Computer development from a user's point of view

By C. R. Smith*

The intention of this paper is to discuss the directions which a mainly commercial user would like to see computer development take. It is written from the point of view of a user with a very large amount of commercial work obviously suitable for computers. This paper was first presented at the Joint Computer Conference held in Edinburgh in April 1964.

This paper is based on the following assumptions:

(*a*) That "large" computers will continue to be better value than "small" ones, and therefore that the best user policy is to employ the "biggest" computer that can be both fully loaded and efficiently managed.

(*b*) That the point of departure is possession of a computer with a core store cycle time of the order of two microseconds served by magnetic tapes with a speed of the order of 100,000 characters per second.

(*c*) That the criterion of a worth-while development is that it reduces the cost per unit of data processed.

The most urgently required development

This, in the writer's view, is something right outside the computer installation. If computers are to be widely used for commercial work a very great deal of data must be put into the machine, and it must be put there as accurately as possible. This data will in general arise as the product of the daily activity of a large number of human beings, each of whom, in computer terms, produces only a very small amount of data.

The most important development required is, therefore, a device for capture—in machine language—of the basic data at its source. Because of the multiplicity of sources and the restricted possibilities of sharing such data-capture machines between the human beings at work, such data-capture machines must therefore be *really* cheap: their prices must be measured in tens of pounds.

Now let us turn to the computer installation and the commercial user.

What the commercial user wants from a computer

All or most clerical processes are effectively overheads on the basic function of any business which will be selling a product or service. What the commercial user is looking for from the computer is the maximum reduction in his overheads either by direct reduction of the cost of clerical processes—for example, billing—or by "better management" or both. Therefore he wants computer installations to work at maximum efficiency to give the greatest possible return on his investment (doing as much of his work as possible at as low a cost as possible). A most informative way of looking at a computer is to regard it as a high-speed store surrounded by peripherals, some of which put data in, some of which take it out *and* some of which process the data. On this view it is possible to define maximum efficiency as being attained when every core cycle is utilized.

The commercial user and today's machines

I think it may be salutary to start by saying what a "large" computer can cost a commercial user. This, assuming double-shift working, can be in the range from 27/– a minute for the installation, unloaded but ready to go, up to 40/– a minute loaded with a job running. So, having got the data into the machine, the commercial user is obviously very keen indeed to get as much as he can of what he has paid for. This in terms, means to make his *effective* through-put as high as possible. There are two main factors in this:

(*a*) equipment reliability,

(*b*) operating efficiency.

Taking first equipment reliability, the commercial user is only interested in his computer while it is being used productively. The number of hours, therefore, for which his computer is in the hands of the engineering staff, either for routine maintenance or fault-finding, represents an overhead and is a reduction in the number of productive hours that can be achieved. The significant statistic of computer performance from this point of view is that defined by the BSI as "availability" and is given by the ratio

$$\frac{\text{serviceable time}}{\text{serviceable} + \text{fault} + \text{scheduled maintenance (times)}}$$

A recent return of this factor for 23 computers of British manufacture in use in Government service shows this ratio to be,

0·91 or better	4·3% cases
0·81 to 0·90	56·5% cases
0·71 to 0·80	17·4% cases
0·61 to 0·70	13·0% cases
0·60 or less	8·7% cases

It is of course true that some of these computers have been in service for four or five years, also that in some

* *G.P.O. Central Organisation and Methods Branch, Armour House, St. Martin's-le-Grand, London E.C.*1.

of them it may not matter if "fault" time is kept down at the cost of extended "maintenance" time during periods when the machine is not needed for work. But to say this is really to evade the point.

A recent article "Faults in Computers" in the *O & M Bulletin*, November, 1963 published by M. Stephenson of the Treasury Technical Support Unit reaches the conclusion (equipment wise)

> ". . . that whilst substantial improvements have already been made in the reliability of the electronic equipment there is much scope for further improvement in the electromechanical peripheral items."

In the same article the author draws a most interesting set of curves relating the probability of a successful work run to the ratio-length of run/mean time between faults on tape decks. For a ratio of 1/200 the expectation of a successful run is given as 0·98 with a 5 tape deck system, and 0·95 with a 10 tape system. This means that for a job duration of half an hour a mean time between faults per tape unit of at least 100 hours is required to give a probability of success not worse than one failure in fifty in the first case, and one failure in twenty jobs in the second (tape unit failures only). The author remarks "it is certain that at the present time some tape units have an average mtbf that is less than this figure."

If the mean time between faults is not long enough, then the length of the job must be reduced. To adopt a system approach of splitting a job into a number of (otherwise unnecessary) sequential processes in order to shorten run time and reduce the number of tape decks involved will tend to be a bad answer in terms of total processing time required and of the ratio of set-up to processing time, to say nothing of the increased scheduling difficulties produced. We are therefore left with the programming complications of short period re-start points in multi-file operations, something we should be well pleased to be without. It may be worth adding as some confirmation of Mr. Stephenson's argument that quite empirically we had already decided to adopt 15 minutes as the maximum length of run without re-start facilities.

