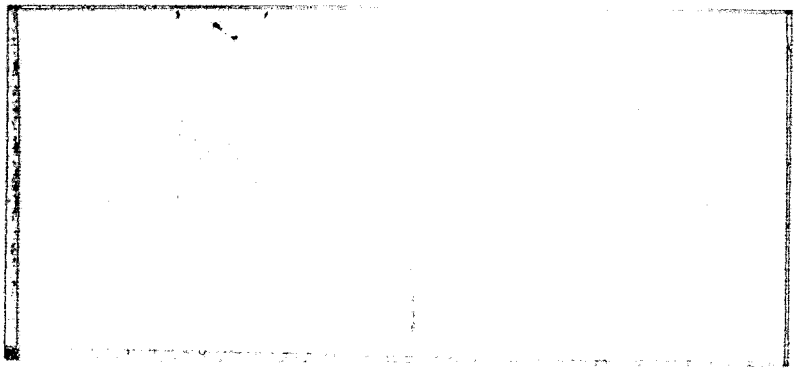
DRA



PROGRAM IN INFORMATION POLICY

ENGINEERING-ECONOMIC SYSTEMS DEPARTMENT

STANFORD UNIVERSITY • STANFORD, CALIFORNIA 94305

THE ECONOMICS OF TIME SHARED COMPUTING: CONGESTION, USER COSTS AND CAPACITY*

By

Carson E. Agnew

Report No. 45

October 1982

Partially Supported by:
National Science Foundation Grant GJ 36392X
National Science Foundation Grant IST-8108350
National Aeronautics and Space Administration Contract NASW-3204

Program in Information Policy

Terman Engineering Center
Engineering-Economic Systems Department
Stanford University Stanford, CA 94305

ABSTRACT

Time shared systems permit the fixed costs of computing resources to be spread over large numbers of users. However, "bottleneck" results in the theory of closed queuing networks can be used to show that this economy of scale will be offset by the increased congestion that results as more users are added to the system. If one considers the total costs, including the congestion cost, there is an optimal number of users for a system which equals the "saturation" value usually used to define system capacity.

Designers of time shared systems face a trade-off between the economies of scale associated with sharing computing resources and the congestion resulting from users' contention for them. Congestion increases the users' total costs because their time could be used for other purposes than console interaction. This paper explores this trade-off using results from the theory of closed queuing networks.[1] This theory has been developed and validated many times in the last decade and more. When combined with a simple model of costs, the theory shows that in the short run--i.e., when the system's configuration is fixed--a curve of average total cost per user plotted against the number of users supported is U-shaped. The minimum point of this curve determines the number of users N^* that can be served at lowest cost by this configuration. Moreover, N^* is equal to the "capacity" given by "bottleneck" models, e.g., [4], [10],[14],[15]. That is, it is always economic fully to utilize the bottleneck server.

System Cost Model

This paper employs a general definition of "cost" used in economics. That is, the cost of a good or service measured by the value given up when it is used for one purpose rather than another. As implied by this definition, cost is a way of comparing alternatives and not an intrinsic property of a good or service. While money is sometimes a convenient metric for costs, goods or services which have no observable money price may still be costly. Here we consider two cost elements which are measured in money terms directly and a third element, user time costs, that can be expressed indirectly in money terms.

The first type of money costs are so-called "fixed" costs, which cannot be varied from one day to the next. Fixed costs depend on the system configuration, including, for instance, equipment rental, power, heat, space and labor. However, in the long run these costs can be varied by changing configuration or location or by hiring and firing employees.

"Variable" costs, which depend on the output of the system, are the second type of money cost. This paper uses a model with a single class of users. Therefore, output can be measured in units of a homogeneous commodity called "computing services."[2] One unit of computing services is comprised of a bundle of goods and services which are consumed by one user. The next section shows how to calculate an index of computing services from the parameters of a queuing network model. In general terms, a unit of computing service includes computing power (measured by CPU cycles, memory allocation, and so forth), communication facilities, and user support in appropriate proportions.

