BELL TELEPHONE LABORATORIES
INCORPORATED

SUBJECT: Throughput on the Dual Processor 645 System - Case 39199-14

DATE: September 23, 1966

FROM: D. L. Stone

MEMORANDUM FOR FILE

I. INTRODUCTION

The throughput of a computing system depends upon the relationships among three parameters:

1. The computing hardware.

2. The operating system (or software).

3. The type of service requested by the input load.

The interactions among these three parameters are so complex that local changes in one parameter often produce an unpredictable change in throughput. In estimating the throughput capacity of a given system, one must consider not only the efficiency of the hardware but also how well the software takes advantage of the hardware and most important, how well the hardware-software combination is adapted to the input load. This discussion compares two IBM 7094 model II's under BESYS-7 with a dual processor 645 system under MULTICS assuming an input load such as currently exists at the Holmdel computing center.

Definitions: For the purpose of this paper, the following definitions are in effect:

batching - dividing a sequence of jobs into groups (batches) which are offered to the operating system one at a time. The jobs are initiated as they are encountered in sequence. Unless multiprogramming is in effect, each job is completed before the initiation of the next job.

multi-programming - running a sequence of jobs in such a fashion that should a job have to wait for some event, another job will be run during the waiting period. The purpose is to utilize all available processor time.

<u>multi-processing</u> - using two or more processors to execute a single job in parallel. May be used with or without multi-programming.

<u>time-sharing</u> - using a computing system to service a number of jobs so that each job receives service within a specified time interval. Execution of the jobs is interleaved as in multi-programming except that real time constraints may force the interruption of a job in order to provide service to another which has waited too long.

II. HARDWARE

The basic configuration of the Holmdel 645 system will be:

2 central processing units (CPU)

2 Generalized Input-Output Controllers (GIOC)

1 Drum Controller

4 banks of 4 way interlaced core (64 K of 36 bit words each)

1 Drum (MSU-32)

8 Disk Modules (DSU-25)

20 Tapes

N Teletypes (M37 and IBM 1050, estimates place N from 40 to 250)

6 Printers (PRT-202)

Whereas, each 7094 II at Holmdel has:

1 CPU (IBM 7094 II)

4 Data Channels (2 7909 disc, 2 7607 tape)

2 Banks of core memory (16 K of 36 bit words each)

4 Disk modules (IBM 1301)

20 Tape frames (IBM 729 VI)

On the basis of these configurations, one may examine the relative efficiencies of the two hardware systems.

A. CPU Activity

The CPU's are basically similar in the 7094 II and the 645, although the floating point arithmetic is faster on the 7094 II. It should be noted that very little timing information has been released regarding the 645.

B. Core Memory Access Time

The memory cycle time on the 7094 II is basically 1.4 microseconds. On the 645 it is one microsecond. This simple comparison does not reflect the actual effective access time for either machine, since both machines are equipped with overlap circuitry which can reduce effective access time. Furthermore, a major difference between a pair of 7094's and one dual processor 645 system is that the 7094's have separate banks of memory. There can be no interference between the processors in accessing core. Even the data channels can compete only to the extent of stealing 2 percent of the CPU's memory cycles each (15,000 words/second). The 645 system, on the other hand, has five active devices competing for four interlaced banks of memory. The General Electric GIOC manual specifies a "judicious" worst case interval between processor accesses of 5.3 microseconds. Divided between two processors, this time becomes an average of approximately 10 microseconds per access for a given CPU.

Since one may safely hope that the "judicious" worst case will not be average, a simpler probability calculation is used to supply a result hopefully closer to the average. The calculations in Appendix A yield an average access time of approximately 2 microseconds. A further warning can be derived from the calculations. The more active devices which are appended to the 645 configuration, the more time it will take them to access memory.

Another factor affecting the access time on the 645 is the segmentation and paging mechanism. This memory addressing scheme requires up to three core accesses in order to locate a single word (the descriptor segment, the page table and the word itself). In order to alleviate the paging overhead, associative memory hardware has been added to the 645. General Electric modeling results* show that the effective

---

* A Proposal for Modeling Associative Memory Performance" - J. E. Shemer (General Electric) p. 14.

memory access on a single processor 645 system with paging and segmentation can be decreased from 2.76 memory cycles to .9 cycles by the addition of 12 associative memory cells. However, these figures are only valid for a single processor system. Two or more processors create a costly overhead, for each time a page is dispossessed of its core location by any processor, the associative memories in all processors must be entirely cleared, for they may contain references to the page which is no longer in core. Clearing the associative memory in another CPU requires the acceptance of a connect call by that CPU, since the CPU's do not operate synchronously. But connect calls are not recognized while the processor is in master mode - hence the average time to clear all associative memories is dependent upon the amount of master mode code in MULTICS, a figure which is not yet available. It is not clear to what extent this problem will influence the performance of the associative memories, but it will certainly raise the average effective access time.

