MANAGEMENT SCIENCE
Vol. 13, No. 6, February, 1967
Printed in U.S.A.

Time Sharing, Management, and Management Science*

G. J. Feeney†

This is going to be a talk: mostly about time sharing, a little bit about the implications for management and an even smaller bit about the implications for management science. Not that these implications are minor. On the contrary, I think time sharing is going to produce profound, irreversible changes in both the practice and the science of management. But instead of the usual inspirational harangue on the revolutionary future that time sharing will bring, let's take a close look at the technical and economic fundamentals which define the invention and the necessity that it answers. Once these fundamentals are clear, you can make your own forecasts.

Time sharing is the natural outgrowth of two technologies: computing and communications. Its *technical feasibility* is based in part on fairly recent and almost explosive progress in computer hardware—particularly in the cost/performance characteristics of large rapid access storage devices and of extremely fast central processors—and in part on longer term progress in the cost/performance characteristics of telephone lines and data transmission equipment. The need for time sharing grows out of fundamental inadequacies in present computing systems which we will discuss later.

Although time sharing effectively began through experimental projects at MIT, RAND, SDC, Dartmouth and a few other similar institutions, it is already well under way on a commercial basis. General Electric has time sharing centers operating as a regular commercial business in New York, Washington, Chicago, Detroit, Cleveland, Schenectady, Phoenix and Los Angeles. (Incidentally, let me emphasize that, except as a very active user for the last two years, my work is totally unrelated to the computer equipment business and the time sharing business in General Electric. This discussion is an expression of my own opinion and not necessarily of the policies or practices of General Electric or any other company.) IBM has, I believe, a number of Quicktran centers operating, and there are many individual centers such as Adams Associates in Boston, CEIR in Washington, Tymshare in San Francisco, Com-share in Detroit either already in operation or about to begin.

Things are happening fast. I would estimate that by the end of *next year* there may be at least 50 centers in operation serving perhaps 10,000 terminals and representing a going rate of expenditures of perhaps $40 million per year.

Why is this happening? Where is it headed? To answer these questions we must look at the economics of time sharing. For this purpose, let's assemble a hypothetical time sharing system and cost it out.

---

* A Talk Delivered at the International Meeting of The Institute of Management Sciences on September 6, 1966.

† General Electric Company, New York, New York.

In order to be entirely impartial, let's consider a hybrid system consisting of components similar to an RCA race file, a Burroughs disc file, an SDS 940 central processor and a GE DataNet 30 communications interface, 103A data set and model 33 teletype terminals supplied by the telephone company. (This seems like a representative cross-section of manufacturers but I may have left out one or two of the smaller companies.) Such a system would handle about 50 terminals simultaneously, serving a total population of say 200 terminals. The system would be on the air 20 hours a day, 7 days a week. Thus, the total capacity is 7000 terminal-hours per week. We will assume that the effective utilization is 2000 terminal-hours per week, anticipating fairly heavy usage during the usual working hours augmented by a significant additional second shift and weekend time. For reasons that will be clearer later, the system has built-in inducements to motivate off-hour usage.

On this basis we can now make a rough estimate of the cost of the system per terminal hour. Let me emphasize that these are very approximate numbers but I believe they are in the right neighborhood. First, the race file, with 250-million-character capacity, would provide permanent central storage for user programs and data, giving the average user more than a million characters of file capacity. This is equivalent to about 12,000 cards—a fairly adequate file base for many applications. On the basis of 2000 user-hours per week this storage would cost users about 75¢ per hour.

The second component, a 48-million-character disc file, would provide fast-access storage of the full file of each of the up to 50 users who are on the system at a given moment. It would also be directly accessible to the communication interface so that input from and output to the user terminals and user file maintenance could be handled without significant demand on the central processor. This file would cost about $1 per user hour. This raises the cost to $1.75 per user-hour.

Next, the central processor with, say, 64 thousand words of core storage and a 2 micro second cycle time, adds another $2.50 to the per-hour cost of the system. Lest this sound unbelievably cheap, remember that there are up to 50 users on the system so that—under very heavy demand the cycle time is effectively 100 micro seconds to each user. Of course, the effective speed goes up as the number of users goes down. Hence the attractiveness of second shift and weekend usage. Moreover, the core available to a given program would probably be restricted to say 16K to allow space for the executive program controlling the entire system which must reside in core all the time and to allow space for the previous active program and, possibly, for the next active program so that swap time—that is time required to move successive programs into core from the disc and back—could be shared with regular processing time as much as possible. Since each sub file the user creates within his total million-character file space can be used by his programs as if it were a tape, the 16K limitation is probably not too serious for many applications. The central processor brings the cost to $4.25 per hour.

The communication interface, consisting of a separate computer to handle the 50 telephone lines connecting active users to the system and of the 50 data sets which convert digital input into signals that can be carried on regular voice-grade phone lines, would cost about 75¢ per user-hour, raising the total to $5.00 per hour.

The central station would require a staff of about 30 people—including operators, system programmers, salesmen, etc. Their cost plus the associated facilities might be about \$3.50 per user-hour, bringing the total central station cost to \$8.50 per hour.

Finally, the line charges and the teletype terminals—assuming only *local* transmission and some sharing of teletypes, say two units for every three users—might come to \$1.50 per user-hour, bringing the total estimate to \$10.00 per hour.

While this is admittedly a very approximate analysis based on somewhat hypothetical components, I think it is a fair characterization of current time sharing economics. We have ignored many potential difficulties such as software development and component compatibility, which could increase the cost, but we also left out a number of factors which might reduce the cost, such as paging, drum swapping, direct communications processing by the central processor, scale economies in larger systems, batch processing in background mode, and the general prospect of improved performance per unit in future hardware.