Because of congestion, however, the number of units of service delivered to the system's users during some time period (i.e., the throughput) varies. More service is delivered per hour per user when the system is lightly loaded than when it is heavily loaded. Rather than measure output (i.e., computing services) so that it varies with congestion, we will take the variable costs to be proportional to the number of users, N, and work with N parametrically. Thus, if variable costs per user per hour, such as terminal and line costs, are c, the variable costs per hour will be cN while throughput is (say) $X(N)$ units of computing service per hour.

The third major source of costs is not always expressed in money terms. This is the cost to users' of the time spent interacting with a time sharing system. Time is costly because the time spent using a time sharing system almost always could be put to some other productive use. Each user's time cost can be thought of as the product of (a) the time required to perform some task on the system and (b) the money cost of a unit of the user's time. In our model, we let w represent the money price of a user's time.

Time costs often show up as productivity losses. For example, consider an on-line order processing application. (Processing one application requires a fixed number of units of computing service.) Hypothetically, suppose it takes one second of machine time to process an application. The total time required to process the application includes this time, plus the "think time" of the human operator. If the think time is, say, 20 seconds the total time per order processed will be 21 seconds, and the rate at which orders are processed is $3600/21 = 171$ per hour. The cost per order processed is clearly $w/171 = w(21)/(3600)$, where the units of w are dollars per hour. Now, suppose congestion increases the system's response time, so that it takes, say, 30 seconds to deliver the same amount of processing capability (i.e., 1 second). The order processing rate drops to $3600/30 = 120$ per hour, which we see as a loss of productivity because the cost per order is now $w/120$.

Delay in a Time Sharing System

The above example illustrates the general point that each user's time costs are proportional to the time required to deliver one unit of computing services.[3] We now consider how long this takes, using the so-called "bottleneck" results of Muntz and Wong [14] and Chang and Lavenberg [4] for closed queuing network models. These results provide a simple, very general relation between the speed of the system, the number of users, and the time required to provide a given amount of service using the system. We summarize the model and the results we need as follows.

A time sharing system with N users can be thought of as a network of multi-server queues. Users are queued or in service at one node of the network, and are dispatched to the next node when they finish service according to fixed probabilities. One of these nodes (with one server for every user) represents the users' terminals, others might be the central processing facility (which might have several CPUs in it), or I/O devices of some kind.

Formally, let v_i be the number of visits at node i for every visit to the terminal node, $1/\mu_i$ be the service requirement at node i per visit, and m_i be the number of servers at node i. To include the infinite-server node representing terminals, let its index be 1, with $v_1 = 1$ and let $1/\mu_1$ be the user "think time." At each node, $\rho_i = v_i/\mu_i$ is the total service requirement. The sum $T(1) = \rho_2 + \ldots + \rho_n$ represents the mean total time a user who never has to queue spends in the system, and implicitly defines a unit of computer services. The mean time required to

deliver a unit of computer services when there is only one user (i.e., when $N = 1$) is therefore $S(1) = \rho_1 + \ldots + \rho_n = 1/\mu_1 + T(1)$.

Now, for any N we can calculate the thoughput $X(N)$ and the mean response time $T(N)$. The mean time to deliver one unit of computer services when there are N users in the system is $S(N) = T(N) + 1/\mu_1$ (i.e., the response time plus the think time). The cycle time $S(N)$ measures the average real time required to deliver one unit of computing services. The cost per unit delivered is thus $(w/3600)\, S(N)$. However, the number of units of service delivered in an hour is $3600\, X(N)$. However, Little's Result $N = X(N)S(N)$ for this system. Hence, user time costs are wN per hour when there are N users and the throughput is $X(N)$.