C. Peripheral Devices

1. Tape

On the 7094 II, tapes are used for:

a. batch input

b. large data input-output (IO)

c. compiler scratch pads (FAP and FORTRAN)

d. external programs (IBSYS)

e. programmer scratch use

f. preparing EDP input (off line printing).

Tape IO is buffered and blocked. A data channel accessing memory for a tape will steal one in 47 memory cycles (at 15000 words/second). In addition, BESYS-7 takes a large percentage of the IO time for bookkeeping. Of the remaining time, all may be used by the programmer if his program can proceed asynchronously.

On the 645, the tapes will be less used due to the greatly expanded capabilities of the MULTICS File System. Common uses will be:

a. foreign data

b. bottom of storage hierarchy (file system)

c. incremental and other dumping

d. programmer use when detachable storage is required.

Blocking and buffering will normally be performed on MULTICS tape IO. The complexity of the IO system, while gaining efficient use of the hardware, will probably cost a larger percentage of the IO time. While running tape jobs with synchronous needs, the throughput of the 645 may be greater due to multiprogramming.

2. Disc

On the 7094 II, disc is used for:

a. operating system storage

b. FAP and FORTRAN compilers

c. programmer semi-permanent use

d. programmer scratch use.

BESYS-7 employs non-optimal strategies in operating the disc. Due to poor choice of logical record length for example, disc time can be 100 percent longer than necessary. This time waste could be recovered within BESYS-7 and will certainly be gained by the MULTICS IO system.

On the 645, the disc occupies second place (after the Drum) in the storage hierarchy. It will get much less frequent use for storing the system and compilers, but much more frequent use by programmers and especially by the file system. In particular, file system disc IO will probably eventually replace most card IO at Holmdel.

3. Cards

On the 7094, all significant card operations are performed off-line by 1460 card-to-tape and tape-to-card conversions.

On the 645, card IO will be done on-line. Input will come from new programs and large scale corrections, but will decrease as keypunching service begins to include teletype input directly to the file system. Output will be effectively nil except for foreign consumption and accounting purposes. This phase of IO will cost the 645 system some amount of time over the 7094 tape inputs.

4. Drum

The 7094 has no drum.

On the 645, the drum will be exceedingly useful for page swapping of the operating system, compilers and running programs. It is quite fast - a word can be transmitted in 2 to 5 microseconds - and would speed throughput if used in comparison to the 7094 using disc. The paging for which it is so useful costs time elsewhere.

5. Printed Output

On the 7094, all printed output is prepared off-line from tapes.

On the 645, the printers are on-line. This decreases the overall time required to print a page, but increases that share which the system must provide. Although the nominal cost of conversion to printable form is only 2 to 3 percent of a CPU's time per printer, total cost to keep a printer busy may exceed 10 percent of a CPU's time. This cost replaces tape IO on the 7094 and will certainly represent a loss in throughput capacity.

D. The GIOC vs. the Data Channel

The basic difference between the 645 GIOC and the IBM data channels is that the GIOC can multiplex input and output from many on-line peripheral devices, whereas the data channel can deal with only one peripheral unit at a time. The more bookkeeping done by the GIOC, the less must be done by the CPU's. The GIOC is one of the most powerful features of the 645 and should be responsible for a considerable increase in IO throughput over the 7094 data channels.

The rated throughput capacity of a GIOC is 3.5×10^6 characters/sec or 600,000 words per second compared to the 4 data channels of a 7094 installation which are limited by the speed of their attached peripheral devices to a maximum of about 15,000 words per second each.

III. THE SOFTWARE-BESYS-7 vs. MULTICS

A. Philosophy

The operating systems at Bell Laboratories have evolved over many years. Their continuity, enforced by the user population, guarantees that they are chained to system design considerations arrived at many years ago. As new features become desirable, new software is added to implement them. The random access disc is a prime example of a new hardware device which was patched into the operating system. It was only as experience was gained with the disc that efficient disc management techniques were discovered. From the viewpoint of device management and overall system compatibility, BESYS-7 can be improved upon.

MULTICS is (or will be) an operating system in which design has been the product of many man-years of labor - all done before a working version had to be installed. Changes will certainly be made after the system is installed, but the important point is that most parts of the system were designed in parallel, each influencing the others. This method of system creation enforces a compatibility and a cleanliness of design difficult to achieve with a serially designed system. The resultant system, it is hoped, will be freer of incompatible interfaces and many times simpler to debug and modify.

