4-way parallel processor partition of an atmospheric primitive-equation prediction model

by E. MORENOFF
Ocean Data Systems, Inc.
Rockville, Maryland

and

W. BECKETT, P. G. KESEL, F. J. WINNINGHOFF and P. M. WOLFF
Fleet Numerical Weather Central
Monterey, California

INTRODUCTION

A principal mission of the Fleet Numerical Weather Central is to provide, on an operational basis, numerical meteorological and oceanographic products peculiar to the needs of the Navy. Toward this end the FNWC is also charged with the development and test of numerical techniques applicable to Navy environmental forecasting problems. A recent achievement of this development program has been the design, development, and beginning in September 1970, operational use of the FNWC five-layer, baroclinic, atmospheric prediction model, based on the so-called "primitive-equations," and herein defined as the Primitive Equation Model (PEM).

The PEM was initially written as a single-processor version to be executed in one of the two FNWC computer systems. In this form the PEM was exercised as a research and development tool subject to improvement and revision to enhance the meteorological forecasts being generated.

The development reached a point in early 1970 where the PEM was skillfully simulating the essential three-dimensional, hemispheric distribution of the atmospheric-state parameters (winds, pressure, temperature, moisture, and precipitation). Its ability to predict the generation of new storms, moreover, was particularly encouraging. The FORTRAN coded program, however, required just over three hours to compute a set of 36 hour predictions. To be of operational utility, it was clear that several types of speed-ups were in order.

The principal effort in the development of the operational version of the PEM was directed at partitioning the model to take advantage of all possible computational parallelism to exploit the four powerful central processing units available in the FNWC computer installation. Additional speed-ups involved machine language coding for routines in which the physics were considered firm, and the substitution of table look-up operations for manufacturer supplied algorithms. The resultant four-processor version of the PEM was considered ready for final testing in August 1970, four months after work was initiated.

The one-processor version of the PEM required 184 minutes of elapsed time for the generation of 36-hour prognoses. The four-processor version, on the other hand, requires only one hour of elapsed time to produce the same results.

This paper summarizes the principal factors involved in the successful operation of the 4-processor version of the PEM. Operating System modifications needed to establish 4-way inter-processor communications through Extended Core Storage (ECS) are described in the second section. The PEM structure is described in the third section. The partitions into which the PEM is divided are examined in the fourth section. The fifth section is devoted to the methods employed for synchronizing the execution of the partitions in each of the multiple processors and the model's mode of operation. The results of the PEM development and reduction to operational use are summarized in the last section.

FNWC COMPUTER SYSTEMS

COMMUNICATIONS

The Fleet Numerical Weather Central operates two large-scale and two medium-scale computer systems as
shown in Figure 1. The two CDC 2200 computer systems communicate with each other through a random access drum. One of the CDC 3200 computers is linked to one of the CDC 6500 computers by a manufacturer-supplied satellite coupler. The two dual-processor CDC 6500 computer systems are linked with each other through the one million words of Extended Core Storage (ECS).

Normally, the ECS is operated in such a manner that 500,000 words are assigned to each of the two CDC 6500 computer systems with no inter-communication permitted. A mechanism was developed by the FNWC technical staff allowing authorized programs in each of the four central processors of the two CDC computer systems to communicate with each other through the one million words of Extended Core Storage (ECS).

There are three classifications of ECS access, normal, master and slave, designated for each job in the system by an appropriate ECS access code and a pass key. For normal ECS access these fields are zero. If the ECS access code field designates a job as a master, then the associated pass key will be interpreted as the name of ECS block storage assigned to that job. A slave has no ECS assigned to it but is able to refer the ECS block named by its pass key.

A master job in one of the CDC 6500's may have slave jobs in the other CDC 6500. A communication mechanism called ISI was established between the operating systems by FNWC technical staff to facilitate implementation of the master-slave ECS access classification. ISI is a pair of bounce PP routines (one in each machine) which provide a software, full duplex block multi-plexing channel between the machines via ECS. Messages and/or blocks of data may be sent over this channel so that ISI may be used to call PP programs in the other machine or to pass data such as tables or files between the machines.