The Bottleneck Model [4],[11],[14],[15]

When N is small the cycle time is approximately $S(1)$ for each user, and throughput is approximately $X(N) = N/S(1)$. When N is large we may concentrate on the "bottleneck" node in the system, with a relative utilization ρ_b/m_b higher than that of any other node (ignoring possible ties). When N is large the rate at which user service is completed at the bottleneck node equals the number of busy servers m_b times the service rate for a server $1/\rho_b = \mu_b/v_b$. This rate, $m_b\mu_b/v_b$, must equal the rate at which users enter the bottleneck, which also must equal the throughput $X(N)$. Solving for the cycle time gives:

$$X(N) = N/S(N) = m_b\mu_b/v_b$$

or:

$$S(N) = Nv_b/(m_b\mu_b) \qquad (2)$$

That is, the cycle time increases with N for large N, and with the average service requirement v_b/μ_b. It decreases as more processing capacity is devoted to user jobs, that is as m_b increases. The throughput $X(N)$ is a constant determined by the capacity of the bottleneck node.

Now, when N is large $S(N) = Nv_b/(m_b\mu_b)$, while when N is small $S(N)$ is approximately $S(1) = \rho_1 + \ldots + \rho_n$. Equating these two expressions gives a critical value for N, N^*:

$$N^* = S(1)m_b\mu_b/v_b$$

$$= m_b \sum_{i=1}^{n} \rho_i/\rho_b \qquad (3)$$

When $N > N^*$ the system, in Kleinrock's terminology [11], is "saturated," and the cycle time grows in proportion to N. On the other hand when $N < N^*$ the cycle time is more or less constant, and its lower bound is $S(1)$. N^* can be thought of as the number of simultaneous users that can be accomodated without queueing if they were each given exactly $S(1)$ seconds of service.

The total costs of time sharing

The three cost components--fixed costs, variable costs and time costs--must be combined for decision making purposes. We call the first two cost elements together the private cost. The private cost is $F + cN$ when

there are N users on the system. F is the fixed cost (in dollars per hour, say) and c the unit variable cost (in dollars per user per hour).

There are N users on the system, each incurring a cost wN per hour. Hence, total time cost for all users is wN^2, and the total cost is:

$$C(N) = F + cN + wN \tag{4}$$

The units of $C(N)$ are dollars per unit time. For this value of N, the throughput $X(N)$ gives the number of units of service provided per unit time. However, conceptually a cost function is a function of output, written e.g., $C(X)$. Since $X(N)$ is an increasing function, we could in principle invert it and find this function. But there is no analytic expression known for $X(N)$, so this would not be a particularly insightful way to proceed. Instead we use the "bottleneck" approximation for $X(N)$ and $S(N)$, and then look at the resulting approximate cost function.

Costs in the Bottleneck Model

The general expression for the average cost per unit of computing service delivered is found by dividing Equation (4) by $X(N)$:

$$A(N) = F/X(N) + (c + w)S(N) \tag{5}$$

In terms of the bottleneck model, this is:

$$A(N) = \begin{cases} (F/N + c + w)S(1) & N \leq N^* \\ (F + (c + w)N)/(m_b\mu_b/v_b) & N \geq N^* \end{cases} \tag{6}$$

Evidently, Equation (6) has a minimum at N^* since it is decreasing with N when $N \leq N^*$ and increasing when $N \geq N^*$. Thus N^* measures the system's economically efficient operating point.

The explanation for this result is that when $N < N^*$ there is idle capacity at the bottleneck node. In the bottleneck model congestion does not begin increasing until $N = N^*$, so adding more users spreads the fixed costs over a larger number of users without increasing congestion costs. However, when $N > N^*$ adding another user adds to everyone else's collective delay, thereby increasing costs.[5]

If we could express $X(N)$ analytically, we could find an approximate value for the minimum by treating N as a continuous variable and differentiating. Thus, $A'(N) = 0$ would imply $X(N)/X'(N) - N = F/(c + w)$. Admittedly, computing $X(N)$ is not computationally difficult, so finding an exact minimum is fairly easy if the parameters of the queuing network model already have been determined. However, the bottleneck model is often used for rule-of-thumb calculations and the cost model used here is presented in a similar spirit.