B. Purpose

BESYS-7 is designed to run a single 7094 system at the Bell Laboratories under batch processing. At Holmdel in particular, the emphasis is on first-in-first-out quick turn-around service. For this purpose, administratively imposed cpu time and output limits are imposed during the day. At Holmdel these limits are 15,000 lines of printed output and .15 hours of cpu time during prime shift. Thus, large demands for time and output are postponed in order to attain lower rates of turn-around.

MULTICS is designed to run the 645 system with a variable number of hardware devices. It will have the ability to cope with up to five cpu's, 512,000 words of core, two high-speed drums, 300 million words of disc, 32 tapes, 1000 typewriter consoles, etc.* Its basic purpose is to provide real time, multiprogrammed, multiprocessed computing to a wide variety of users with varying needs. In order to serve a diverse set of customers with some guarantee of reasonable turn-around and response times, administratively imposed limits on the use of system resources will undoubtedly exist. They may take a different form from those now in effect, but their purpose will be the same - to increase what is considered to be a desirable service at the cost of restricting the set of users who can be served. Since MULTICS

---

* MULTICS Repository Document B0008 p. 7; and others.

represents a considerable advance in computer software, it is not clear just what are the most desirable services it can offer - nor is it clear to whatset of users it can be restricted in order to attain a given level of that service. Hopefully, the system is sufficiently flexible that tuning of this sort will be relatively easy and will allow the systems programmers to adapt the system to the offered load.

IV. <u>THE INPUT LOAD</u>

At Holmdel, the two 7094 II systems are being worked hard. Currently, they are both operating at least fourteen hours per day and generating about 1.5 million printed lines per day. Although statistics about this load do exist (as to core, disc, tape and time used), there are two very important questions still to be answered:

1. How amenable to time sharing is the current Holmdel load?

2. In what way has the design of the programs at Holmdel been influenced by the computing service available rather than the problems to be solved?

Upon the answers to these questions hangs the nature of the input load which will be presented to the 645 system under MULTICS.

A. <u>Examples of Input Load</u>

To facilitate discussion of the input load, three examples are given below. Each represents a type of computing load which requires a specific mixture of hardware and software systems for most efficient computing. Deviations from that mixture would produce decreased throughput.

1. Consider a computer input load involving a reasonable amount of data and an unreasonable amount of computing time. Traveling salesmen routes, inverted matrices and detailed simulation results are examples of this type of load. Computer programs which obtain these results tend to run for a very long time, utilizing core memory access and cpu speed far more than Input-Output speed. If the input load consisted entirely of such programs, then it would be important in terms of efficiency to obtain hardware which performed calculations quickly and had adequate core memory,

and (batched) software which devoted the entire resources of the system to one program at a time. Since brute computing power would be essential, overhead should be reduced so as to maintain maximum use of the hardware. A multiprogrammed operating system to reduce delays caused by IO might be useful, provided that a sufficient amount of cpu time was being wasted in IO delay to pay for the overhead involved in multiprogramming. Otherwise, a simple batched system would be more efficient.

2. Another part of the computing spectrum is a job load generated entirely by people desirous of quick numerical answers to small problems. There the speed with which computing service is available is more important than the power of the service. Used as a glorified slide rule, a computer must be freely accessible in order to avoid man hours wasted waiting for service. Teletype input with low response time is required and therefore a time sharing software system must be used. The amount of useful computing done is reduced by the overhead necessary to service consoles with low response time. An average occupancy as low as 30 to 40 percent might be desirable to assure the necessary degree of service.

3. A third type of load is primarily one of input-output. Data is presented in one form; relatively trivial calculations are made; and the results are produced in another form. The transformation from time cards to Bell Laboratories payroll checks is an example. The limiting factor is the ability of the system to input and output data rather than to process it. For such a load, fast peripheral hardware is necessary. In addition, the hardware interface between the peripherals and the cpu must be efficient so that the software can utilize the peripherals to their full capacity. Finally, the software must be written so as to minimize IO delay. Computing speed is not important - large parts of the operating system can be devoted to running the IO gear efficiently. A straight batched or multiprogrammed batched system would be appropriate, depending upon the relative speed of the IO equipment and the cpu.

These three examples require distinctly different operating systems and hardware configurations to be served efficiently. The basic problem in designing an operating system for Holmdel is that our load combines the features of all of the examples and will be of sufficient quantity to tax our computing system for reasons of efficiency. The question then arises which part of the load is dominant and which part will be served best. It is clear that if loads of type 1. and 3. are not given administrative priority they will be seriously hampered by load 2. in a heavily loaded environment. In order to service consoles at all, a time-sharing system must be installed, with its large overhead. On the other hand, if a time-sharing system is installed which is forced to give fast service to large loads of types 1. and 3. by administrative fiat, then load 2. will suffer from long response time and a decreased number of active consoles or even periods of complete rejection while 1. and 3. are being serviced. Unless the quantity of hardware is so large that it never becomes overloaded, the system will not be able to cope with the mixed load.

