adams associates

AN APPRAISAL

of

THE STRETCH SYSTEM

Submitted to

INTERNATIONAL BUSINESS MACHINES CORPORATION
112 East Post Road
White Plains, New York

by

CHARLES W. ADAMS ASSOCIATES, INC.
142 the Great Road
Bedford, Massachusetts

May 31, 1961

adams associates

Abstract

The performance of a data processing system can only be properly evaluated with reference to specific areas of application. In many of the problems for which it was intended the Stretch system outperforms the 704 by a factor of about 35 to one, while in some arithmetical problems the factor rises to perhaps 50 and in certain logical problems drops to about five. In a few problems which urgently require its large storage, long word-length and built-in checking, Stretch more than meets its design objective of outperforming the 704 by more than a hundred to one.

Most of the shortcomings in the logical design of Stretch can be traced to poor communication between the planners and the designers and to setting quite specific goals in contracts and public announcements for what was really a "best effort" development project. The planners should have stayed with the project longer. The trial programming and simulation efforts should have been started sooner, carried on longer, and taken more seriously. Had it not been for the publicity and the competition provided by the 7090 (itself based on Stretch technology), Stretch might well have received unqualified acclaim.

adams associates

Contents

Introduction

The object of this report is to discuss general criteria for evaluating computer systems, apply these criteria to the IBM Stretch system, and comment on the objectives set and the decisions made in the development of that system. In the preparation of the report, many of the people involved in Project Stretch were interviewed and a large number of written and verbal reports were reviewed. Every effort has been made to evaluate fairly the Stretch system and to correlate its ultimate performance with the design decisions on which it is based.

General Criteria for Systems Evaluation

The ultimate criterion in the evaluation of a data processing system is the overall cost entailed in performing the work it is to do. The principal elements in the cost are:

1. Preparation of the program initially.
2. Preparation of data for processing.
3. Machine running time.
4. Machine down time (maintenance, repair, rerun time).
5. Machine rental per unit time.

Of these five major elements of cost, all but the rental of the machine are dependent not only on the machine design but also on both the nature of the problems to be handled and the skill of the programmers who prepare the problems for running. Consequently it is possible to evaluate a data processing system accurately only with reference to the type of problems for which it is to be used by a particular customer.

An approximate evaluation may be obtained by neglecting all factors except the machine running time and the rental. This may be justified on the grounds that the cost of preparing programs is dependent more on the available software than on machine design and that data preparation and down time do not usually differ substantially from one machine to another. By establishing as a standard of measurement some reasonable mixture of typical problems, it becomes possible to evaluate an "average" machine running time and thereby obtain a figure which is useful as a first approximation to the overall performance of a given system. In comparing two systems which are logically similar, such a rough "average" figure of merit will be much more meaningful than it will in comparing systems which differ greatly in their logical structures and in which the figure therefore varies greatly from one problem to another.

Evaluation of Stretch

In light of the remarks of the previous section, no single quantitative measure of performance has been sought. The effort has been rather to establish a qualitative evaluation by considering many factors.

While the most important single criterion is speed, no independent tests of relative speed were run during this study. For one thing, there appears to be no reason to doubt the figures obtained by IBM and Los Alamos personnel. Then, too, the uncertainties as to what problem mix should be considered greatly exceed the uncertainty in the timing of individual runs. One of the most interesting sets of results obtained by others show the following ratios in comparison to the 704:

matrix multiplication	33
Laplace equation	32
Bengt Carlson loop	37
polynomial, many terms	52
part of weather prediction	35

The "average" might be said to be about 35 times the 704 speed. The uncertainty in this figure is clearly shown by the fact that in the evaluation of a polynomial (and other problems involving a large proportion of floating point arithmetic) Stretch shows performance 50 times that of the 704, while on problems which require constant use of branching and do not involve any appreciable amount of arithmetic the figure is in the vicinity of five. (Stretch is actually slower than the 7090 in such logical problems as binary search).

In addition to the question of speed, Stretch offers numerous impressive features which are difficult to evaluate quantitatively. For example:

<u>the large capacity and high speed</u> permit the handling of problems which could not otherwise be solved on

any existing computer -- such problems as large hydrodynamical problems and extensive simulations which would require months to complete on a 7090, and large data handling jobs like weather forecasting where a time deadline must be met. The number of problems which cannot be handled on the 7090 is not substantial at present but this is partly due to the fact that until Stretch became available there was little point for anyone in studying problems which would fall into this class.

the elaborate built-in checking system eliminates the need for much of the program checking which would otherwise be required and minimizes the amount of time likely to be required in rerunning in the event of a machine failure. To some users (apparently to Carlson at Los Alamos, for example) this could be equivalent to having the computer be twice as fast as it is.

the long word-length permits more accurate results to be obtained than are normally obtained on the 7090 unless double precision is employed. Since double precision slows the 7090 down substantially, this too could easily be worth a factor of two or more in some cases. In most other cases the results obtained from the Stretch would have an intangible added value in being more precise than those from the 7090 (although this extra precision may not always be of value).

the extensive instruction repertoire and other logical features of Stretch permit a very substantial reduction in the amount of memory required for many kinds of programs. This reduction in the number of instructions is also one of the means by which the effective speed of the machine was increased. On the other hand, the attempt to make the system easy to use has backfired to some extent: the speed of some of the VFL instructions is so low that their use is to be avoided; the index and floating arithmetic must be carefully interspersed or speed is lost; and the conventional (e.g., 7090) situation in which branch and index instructions are very fast has been completely reversed in Stretch so that new coding techniques are required.

