Met O 11 Technical Note No 152

Optimizing numerical weather forecasting models for the CRAY-1 and CYBER 205 Computers

by

A. Dickinson

Meteorological Office
Forecasting Research Branch
London Road
Bracknell
Berkshire
United Kingdom

November 1981

F42A.

ABSTRACT

The Meteorological Office has recently acquired a CYBER 205 computer system which will be used to run large numerical models of the atmosphere and weather forecasting models. In particular a new operational forecast model has been developed for this computer and this is being coded to take full advantage of the vector capabilities of this machine. Development versions of this model have also been run on a CRAY-1 at the European Centre for Medium Range Weather Forecasts. This has presented the opportunity to compare some of the features of these two computers.

It is shown that in general numerical weather prediction models can be split into two sections, the dynamical and physical processes. In the dynamical processes the same operations are applied at every point in the integration domain and so these are readily vectorizable. However, the physical processes are conditional and intermittent in nature and use must be made of the mask/merge compress/expand and gather/scatter class of instructions in order that these may be vectorized.

1. Introduction

The operational work and scientific research carried out by the Meteorological Office has for many years been highly dependent on the use of numerical computer models to simulate the motion of the earth's atmosphere. These models which are used primarily for weather forecasting and general circulation studies have increased in size and complexity as the speed and memory capacity of computers have increased. Finer horizontal resolutions and more levels have been introduced along with a more detailed description of the physical processes in the atmosphere. Nevertheless, there are still many important applications of weather forecasting and general circulation modelling which are not possible on the current generation of serial computers and only now with the advent of the most powerful of the vector and parallel processors can these projects be attempted.

Perhaps the most important of these applications is research into possible techniques for forecasting fluctuations in the earth's climate using general circulation models. Recently, through the setting up of the World Climate Program, many governments throughout the world have shown a greater awareness of the possible effects of climatic fluctuations on the supply and demand of vital resources, particularly food and energy. The number of days supply of food reserves, for example, has dropped sharply, partly in response to severe fluctuations in weather and climatic conditions in recent years. But it is known that even more severe fluctuations have occurred in the past hundred years. Thus, there is an urgent need to assess the probabilities for the occurrence of human catastrophes induced by climatic variations.

In order that this and other important topics of research may be carried out using large numerical models of the atmosphere, the Meteorological Office has recently acquired a CDC Cyber 205 computer system. The system comprises a two pipe version of the computer with one million 64-bit words of memory and four on-line disks each with a capacity of approximately 80 million 64-bit words. It is anticipated that this machine will give more than an order of magnitude increase in power over that of the IBM 360/195 computer currently in use at the Meteorological Office.

The first numerical model to be introduced on the Cyber 205, however, will be a new operational weather forecast model. The current operational forecast suite uses two versions of a 10-level numerical model run on the IBM 360/195; one has a coarse horizontal resolution of 300 km and covers most of the northern hemisphere and the other has a smaller rectangular area covering only the North Atlantic and Western Europe with a finer 100 km resolution. After the final acceptance of the Cyber 205 these models will be replaced by versions of the new forecast model which have been specially coded to take full advantage of the architecture of the Cyber 205. The opportunity has also been taken to improve and enhance the formulations of the models. Finer horizontal and vertical resolutions will be used and the size of the forecast areas increased. The new coarse mesh version, for example, has been designed so that it may be run with 15 levels and a horizontal resolution of 150 km over the area of the globe north of 30^{o} south.

An important factor which will contribute to the high execution speed required of this new model is the use of 32-bit arithmetic precision throughout the program code; on the Cyber 205 32-bit vector arithmetic operates at speeds up to twice that of 64-bit vector arithmetic. The current operational model run on the IBM

360/195 uses 32-bit precision and tests have shown that there is little to be gained in terms of forecast accuracy with the increased precision of 64-bits except perhaps in the area of moist convection modelling[1]. A second consideration is that the memory of the Cyber 205 is effectively doubled so that the version bought by the Meteorological Office may be considered to have two million words of memory.

At present no 32-bit Fortran compiler is available on the Cyber 205. However, Cyber 200 Fortran allows any machine instruction to be inserted directly into a program by means of CALL statements to subroutines with special reserved names. This technique has been used to generate 32-bit vector instructions within the structure of Fortran making use of the facilities given by Fortran linkage and program declaratives.