B. The Holmdel Load

It is important to notice that this memorandum does not deal with the desirability of a time-sharing system in general nor even with the ultimate desirability of MULTICS at Holmdel. The sole purpose of the memorandum is to indicate the expected throughput performance of the 645 system under MULTICS at Holmdel assuming a load similar to the one presently being observed. Major changes in the input load, however caused, would require changes in the estimates which follow.

One short term change in load will be the influence of the new operating system on new or rewritten programs. Since most existing programs will have to be rewritten in order to run at all under MULTICS, it is logical to assume that at least some will be rewritten in such a way as to take advantage of new system capabilities. Although no jobs run interactively on the 7094's at Holmdel because the operating system is batched, some jobs would run more efficiently under a time-sharing system (file editing and program testing for example). It is therefore reasonable to expect eventually some increase in capacity due to the adaptation of the load to the system. That this increase will be large initially is very unlikely due to the existence of a sizable body of programs at Holmdel which demand brute computing power. These programs are antithetic to timesharing and may increase their running time under MULTICS. In this light, an overall increase in throughput of ten percent for job adaptation is a reasonable upper bound.

Another change in offered load will result from the elimination of wasted Input-Output waiting time. A study of the Holmdel load showed that approximately ten percent of the 7094's running time is spent in blocked status waiting for some kind of IO completion. This time will be gained by the MULTICS system due to its multiprogramming capabilities. However, the ability to multiprogram is only part of the real-time operations of which MULTICS is capable. The total cost in overhead will more than balance the ten percent gain.

These two characteristics of the Holmdel load are revealing in that they do not invite time sharing. The relatively small gains fail to counterbalance inefficiencies inherent in time-shared operation. For the current Holmdel load, most efficient operation would be obtained from a batched system similar in concept to the one presently employed.

V. ESTIMATE OF THROUGHPUT

The estimates offered below take into account the various facets of throughput which have been discussed. The particular categories chosen are treated as though they were independent in order to make the conclusions as accessible to analysis as possible. Each category is assigned a multiplicative factor representing the estimated effect which that category has on the throughput capacity of the 645 system. The present capabilities of the two 7094's under BESYS-7 at Holmdel are taken as the norm.

1. cpu: Basically the same. factor: 1.0.

2. core: 1 microsecond vs. 1.4 microseconds. factor: 1.4.

3. Input-output capabilities. The GIOC is capable of ten times the throughput, but there is not sufficient computing power to utilize this capacity for Holmdel load. factor: 1.2.

4. Efficiency in handling peripherals. Elegant MULTICS IO design is easier to use but more expensive. factor: 0.9.

5. Multiprogramming capabilities. The gain is less than 20 percent (pessimistic figures obtained at Holmdel yield IO wait time of less than 10 percent on the average). factor: 1.1.

6. Real time overhead. Includes effective core access and multiprogramming overhead. factor: 0.5.

7. Jobs which should be time-shared. Decreased number of runs. factor: 1.1.

Multiplying these factors together we have approximately .9 as the compound multiplicative factor representing the 645 system's throughput relative to the two 7094's at Holmdel.

Since this estimate of throughput is subject to errors of judgment, it is appropriate to specify the range in which the true answer lies. On the optimistic side, the relative throughput factor might be as high as 1.8 allowing for an error of 100 percent maximum. Pessimistic calculations might show a figure of .5. Using this range and load figures from the Holmdel computer department,* one can estimate the adequacy of the 645 system under MULTICS to bear the offered load. The 7094's are currently loaded at close to their capacity for two shifts; the load is expected to increase yearly at a rate between 25 percent and 45 percent. At 45 percent, the load will increase to 1.8 its current value in one and one-half years. At 25 percent it would take two and one-half years. Hence, even combining the maximum throughput estimate with the minimum load growth, the 645 system will be loaded to capacity in two and one-half years. Since the Holmdel 645 will not be installed before one year has elapsed and will not bear appreciable load for a year and one-half, consideration of adding more computing power seems advisable.

HO-3343-DLS-MO

D. L. STONE

Att.
As above

Copy to
Messrs. T. H. Crowley - MH
A. A. Currie - HO
Mrs. E. S. Hoover - HO
Messrs. W. Keister - HO
C. Y. Lee - HO
J. Poage - HO
V. Vyssotsky - MH
J. M. Wier - HO
E. Wolman - HO

---

* Computer Department Load Estimates, Memorandum for File, July 28, 1966, A. A. Currie.