Obtaining a master/slave ECS access code is accomplished by two PP programs: ECS and IEC. A job wishing to establish itself as a master first requests a block of ECS storage in the same manner of a normal access job. Once obtained, the labeling of this block of ECS storage is requested by calling the PP program ECS with the argument specifying the desired pass key and the access code for a master. The program ECS searches the resident control point exchange areas (CPEA) for a master with the same pass key. If one is found the requesting job is aborted even if the program ECS used ISI to call IEC in the other machine. IEC will perform a similar search of the CPEA in its own machine and return its findings to the program ECS via ISI. If the other machine is down, or if no matching key can be found, the label is established, otherwise the requesting job is aborted. Before returning control to the requesting job, the program ECS increments the ECS parity error flag and monitors via a special monitor function developed at FNWC. A non-zero value of this flag has the effect of preventing ECS storage moves in the half of ECS assigned to the particular machine.

Similarly, a job wishing to establish itself as a slave calls the PP program ECS with the appropriate pass key and access code. ECS searches its own machine's CPEA for a master with a matching key. If none is found, IEC is called on the other machine via ISI and the search is repeated in the other CPEA. If still none is found, this fact is indicated to the requesting job. If a match should exist in either machine, the original ECS will have the address (ECRA) and field links (ECFL) of the requesting job saved in its CPEA and will be given the ECRA and ECFL of the matching master.

Modifications made to the ECS storage move program allow ECS storage moves in a machine with no master present. Modifications to the end of job processor reset the ECRA and ECFL of slaves to their values and decrement the ECS parity error flag in the monitor when a master terminates.

ATMOSPHERIC PREDICTION MODEL STRUCTURE

Several developmental variations of a five-layer baroclinic atmospheric prediction model, based on integrations of the so-called primitive equations, were designed and developed by Kesel and Winninghoff in the 1969-1970 period at FNWC Monterey.

The governing equations are written in flux form in a
manner similar to Smagorinsky et al., and Arakawa. The corresponding difference equations are based on the Arakawa technique. This type of scheme precludes nonlinear computational instability by requiring that the flux terms conserve the square of an advected parameter, assuming continuous time derivatives. Total energy is conserved because of requirements placed upon the vertical differencing; specifically, the special form of the hydrostatic equation. Total mass is conserved, when integrated over the entire domain. Linear instability is avoided by meeting the Courant-Friedrich-Lewy criterion.

The Phillips sigma coordinate system is employed in which pressure, $P$, is normalized with the underlying terrain pressure, $\pi$. At levels where sigma equals 0.9, 0.7, 0.5, 0.3, and 0.1, the horizontal wind components, $u$ and $v$, the temperature, $T$, and the height, $Z$, are carried. The moisture variable, $q$, is carried at the lowest three of these levels. Vertical velocity, $w = -\frac{\partial}{\partial z}$, is carried at the layer interfaces, and calculated diagnostically from the continuity equation. See Figure 2.

The Clarke-Berkovsky mountains are used in conjunction with a Kurihara form of the pressure-force terms in the momentum equations to reduce stationary "noise" patterns over high, irregular terrain.

The Richtmyer centered time-differencing method is used with a ten-minute time step, but integrations are recycled every six hours with a Matsuno (Euler backward) step to greatly reduce solution separation. The mesh length of the grid is 381 kilometers at 60 North. The earth is mapped onto a polar stereographic projection for the Northern Hemisphere. In the calculation of map factor and the Coriolis parameter, the sine of the latitude is not permitted to take on values less than that value corresponding to 23 degrees North.

Lateral diffusion is applied at all levels (sparingly) in order to redistribute high frequency components in the mass and motion fields. Surface stress is computed at the lowest layer only.

