of these systems grows and as the data bases grow and as the number of users requesting access to these resources grows, so also must grow the rate of conflict among these users for simultaneous access. A means for allocating these resources in order to resolve the conflicting demands is one of the most important aspects of today's system design and operation. In fact, resource allocation is at the root of most of the technical (and non-technical) problems we face today in and beyond the information processing industry. These problems occur in any multi-access system in which the arrival of demands as well as the size of the demands made upon the resources are unpredictable. The resource allocation problem in fact becomes that of resource sharing and one must find a means to effect this sharing among the users in a fashion which produces an acceptable level of performance.

The need for sharing arises due to the unpredictability of the demands. Specifically, as shown in the measurements of Fuchs, Jackson and Stubbs [4,5], the behavior of users accessing computing power is extremely bursty; that is, the typical user may be characterized as one who makes demands rarely but when he does, he requires a high bandwidth in terms of communications as well as in terms of processing. Clearly, it is uneconomical to provide the full time use of a high capacity resource to such a user; on the other hand, to provide him with a smaller resource would be inadequate for his needs when they do arise. The clear solution to this dilemma is to provide a large resource which many users share in some multi-access mode (e.g., time-sharing, multiprogramming, multiplexing, etc.). If done properly this will be an economical means for satisfying the needs of bursty traffic. It is our purpose in this paper to espouse that point of view and to point out the gains which are available as the system resources grow in size to satisfy an ever increasing population of users. That which gives rise to this gain in performance is nothing more sophisticated than the law of large numbers [6]! As is well known, this law states that the collective demand of a large population of random users is very well approximated by the sum of the average demands required by that population. That is, the statistical fluctuations in any individual's demands are smoothed out in the large population case so that the total demand process appears as a deterministic (i.e., predictable) demand process.

In this paper we give quantitative measures of the gains which are available from large resources shared by large populations. The gains we discuss are beyond any effects of "economy or loss due to scale" in the management, maintenance, operation or quantity discounts associated with large systems, but rather arise due to the statistical nature of the demand. We take the system cost to be proportional to the total system capacity. This cost assumption neglects any possible "economy of scale" which would only strengthen our case in showing that large resources give improved performance. On the other hand, rather than an economy, there may be a "loss due to scale" as a result of various forms of overhead which sometimes manifest themselves in large systems. For example, the advancing semiconductor technology has produced some unusual cost structures. We intentionally omit any serious consideration of those important issues since we wish to isolate the impact of scale on system performance.

These results have clear applications to computer operating systems construction, terminal network design and computer-communication networks [7]. In the operating systems environment one is concerned with providing high performance to a population of users who attempt to share the CPU, main memory, secondary memory devices, printers, plotters, readers, punches, terminals and other system devices. In the terminal network environment, one is anxious to share the data communications capacity required to provide access to and from the terminals and the main processing facility. In computer-communication networks, we are interested in sharing both the processing facilities which are geographically distributed as well as the data communications required to connect these systems among themselves and among the terminals. Our goal, however, is not to elaborate upon the applications

*This work was supported by the Advanced Research Projects Agency of the Department of Defense (DAHC-15-73-C-0368).

but rather to focus on the effect of resource sharing in general.

## 2. RESOURCE SHARING: TRADEOFFS AND STRUCTURE

The basic performance parameters of any resource sharing system include the following:

a) the system response time or delay
b) the throughput
c) the resource capacity
d) the resource utilization

In what follows, we take the point of view that there is a stream of job requests accessing the system resource, each of which requires some number of operations from that resource. We let C denote the capacity of the resource in operations per second. Further, we let $1/\mu$ represent the average number of operations required by a job. Thus we see that the average number of seconds a job requires from the resource is simply $1/\mu C$. We let $\lambda$ denote the average number of jobs per second accessing the resource. The response time of the system is simply the time from when the job arrives until that time when its complete request has been satisfied; the average response time will be denoted by T. Similarly, the average waiting time for a job (response time minus processing time) will be denoted by W; it is clear that

$$T = W + 1/\mu C \tag{1}$$

We assume a first-come-first-served queueing discipline (although most of our results do not require this assumption). The utilization of the system resource will be denoted by $\rho$ and represents the fraction of time that the resource is busy processing jobs. A resource of capacity C, under demand by jobs whose input rate is $\lambda$ per second each of which requires $1/\mu$ operations on the average, has a utilization given simply by

$$\rho = \lambda/\mu C \tag{2}$$

Clearly, we must have $0 \le \rho < 1$ if we are to prevent the formation of infinite queues. As is well known from queueing theory [8], as the load ($\rho$) on the system grows, so grows the queue length and the response time; in fact, the average response time (T) and the average queue length are inversely proportional to $1 - \rho$. At $\rho = 1$ these quantities climb to infinity and the system is said to be unstable. The only exception to this is the case of a purely deterministic system in which the arrivals are uniformly spaced in time, each of which requires a constant amount of service; in such a case we do permit $\rho = 1$. For $\rho > 1$, these averages grow linearly with time in proportion to $\rho - 1$.

We are interested in the tradeoff relations among the response time T, the throughput $\lambda$, the resource utilization $\rho$ and the system capacity C. The system structure will affect the relationship among these performance variables in a significant way, and it is our purpose to demonstrate that the simplest structure is often superior to the others. We shall also show that large systems give significantly improved performance compared to smaller ones. Suppose we have a system in operation and find that its average response time T is larger than we desire. We may reduce T by reducing $\rho$ in one of two disagreeable ways: either by increasing the system capacity C or by reducing the system throughput $\lambda$. On the other hand, it is not generally known that this reduction in T can be obtained at constant $\rho$ if we merely scale up both the throughput $\lambda$ and the capacity C! A related tradeoff is that of attempting to increase the throughput of the system. If we simply increase $\lambda$ then indeed T will increase. However, we can maintain a constant T as both $\lambda$ and $\rho$ increase if we permit C to grow less quickly than $\lambda$. These effects and the obvious and important tradeoffs among them are investigated below.