Much of the scientific development work for the new operational forecast model has been done on the Cray-1 computer at the European Centre for Medium Range Weather Forecasts (ECMWF). Most of this version of the new forecast model has been written in vectorizable Fortran. Experiences from running this program have allowed some conclusions to be made about efficient data organisation and coding for the Cray-1.

The Meteorological Office also has experience of coding up a full forecast model in Cray Assembler Language (CAL). A small team of scientists was set up in 1976 to examine the problems and advantages of using vector computers, including the Cray-1 computer, for numerical weather prediction computations. At that time the Fortran compiler was at an early stage of development and it was decided to write a vectorized CAL version of the limited area fine mesh rectangle version of the 10-level model. The timings obtained from this investigation are presented later along with the equivalent timings from the same model run on the IBM 360/195.

2. Numerical weather prediction models

Numerical weather prediction usually means the prediction of meteorological parameters by the numerical solution of the fluid dynamic equations governing atmospheric motions. The principal equations which are relevant to motions in the atmosphere are Newton's second law of motion, the first law of thermodynamics, the law of conservation of mass, the equation of state and the conservation equation for water substance.

For large-scale motions of the air, the atmosphere may be assumed to be in hydrostatic equilibrium. In which case, the vertical acceleration may be neglected along with the vertical components of the coriolis force. When the hydrostatic approximation is applied the complete system of equations may be written as shown below. The (x,y,p,t) co-ordinate system is assumed where pressure p, is the vertical co-ordinate. The prognostic variables are the horizontal velocity vector $\underline{v}$, the vertical velocity $\underline{w}$, potential temperature θ and the humidity mixing ratio r. The vector operator ∇ is used in a two dimensional, horizontal sense. $\underline{k}$ is the unit vertical vector.

(i) Momentum equation

$$\frac{\partial \underline{v}}{\partial t} + (\underline{v}.\nabla)\underline{v} + w\frac{\partial \underline{v}}{\partial p} + \nabla\Phi + f\underline{k}\times\underline{v} = \underline{F}_H \qquad (1)$$

(ii) Thermodynamic equation

$$\frac{\partial \theta}{\partial t} + \underline{v}.\nabla\theta + w\frac{\partial \theta}{\partial p} = Q/\pi c_p \qquad (2)$$

(iii) Humidity equation

$$\frac{\partial r}{\partial t} + \underline{v}.\nabla r + w\frac{\partial r}{\partial p} = W \qquad (3)$$

(iv) Continuity equation

$$\nabla.\underline{v} + \frac{\partial w}{\partial p} = 0 \qquad (4)$$

(v) Hydrostatic equation

$$\frac{\partial \Phi}{\partial p} + \frac{R\theta\pi}{p} = 0 \qquad (5)$$

Here f is the coriolis parameter, $\Pi=\left(\frac{p}{1000}\right)^{\kappa}$ where $\kappa=\frac{R}{C_p}$, R being the gas constant, C_p the specific heat at constant pressure, and Φ is the geopotential.

The terms on the left hand side of equations (1)-(5) represent the effects of the earth's rotation and gravity on the internal dynamics of the atmosphere. These are termed the dynamical processes. The terms on the right hand side are forcing terms associated with the physical processes. $\underline{F}_H$ is the horizontal component of the frictional force, W is the source and sink of water vapour through evaporation and condensation and Q is the source and sink of sensible heat through the effects of solar and long wave radiation and the release of latent heat.

The solution of the equations describing the dynamical processes is normally accomplished by finite difference techniques. The Meteorological Office uses an explicit grid point integration scheme for its operational numerical weather prediction models. The dynamical equations are split so that the horizontal advection terms are integrated with a timestep limited by the wind speed, whilst the terms which describe gravity inertia oscillations are integrated in a succession of shorter adjustment steps [2]. The advection stage is futher split by the use of a two level Lax-Wendroff scheme [3]. The split explicit scheme offers computational economy and improved accuracy over integration schemes used in earlier operational models.

In simple terms, the integration of the dynamical equations by explicit grid point and other similar techniques requires each level of the forecast area to be covered by a regular grid of points at which the current values of the prognostic variables are stored. A set of linear equations, derived from finite difference approximations to (1)-(5) are then used to step the forecast forwards one timestep at a time by modifying the values stored at each grid point and at each level of the integration domain. The important point here is that the same calculations are generally applied at every point and this makes the solution of the dynamical equations highly vectorizable.