An Example

To illustrate how well the bottleneck approximation works in a cost function, we extend an example used by Denning and Buzen [7]. This example has three devices (CPU, drum, and disk) with the queuing network parameters shown in Table 1. In this example the CPU is the bottleneck node, with $v_b/(m_b\mu_b) = 1$ second. The total time required to deliver a unit of service

is $T(1) = 2.2$ seconds, and the think time is 20 seconds. Hence $S(1) = 22.2$ seconds and $N^* = 22.2$ users. Figure 1 shows the value of $S(N)$ calculated by the bottleneck approximation (solid line), as well as the exact solution (dashed lines).

Inspection of Equation (3) shows that the true minimum of cost per unit of service depends only on the cost ratio $F/(c + w)$. Table 2 shows the location of the minimum cost point for values of this ratio between 5 and 50.[6] As can be seen, although the cost ratio varies by a factor of ten, the true minimum stays close to the approximation $N^* = 22.2$. This implies that the bottleneck approximation can be used to specify the system's capacity without causing large errors.

Also, the exactly computed cost curve is flat in the region of the minimum. This is also shown in Table 2, where the range of N for which unit costs are within 5% of the minimum is shown. This range includes $N^* = 22.2$ in all four cases. Figures 2 and 3 show the full cost curve for the first and last cases shown in the table. These figures also illustrate the fact that cost curve is flat near its minimum value.

Table 1

Queuing Network Parameters Used in Example

i	Node	Number of Servers, m_i	Visit Ratio, v_i	Mean Service Time, $1/\mu_i$ (sec)	Mean time required per cycle
1.	Terminals	N	1	20	-
2.	CPU	1	20	0.05	1.0
3.	Disk	1	11	0.08	0.88
4.	Drum	1	8	0.04	0.32

Table 2

Minimum Cost Values of N for Example

F/(c + w)	Minimum Cost N	Upper and Lower Values of N for Costs Within 5% of Minimum	
5	17	12	24
10	20	16	27
20	24	18	30
50	28	22	36