It may not be out of place to emphasize that in a commercial installation the most important piece of equipment is the clock. Work must be done to a timetable, and the first practical consequence of unreliability is a loss of confidence by the computer staff in the equipment. The second is that the user will have to reduce his planned loading, perhaps to that of the lowest daily reliability level obtained in practice. His throughput goes down and his cost per unit of work goes up.

Now let us turn to the second factor—operating efficiency. I do not think that the real nature of a computer installation doing commercial work is widely enough realized amongst engineers and designers. It is not a machine installation; it is a man/machine installation in which the "man" component has, at present, a very significant effect on overall performance. Time lost by human failure can cost as much as £2 a minute. There is therefore an urgent need for a line of development which will progressively reduce the influence of the man in the man/machine complex. This has both an hardware and a software side. On the hardware side one might plead for a little work-study engineering to be applied to operating controls on tape decks, printers and consoles—we have seen some grisly specimens. One of the bogies of the faster machine is that the set-up time begins to assume a rapidly increasing importance because it stays put while processing time decreases. Magazine-loading tape decks—at of course the right kind of price—could make a significant contribution to many of the jobs we do in the Post Office. The software contributions must be aimed at the basic management problem in running a commercial computer installation —namely the organization of the day's work through the machine in the most efficient way. The commercial user would be glad to see the efficient self managing machine.

An equipment defect of today's (the 2 microsecond) computers

This basic fact is that for most commercial jobs a 2 microsecond computer is inefficient because it is out of balance, if an attempt is made to run it in a simple way with 90K tapes. The data transfer rates over much of our work needed to keep the central processor fully occupied in fact start somewhere about 90K and go up to 320K (the searching of low-activity files containing millions of records).

The current answer to this problem is "time sharing", "split file working", or "parallel processing" and it is in this context that the expression "in a simple way" was used in the previous paragraph. The writer's definition of a simple way would be not to have more than one input program (say, card to tape), one output program (say, tape to printer), and one main process (tape to tape) running simultaneously. Taking parallel processing of main tape to tape programs much beyond this point increases the management difficulties—of which complaint has already been made—considerably.

A much better answer would be to raise the maximum data transfer rate from magnetic tapes to something in the region of 300K. But in doing this the already stipulated requirement for reliability must not be ignored.

Consolidation operations required for 2 microsecond computers

Accepting the data capture requirement, although of prime importance, as being outside the computer installation, we are now in a position to re-cap on the developments required, and their order of priority, to make the 2 microsecond computer thoroughly acceptable to the commercial user. They are:

Priority 1—Development of existing machines for greater reliability (particularly applicable to peripherals—mean time between faults of several hundred hours).

Priority 2—Development of existing machines for greater self management, easier and simpler operation (software and/or hardware).

Priority 3—Development of cheap and reliable tape units which will enable existing 2 microsecond computers to be operated in balance on commercial work with no more parallel processing than one input, one output and one main program.

These requirements are, in the writer's view, necessary to enable the commercial user to get the best out of what he has paid for. And let no one who is not a commercial user of computers underrate what such a user pays out before he gets his job running on the machine, nor of the time and effort required to get his money back.

My five priorities so far may not seem very exciting to a technical audience, but I think the really significant thing is that in the year 1964 someone should have to make these points.

What does the user want for the next generation (post 2 microsecond machine)?

The reader may by now have formed the view that the first need of the commercial user is a consolidation operation effected on the 2 microsecond type of machine, and that he is mainly interested in computing as a way of making money. If so the reader is right. Development so far seems to have been a frantic affair mainly directed to putting gimmicks in the hands of salesmen. Little, if any, attention seems to have been paid to the overall balance of the installation or to the really effective exploitation of the potential already available in the machine.

If the user is given a choice between "larger", "faster" or "cheaper" (per unit processed) machines his choice will unhesitatingly be for "cheaper". He is not really interested in size or speed except in so far as these qualities lead to lower costs. What the engineer and designer so often overlook is that increases in speed and/or size introduce such proportionately greater difficulties into the management and operation of the (man/machine) system that in the end the larger, faster machine runs on a throughput so much below its potential that the cost per unit processed goes up.

The first priority for forward development is therefore quite clear, it is

Priority 4—(Priority No. 1 for new machine development). New machines embodying the points called for in priorities 1, 2 and 3 having access speeds in the 2 microsecond region, but which cost less than present machines.

Given this, the commercial user can apply the techniques of management and operation, which he will have established by practice, to his new machine in the confident expectation of seeing his cost per unit go down.

Does this attitude mean stagnation?

The writer would answer this question with a decided *no*. It has already been said that the first need of the moment is consolidation; giving—if one cares to put it that way—the user a chance to catch up with the technological progress already made. Let there be no doubt that the problems of the user are greater than those of the designer. The one has to produce something that will work, the other has to make it work to produce a profit. (An over-simplification of course but containing a basic truth.)