Let us now examine some alternative structures for resource allocation and sharing. We begin by considering a collection of m resources each of which has capacity C/m and each of which is individually accessed by a job stream at rate $\lambda/m$; this structure is depicted in part (a) of fig. 1. Our intuition suggests that this system is inefficient, since there may be jobs queued up in front of one of the resources when another one is idle; therefore, let us consider part (b) of the figure in which we have a single queue accessing the collection of m resources at a total rate equal to $\lambda$. Here we expect an improvement, since no job will wait if any resource is idle. Note that both configurations have the same total utilization $\rho = \lambda/\mu C$; in part (b) the appropriate interpretation is that $\rho$ is merely the expected number of busy resources [9, Theorem 4.1].

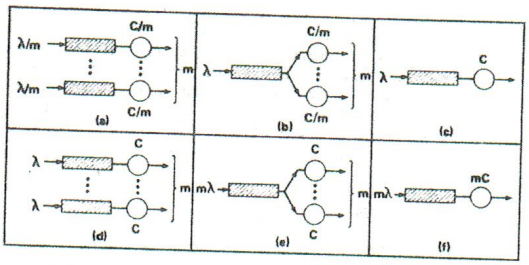


Fig. 1. Evolution of queueing structures.

Whereas the merged queue system provides an improvement over the m separate facilities, there still remains an inefficiency when there is no queue but less than all the resources are busy; in this case, some resources remain idle at a time when their services could be used in speeding the work of the other busy resources. To overcome this inefficiency, we consider part (c) in which we have now merged the resources as well as the job stream to produce a single queue whose input rate is $\lambda$ and whose capacity is C. In part (d) we show a collection of m such single resource systems similar to that in part (a) but now each has m times the input rate and m times the resource capacity and therefore the system is capable of handling more jobs per second. We have come full circle and are back to the inefficient structure of part (a) which therefore suggests we consider the system of part (e) in which we have a merged job stream at a rate $m\lambda$ which itself can be improved to the merged-queue merged-resource system of part (f). That which distinguishes the two single resource facilities in parts (c) and (f) is that in the latter we have scaled up the input and the capacity by a factor of m while maintaining a constant load $\rho = \lambda/\mu C$. Of the six systems shown, all of which have the same value for $\rho$, the last is often superior in that it has the smallest response time T. In fact if we were to further scale the input and capacity of the system, we would see yet further improvement. Below we obtain the quantitative measures of this improvement. (In fig. 2 we introduce a shorthand notation for these various structures.)

Below we use the standard notation A/B/m from queueing theory, where the symbol A describes the distribution of the time between arrivals (assumed to come from a renewal process), where B describes the distribution of service time and where m identifies the number of servers in a multiple-server system (i.e., multiple resource) such as the structure shown in fig. 1 part (b). Both A and B may take on the values M, D, G (and others) in which M describes an exponential distribution (the Markovian or memoryless

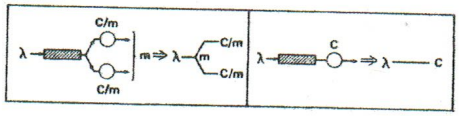


Fig. 2. Shorthand notation.

distribution), in which D describes a deterministic distribution (constant value for the random variable) and where G implies no restriction on the distribution (that is, a general distribution). In the next section, we begin by discussing the queueing system M/M/m and in the section following that we consider the system G/G/m.

## 3. THE CLASS OF SIMPLE QUEUEING SYSTEMS--M/M/m

Here we consider the system M/M/m as shown in fig. 1 part (b). The interarrival time between jobs is exponentially distributed producing a mean input rate of $\lambda$ jobs per second. Each resource has a capacity of C/m operations per second and the job lengths are exponentially distributed with mean $1/\mu$ operations per job; this produces a mean service time of $m/\mu C$ sec. per job and a system load $\rho = \lambda/\mu C$. We are interested in the performance of this system as $\lambda$, C and m vary.

The basic equations describing the behavior of this system are well known [8]. We find that $p_k$, the probability that the system contains k jobs (counting those in queue as well as those being processed), is given by

$$p_k = \begin{cases} p_o(m\rho)^k/k! & k \le m \\ p_o \rho^k m^m/m! & k \ge m \end{cases} \tag{3}$$

where

$$p_o = \left[ \frac{(m\rho)^m}{(1-\rho)m!} + \sum_{k=0}^{m-1} (m\rho)^k/k! \right]^{-1} \tag{4}$$

From these basic equilibrium probabilities one may easily calculate the average response time T as

$$T = m/\mu C + \frac{P_m}{\mu C(1-\rho)} \tag{5}$$

where $P_m$ is simply the probability that the system contains m or more jobs and is given by

$$P_m = \frac{p_o(m\rho)^m}{(1-\rho)m!} \tag{6}$$

From these equations we may examine the relationship among T, $\lambda$, C, $\rho$ and m. In particular, it was shown by the author [9, Theorem 4.2] that the value of m which minimizes T at constant $\rho$ is m = 1. In addition, it was demonstrated that T could be reduced at constant $\rho$ by increasing both $\lambda$ and C. (Results similar to these were discussed by Morse [10] and Feller [11]). There are numerous ways to display this tradeoff, some of which we now present.