A considerable part of the heating "package" is fashioned after Mintz and Arakawa, as described by Langlois and Kwok. The albedo is determined as a function of the mean monthly temperature at the earth's surface. A Smagorinsky parameterization of cloudiness is used at one layer (sigma equals 0.7), but based on the relative humidity for the layer between 0.7 and 0.4. Dry convective adjustment precludes hydrostatic instability. Moisture and heat are redistributed in the lowest three layers by use of an Arakawa-Mintz small-scale convection parameterization technique. Small-scale convective precipitation occurs in two of the three types of convection so simulated. Evaporation and large-scale condensation are the main source-sink terms in the moisture conservation equation. Evaporation over land is based on a Bowen ratio, using data from Budyko.

In the computation of sensible heat flux over water, the FNWC-produced sea surface temperature distribution is held constant in time. Over land, the required surface temperature is obtained from a heat balance equation. Both long- and short-wave radiative fluxes are computed for two gross layers (sigma = 1.0 to 0.6 and from 0.6 to 0.2). The rates for the upper gross layer are assigned to the upper three computational levels. Those rates for the lower gross layer are assigned to the lower two computational levels.

The type of lateral boundary conditions which led to the over-all best results is a constant-flux restoration technique devised by Kesel and Winninghoff, and implemented in January 1970.

The technique was designed to accomplish the following objectives:

a. To eliminate the necessity of altering the initial mass structure of the tropical-subtropical atmosphere as is the case when cyclic continuity is used.

b. To eliminate the problems associated with the imposition of rigid, slippery, insulated-wall boundary conditions; particularly those concerning the false reflection of the computational mode at outflow boundaries.

c. To preserve the perturbation component in the
The aforementioned areas in the prognostic period (although no dynamic prediction is attempted south of 4 North the output is much more meteorological than fields which have been fattened as required by cyclic continuity).

The procedure is as follows: All of the distributions of temperature, moisture, wind, and terrain pressure are preserved at initial time. A field of restoration coefficients which vary continuously from unity at and south of 4 North to zero at and north of 17 North is computed. At the end of each ten minute integration step the new values of the state variables are restored back toward their initial values (in the area south of 17 North) according to the amount specified by the field of restoration coefficients. The net effect of this procedure is to produce a fully dynamic forecast north of 17 North, a persistence forecast south of 4 North, and a blend in between. The mathematical-physical effect is that the region acts as an energy sponge for externally (outwardly) propagating inertia-gravity oscillations.

The basic inputs associated with the initialization procedure are the virtual temperature analyses for the Northern Hemisphere at 12 constant pressure levels distributed from 1000 MBS to 50 MBS, height analyses at seven of these pressure levels, moisture analyses at four levels from the surface up to 500 MBS. In addition, the terrain height, sea level pressure and sea surface temperature analyses are used.

Several types of wind initialization have been tried: geostrophic winds (using constant Coriolis parameter); linear balance winds; full balance winds; winds obtained by use of an iterative technique. Aside from geostrophic winds the quickest to compute is the set of non-divergent winds derived from solution of the so called linear balance equation. These are entirely satisfactory for short-range forecasts (up to three days). The degree of prediction skill currently being observed from the tests is very gratifying. It is clear that little or nothing is known about the initial specification of these parameters over large areas of the Northern Hemisphere, particularly over oceans and at high altitudes.

As noted at the start of the section, the equations are written in flux form and an Arakawa-type conservative differencing scheme is employed. No attempt will be made to exhibit herein a complete set of the corresponding difference equations, since it is well beyond the scope of this paper to do so. Rather, it will suffice to show the main continuous equation forms (using only symbols such as $H$, $Q$, and $F$, to denote all of the diabatic heating effects, moisture source and sink terms, and surface stress, respectively).