On balance, Stretch certainly does not outperform the 704 by a factor of 100 as originally predicted. The factor is, however, at least 30 and in some cases will be as high as 50 or 60. Stretch consequently represents a giant step from the technology of the time in which it was originally proposed.

The 7090, making use of all of the circuit and core memory technology of the Stretch together with the logic of the 704, has provided such a strong competitor that a reduction in the price of the Stretch system has been made in an effort to keep the two systems properly competitive. It seems likely, however, that the price reduction actually established was unnecessarily great and that Stretch at its new price is a considerable bargain for many types of problems.

Objectives of Project Stretch

The objectives of Project Stretch were broad and ambitious. That they were very nearly reached is indeed an impressive achievement. That the final product of the effort has been the subject of much criticism results from an ambivalence of purpose: despite the specific and detailed nature of the Los Alamos and Bureau of Ships contracts, Stretch had many of the aspects of a best effort rather than a product development project.

Not only were the two contracts quite dissimilar in their requirements, but a commercial system was also part of the goal. There was, in fact, a conscious effort to produce a jack-of-all-trades, without sufficient realization that it could well end up a master of none.

The statement of purpose and the justification offered by F. P. Brooks, Jr., in Chapter 2 of the superb chronicle of the Stretch project compiled by W. Buchholz (to be published under the title Planning a Computer System) is revealing.

> "In addition to being fast, the Stretch computer was to be truly a general purpose computer, readily applicable to scientific computing, business data processing, and various large information processing tasks encountered by the military. In 1955-56, when the general objectives of Project Stretch were set, it was apparent that there existed a few applications for a very high performance computer in each of these areas. There is no question but that the computer could have been made at least twice as fast, with perhaps no more hardware, were it specialized for the performance of one or a very few specific computing algerithms. This possibility was rejected in favor of a general purpose computer for four reasons, each of which would have sufficed:
>
> 1. No prospective user had all of his work confined to so few programs, nor could any user be sure that his needs would not change significantly during the life of the machine."

While this statement is incontestable, one can seriously question the implication that a powerful general purpose system aimed at a specific class of problems would not fill a need.

> "2. If a computer were designed to perform well on the entire class of problems encountered by any one user, the shift in balance required to make it readily applicable to other users would be quite small."

Even disregarding the fact that much of the difficulty is wrapped up in the big "if," the truth of this is hardly self-evident.

> "3. Since there existed only a few applications in each specialized area, and since the development costs of a computer of very high performance are several times the fabrication costs, each user could in fact acquire a general purpose computer, containing hardware he did not especially need, more cheaply than he could acquire a machine more precisely specialized for his needs."

In most cases very highly specialized systems are uneconomical as stated, but extending the reasoning to justify an attempt at an <u>all-purpose</u> machine represents, in hindsight, a serious underestimate both of the market for more specialized large systems and of the cost of including powerful features aimed at radically different areas of application.

> "4. Since there are real limitations on the skilled manpower and other facilities available for development efforts, it would not have been possible to develop several substantially different machines of this performance class at once, whereas it was possible to meet a variety of needs for very high performance computers with a single machine."

Stretch (and expecially Harvest)* is little more than a melding of several substantially different machines into one, which would seem on the surface to be even more difficult than designing several different systems for more restricted purposes.

Comments on the Design of Stretch

Stretch was intended to achieve its power through faster circuitry, faster logical implementation of the built-in arithmetic and logical operations, multiplexing on a micro level through the exchange and look-ahead features, multiplexing on a macro level through the interrupt feature, and the introduction of new instructions to reduce the number of steps required in the solution of common problems. In addition, elaborate error detecting and error correcting techniques are used, primarily to facilitate maintenance, and the logic of the instruction repertoire, interrupt facility and look-ahead scheme are all required to be foolproof in use (the programmer-user is not required to adhere to any rules of program syntax merely for convenience of the machine designers). Each of these aspects of design deserves individual comment here.

Circuits and components: the authors of this report are not competent to evaluate critically the design of the circuits nor the magnetic core memories in Stretch. Clearly, if the circuits had been faster, the computer would have been faster and the original objectives might very well have been met. That they were not faster apparently reflects over-optimism in the original specifications for Stretch set at a time when transistors were still a laboratory curiosity. Evidently twenty-nanosecond circuitry was a real achievement; ten or five was asking for too much all at once. Similarly, two-microsecond memories were an impressive development; and the slow-down in circuits and logic obviated the need for a half-microsecond memory so that difficulties in that area are not highlighted.