The physical processes embodied by the terms on the right hand side of equations (1) - (5) take place on spatial scales which are considerably shorter than the grid length of numerical weather prediction models. These sub-grid scale processes cannot be dealt with explicitly. Instead their statistical effect is represented in terms of the grid point variables.

The schemes for modelling these processes are characterised by sets of calculations which generally apply only at non-contiguous subsets of the forecast domain. For example, when calculating the transfer of moisture from the earth's surface into the atmosphere it is necessary to distinguish between land, sea and surface ice gridpoints. More usually these subsets vary in position and size from timestep to timestep as in the case of developing areas of rainfall.

3. Programming considerations

This section considers how a numerical weather prediction model may best be coded to take full advantage of the design features of the Cray-1 and Cyber 205 computers. Table 1 lists some of the features of the architectures of these machines but a more detailed description may be obtained from references [4] and [5].

Data organisation

The correct organisation of the forecast data is the first and perhaps the most important step in designing an efficient forecast model for a vector computer. The method chosen should take account of the architecture of the computer, the amount of memory available, the speed of access to the backing store and, of course, the details of the mathematical equations to be solved.

On the Cyber 205 it is desirable to maximise the length of the vectors used in the calculations so as to minimise the start up time (see table 2) which reflects the time taken for the first pair of data elements to pass through the vector pipes. For 32-bit floating point adds or multiplies a vector of length 200 is only 50% efficient whilst a vector of length 10,000 is 95% efficient.

Large vectors are also efficient on the Cray-1, although any vector length which is an exact multiple of 64 gives the optimum result rate. Computation may remain efficient for quite short vectors though it is better if the data is structured so that its vector length is equal to or just less than a multiple of 64.

The code for the current operational model run on the IBM 360/195 uses a data organisation which has all the information at and above each surface grid point arranged in 'columns' of consecutive core store locations. This is useful since it enables all information required at a given locality to be retained in the 'Cache Store', thus reducing the load and store times. However, this a poor method of data organisation for a vector processor since the maximum vector length is 10, the number of levels in the model.

A better method of data organisation is to store the grid point values of each variable in fields of contiguous storage locations. Two such methods have been devised for the Cyber 205 version of the new forecast model. The first organises the data so that the grid point values of each variable are stored as horizontal fields. Thus, in this case, a vector spans one level of the model giving a vector length of around 20,000 for most calculations in the dynamics. However, this method is best suited to the case when the total memory requirements of the model are less than that available on the computer so that no input/output or paging is required except at those times when the forecast results need to be output to disk.

On the Cray-1 the memory is 64-bit addressable and consequently all of the model will not fit into the 1 million words of memory available on the ECMWF machine. There is also a requirement for a version of the new model that will forecast for the whole of the globe and this will require more than 2 million words. Thus the second method of data organisation allows for the efficient use of backing store when the forecast is too large for the memory of the computer. The data is organised into records each of which contains the grid point values for each variable stored in vertical slices. For dynamical calculations vectors span a vertical cross section of the forecast area giving vectors of length 3,000. Each record is read and written to backing store using concurrent input/output. At any stage in the integration of a timestep several adjacent slices are present in memory each undergoing a different step in the integration. This allows for the masking of the input/output with the CPU except at the beginning and end of a timestep when there is a slight 'startup' and 'wind down' effect.

Dynamical calculations

Although very long vectors can be constructed for the dynamical calculations, these ignore the effects at the edge of the forecast area or the cyclic nature of the hemispheric and global forecasts. Special measures must be taken to modify these boundary elements whenever necessary. On the Cray-1 it is more efficient to restructure the data to allow for these boundary effects. For example, the ECMWF grid point model ([6] and [7]) typically has 192 distinct points on a latitude circle, but the code carries 194, repeating the first point at the end, and with a copy of the last point preceding the first point. This has two benefits. When taking east-west differences, the first and last points are not special cases, and also 192 is a multiple of 8 leading to potential bank conflicts, while 194 is not. The number of east-west points (192) is chosen to be suitable for Fast Fourier Transform routines.

The new operational forecast model also has 192 points along a line of latitude. But for 32-bit arithmetic on the Cyber 205 there are 256 banks and increments of 192 will not lead to bank conflicts. It is also convenient to minimise the storage requirements of the model.