There are five prognostic equations, one of which must be integrated prior to parallel integration of the remaining four. These are the continuity equation, the east-west momentum equation, the thermodynamic energy equation, and the moisture conservation equation. Heights (geopotentials) are computed diagnostically from the hydrostatic equation (the scaled vertical equation of motion). Vertical velocities are calculated from a form of the continuity equation. The pressure-force terms are shown in their original forms. [The pressure surfaces are actually synthesized “locally” about each point, by means of the hypsometric conversion of pressure changes to geopotential changes; and geopotential differences are computed on these pressure surfaces.] This Kurihara-type modification tends to reduce inconsistent truncation error when differentiating the terrain pressure (which remains fixed in any column) and geopotentials of sigma surfaces (the “smoothness” of which varies with height).

A. East-West Momentum Equation

\[
\frac{\partial \pi u}{\partial t} = -m \left( \frac{\partial (u \pi)}{\partial x} + \frac{\partial (u \pi)}{\partial y} \right) + \pi \frac{\partial (wu)}{\partial \sigma} \\
+ \pi \nu - m \left( \frac{\partial \phi}{\partial x} + R \frac{\partial \sigma}{\partial x} \right) + K \pi \nu + F_x
\]

B. North-South Momentum Equation

\[
\frac{\partial \pi v}{\partial t} = -m \left( \frac{\partial (w \pi)}{\partial y} + \frac{\partial (w \pi)}{\partial x} \right) + \pi \frac{\partial (vw)}{\partial \sigma} \\
+ \pi \nu - m \left( \frac{\partial \phi}{\partial y} + R \frac{\partial \sigma}{\partial y} \right) + K \pi \nu + F_y
\]

C. Thermodynamic Energy Equation

\[
\frac{\partial \pi T}{\partial t} = -m \left( \frac{\partial (\pi u T)}{\partial x} + \frac{\partial (\pi v T)}{\partial y} \right) \\
+ \pi \frac{\partial (\pi T)}{\partial \sigma} + H \pi + K \pi T
\]

D. Moisture Conservation Equation

\[
\frac{\partial \pi e}{\partial t} = -m \left( \frac{\partial (\pi u e)}{\partial x} + \frac{\partial (\pi v e)}{\partial y} \right) + \pi \frac{\partial (\pi e)}{\partial \sigma} + Q \pi
\]

where $Q$ = moisture source/sink term.
E. Continuity Equation
\[ \frac{\partial \mathbf{u}}{\partial t} = -m \left( \frac{\partial (u \mathbf{u})}{\partial x} + \frac{\partial (v \mathbf{u})}{\partial y} \right) + \mathbf{w} \cdot \nabla \mathbf{u} \]

F. Hydrostatic Equation
\[ \frac{\partial \phi}{\partial \sigma} = -\frac{RT}{\sigma} \]

PARTITIONING THE MODEL

The PEM may be considered in three distinct sections: the data input and initialization section; the integration section repeated in each forecast time step; and the output section. Each sixth time step, the basic integration section is modified to take into consideration the effects of diabatic heating. This includes incoming solar radiation, outgoing terrestrial radiation, sensible heat exchange at the air-earth interface, and evaporation. Condensation processes, in contrast, are considered every time-step. Each thirty-sixth time step, the results of the preceding forecast hours are output and the integrations reiterated.

The basic structure of the PEM, as represented by the governing set of difference equations and the method of their solution, is naturally suited for partitioning for parallel operation and concurrent execution in multiple processors. The particular partitioning implemented was selected in order to ensure approximately equal elapsed time for the execution of concurrently operating partitions. Four-way partitions were principally employed, although both three-way and two-way partitions were introduced where appropriate.

The basic partition of the model was based on the observation that during each time step in the forecast process the momentum equations in the east-west and north-south directions, the thermodynamic energy equation, and the moisture equation could each be executed concurrently in each of four different processors. By virtue of the centered time-differencing method, the forcing functions to be evaluated in the solution of each of these equations require data generated during the preceding time step and accessed on a read only basis during the current time step. Hence parallel processing could be achieved by providing separate temporary locations for storage of intermediate results during execution of a time step by each processor and by providing a mechanism to insure that each processor is at the same time step in the solution of its assigned equation and, where required, at the same level within that time step.