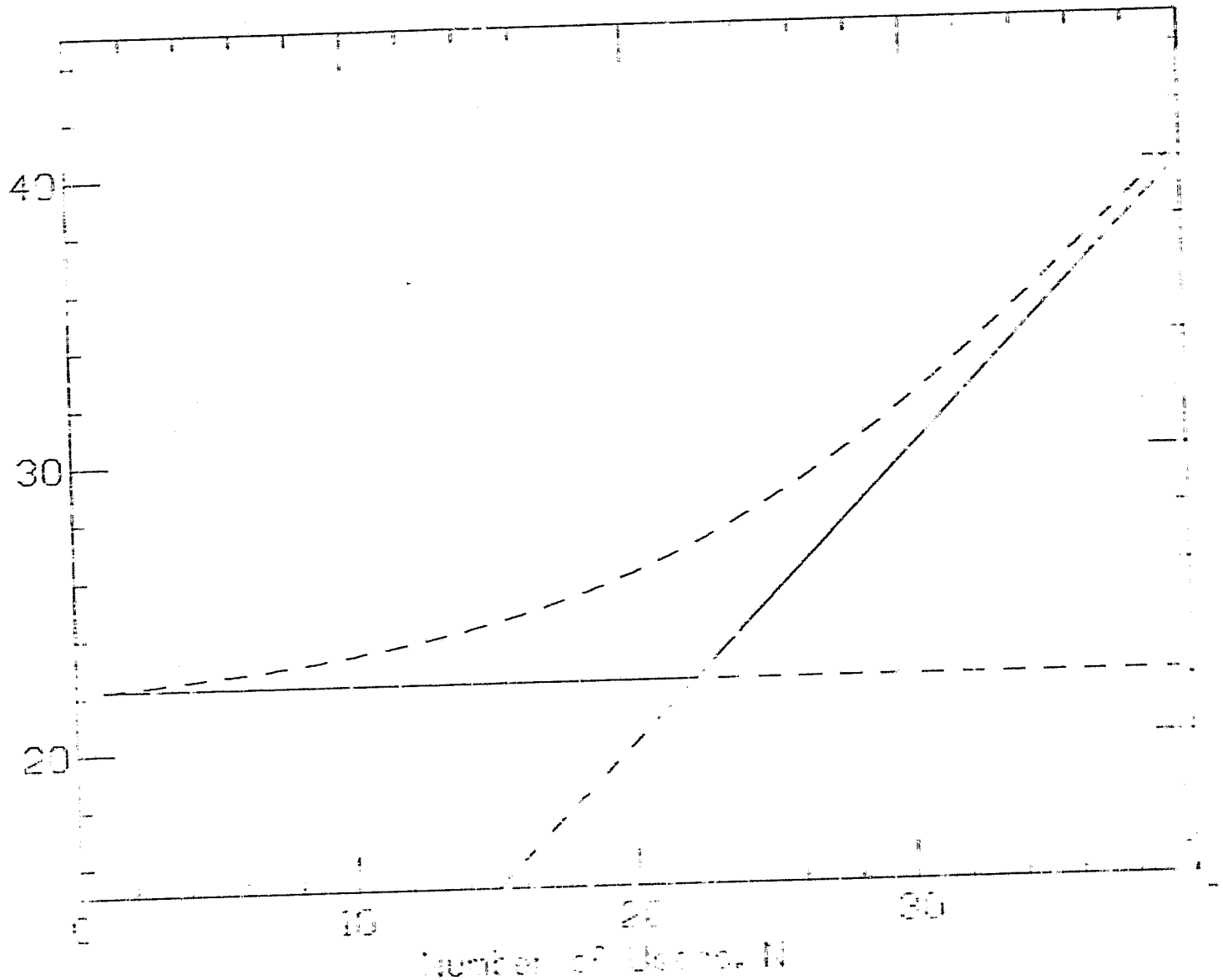


Figure 1 — Cycle time vs. number of users for example.