We therefore gather the cyclic boundary values into contiguous vectors, perform the arithmetic and scatter the results back to the appropriate storage locations. This adds about 8% to the time taken to do the original instruction, but of course it is only applied where necessary.

As an illustration of the amenability of the dynamical calculations to vectorization it is useful to examine a small piece of coding from the new forecast model. Let us suppose that we wish to calculate $\partial u/\partial y$ at all points on a model level. Figure 1 shows the position of the horizontal grid points. The current values of the u field are stored at the points marked $\times$. The value of $\frac{\partial u}{\partial y}$ is required at the points marked $\circ$.

In Fortran the coding may take the form

```
        DO 1 I = 1,M
        TEMP(I+1)= U(I)-U(I+ISKIP)
      1 DUDY(I)  = (TEMP(I)+TEMP(I+1))*HALFGRID
```

where M+ISKIP is the number of points along a model level, ISKIP is the number of points across the forecast area and HALFGRID is the reciprocal of half the gridlength. This code would be vectorized by the Cray-1 and Cyber 205 compilers. However, it is never easy to see how efficient a piece of code is just by looking at Fortran statements and indeed more efficient code can usually be obtained by coding in assembler language.

On the Cray-1 several sweeps of the instructions shown below are required to obtain M results. At each sweep there are a maximum of 63 results.

```
1.  V1 <---- U(I)              load
2.  V2 <---- U(I+ISKIP)        load
3.  V3 <---- V1-V2             subtract
4.  V4 <---- V3<64             shift vector left by one element
5.  V5 <---- V3+V4             add
6.  V6 <---- V5*HALFGRID       multiply
```

Instructions 2, 3 and 4 chain as do instructions 5 and 6. Ignoring vector start ups the time required to execute the above code is therefore 3M clock periods.

On the Cyber 205 the code reduces to just three vector instructions

```
    1.  SUBTRACT     U(1;M)-U(1+ISKIP;M)
    2.  ADD          TEMP(1;M)+TEMP(2;M)
    3.  MULTIPLY     *HALFGRID
```

of which instructions 2 and 3 may be linked giving an execution time of M/2 clock periods plus start ups for 32-bit arithmetic on a two pipe machine. It is clear that at least in the case of this example the Cyber 205 will be approximately three times faster than the Cray-1 even though several Cray-1 instructions are linked. It is also true to say that the Cray-1 Fortran compiler will produce less efficient code than that shown above producing extra loads in place of instruction 4.

<u>Calculation of the physical processes</u>

Both the Cray-1 and Cyber 205 instruction sets provide instructions which allow the intermittent and conditional structure of the physical processes to be vectorized. On the Cray-1 there is the vector mask instruction which allows two vectors to be merged under the control of a bit mask. This enables computations done at inappropriate grid points to be omitted from the final results. Of course this often means that a large number of unnecessary calculations are performed particularly when the number of active points is small compared with the vector length.

The designers of the Cyber 205, however, have provided the programmer with three alternative methods of vectorizing calculations that only apply at points which form a non-contiguous subset of a larger vector. All three methods first require the setting up of a bit string which points to the active elements of the vector. This may be done by means of a vector compare instruction. The methods are as follows :-

(a) <u>Control Store</u>

Here the required set of operations is done on all elements of the full vector, but the results are controlled by the bit string so that only the active elements of the vector are changed. This is fundamentally the same as the vector merge facility on the Cray-1, but in this case it operates as part of any vector arithmetic instruction.

(b) Compress and expand

The compress instruction uses the bit string to extract the active elements from a vector to form a contiguous subvector. After the calculations have been performed at the shorter vector length, the expand instruction allows the answers to be inserted back into their original place in the vector.

(c) Gather and scatter

The third method available for dealing with this type of construct uses the gather and scatter instructions. Here an integer index list is used to gather active elements from a vector to form a contiguous subvector and to scatter a subvector back into its parent vector. Unlike methods (a) and (b) the time taken to execute these instructions is proportional to the number of active elements in the vector rather than the total vector length (see table 2). There is an extra overhead with this method, however, the setting up of the index list. This may be done efficiently by using the bit string to compress a vector containing the integers 1 to M into the index list.

