H. G. Kolsky

POUGHKEEPSIE

POUGHKEEPSIE
Dept. 537 - Bldg. 703
February 6, 1958

MEMO TO: Mr. S. W. Dunwell

SUBJECT: Linear Programming Problems

On Wednesday, January 22, 1958, Mr. William Orchard-Hayes presented a lecture in the Research Auditorium on "Computer Requirements for Linear Programming Problems." Following a luncheon, Mr. Orchard-Hayes discussed such problems in detail with a group consisting of Drs. Brooks and Cocke from Project 7000; Drs. Chen, Hyman, Willoughby, Petrie, and Tuckerman from Research; Mr. Crumpelman from Military Products; Dr. Branin from Dept. 545, and myself. Our object, as suggested originally by Dr. C. C. Hurd, was to determine precisely what the Air Force linear programming requirements were, and to evaluate potential Stretch performance in terms of these requirements.

On the basis of Mr. Orchard-Hayes' presentation, and of admittedly hasty estimates which I have since made for these particular problems, it appears that the Stretch System, as defined by the Los Alamos Contract and by the Preliminary Manual of Operation of November 13, 1957, is at least two hundred times as fast as the 704 with respect to internal computing time, and that this ratio would probably maintain itself for total problem-solving time. A more precise estimate demands some actual simulation and more sophisticated consideration of the problem.

It should be emphasized that the Stretch System as defined here is relatively much more powerful on this type problem than it is on more conventional scientific or data processing computation. Although this System was designed to handle problems which are essentially large matrix problems, it was not designed with this particular application in mind. The major reasons for Stretch superiority, roughly in order of their importance, are as follows:

(1) long word length

(2) flexible, fast, complete parallel logic

(3) flexible and fast floating point arithmetic (particularly cumulative multiply operation)

(4) relatively large number of fast transistor registers

(5) flexible and fast indexing

(6) ease of performing double precision arithmetic

(7) look-ahead feature (virtual memory)

(1) By far the most important feature here is the 64-bit word length. This means that almost all linear programming work done on Stretch can be done with single precision operations. On the 704, almost all calculation is done in double precision, which means that arithmetic speeds are in the millisecond range. Stretch would thus do the actual arithmetic necessary to solve such a problem faster by a factor of more than 1000:1, but this factor of performance increase cannot, of course, be maintained by the other components of the system.

(2) The general problem here - treatment of collapsed matrices - is more logical than arithmetical. Stretch logic is flexible and fast enough to permit the use of more efficient methods which 704 logic will not accommodate.

(3) Total arithmetic time depends directly upon the brute speed of the floating point operations, and more indirectly upon their flexibility. The cumulative multiply operation is particularly valuable because it avoids the use of double precision operations whenever possible.

(4) Since the assumed Stretch memories are only a few times as fast as 704 memories, it is obvious that the arithmetical-logical structure must carry most of the load in effecting performance improvement by a factor of 200. Any high-order improvement attributable to memory performance must be achieved by the use of fast transistor registers; such improvement will depend almost directly upon the speed of such registers, and in a much more complex way, upon their number. A quick estimate shows that a set of at least eight registers with speeds of from .1us to .2 us is a minimal requirement for this problem.

(5) The indexing used to get from one non-zero element of a sparse matrix to the next one needs to be both fast and flexible, and full advantage must be taken of the ability to do much of the indexing in parallel with other operations.

(6) Although double precision arithmetic is necessary on the Stretch System much less often than on the 704, it will still be required for some linear programming work. When this happens, the Stretch System is much better designed for double precision operations than is the 704.

(7) The contribution of the virtual memory is considerable, but appears to be less for this particular application than for most scientific applications; however, the effect of virtual memory demands a much more detailed study, and is probably more important than some of the other features listed.

Tentative Conclusions.

1. For the solution of the type linear programming problems discussed by Mr. Orchard-Hayes, the 701 was a very poorly designed machine; the 704 was somewhat better, largely because it had built-in floating point and indexing operations - two steps in the right direction; and the 709 has almost marginal superiority over the 704. The greatest single improvement provided by the Stretch System comes solely from the use of longer word length with parallel arithmetic operations, although a very respectable further improvement comes from the increased speed and sophistication of the system, as indicated above.

2. The Stretch System itself is well-designed for these problems only in comparison with the 704. By design modifications involving a moderate amount of hardware, the performance of the Stretch System could be improved by a further factor of from 3 to 5. Such additions would be mostly useless in the scientific problems for which Stretch was designed; similarly, a great deal of Stretch hardware is useless so far as linear programming problems are concerned. Stretch parallelism (or multiplexing) is not particularly effective on these problems, and individual unit efficiency is low.

3. It would be possible, using Stretch circuits and philosophy, to build a machine which would be perhaps 20 times as fast as Stretch for linear programming problems. The cost of such a machine would be staggering, and construction would not be realistic using present components. However, developments in areas such as cryogenics may make the construction of such a machine feasible.

4. In this particular application, the unmodified Stretch System as assumed here would have relatively high superiority over its possible competitors, such as LARC.

5. There is no point in designing a computer specifically for linear programming problems, since such a machine would require little additional hardware to make it a very powerful general prupose computer; on the other hand, a very powerful general purpose computer such as Stretch could be made several times as efficient for linear programming problems at a moderate increase in cost (the look-ahead concept would have to be extended, and a considerable amount of extra hardware (control, logic, and arithmetic) would have to be added under the direct supervision of look-ahead).

On the basis of these considerations, I feel we can conclude that the Stretch System, as defined here, is a powerful machine for the solution of linear programming problems of the sort described by Orchard-Hayes, that it is indeed the only machine now under development this side of the iron curtain which is capable of solving such problems at all, and that the improvement attainable by modifying the hardware would not make good economic sense unless the computer were expected to devote a considerable part (perhaps 10%) of its expected life to the solution of such problems. (To determine precisely where the break-even point would be is, of course, a linear programming problem!)

S. G. Campbell

SGC/img

cc: Messrs. F. P. Brooks
W. Buchholz
J. Cocke
A. B. Credle
H. Kolsky
G. Petrie
D. W. Sweeney