With this four-way partitioning within the basic time step as a starting point, additional possibilities for simultaneity in the model's operation were observed and further partitions developed. For example, prior to the execution of the four-way partitioning within each time step a three-way partition was implemented which allowed the continuity equation to be solved for the interface vertical velocities and the local change of lower boundary pressure at the same time that geopotential-field correction terms are generated. The model's initialization section was similarly partitioned three ways and the output section two ways. Finally, the heating effects computations were implemented as a three-way partition.

The four-way, three-way and two-way partitions were packaged and compiled as four separate programs, one for each of the four FNWC processors. The overall structure of the partitioned model is illustrated in Figure 3. Following completion of the output section at time step (36), the integration sequence is recycled from time step (1) as shown.

Processor 1 is designated as the "master" processor and Processors 2, 3, and 4 as the "slave" processors, both in the sense described in the inter-computer communications section and in the sense that each time step is initiated by command from Processor 1 and termi-
nated by Processor 1 acknowledgment of a “complete” signal emanating from each of Processors 2, 3, and 4. At the completion of each step, results from the computations of that time step are transferred from temporary to permanent locations in storage and the next time step initiated. Once again, the transfer is initiated by command from Processor 1 and terminated by Processor 1 acknowledgment of a transfer complete signal received from Processors 2, 3, and 4.

The structure of a typical time step partition is illustrated in Figure 4. At the start of the time step a three-way split is initiated by Processor 1 during which time Processor 1 integrates the continuity equation to obtain vertical velocities and Processors 2 and 3 compute the ten pressure-force-term geopotential correction fields in the east-west and north-south directions, respectively. At this time Processor 4 is not executing a portion of the model and may either be idling or operating on an independent program in a multi-programmed mode. The completion of the assigned tasks by Processors 2 and 3 are signaled to Processor 1 which then initiates the basic four-way split. The variables $u$, $v$, $T$ and $e$ represent the new values of the variables obtained through integration of the east-west and north-south oriented momentum equations, the thermodynamic energy equation and the moisture conservation equation, respectively. The variable $L$ represents the computation of the effects of the large scale condensation process.

Once the computations of the $u_i$, $v_i$ and $T_i$ ($i=1, 2, 3, 4, 5$) are initiated in Processors 1, 2, and 3, respectively, they proceed independently of one another to the end of the time step. Each “$i$” value represents another layer in the five-layer atmospheric model.

An added consideration is introduced into the computations of Processor 4, however. Before the effects of the large scale condensation process can be computed for a layer, both the Thermodynamic Energy equation and the Moisture Conservation equation must be solved at that layer. Hence, a level of control is required to synchronize the execution of Processor 4 with Processor 3 within the individual time step computations. Further, the Dry Convective Adjustment computation in Processor 4 requires the completion of all five layers of the Thermodynamic Energy equation before it can be initiated so that a second level of intra-step control is required. At the conclusion of the Dry Convective Adjustment computation, the Hydrostatic equation is integrated in Processor 4 to obtain the new geopotential fields. The time step is concluded with the transfer of intermediate time step results from temporary to permanent storage.

The basic time-step partition structure is modified each sixth time step to include the effects of a diabatic heating. The heating section was implemented as a three-way partition illustrated in Figure 5. Additional intra-step level control is required to synchronize the execution of each of the partitions as shown in the figure. Note that the heating partition in Processor 3 is itself divided to allow as great a degree of simultaneity as possible with the execution of partitions in Processors 1 and 2.