In order to determine which of the above methods is most appropriate for a given set of calculations use must be made of the timings presented in table 2. It is convenient to consider each vector instruction in terms of its speed relative to the speed of the vector add instruction. Thus a 64-bit floating point divide, for example, is equivalent to 8 64-bit adds. We may then define A as the number of vector instructions, expressed in terms of the vector add instruction, which are to be executed for N results from a vector of length M. Also define B as the number of gathers and scatters or compresses and expands which are needed to reduce the input vectors to length N and to return the result vectors back to length M. In the following all the timings assume 32-bit arithmetic on a two pipe Cyber 205.

(i) Control store v compress/expand

The time in cycles for the control store technique may be expressed as

$$AM/4+X$$

where X is the total start up time. Similarly the compress and expand technique will take

$$B(M/4+55)+AN/4+X$$

cycles where the start up time for the compress and expand instructions has been given the average value of 55 cycles. It is clear that the compress and expand method will be faster than the control store method when

$$B(M/4+55)+AN/4 < AM/4$$

that is when

$$N < (1-C)M-220C \qquad (6).$$

where C=B/A. A and B are constant for any given problem since they are directly related to the number of vector instructions coded by the programmer. N and possibly M may vary at each sweep of code since they could depend on the results of some earlier conditional tests.

(ii) Gather/scatter v compress/expand

The time in cycles using gather and scatter may be expressed as

$$B(1.25N+76)+AN/4+X+M/2+52$$

where 76 cycles is the average start up time for a gather or scatter and an extra term is included to represent the time taken to produce the index list needed. Thus gather and scatter are faster than compress and expand when

$$N < M(1/5-2/5B)-17-40/B \qquad (7)$$

This formula assumes that there are no bank conflicts during the gathers or scatters. It is immediately clear that unless there are more than two instructions the compress and expand technique is always better. This reflects the time taken to do the integer compress initially.

(iii) Gather/scatter v control store

By using the above expressions we can also say that the gather and scatter method is faster than the control store technique when

$$N < M\left(\frac{1-2/A}{5C+1}\right) - \frac{304}{(5+1/C)} - \frac{208}{A+5B} \qquad (8)$$

Figure 2 shows the areas of influence of these three techniques for large values of A, B and M. As A or B become smaller the area of influence of the gather and scatter technique becomes less.

Let us now look at an example to which methods (a), (b) or (c) might be applied. In the part of the model that calculates rainfall from large scale dynamic effects the humidity mixing ratio, r, is tested at each grid point against the saturated mixing ratio for that temperature and pressure, r_s. If $r \leq r_s$ there is no production of rain, but if $r > r_s$ precipitation forms according to the formula

$$P = (r-r_s)T^2/(T^2+\alpha r_s) \qquad (9)$$

where α is a physical constant and T is the temperature.

An analysis of the vector instructions needed to compute P from (9) shows there to be six in all of which one is a divide and so A=11.5. Application of the compress and expand method would require three compresses and one expand so that B=4 as it does in the case of the gather and scatter technique. Relations (6) - (8) now suggest that we use control store when $N \geq .65M-76.5$, compress and expand when $.1M-27 < N \leq .65M-76.5$ and gather and scatter when $N \leq .1M-27$.

In the model atmosphere N represents those points at which supersaturation has occurred during a model timestep through dynamical causes. M represents the total number of points along a model level. We would expect supersaturation to occur in areas where near saturated air is undergoing large horizontal convergence such as is found in regions near depressions. At upper levels little or no convergence of moist air takes place and we might expect only a few points to reach supersaturation. At lower levels greater numbers of points will be active. On the global scale it is thought that the maximum number of saturated points at lower levels would be no greater than 50%.

Thus for models which cover large areas of the globe a combination of methods (b) and (c) is needed. Both methods can be included in the code and the choice of which one to use given by the simple scalar tests (7) and (8).

4. Timings

Two sets of timings are presented here. The first compares the CPU time taken for the same 12 hour forecast run on the Cray-1 and IBM 360/195 computers. The model used here was an early version of the 10-level model[3] . Both versions of this forecast model were coded in assembler language to ensure that the best use was made of the hardware features available on the two machines. Table 3 presents the results. It can be seen that there is a very wide range of relative speeds depending on the precise nature of the the computation. However, the dynamics, which constitutes the largest part of the calculation, is about 20 times faster on the Cray-1 than on the IBM 360/195, and overall the Cray-1 is about 15 times faster. The very much higher speeds obtained for routines such as friction, diffusion and surface exchanges were partly due to a coding reorganisation in which parameters were precalculated and stored in an array instead of being worked out each timestep.