Perhaps the most striking display of the effect of large systems may be seen in fig. 3 where we plot the normalized average response time

$$\frac{\mu CT}{m} = 1 + \frac{P_m}{m(1-\rho)} \tag{7}$$

The normalization is simply the average service time for a job in one of the m servers, and this normalization successfully removes all parameters from the expression leaving only m and $\rho$. In this figure, we see that all the curves begin at unity for the value $\rho = 0$ since at this point $P_m = 0$ (m = 1,2,...). As m increases for a given value of $\rho$, we see that the normalized delay decreases in a dramatic fashion and as $m \to \infty$ we see that the behavior approaches that of the pure deterministic system (D/D/1) in which no queues form until we exceed the value $\rho = 1$. This figure, however, does not permit one to compare the various structures from fig. 1 equitably since we cannot investigate the behavior as $\lambda$ and C vary in some observable way; the difficulty of course is that

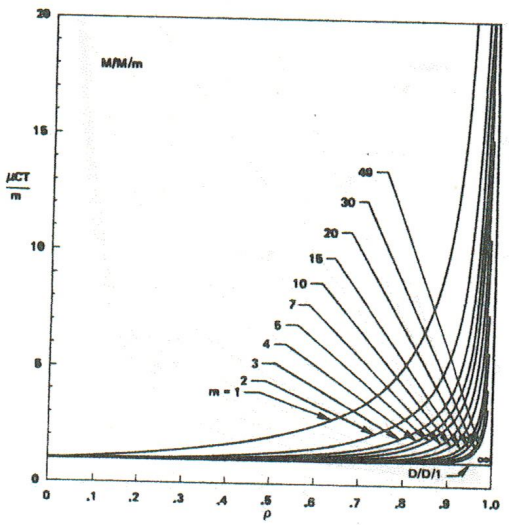


Fig. 3. Normalized average response time (eq.(7)).

we have normalized the response time and have therefore lost an essential parameter in our performance evaluation. If we return to eq.(5), we see the manner in which the average service time $m/\mu C$ affects the response. In fig. 4 we plot eq.(5) under the constant conditions $\mu = \mu_o = 1$ and $C = C_o = 1$; thus we assume that the total capacity of our M/M/m system is held constant at $C_o = 1$ and this is shared equally among the m resources as in fig. 1, part (b). In fig. 4 we see quite the opposite behavior from that in fig. 3, namely that the response time degrades as m increases at constant load $\rho$. This is simply a demonstration that the system of fig. 1 (c) is superior to that of fig. 1 (b). Since $\mu C$ was held constant in fig. 4, we obviously were varying $\rho$ by a variation in $\lambda$. We need not have held $\mu C$ constant but rather could have held $\lambda$ constant, in which case we choose to rewrite eq.(5) as

$$T = \frac{m\rho}{\lambda} + \frac{\rho P_m}{\lambda(1-\rho)} \tag{8}$$

In fig. 5 we select $\lambda = \lambda_o = 0.8$ as we permit $\rho$ to vary by changing $\mu C$. Clearly, the response function will now branch out from the origin and again we see

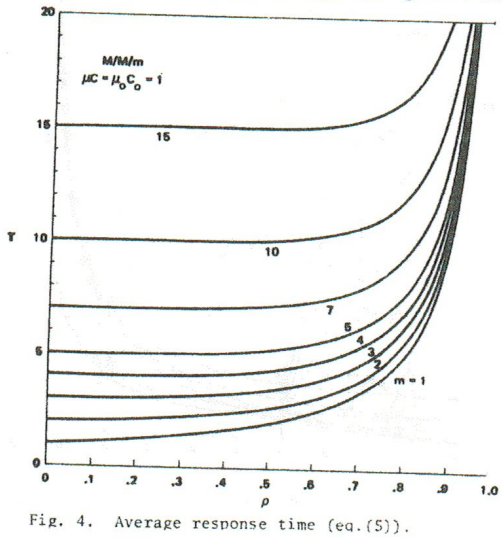


Fig. 4. Average response time (eq.(5)).

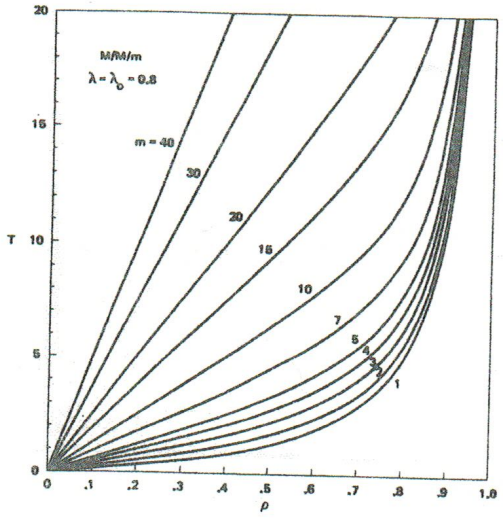


Fig. 5. Average response time (eq.(8)).

the degradation in response time as m increases at a constant ρ; this too demonstrates the superiority of a single resource as opposed to multiple resources at constant total resource capacity.