The output section, executed each thirty-sixth time step (at the completion of six forecast hours), is partitioned as shown in Figure 6. The output section partitions were placed in Processors 3 and 4 principally for central memory space considerations, more central memory being available in these processors than in Processors 1 and 2. The basic function of each output partition is co-ordinate transformation of the forecast variables and conversion to forms suitable for the user community.

Each output partition is initiated by command from Processor 1. Processor 4 may immediately begin processing of the east-west and north-south momentum equation variables but must wait on the transformation of the Phi fields until Processor 3 has completed the
Preprocessor program. A three-way partition was not implemented since the Preprocessor must be completed prior to the transformation of the Thermodynamic energy equation and moisture conservation equation variables.

To increase total system reliability a checkpoint restart procedure was designed and coded. At each output step (6, 12, 18, ..., 72 hours) all of those data fields required to restart the PEM are duplicated from their permanent ECS locations onto a magnetic tape by Processor 1, at the same time that Processors 3 and 4 are processing the output forecast fields. The essential difference between these two data sets is that the restart fields contain the variables on sigma surfaces as opposed to the pressure surface distributions required by the consumers.

The “restart” procedure itself requires less than a minute. If the prediction model run is terminated for any type of failure (hardware, software, electric power, bad input data, etc.), the restart capability ensures that the real time loss will be less than ten minutes.

In addition to the four processor version of the Atmospheric Prediction Model a two-processor version was also implemented. The primary motivation for the second implementation was to provide a back-up capability with graceful degradation which could be operated in the event one or two of the central processing units were down for extended periods. The two-processor version will also be used as the vehicle for further research and development efforts to improve the meteorological and numerical aspects of the model, and the quality (skill) of the resultant forecasts.

**PARTITION SYNCHRONIZATION AND EXECUTION**

The parallel execution of the multiple partitions is realizable because it is possible to postulate a mechanism by which the operation of each partition in each of the multiple processors can be exactly synchronized. This mechanism is an adaptation to the requirements of the PEM and the characteristics of the FNWC computer installation of a general program linkage mechanism known as the Buffer File Mode of Operation.8,9,10

Implicit in the Buffer File Mode of Operation is the concentration of all inter-program communications through Buffer Files. A Buffer File is a set of fixed length blocks organized in a ring structure and placed in each data path from one program to another. The program generating the data to be passed places the data into the Buffer File. The program to receive the data finds the data to be operated on in the Buffer File.

The flow of data through the Buffer File is unidirectional; that is, one program may only write data to the Buffer File and the other may only read data from the
Buffer File. Pointers are maintained which indicate which blocks in the Buffer File have last been written into and read from by the two programs involved in the data transfer. The Buffer File Mode of Operation can be used to synchronize the operation of otherwise asynchronously operating programs in the same or different processors by either of two methods.

In the first instance, program synchronization is effected by regulating the streaming of data through the Buffer File from one program to another. The program writing data to a Buffer File cannot proceed beyond the point in its execution when it is necessary to place data into the Buffer File and there is no room for additional data in the Buffer File. Similarly, a program reading data from the Buffer File cannot proceed beyond the point in its execution when it requires data from the Buffer File and there is no additional data in the Buffer File. The execution of a program, either waiting for additional data in its input Buffer File or for additional space in its output Buffer File, is temporarily delayed, and thereby brought into synchronization with the execution of the other program.

In the second instance, program synchronization is effected by conveying “change of state” or “condition” information from one program to the other. The Buffer File block size is chosen on the basis of the quantity of data in that program’s output Buffer File. The fact that there has been a change in state of the program can readily be sensed by the other program which then can read the block of data from the Buffer File. The second program can determine the nature of the change in state of the first program by examination of the data in the block it has read from the Buffer File.

The bi-directional transfer of the program state information is realized by the introduction of Buffer File pairs. The first Buffer File can only be read from by the first program and written to by the second program, while the second can only be read from by the second program and written to by the first program. This method of exchanging state information between programs not only provides a mechanism for synchronizing the execution of two otherwise asynchronously executed programs, but also eliminates the internal program housekeeping which would normally be needed to coordinate the accesses and the sequences of such accesses of the programs to the program state information.