The second set of timings compares versions of the dynamics of the new forecast model run on the Cray-1 at ECMWF and the Cyber 205. The Cray-1 version is coded in vectorized Fortran and it is thought that the timings obtained for this version may be improved by as much as a factor of two by recoding in CAL. The Cyber 205 version is coded using the 'special call' technique refered to earlier. These timings are listed in table 4.

The times obtained from the Cyber 205 are consistently faster than those obtained from the Cray-1, showing a total speed ratio of 3 times the Cray-1 in 64-bits and 7 times the Cray-1 in 32-bits. (Note that if CAL had been used for the Cray-1 program we might have obtained speed ratios of 1½ and 3½ respectively, a result that is consistent with the simple coding discussed earlier.) The extra speed when using the 32-bit code over and above the two times that might be expected is associated with the much longer vectors available in the 'model S' version of the forecast. Also the 'model T' version of the forecast suffers from bank conflicts in 64-bit arithmetic since there are 128 banks on the Cyber 205 when using 64-bit arithmetic.

It is clear from these timings that both the Cray-1 and Cyber 205 computers are capable of giving large improvements in the speed of numerical weather prediction models over that currently available on serial machines. For completely vectorized dynamical problems 32-bit arithmetic on the Cyber 205 will give speeds greater than 40 times those obtainable on an IBM 360/195.

The speed increases that may be obtained from vectorizing the physical processes will be data dependent and will usually be related to the number of active points in the computation. Vectorization of these types of problems can sometimes lead to the extra expense of large numbers of unnecessary computations. However, as we have seen, the compress, expand, gather and scatter instructions on the Cyber 205 may be used to remove these extra unwanted calculations when their cost becomes too expensive.

References

[1] A.P.M. Baede, D.Dent and A.Hollingsworth, ECMWF Tech. Report, 2 (1976)

[2] A.J.Gadd, Q.J.R.Met.Soc., 441, 569 (1978)

[3] G.R.R.Benwell, A.J.Gadd, J.F.Keers, M.S.Timpson and P.W.White, Met Office Scientific Paper, 32, HMSO, London (1971)

[4] P.M.Johnson, Cray Research Inc., Publication 2240002 A (1977)

[5] M.J.Kascic, 'Supercomputers', Infotech State of the Art Report, Infotech International Limited, U.K. (1979)

[6] D.M.Burridge and J.Haseler, ECMWF Tech. Report, 4 (1977)

[7] M.Tiedtke, J-F.Geleyn, A.Hollingsworth and J-F.Louis, ECMWF Tech. Report, 10 (1979)

Table and figure captions.

Table 1. Some of the design features of the Meteorological Office's Cyber 205 and ECMWF's Cray-1.

Table 2. Timings for a selection of vector instructions in cycles. M is the vector length and N is the number of elements gathered from a vector of length M. The Cyber 205 times are for 32-bit arithmetic on a two pipe version of the computer (personal communication from CDC).

Table 3. Comparison of the CPU time taken by a 12 hour forecast using the explicit version of the 10 level model on a Cray-1 and an IBM 360/195 computers. The number of points per level is 576.

Table 4. A comparison of times taken for one timestep of the dynamical part of the new operational forecast model on the Cray-1 and Cyber 205 computers. Model S has 15 levels, 192 points round each latitude and 48 points between equator and pole. Model T has 11 levels, 128 points around each latitude and 45 points between equator and pole.

Figure 1. A section of the horizontal forecast grid used by the new operational model.

Figure 2. Areas of influence of the control store, compress and expand and gather and scatter techniques for large values of A, B and M. N is the number of active points in a vector of length N. C = B/A. A and B are defined in the text.

TABLE 1.

Cray-1	Cyber 205
12 fully segmented functional units.	2 general purpose functional units (pipes).
8 x 64 64-bit vector registers requiring loads from and stores to memory	Data streamed from memory through pipes and back to memory for each vector instruction.
Results from one vector instruction used as input to the next <u>chain</u> provided there are no register or functional unit conflicts. These execute as one instruction. Up to four vector instructions can be chained.	Triadic statement involving a scalar and two vectors may execute as one instruction. This is termed <u>link</u>ing.
1 million words of 64-bit memory.	1 million words of 64-bit memory.
Memory is word addressable.	Memory is bit addressable.
64-bit arithmetic only.	64-bit and 32-bit arithmetic. 32-bit vector instructions execute at speeds up to twice those of 64-bit vector instructions.
Memory has 16 banks.	Memory has 128 banks for 64-bit arithmetic and 256 banks for 32-bit arithmetic.
12.5 nsec clock period.	20 nsec clock period.