It is convenient to superimpose portions of these last three figures, and this we have done in fig. 6, where we show the curves for m = 1,2 from figs. 3, 4 and 5 (note that the curve m = 1 from figs. 3 and 4 are identical). In addition, we show a curve for m = 1 in the case when $\mu C = 2\mu_0 C_0 = 2$ and $\lambda = 2\lambda_0 = 1.6$. Also in fig. 6 we have used our shorthand notation from fig. 2 to identify the exact system structure and system parameters for the cases ρ = 0.4 and ρ = 0.8. This allows us to see the proper tradeoff relations among the various structures of fig. 1. We draw attention to the value ρ = 0.8. The m = 1 system at point B gives a response time of five seconds, and this is seen to be superior to the M/M/2 system with the same input rate and the same total capacity at point A; again this demonstrates the superiority of single resource systems. On the other hand, doubling the input rate and doubling the total system capacity for M/M/2 brings us to point C and provides a large improvement over that of point B;

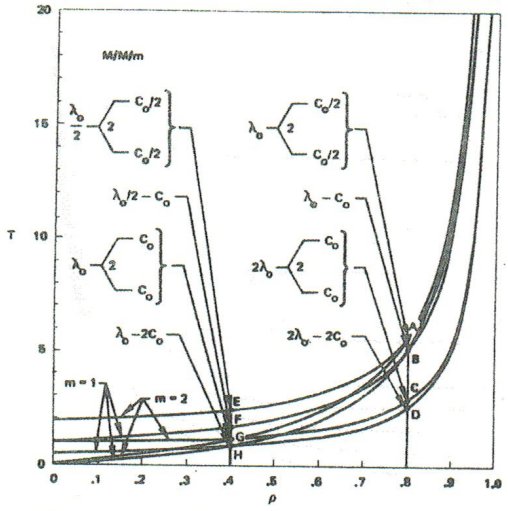


Fig. 6. Comparison of average response times.

this is an interesting observation since we have really done two things simultaneously, namely we have increased the size of the system (from this we expect an improvement) and we have also split the resource into two (from this we expect a degradation). The overall effect is a large improvement, but we do see at point D that if we were to merge the two resources into one of equivalent size then we gain further. (A similar behavior may be found at ρ = 0.4.) The message is clear in this case; large single shared systems yield improvements in the average response time!

Let us quantify these observations analytically. We find it convenient to expose the parameters of the response time and so we temporarily write T = T(m,λ,C) where once again C denotes the total capacity of the m resource system (throughout we assume that μ is constant). As mentioned above, we have the important result

$$T(1,\lambda,C) \le T(m,\lambda,C) \qquad m = 1,2,3,\ldots \tag{9}$$

demonstrating the improvement due to single resource systems. Furthermore, since $P_m$ and $p_0$ depend only upon m and ρ (and not upon λ and C separately), then from eqs.(5 and 8) we see that scaling λ and C together must give an improvement in T proportional to the scaling factor, namely, for a > 0

$$T(m,a\lambda,aC) = \frac{1}{a}\,T(m,\lambda,C) \tag{10}$$

Of course the same is true for the average waiting time in the queue W = W(m,λ,C), that is

$$W(m,a\lambda,aC) = \frac{1}{a}\,W(m,\lambda,C) \tag{11}$$

Thus again we see the significant gains due to scaling up the system parameters. These observations were made by the author [9, Sect. 4.3] where he also generalized the single server case to yield the result

$$T(1,a\lambda,bC) = \frac{(1-\rho)}{b[1-\rho(a/b)]}\,T(1,\lambda,C) \tag{12}$$

$$W(1,a\lambda,bC) = \frac{a(1-\rho)}{b^2[1-\rho(a/b)]}\,W(1,\lambda,C) \tag{13}$$

where $\rho = \lambda/\mu C$ and $a < \mu Cb/\lambda$. We note that when a = b, then these last equations reduce to our earlier results for m = 1. Whereas T and W vary as we scale λ and C as just described, we find by using Little's result [8] in eq.(8) that

$$\lambda T(m,\lambda,C) = \bar{N}(m,\lambda,C) = \bar{N}(m,a\lambda,aC) \tag{14}$$

and

$$\lambda W(m,\lambda,C) = \bar{N}_q(m,\lambda,C) = \bar{N}_q(m,a\lambda,aC) \tag{15}$$

where $\bar{N}$ and $\bar{N}_q$ are the average number of jobs in the system and in the queue respectively. This tells us that the average number of jobs does not depend upon the scaling parameters, but remains constant for a given value of ρ.

Eqs.(11, 13, and 15) give the corresponding results for W as eqs.(10, 12, and 14) give for T. It remains for us to find the relationship for W which corresponds to the relationship in eq.(9). First we observe from eq.(5) that

$$W(m,\lambda,C) = \frac{P_m}{\mu C(1-\rho)} \tag{16}$$

Our concern is with the behavior of W as m varies. This dependence is contained in the expression $P_m$ which is given in eq.(6) and which represents the probability that an M/M/m system contains m jobs or more, that is

$$P_m = \sum_{k=m}^{\infty} P_k \tag{17}$$

Now from Theorem 4.1 of [9] we know that ρ is simply the average fraction of busy resources which may be written as

$$\rho = \sum_{k=0}^{m-1} \frac{k}{m} p_k + \sum_{k=m}^{\infty} p_k \qquad (18)$$

Since the first of these two sums must be non-negative we have immediately that

$$\rho \geq p_m \qquad (19)$$

Further we see for the single resource system that

$$W(1,\lambda,C) = \frac{\rho}{\mu C(1-\rho)} \qquad (20)$$

Thus from eqs.(16, 19, and 20) we have the relation we were seeking, namely

$$W(1,\lambda,C) \geq W(m,\lambda,C) \qquad m = 1,2,3,\ldots \qquad (21)$$

This tells us that the average waiting time in an M/M/m system decreases with m whereas the average response time increases with m! This effect may be seen in fig. 7 where we plot T and W at constant values of $\rho$ as a function of m with $\mu C = 1$. Note that $T(m,\lambda,C) - W(m,\lambda,C) = m/\mu C$ as shown in that figure. Thus if average queueing time is the performance measure rather than average response time, then partitioned systems are superior; in the computer systems under study, we take the point of view that the total response time is the appropriate performance measure and so the single resource systems are preferred.