The PEM synchronization mechanism, referred to herein as the Partition Synchronization Mechanism (PSM), is based on the latter alternative. The application of the PSM to the multi-processor FNWC computer environment requires the Buffer Files to reside in some random access storage device jointly accessible by each of the processors. The device which satisfies this requirement is the ECS, operated in the manner previously described.

A pair of Buffer Files is assigned between each two partitions for which bi-directional transfer of state information is required. Hence in the typical time step partition structure illustrated in Figure 4 and amplified in Figure 5, Buffer File pairs are assigned between partitions resident in Processors 1 and 2, 1 and 3, 1 and 4, 2 and 4, and 3 and 4.

The nature of the change of state information to be passed between any pair of partitions in the PEM is whether or not one partition has reached a point in its execution where sufficient data has been developed to allow the other partition to initiate or continue its own execution. This can be represented as a single “GO-NO GO” flag to be sensed by the second partition. Hence, in the PEM the Buffer File recirculating ring structure reduced to a simple single one word block maintained in ECS.

Referring to Figure 3, it can be seen that the issuance of a “GO-NO GO” signal by a partition is equivalent to either a command to “split” the straight line execution of the model into multiple partitions or to “join” the execution of the multiple partitions into a lesser number of partitions. A five character Buffer File naming convention was established to facilitate identification of which process was involved.

The first two characters of the name serve to identify whether the Buffer File is associated with an inter-step or inter-level signal; the former is designated by the characters “IS” and the latter by the characters “IL”. The third character specifies whether a split (“S”) or a join (“J”) is being signaled. The fourth and fifth characters specify the processors in which the partitions writing and reading the Buffer File are located respectively. Hence Buffer File ISS12 is used by the partition resident to Processor 1 to split its operation by initiating execution in Processor 2 in going from one time step to another.

When the PEM is to be executed the four programs of which it is comprised are loaded, one into each of the four Processors. The programs in Processors 2, 3 and 4 are immediately halted upon initiation and manually delayed until the program in Processor 1, the master Processor, has been assigned the necessary ECS for the model’s execution and has initialized all Buffer Files to reflect a NO GO condition. Processors 2, 3 and 4 are then permitted to enter a programmed loop in which each periodically tests a Buffer File to determine when it may initiate processing of its first partition.
While in this programmed loop the slave Processors may either be engaged in the execution of unrelated programs or simply remain in a local counting loop.

Upon completion of the data input phase of its operation, Processor 1 removes the hold on the execution of Processors 2 and 3 which then proceed with the initialization phase while Processor 4 remains at the hold condition. At the completion of its portion of the initialization phase, Processor 1 holds until receipt of a GO signal from Processors 2 and 3, signifying the completion of their assigned partitions. Processors 2 and 3 again enter a hold status after providing the GO signal to Processor 1. Finally, Processor 1 initiates the iterative integration section by signaling the GO condition for Processors 2, 3 and 4. At the completion of the execution of the partitions in Processors 2, 3 and 4 the master Processor is notified via the appropriate Buffer Files and each once more enters the hold condition and remains there until Processor 1, having verified that each partition has been completed, signals the transfer of the time step results from temporary to permanent storage. This process then continues to repeat itself, modified as previously described in each sixth and thirty-sixth time step.

Inter-level holds and go's are generally implemented in the same manner as the inter-step holds and go's described in the preceding paragraph. There is one exception, however. In the partition executed in Processor 4 in the iterative integration section, a separate Buffer File is provided to control the initiation of the execution of the large scale condensation effects computation at each of levels 1, 2 and 3. The separate Buffer file at each level is predicated on the need to allow the partition in Processor 3 to proceed on with its execution after signaling the start of execution of Processor 4 at each level without waiting for an acknowledgment of completion of that level by Processor 4.