TABLE 2.

INSTRUCTION	Cyber 205		Cray-1	
	START UP	RESULT RATE	START UP	RESULT RATE
FLOATING POINT ADD	51	M/4	6*(M/64+1)	M
FLOATING POINT MULTIPLY	52	M/4	7*(M/64+1)	M
FLOATING POINT DIVIDE	80	M/.61	35*(M/64+1)	3M
COMPRESS	52	M/4		
EXPAND	58	M/4		
GATHER	69	N/.8		
SCATTER	83	N/.8		
FLOATING POINT COMPARE TO PRODUCE MASK	56	M/4	9*(M/64+1)	M

TABLE 3.

	IBM 360/195 ASSEMBLER	CRAY -1 ASSEMBLER
Dynamics	93.47	4.73 (X19.76)
Diffusion	16.57	0.59 (X28.08)
Friction	3.61	0.1 (X36.1)
Dynamic rain	17.73	2.25 (X 7.88)
Surface exchanges	3.12	0.07 (X44.57)
Convective adjustment	8.26	2.31 (X3.57)
Complete model	150.97	9.91 (X15.23)

TABLE 4.

DESCRIPTION OF SUBROUTINE	MODEL T HEMISPHERIC 64 BITS		MODEL S HEMISPHERIC 64 BITS CRAY 32 BITS CYBER	
	CRAY-1 FORTRAN	CYBER 205 SPECIAL CALLS	CRAY-1 FORTRAN	CYBER 205 SPECIAL CALLS
PRESSURE, TEMPERATURE AND HMR ADJUSTMENT	.361	.166	.752	.134
REMOVE GRID-SEPARATION INSTABILITY	.379	.101	.813	.113
WIND FIELD ADJUSTMENT	.366	.159	.787	.167
FIRST ADVECTION STEP FOR WIND FIELD	.048	.016	.104	.014
FIRST ADVECTION STEP FOR TEMPERATURE AND HMR	.057	.017	.122	.020
TEMPERATURE AND HMR DIFFUSION	.050	.023	.107	.026
SECOND ADVECTION STEP FOR TEMPERATURE AND HMR	.208	.044	.436	.042
WIND FIELD DIFFUSION	.050	.023	.105	.027
SECOND ADVECTION STEP FOR WIND FIELD	.148	.030	.314	.029
FOURTH ORDER ADVECTION FOR WIND FIELD	.058	.015	.124	.011
TOTAL DYNAMICS	1.725	.594	3.664	.583

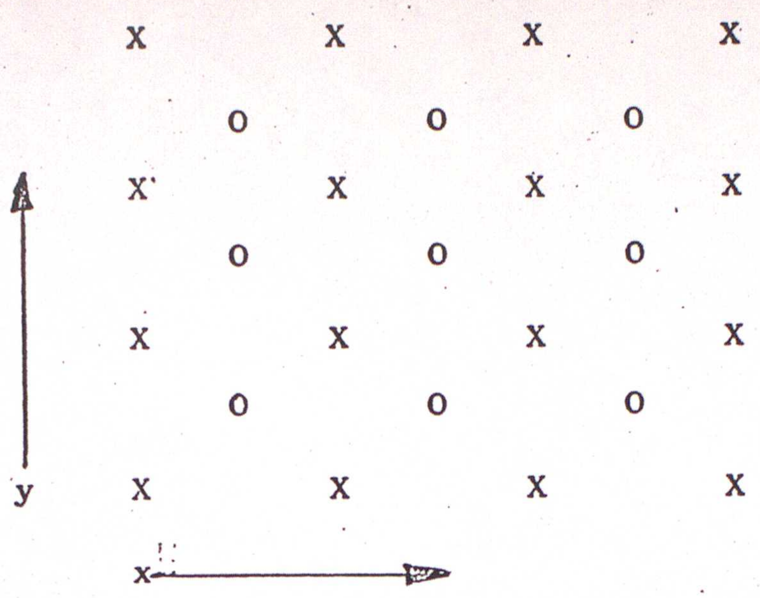

FIGURE 1.