We may display these improvements in yet another way if we focus on the single resource system M/M/1. For example, in fig. 8 we show the effect of increasing $\lambda$ on the average response time for m = 1; the appropriate expression for T is

$$T = \frac{\rho/\lambda}{1-\rho} \qquad (22)$$

At constant $\rho$ we observe the reduction in delay as we increase $\lambda$ (and therefore also C). In fact, we observe that the average response time improves by a factor of 50 as we pass from $\lambda = .1$ to $\lambda = 5$ as anticipated from eq.(10). Similarly, in fig. 9 we show the improvement in efficiency (i.e., resource utilization) at constant average response time as the system is scaled up. The function plotted there is simply the solution to eq.(22) for $\rho$, namely

$$\rho = \frac{\lambda T}{1+\lambda T} \qquad (23)$$

A further result of this sort was given by the author [9, Theorem 4.4] for a system of m separate M/M/1 queues, the ith of which has an input rate $\lambda_i$ and a capacity $C_i$. Subject to the constraints $C = C_1 + C_2 + \cdots + C_m$ and $\lambda = \lambda_1 + \lambda_2 + \cdots + \lambda_m$, it was shown that the assignment of capacity and traffic which minimizes the overall average response time $T = \lambda_1 T_1/\lambda + \lambda_2 T_2/\lambda + \cdots + \lambda_m T_m/\lambda$ (where $T_i$ is the average response for the ith system) is $\lambda_i = \lambda$ and $C_i = C$ for $i = i_o$ (and $\lambda_i = 0$ and $C_i = 0$ for $i \neq i_o$); this is for any value of $i_o$. Again, a single large shared resource is optimum.

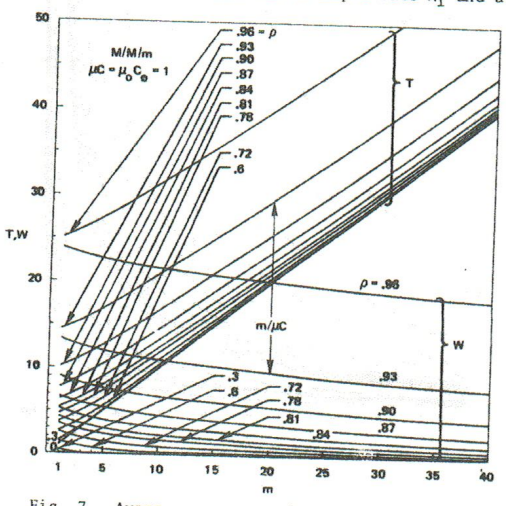


Fig. 7. Average response time and average wait at constant loads (eqs.(5 and 16)).

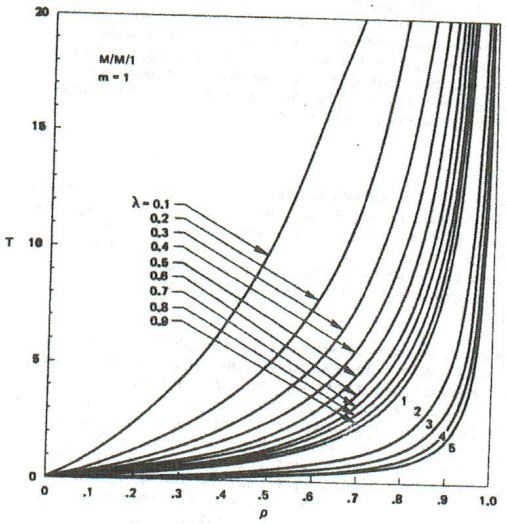


Fig. 8. Average response time for various input rates for m = 1 (eq.(22)).

Thus, in summary, for the system M/M/m we have the two basic relationships given in eqs.(9 and 10). Eq.(10) tells us that large systems (scaling up the input rate and the system capacity) yield improvements in average response times which are proportional to the scaling factor. Eq.(9) tells us that for a given scale factor the single resource system is superior to the multiple resource system. Let us now investigate these trading relationships for other than these simple queueing systems and attempt to extend them to more general distributions.

## 4. MORE COMPLEX QUEUEING SYSTEMS--G/G/m

Computer systems seldom display the simple statistical behavior which we have assumed for the system M/M/m in the previous section. In particular, job processing times are seldom exponentially distributed and so it behooves us to remove this constraint. The exponential assumption on the interarrival times (that is, assuming a Poisson arrival process) is the

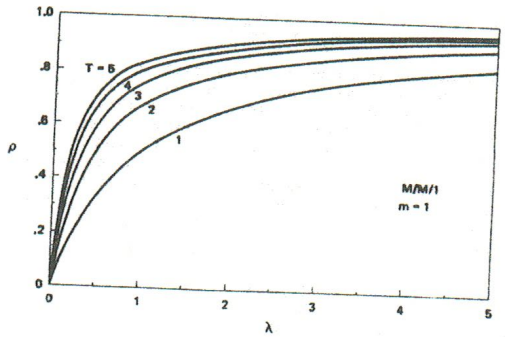


Fig. 9. Throughput versus input for m = 1 and various response times (eq.(23)).

lesser of the two evils in Sect. 3; if we temporarily accept the Poisson assumption, then we are led to considerations of the theory for M/G/m queues. Unfortunately, the theory is badly lacking in this case; however, the famous Pollaczek-Khinchin mean value formula for the average response time is available for the M/G/1 queue, namely [8]

$$T = \frac{1}{\mu C} + \frac{W_o}{1-\rho} \tag{24}$$

where $W_o = \lambda b_2/2$ and where $b_2$ is the second moment of a job's processing time. We note that this expression for T is similar to that given in eq.(5) when m = 1, namely $T = 1/\mu C + \rho/[\mu C(1-\rho)]$. Thus the improvements discussed with reference to figs. 8 and 9 as the capacity and input are scaled up also applies to this case.