Thus, in terms of technology that exists right now, if 200 individuals get together—or are brought together by an entrepreneur—they can have desk-side access to a system that will look to each of them like a 16K—50-100 micro second processor with a million-character supporting file system, a comprehensive compiler, enough software to get into and out of the system and maintain program and data files with relative ease, and a teletype keyboard and printer with paper tape attachment as the input-output device, all for about \$10 per hour. While no existing system quite comes up to these performance *and* cost specifications it is safe to say that such systems will be fairly common next year and will be very widely available the year after. And we can anticipate that they will provide increasing performance scope and decreasing unit cost as time passes.

Now let's look at the implications. For convenience, let's distinguish between the *actual* system—the hardware and software in the central station and the communication lines—and the *virtual* system—the apparent system of hardware and software available to the ordinary user. I will argue that even for conventional applications—or at least a large share of conventional applications—the virtual system with its limited capacity and relatively high cost per unit of computation still stacks up very competitively against batch processors *if* we consider *all* the aspects of cost and performance. Remember that not so long ago people were paying \$80 per hour to use an IBM 650 with less than one-tenth the speed and one-fourth the core capacity. The argument usually raised against the use of time sharing for conventional computing is simple and, at first glance, fairly conclusive. Our hypothetical system serves 200 customers and even if we assume a certain amount of second shift and weekend usage, there are really only about 200 hours in the month or about one total system hour per user per month. Therefore, the average user who pays \$10 per hour for 40 hours of virtual system usage per month is really paying \$400 per hour for the actual system. And, so the argument goes, the computation that can be performed by this system could be bought on the newer, large batch processors for \$200 per hour or even less. Thus, all this fancy time sharing does is double the cost of computing—or worse.

That's a pretty convincing argument and, in a sense, it is absolutely true. But this same argument would have proved that there was no hope for the private automobile since busses are not merely half as expensive per man-mile but more like one-tenth as expensive. A similar argument proves that business will never have any serious use for telephones since letters are so much more efficient in terms of cost per unit of information transmitted. In spite of these apparent disadvantages, cars and telephones have massive use. We will use time shared computing very extensively for the same basic reason: it is a much better solution to the *whole* problem. Conventional computing is very well suited to one class of problems: routine application of a production program with high volume on a regularly-scheduled basis. Linear Programming is a perfect example of this kind of application. Indeed, I sometimes think the most important problem that has been solved by Linear Programming is the problem of justifying large scale, limited access computers. But let's look at the rest of the computer user's problem including data acquisition, file maintenance, program development, program maintenance, and man-machine interaction.

Consider, in particular, management applications. In spite of years of high hopes and extravagant promises, conventional computers are still practically useless to managers. It's ironic that the individual responsible for programming a business gets so little direct help from an activity whose principal professionals are called programmers. But these systems are ill-equipped to deal with non-routine problems—speculative inquiry—which are fundamental to managerial analysis. Those of you who have tried to penetrate the tab-card curtain to attempt to get a non-routine job developed and run know from experience the following simple rule: if it takes 10 errors to get a program completely debugged and operative and if the average turn around cycle—including time to make changes and resubmit the job is one day,. *then* any answers you want in less than 2 weeks you'd better get with pencil and paper.

Time sharing is going to change all this. Instead of two weeks, the non-routine program will be running in two hours. We do it every day—and turn out work we couldn't produce on a conventional system at any price.

Under time sharing the computer is a personal tool available and responsive when *you* want it. The implications for management and management science are enormous. First, the computer becomes an inexpensive and extremely efficient *laboratory* in which to experiment with new ideas. Programs and data can be modified by simple entry from your terminal so that different approaches can be investigated quickly while previous steps are still fresh in your mind. If you think about this in connection with, say, the development of sales forecasting techniques, for example, you can see what the kind of gain produced by immediate, desk-side access would be.

Second, think of the implications in connection with implementation of research. Systems can be developed which really do give the manager an active, creative role in the operation of the process.

So far, we've discussed only more or less conventional applications in which users employ the system *individually*. But time sharing provides a communication network as well as a computation capability. This means that central data and program resources can be supported and used by many terminals and that terminals can interact directly with each other. The full implications of these capabilities are absolutely staggering.

Consider, for example, a sales forecasting system which includes hundreds of economic time series—automatically updated centrally—plus current expert forecasts of each series, together with a family of analytic programs which permit the user to develop his own approach to his particular problem. Included in the system would be an on-line documentation package which would conversationally show the user how to use the programs and how to interpret the results. Such a system is currently under development as one of the key products of a newly-organized information services company.

Similarly, a Wall Street house—White, Weld—has announced a time sharing system for financial analysis that will include complete financial records of hundreds of listed companies and an extensive set of programs for security analysis.

We can anticipate that there will be many more of these services in the future, since time sharing not only makes such services technically feasible but it also represents a completely new channel of distribution through which enterprising individuals can take data and programs to market and get paid for their efforts.

Looking further off, one can see that even more substantial changes are coming. For example, there are nearly a million industrial salesmen in this country who cost eight or ten billion dollars a year. Much of their time (some say all) is spent serving as a communicator between the engineering and production operations of the customer and the supplier. Imagine a time sharing system with terminals in customers' engineering, manufacturing, and purchasing offices which contains product specification data (updated centrally by a typewriter instead of a printing press), technical programs for engineering design calculations, availability and cost information and automatic order acceptance, delivery status notice and billing. This is just one example of the communicator task. If we look more deeply, probably half the total manpower in this country is primarily concerned with collating and communicating information. When the cost of time sharing gets down to $2 or $3 an hour, time shared computing will become the central technology of all administrative activities.