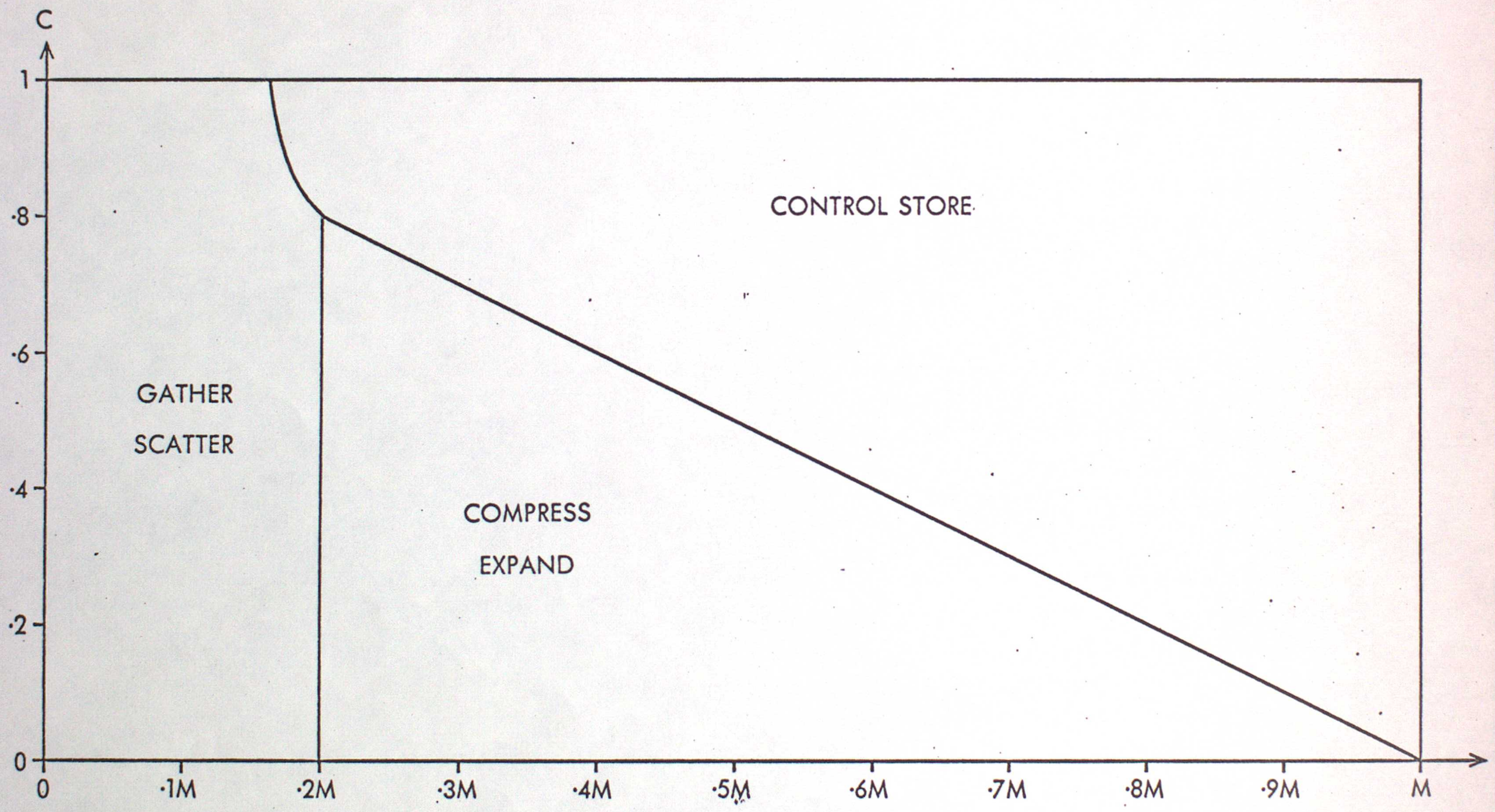
C
1
·8
·6
·4
·2
0
CONTROL STORE
GATHER
SCATTER
COMPRESS
EXPAND
0
·1M
·2M
·3M
·4M
·5M
·6M
·7M
·8M
·9M
M

FIGURE 2.