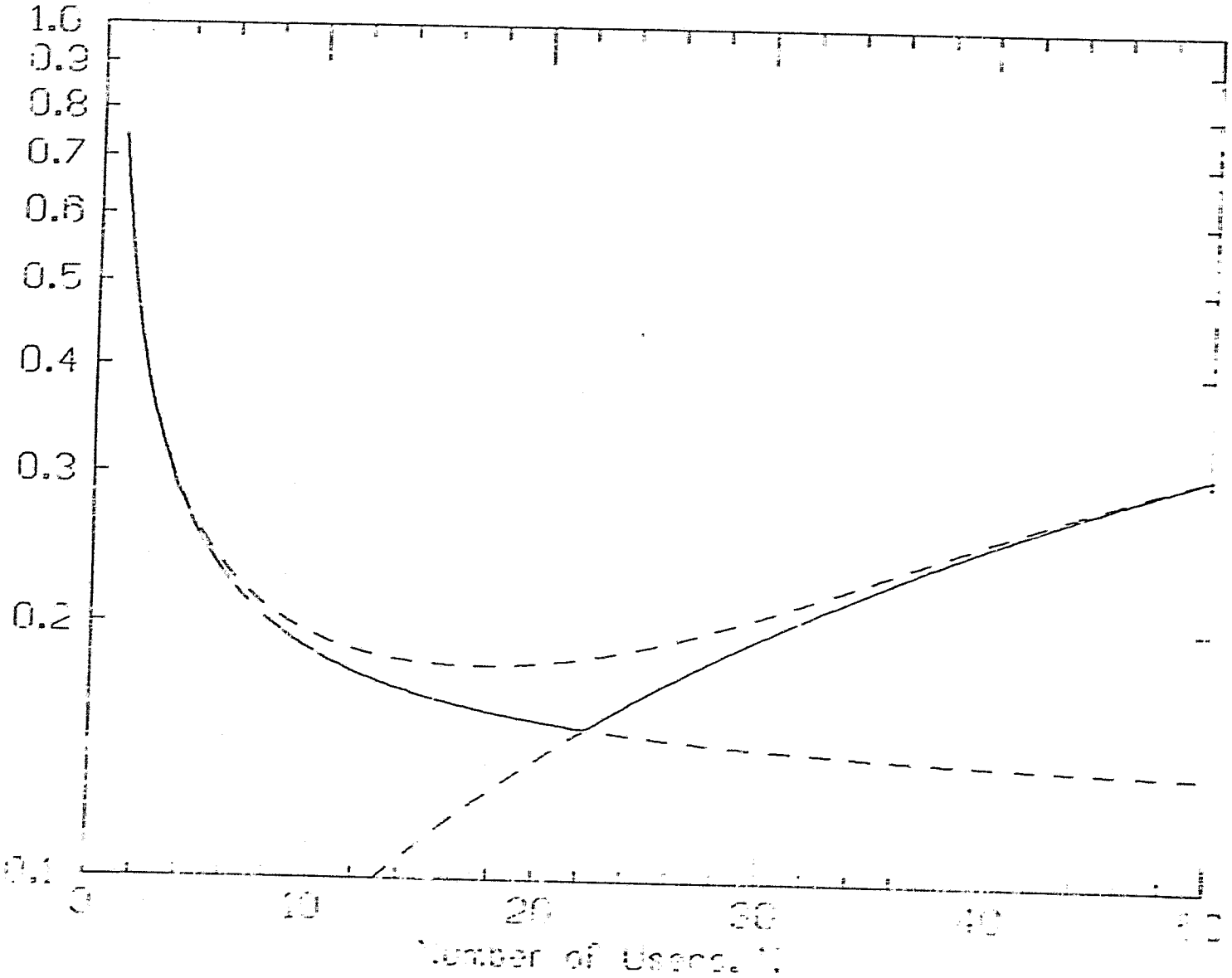


Figure 2 — Cost per unit of throughput versus number of users for example, $F/(c + w) = 5$.