However, we are interested in the much more general system G/G/m. We now permit an arbitary distribution of interarrival times and of service times for jobs in a multiple server system. We seek to obtain trade-off relationships similar to those we obtained in the previous section. Here, the theory is in worse shape than for M/G/m; we cannot even give an exact expression for T in G/G/1! The best we can do is to work with known bounds on the system performance and we will show that these bounds predict behavior not unlike that which we were able to obtain above.

We begin with the single server system G/G/1. Once again, we assume that the average arrival rate of jobs is given by $\lambda$ and that the variance of these jobs' interarrival times is given by $\sigma_a^2$ (in the case of a Poisson job stream we have $\sigma_a^2 = 1/\lambda^2$). Also the average number of operations per job will again be denoted by $1/\mu$ and with a variance given by $\sigma_p^2$. The single resource is assumed to have a processing capacity of C operations per second. Thus the average processing time for a job is again given by $1/\mu C$ and with a variance which we denote by $\sigma_b^2$. We see that $\sigma_b^2 = \sigma_p^2/C^2$. A useful descriptor for a random variable, for example the interarrival time, is its squared coefficient of variation which we shall denote by $C_a^2$ and which equals

$$C_a^2 = \lambda^2\sigma_a^2 \tag{25}$$

Namely, it is the ratio of the variance to the mean squared. For the processing time we have the corresponding quantity

$$C_b^2 = \mu^2C^2\sigma_b^2 = \mu^2\sigma_p^2 = C_p^2 \tag{26}$$

where $C_p^2$ is the squared coefficient of variation for the number of operations required by a job. Whereas T is in general unknown for G/G/1 there does exist an upper bound for this quantity first derived by Kingman [12] and which is given by

$$T \le \frac{1}{\mu C} + \frac{\lambda\left(\sigma_a^2+\sigma_b^2\right)}{2(1-\rho)} \tag{27}$$

This bound improves as $\rho \to 1$; in fact it is known in this limit that the response time itself is exponentially distributed with a mean given by this upper bound. We are interested in observing the behavior of this expression as we scale up both $\lambda$ and C as earlier. In so doing, of course, we will change the interarrival time parameters but will maintain constant all coefficients of variation. In order to see the effects of this scaling, we rewrite eq.(27) in terms of these coefficients, namely

$$T \le \frac{\rho}{\lambda} + \frac{C_a^2+\rho^2C_p^2}{2\lambda(1-\rho)} = T_U(1,\lambda,C) \tag{28}$$

where once again $\rho = \lambda/\mu C$. This last expression is very similar to that given in eq.(8) for m = 1 (we could just as well have written it in terms of $\mu C$).

Now we clearly see the effect of scaling $\lambda$ and C simultaneously, namely the (bound on the) average response time drops in inverse proportion to this scaling factor. Using our earlier notation and applying it to $T_U$ we have

$$T_U(1,a\lambda,aC) = \frac{1}{a}T_U(1,\lambda,C) \tag{29}$$

In fig. 10 we plot a family of upper bounds on T for G/G/1 holding $\lambda$ constant for each member of the family; this produces a set of curves similar to that in fig. 8 for M/M/1 and once again shows the significant improvement due to scaling as given in eq.(29). In fig. 10 we have taken $C_a^2 = C_p^2 = 1$ which happen to be the values corresponding to the system M/M/m; therefore we may compare the bounds in fig. 10 with the true values for M/M/1.

For the behavior of the multiple resource system, we now turn our attention to G/G/m. Exact values for T again are unknown and so we resort to the known bounds derived by Kingman [13] and Brumelle [14], namely

$$T \le \frac{m}{\mu C} + \frac{\lambda\left[\sigma_a^2+\left(\sigma_b^2/m\right)+(m-1)/\mu^2C^2\right]}{2(1-\rho)} \tag{30}$$

If we rewrite this equation in terms of the squared coefficients of variation we then obtain

$$T \le \frac{m\rho}{\lambda} + \frac{C_a^2+m\rho^2C_p^2+(m-1)\rho^2}{2\lambda(1-\rho)} = T_U(m,\lambda,C) \tag{31}$$

Again we see the improvement due to scaling at a constant $\rho$, that is

$$T_U(m,a\lambda,aC) = \frac{1}{a}T_U(m,\lambda,C) \tag{32}$$

Unfortunately, the bound for G/G/m in eq.(31) is not especially tight and so it is difficult to use it in showing that m = 1 is the optimum system as we were able to do for M/M/m as in eq.(9). However, it has been shown by Brumelle [14] that in the case when $C_p^2 \le 1$ then m = 1 is optimum, that is

$$T(1,\lambda,C) \le T(m,\lambda,C) \tag{33}$$

This result extended the results of Stidham [15] for G/M/m, G/D/m and $G/E_k/m$ where $E_k$ is a k-stage Erlangian distribution [8]. Thus we see that the single large shared resource is superior in terms of response time for a large class of G/G/m systems. However, Brumelle did give an example which showed that G/G/2 can be superior to G/G/1 when $C_p^2 > 1$.

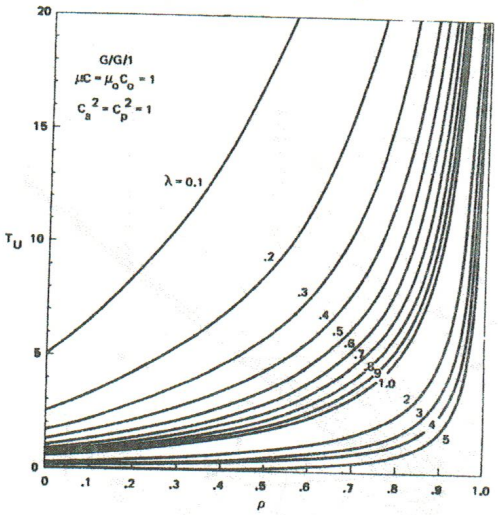


Fig. 10. The upper bound on average response time at constant input rates G/G/1 (eq.(28)).