This emphasizes a particularly important aspect of the operation of the PEM. The execution of the partitions in the different processors cannot get out of synchronization with one another. Each is always working on the same time step at the same time. If the partition in one of the Processors is delayed, for example, while that Processor solves a higher priority problem, then all the Processors at the completion of the processing of their partitions will hold until the delayed Processor "catches-up." The execution of the partitions will not fall out of synchronization.

CONCLUSIONS

The Atmospheric Prediction Model developed at FNWC was partitioned to be operated in a 4-Processor and a 2-Processor configuration, in addition to the 1-Processor configuration for which it was initially designed. The 4-Processor version is currently in operational use at FNWC while the 2-Processor version provides a back-up capability in the event of equipment malfunction and a new research and development tool.

A Partition Synchronization Mechanism was developed for purposes of synchronizing the execution of the partitions being executed in each of the multiple processors. The nature of PSM is such as to ensure that each partition is always operating on data in the same time step. The ability to guarantee this synchronization implies it is possible to allow other independent jobs to co-exist and share what computer resources are available with the Partitioned Atmospheric Prediction Model.

The PSM fully utilizes modifications to the operating systems of each of the two CDC 6500 dual processor computers to allow programs in each of the four processors to communicate with each other using ECS. In addition to the intercomputer communications the FNWC operating system modifications insure software protection from interference by non-authorized programs.

As a consequence of employing the 4-Processor version of the Atmospheric Prediction Model, the same meteorological products were generated in 60 minutes rather than the 184 minutes required of the 1-Processor version. This reduction in time allowed the incorporation of a new and more powerful output section and the extension of the basic forecast period from 36 hours to 72 hours. The 72 hour forecast is produced in an elapsed time of 2 hours.

The next step in the evolution of the FNWC PEM involves expanding grid size from 63x63 points to 89x89 points. To accommodate the additional central memory and processing requirements required of such a shift in grid size, partitioning of the horizontal domain rather than the computational burden is under consideration. It is estimated that partitioning the horizontal domain will reduce overall central memory requirements by one-half and allow the 72 hour forecast on the expanded grid to be performed in only four hours as opposed to the five and one-third hours required by the current partitioning method. The results of these new efforts will be reported on in a later paper.

REFERENCES

1 P G KESEL F J WINNINGHOFF
Development of a multi-processor primitive equation atmospheric prediction model
Fleet Numerical Weather Central Monterey California
Unpublished manuscript 1970
2 J SMAGORINSKY S MANAGE
L L HOLLOWAY JR
Numerical results from a 9-level general circulation model of
the atmosphere
Monthly Weather Review Vol 93 No 12 pp 727-768 1965
3 A ARAKAWA
Computational design for long term numerical integration of
the equations of fluid motion: Two dimensional incompressible flow
Journal of Computer Physics Vol 1 pp 119-143 1966
4 N A PHILLIPS
A coordinate system having some special advantages for
numerical forecasting
Journal of Meteorology Vol 14 1957
5 Y KURIHARA
Note on finite difference expression for the hydrostatic relation
and pressure gradient force
Monthly Weather Review Vol 96 No 9 1968
6 A ARAKAWA A KATAYAMA Y MINTZ
Numerical simulation of the general circulation of the atmosphere
Proceedings of WMO/IUGG Symposium of NWP Tokyo 1968
7 W E LANGLOIS H C W KWOK
Description of the Mintz-Arakawa numerical general circulation model
UCLA Dept of Meteorology Technical Report No 3 1969
8 E MORENOFF J B McLEAN
Job linkages and program strings
9 E MORENOFF J B McLEAN
Inter-program communications, program string structures and
buffer files
10 E MORENOFF
The table driven augmented programming environment: A
general purpose user-oriented program for extending the
capabilities of operating systems
Rome Air Development Center Technical Report
TR-69-108 1969