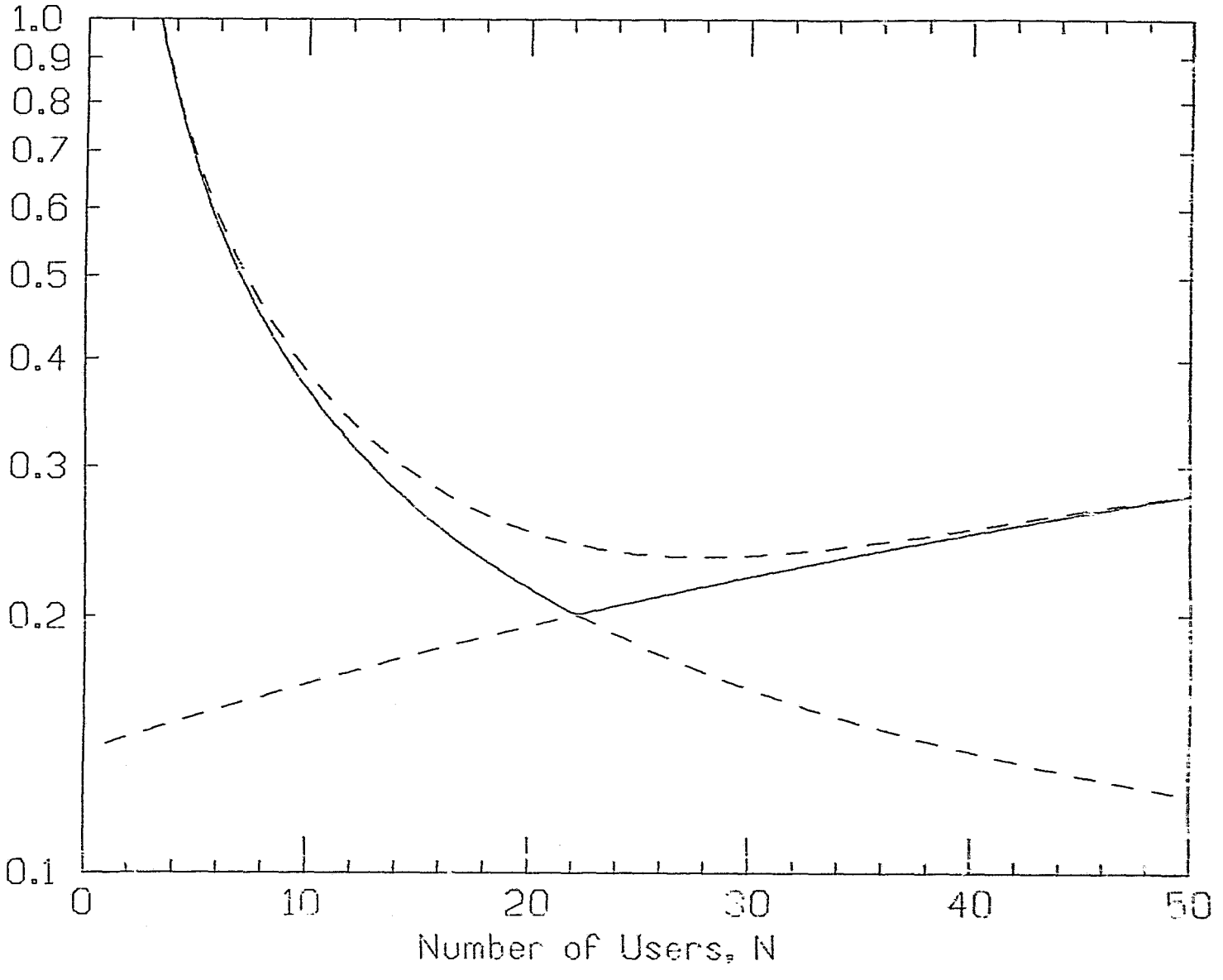


Figure 3 — Cost per unit of throughput versus number of users for example, $F/(c + w) = 50$.

Conclusion

This paper has discussed some of the technical-economic issues caused by congestion in time shared systems. The closed queueing network model has been used to show how to measure output and costs, and the bottleneck approximation derived from that model allows us to express the cost function simply. Using the approximation, we find that the minimum cost per unit of services delivered to users occurs at the "saturation" value N^*, which is the user load just sufficient fully to utilize the bottleneck server. Put differently, we have shown that it is economically efficient to saturate the bottleneck, compared either to under- or over-saturation. Intuitively, this is because in an under-saturated system ($N < N^*$) additional users cause little congestion but reduce the fixed costs per user. In an over-saturated system ($N > N^*$) additional users delay the other users, causing their costs to rise.

Based on the example, this result based on the bottleneck theory appears to be a good approximation to the exact queueing network solution. This is because the cost function is flat near its minimum. Thus, using the bottleneck value of capacity (N^*) should provide a quick check on for system designers on the economic efficiency of their system.

FOOTNOTES

* Partially supported by the National Science Foundation's grants GJ 36392X and IST-8108350, and the National Aeronautics and Space Administration's contract NASW 3204. This is a heavily revised version of "The Economics of Time Shared Computing: Congestion Costs and Economies of Scale," Report No. 12, Program in Information Technology and Telecommunications, Stanford Univerity, October 1974.

1. The special issue of Computing Surveys [9] contains several articles. See also [8] and [12].

2. Multiple job classes could be introduced, but their presence would not change the "bottleneck" results dealt with below except insofar as different classes have different bottlenecks. A multi-class bottleneck model (without congestion) is presented in Kriebel and Raviv [13] (see also [6]).

3. In some computing environments (e.g., academic) productivity may be harder to measure than in this example because it is more difficult to see what alternative use of time could have been made. Of course, this measurement problem does not invalidate the concept of user costs, and the idea that they need to be taken into account.