When that consolidation has taken place progress from then on ought to be a step function where each step ought to be fully consolidated before the next one is taken. The steps in this function are not necessarily those of technology, but of a resulting facility leading to a worth-while reduction of the cost per unit of work done. Probably the next worth-while facility step is,

Priority 5—(No. 2 for new machine development.) Development of random (immediate) access memories having very large capacities with (head of record) access time measured in microseconds.

The writer feels a certain diffidence in tabling a specific requirement here because of the probably much greater scale of government activities and because, by reason of the inadequacy of existing devices, his organization has not yet done a lot of work on the exploitation of random-access memories. Nevertheless, it is becoming quite clear that the availability of random-access memories of several thousand million bits with (head of record) access times measured in microseconds would make for a breakthrough in computing techniques.

We are beginning to move into the field of really integrated systems. One of our jobs under analysis at the moment has been broken down into fourteen systems, each of which is capable of an independent treatment, but all of which are capable of being keyed into the use of a common set of files. One would very much like to see all or most of this file information permanently on store in a random access file. It solves the interrogation problem, often resolved so far by producing otherwise unnecessary mountains of paper. It could also offer significant economies in the processing time required for large low-activity files, and of files where one piece of input affects two records in the file (the continental giro system, for example, requires one piece of input first to be debited to one account number and then to be credited to a different account number).

Whither next?

If engineers, designers and manufacturers have delivered to the commercial user machines which embody all the developments called for in Priorities 1–5, computers will probably have gone about as far as they can go against the background of existing commercial and computer organization.

It is perfectly possible even for someone not an engineer to visualize technical developments which will get data into computers, and when in, to handle it with much greater speed and greater reliability at costs which *providing the potential throughput of the machine can be realized* would be substantially reduced. But the qualification is important. It is very difficult to see the changes taking place in commercial organizations throughout the country which would allow sufficient work being gathered together at the very fast computer to make a worth-while run (we are already down to minutes for some jobs on a 2 microsecond machine). Unless this condition is realized the cost per unit of work done will probably rise.

The manufacturer's problem may well lie in the very wide range of daily throughput required by his customers. The big user wants the kind of development called for in this paper in order to bring his actual cost per unit of work processed down as much as possible for a large daily throughput. The small user would like to enjoy a unit cost of work done comparable with that of the big user, but for a very much smaller daily throughput. Is it possible to visualize a line of development which could meet both these needs from a common basic kit of parts? I do not know, but if both production and development costs can be spread over a greater number of installations, everyone is going to be very much better off. So even if it represents an unattainable millennium I would make my last priority

Priority 6—(No. 3 for new machine development.) Computers which in large throughput configurations have the lowest possible cost per unit of work done, and which yield approximately the same unit cost over a wide range of throughput configurations.

If this were ever substantially realized its most important effect would be to invalidate the first assumption on which this paper was written, namely, that "large" computers are better value than "small" ones. A computer utilization policy built on smaller machines would be very different from, and probably a great deal easier to implement, in a big organization than a similar policy based on the big machine. The advantages in the smaller organization are obvious.

A tail-piece on programming

It may have been remarked with a certain surprise that there has only been one oblique reference to programming in this paper—that made in connection with Priority 2. This has been mainly because program development parallels that of hardware and does not therefore fit into priorities laid down for equipment, but partly perhaps because the writer is profoundly cynical of many of the claims made in the programming field.

By way of definition it seems that the present position is that there are three levels of program language. The lowest is the "machine" language, then comes the "autocode" or "intercode" level, and lastly comes the so-called "clear" language (COBOL, FACT, CLEO, etc.). No one nowadays seems to write programs in "machine" language. The view held in my organization is that for a program of about the same processing efficiency the number of program instructions which have to be written at the second level is about 70% of those which used to be necessary at the machine level, and that this is the only concrete advance so far made in programming. What has not yet been effectively demonstrated (so far as I know) is the price in permanently increased processing time which has to be paid for further reduction in time secured by writing programs in "clear" language.

Having said that let me now say what I think the commercial user wants from programmers. His first need is that his "backbone" programs, run hour after hour, day after day, shall produce the minimum duration of processing time. A 1 per cent shortening in this kind of program can, over the installation's life, show enough saving to pay for an awful lot of programming effort.

His second need is an ability to make changes in established programs at very short notice, for example an alteration in a wages structure. He also needs to be able to write programs for "one off" or very infrequent (say yearly) jobs with the minimum of effort in a very short time. In this latter case the efficiency of the program (measured in running time) does not matter.

It looks, therefore, as though the commercial user will write his main programs at the second level, even perhaps in subsequent "Mark II" versions of the same program, dropping sometimes down to machine level to obtain maximum efficiency (shortest running time). He will need to use the "clear" language for temporary major modifications of these programs if required very quickly—later to be rewritten at lower levels for greater efficiency—or for infrequent programs to save programming effort.

In programming terms this seems to call for a "clear" language which compiles down to "autocode" level which in turn compiles down to machine level. Adequate print-out facilities to enable the programmer readily to find his way between levels are also needed.

Any system of programming languages which does not give these facilities would therefore call for a line of development directed towards realizing them.