An improvement for the bound in eq.(31) for G/G/m was conjectured by Kingman [13, 16] and takes the form of the following approximation

$$T \cong \frac{m}{\mu C} + \frac{\lambda\left[\sigma_a^2+\sigma_b^2/m^2\right]}{2(1-\rho)} \qquad (34)$$

In fact it has recently been shown by Brumelle [17] that this approximation forms an upper bound for the system G/M/m. Recently, Schrage [18] has been able to show that so long as the processing time is bounded by some finite quantity then Kingman's conjecture holds as an approximation for the system G/G/m as $\rho \to 1$. If once again we express this new upper bound (or approximation) in terms of the coefficients of variation we find

$$T \le \frac{m\rho}{\lambda} + \frac{C_a^2+\rho^2 C_p^2}{2\lambda(1-\rho)} = T_U' \qquad (35)$$

or, in terms of $\mu C$ we have

$$T \le \frac{m}{\mu C} + \frac{\left(C_a^2/\rho\right)+\rho C_p^2}{2\mu C(1-\rho)} = T_U' \qquad (36)$$

Once again we have the relationship given in eq.(32) regarding the scaling effect. We note further that the portion of $T_U'$ which corresponds to the waiting time in queue (the second term) is independent of m! $T_U'$ is far superior to the bound given in eq.(31) for G/G/m. We may reasonably assume that processing times are bounded and may therefore use this improved expression for the mean response time especially when $\rho \to 1$. Furthermore $T_U'$ is far simpler than the exact expression for M/M/m given in eq.(5) as regards its dependence upon m. In fig. 11 we plot the upper bound for T and W as a function of m for various $\rho$ in a fashion similar to that of fig. 7; we take the case $\mu C = \mu_0 C_0 = 1$ and $C_a^2 = C_p^2 = 1$. Again we note that these values for the coefficients of variation are the appropriate ones to use for the system M/M/m; thus we may compare figs. 7 and 11 in order to see the effectiveness of the bound in this one case. In any event, we make the observation that the response time degrades with increasing m at constant $\rho$.

Thus in this section we have shown that the concept of large shared single resources is once again the direction in which we find improvements in the mean response time of the system. We may further point out by observing eqs.(5, 8 and 22) for the M/M/m system and eqs.(28, 31, and 36) for the G/G/m system, that forming the product $\lambda T$ yields equations which are independent of the scaled parameters; this product is simply the average number of jobs in the system (by Little's result [15]). Simply stated, then, if $\lambda T$ is to be constant, then an increase in $\lambda$ must give a corresponding decrease in T.

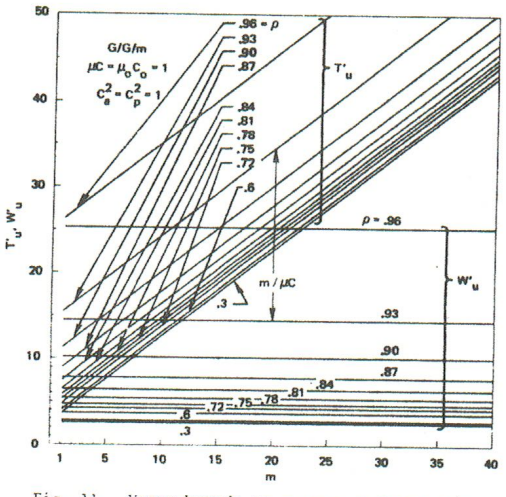


Fig. 11. Upper bounds on average response time and average wait for G/G/m (eq.(36)).

## 5. APPLICATIONS

In this section, we briefly draw attention to the application of our results to computer systems and computer-communication networks.

In the case of single resource time-shared computer systems [7], we expect the results to carry over directly. In this case, each scheduling algorithm must be examined in the limit as the resource capacity and input rate are scaled up. The flavor of these results can be seen by reference to the author's paper (jointly with Muntz and Hsu) published in previous IFIP Congress Proceedings [19] in which tight upper and lower bounds were given for the response time as a function of required service time in the CPU. If one examines these bounds in the scaled limit of interest to this current paper, then one finds that the upper bound approaches the response in a first-come-first-served system and the lower bound approaches zero; moreover, the first-come-first-served system yields a delay which itself approaches zero in our limit, thus showing that all response functions improve with the scaling of input and capacity. In the 1968 meeting of this Congress, the author delivered a paper in which a finite console model for time-shared systems was studied [20] under exponential assumptions. There it was shown that partitioned systems were inferior to single resource systems (the comparison was between the systems of part (a) and part (c) in fig. 1). In a finite population model such as discussed there, an appropriate scaling will be obtained if we hold the "saturation number" constant as we scale the system capacity and the "thinking rate" of customers proportionally (by maintaining a constant saturation number we are in a sense holding the system load constant). It can be seen there that the expression for T behaves as in eq.(10) here, namely the improvement is proportional to the scaling factor. The results of Jackson [21] and Gordon and Newell [22] on queueing networks have recently been applied quite successfully in the modeling of multiple resource time-shared systems [23, 24, 25] and it is interesting to note, for example, that Jackson's model yields resource behavior which is characterized as an M/M/m system and so these resources also enjoy the benefits of scaling which we discussed in Sect. 3 above.