4. Time sharing systems have been studied analytically and emprically with this model for over a decade. In particular, the machine repair model, adapted by Scherr [15] and extended by Greenberger [10], Kleinrock [11], and Adiri and Avi-Itzhak [1] among others, has been used to describe the interaction between a central processor and a finite user

population. The general queuing network [8] has been used by Buzen [3], Baskett and Muntz [2], and Chang et al. [5] to model systems with several sources of congestion. See also the references cited in footnote 1.

5. Because one user by his actions imposes costs on others the costs are said to be external. The general term for phenomena such as congestion where these costs are imposed is "externality."

6. The values of $F/(c + w)$ were chosen to reflect the probable range of costs for the system used in the example. For example, if terminal and line charges are considered the primary variable costs, we might take $c = 3$ \$/hr. Users with a high value of time might have $w = 17$ \$/hr (i.e., $c + w = 20$ \$/hr). If the fixed costs are 100 \$/hr this would give $F/(c + w) = 5$. On the other hand, if valuable costs and user time costs were very low, e.g., $c + w = 5$ \$/hr, and fixed costs were 250 \$/hr, we would have $F/(c + w) = 50$. Intermediate values (e.g., $c + w = 10$ \$/hr and $F = 200$ \$/hr) also seem reasonable.

REFERENCES

1. Adiri, I., and Avi-Itzhak, B., "A time-sharing queue with a finite number of customers," J. Assoc. for Computing Machinery, Vol. 16, No. 2 (April 1969), pp. 315-323.

2. Baskett, F., and Muntz, R. R., "Queuing network models with different classes of customers," Proc. COMPCON 72 (Sept. 1972), pp. 205-209.

3. Buzen, J., "Analysis of bottlenecks using a queuing network model," Proc. ACM SIGOPS Workshop on System Performance Evaluation (April 1971), pp. 82-103.

4. Chang, A. and S. Lavenberg, " Work Rates in Closed Queueing Ntworks with General Independent Servers," Operations Research, Vol. 22, No. 4 (1974), pp. 838-847.

5. Chang, W., Paternot, Y. J., and Ray, J., "Throughput analysis of computer systems--multiprogramming vs. multiprocessing," Proc. ACM SIGOPS Workshop on System Performance Evaluation (April 1971), pp. 59-81.

6. Chismar, W. and C. H. Kriebel, "Comment on Modeling the Productivity of Computer Systems," Management Science, Vol. 28, No. 4 (April 1982), pp. 446-447.

7. Denning, P. J. and J. Buzen, "Operational Analysis of Queueing Network Models," in [9], pp. 225-262.

8. Gordon, W. J., and Newell, G. F., "Closed queuing systems with exponential servers," Operations Research, Vol. 15 (1967), pp. 254-265.

9. Graham, G. S. (ed), "ACM Computing Surveys: Special Issue on Queueing Network Models of Computer System Performance," Vol. 10, No. 3 (September 1978).

10. Greenberger, M., "The priority problem and computer time sharing," Management Science, Vol. 12, No. 11 (July 1966), pp. 888-906.

11. Kleinrock, L., "Certain analytic results for time-shared processors," Proc. 1968 IFIP Congress (Edinburgh, U.K.) pp. D119-D125.

12. Kleinrock, L., Queueing Systems, Vol. 2 (New York: John Wiley and Sons, 1976).

13. Kriebel, C. H. and A. Raviv, "An Economics Approach to Modeling the Productivity of Computer Systems," Management Science, Vol. 26, No. 3 (March 1980), pp. 297-311.

14. Muntz, R. R. and J. W. Wong, "Asymptotic Properties of Closed Queueing Network Models," Proceedings of the Eighth Princeton Conference on Information Sciences and Systems, 1974, Princeton University, pp. 348-352.

15. Scherr, A. L., An Analysis of Time-Shared Computer Systems, MIT Press, Cambridge, Mass. (1967).

Figure Captions

Figure 1 — Cycle time vs. number of users for example.

Figure 2 — Cost per unit of throughput versus number of users for example, $F/(c + w) = 5$.

Figure 3 — Cost per unit of throughput versus number of users for example, $F/(c + w) = 50$.