Fast arithmetic and logic: the speed of the floating point arithmetic unit is quite impressive in view of the clock interval which is now 0.3 microseconds rather than the intended 0.1 microseconds. The original specifications for division were set much too high, and the final cost of the multiplication operation is almost absurdly high; nonetheless the logical design does much to keep Stretch within near reach of its goals.

The variable-field-length arithmetic, on the other hand, is startlingly slow by contrast - so much so that the value of having the VFL operations at all is somewhat questionable (as discussed below in reference to the "foolproof" design).

The Branch-on-Bit instruction is perhaps the most notorious of all. Intended as an ultra-fast, versatile logical operation, it was mechanized as a VFL instruction requiring four data storage accesses (where usually one and at most two are required), uses 0.6 microsecond VFL data transfers, and gets badly entangled with the look-ahead facility, resulting in the ludicrous average execution time of nearly 15 microseconds if the branch occurs. So little consideration was given to the speed of this important instruction that the time it requires was not appreciated by anyone until January 1961 when the computer was actually in operation and speed tests were made. So little analysis of timing was made that it was not until May 18, 1961, that anyone worked out where all the time goes in the performance of this operation.

<u>Exchange and look-ahead</u>: the basic concepts of the exchange and of the look-ahead unit both represent major developments in computer technology and contribute mightily to the effectiveness of Stretch. Unfortunately, practical difficulties were encountered

in the design of the look-ahead unit - due in part to conflict with the interrupt feature as described below and in part to inadequate handling of store instructions - which limit its effectiveness; and the unit is perhaps more elaborate and expensive than it needs to be.

Interrupt feature: the interrupt feature serves two distinct purposes -- it facilitates macro-level multiplexing and it provides a means for dealing with such exceptions as arithmetic overflow and such malfunctions as parity or arithmetic errors. For the second purpose, the designers insisted on instantaneous interruption, which requires recovering from whatever has already happened in the look-ahead unit, even though this is not a particularly important feature and is not even required in providing for concurrent operation. The tremendous cost of this instantaneous interruption in view of its conflict with the look-ahead logic was not recognized at the outset and was not acted upon at the later stages of the design.

Instruction repertoire: the logical power of the instructions provided in Stretch are a pleasure to behold, until one learns the details of the operating times. The three-part index words appreciably reduce the number of instructions required in some loops, but unfortunately were so complex to implement that they run rather slowly and may, on balance, be less efficient than would a simpler but faster indexing structure. Provision for addresses of various lengths, for variable byte-sizes, and for full-word instructions straddling a word boundary are pleasing features to the user, until the price in speed and cost is examined. The difficulty all seems to come from setting specifications without regard for speed and cost, then failing to modify these specifications when the effect on speed and cost became known.

Checking: with hindsight, it seems likely that built-in checking and error-correcting facilities were overdone; the value is not worth the cost. On the other hand, the decision was probably a sound one when it was made since there was little assurance that so large a system could be made to run at all without it. And the checking does reduce machine time used in programmed checking and lost in maintenance, perhaps increasing machine effectiveness by as much as a factor of two in some cases.

Foolproof sophistication: the combination of variable field length arithmetic and a powerful instruction repertoire with a requirement that the instruction set be complete, symmetrical and not burdened with special rules for use dictates excessive complexity which results in deterioration in terms of speed and cost. This in turn defeats the purpose. Far from avoiding a burden on the user, it necessitates use of a complex manual (still in preparation) to show how to mix various instruction types for optimum speed. The VFL instructions are intended to simplify data handling and logical operations such as arise in writing compilers as well as in business-type problems. Yet Dr. Campbell's group is finding it necessary to use floating-point instructions to do fixed-point work because the VFL operations are too slow -- a crowning irony. The basic problem is that expert programmers do not really need elaborate nor foolproof logic, while novices will not make effective use of it.

Conclusions

The Stretch system comes reasonably close to meeting its goals, missing by perhaps a factor of three when problems well suited for the 704 or 7090 are considered, but doing much better on problems which require its large storage, long word-length and built-in checking.

The two general problems - lack of communication between the representatives of the various aspects of design, and the lack of a specific goal of which all were aware - were certainly the root of whatever difficulty was encountered in Project Stretch. Equally certainly, the specifications originally given were extremely optimistic. Even though a development effort will seldom reach great heights without setting distant goals, it was inadvisable to announce design objectives based on conjectures of new technological achievements as confidently as was done in this case.

Many of the difficulties commented on in the preceding section would have been avoided if the planners had stayed with the design until the detailed logical design was complete. Much more sample programming should have been done early in the game to prove the desirability of some of the novel features of the specifications. The limited simulation work carried out in 1958 should have started earlier and been carried on longer. The simulation results obtained should have had more effect on the design than they did.

It seems evident to the authors that the general problems encountered in the design of Stretch must have been encountered in the design of most other new systems by IBM (as well as by its competitors) and have perhaps been no better solved. Very probably the improvements that could have been achieved through better communications and better goal-setting in the design of systems such as the 7070 and even the 650 might have been just as spectacular as those which could have been achieved in the case of the Stretch. In the absence of the advance publicity and the 7090 competition, the Stretch might well have received unqualified acclaim.