In the case of computer-communication networks, the basic characterization for the analysis of network delay has been to decompose it into a weighted sum of single resource delays (each communication channel is a resource); that is, [7, 9]

$$T = \sum_i \frac{\lambda_i}{\gamma} T_i \qquad (37)$$

Here T is the average delay to messages passing through the network, $T_i$ is the average delay messages experience in accessing the ith communication channel in the network, $\lambda_i$ is the traffic carried by the ith channel and $\gamma$ is the total message traffic entering the network. In this case the appropriate scaling to consider is at the network level; that is, the quantities to scale up are the network throughput, $\gamma$ and the capacity of each of the communication channels. In so doing, the ratios $\lambda_i/\gamma$ remain fixed (the $\lambda_i$'s also scale) but the channel delays $T_i$ each behave as do the resources considered in Sects. 3 and 4 above, thereby yielding the same benefit in the performance measure T (that is, an improvement equal to the scaling factor). Another interesting effect in packet switched networks has been observed in the behavior of T in these networks as described in [7], namely that the message delay behaves almost as if the network were a D/D/1 system; that is, the performance is approximately that of an unloaded system until a

critical load value is reached, at which point the system goes unstable. This is an example of the deterministic behavior due to the law of large numbers we had discussed earlier. In this case, the phenomenon is due to the many terms in eq.(37) which contribute to T (one term for each communication channel).

## 6. CONCLUSIONS

Our goal in this paper has been to isolate the effect of scaling the input rate and system capacity of a shared resource. We found that an important performance measure, the average response time, improved significantly in the case of a single large shared resource. The improvement came about for two reasons; first, and principally, due to the simultaneous increase in the system throughput and capacity and secondly due to the merging of the resources into one common resource. We found that these results apply to rather general queueing systems in the class G/G/m and have given some indication as to where these results have application in the field of information processing.

Whereas our theme has been that "bigger is better," the reader is cautioned that this statement has been established here only when one studies the effects of scaling the system parameters in isolation. As we had mentioned earlier, the practical problems of overhead, technology and large systems management may dominate the performance variables and it is those which would then bear careful investigation. Our goal has been to point out the possible gains in the clean environment of our study.

## ACKNOWLEDGEMENT

The author is pleased to acknowledge Fouad Tobagi and Brent Ellerbroek for their assistance in preparing the graphs.

## REFERENCES

[1] Data channels, vol. 1, no. 3, Verona, New Jersey, November 1973.

[2] Eurodata--a market study on data communications in Europe, 1972-1985, a study sponsored by the European Conference of Postal and Telecommunications Administrations, 1973.

[3] R. A. Peters and K. M. Simpson, Eurodata: data communications in Europe 1972-1985, Datamation, vol. 19, no. 12, December 1973, 76-80.

[4] E. Fuchs and P. E. Jackson, Estimates of distributions of random variables for certain computer communications traffic models, Communications of the ACM, vol. 13, no. 12, December 1970, 752-757.

[5] P. E. Jackson and C. D. Stubbs, A study of multi-access computer communications, SJCC, AFIPS Conference Proc., vol. 34, 1969, 491-504.

[6] H. Cramer, Mathematical methods of statistics, Princeton University Press, 1946.

[7] L. Kleinrock, Queueing systems, vol. II: computer applications, to be published by Wiley Interscience Fall 1974.

[8] L. Kleinrock, Queueing systems, vol. I: theory, to be published by Wiley Interscience Summer 1974.

[9] L. Kleinrock, Communication nets: stochastic message flow and delay, McGraw-Hill, New York, 1964, out of print. Reprinted by Dover Publications, 1972.

[10] P. M. Morse, Queues, inventories and maintenance, John Wiley & Sons, Inc., New York, 1958.

[11] W. Feller, An introduction to probability theory and its applications, John Wiley & Sons, Inc., New York, 1950.

[12] J.F.C. Kingman, Some inequalities for the GI/G/1 queue, Biometrika, vol. 49, 1962, 315-324.

[13] J.F.C. Kingman, Inequalities in the theory of queues, J. Royal Statistical Society, Series B, vol. 32, 1970, 102-110.

[14] S. L. Brumelle, Some inequalities for parallel server queues, Oper. Res., vol. 19, 1971, 402-413.

[15] S. Stidham Jr., On the optimality of single-server queueing systems, J. ORSA, vol. 18, no. 4, July-August 1970, 708-732.

[16] J.F.C. Kingman, The heavy traffic approximation in the theory of queues, in W. L. Smith and W. E. Wilkinson (eds.), Proc. Symposium on Congestion Theory, University of North Carolina, Chapel Hill, 1964, 137-169.

[17] S. L. Brumelle, Bounds on the wait in a G/M/k queue, Management Science, vol. 19, no. 7, 1973, 773-777.

[18] L. Schrage, private communication, February 1974.

[19] L. Kleinrock, R. R. Muntz, and J. Hsu, Tight bounds on average response time for processor-sharing models of time-shared computer systems, Information Processing 71, August 1971, TA-2, 50-58.

[20] L. Kleinrock, Certain analytic results for time-shared processors, Information Processing 68, August 1968, 838-845.

[21] J. R. Jackson, Jobshop-like queueing systems, Management Science, vol. 10, no. 1, October 1963, 131-142.

[22] W. J. Gordon and G. F. Newell, Closed queueing systems with exponential servers, Oper. Res., vol. 15, 1967, 254-265.

[23] J. Buzen, Queueing network models of multi-programming, Ph.D. Thesis, Div. of Engineering and Applied Science, Harvard University, Cambridge, Massachusetts, 1971.

[24] C. G. Moore III, Network models for large-scale time-sharing systems, Technical Report no. 71-1, Dept. of Industrial Engineering, University of Michigan, Ann Arbor, Michigan, April 1971.

[25] R. R. Muntz and F. Baskett, Open, closed and mixed networks of queues with different classes of customers, Technical Report no. 33, Stanford Electronics Laboratories, Stanford University, August 1972.