Parallel I/O Requirements of Four Oceanography Applications

Jason A. Moore
Department of Computer Science
Oregon State University
Corvallis, Oregon 97331-3202
(503) 737-4052, FAX: (503) 737-3014
moorej@research.cs.orst.edu

Abstract

Brief descriptions of the I/O requirements for four production oceanography programs running at Oregon State University are presented. The applications all rely exclusively on array-oriented, sequential file operations. Persistent files are used for checkpointing and movie making, while temporary files are used to store out-of-core data.

1 Introduction

Originally all but ignored by parallel computer makers and users, parallel I/O has received much attention in recent years. Parallel I/O hardware performance has improved with the advent of disk arrays and the addition of I/O nodes to parallel computers. The programmer's interface to parallel I/O routines has not improved significantly. The syntax and semantics of I/O routines vary widely from machine to machine and language to language. Before standards for programming parallel I/O begin to appear, the users' needs must be fully understood.

To help in this understanding, we present here brief descriptions of the parallel I/O requirements of four production oceanography applications running in the College of Oceanography at Oregon State University.

2 Tropical Cyclone Modeling

The tropical cyclone model used by Boon Chua [1] uses the finite difference method on a four-dimensional array of data. The sizes along all dimensions are shown in the table below, although higher or lower resolution models are also used depending on the computing power available.

Dimension	Represents	Size
1	Number of time steps	360
2	Number of vertical grid points	9
3	Number of x grid points	256
4	Number of y grid points	132

The program, written in CM-FORTRAN, runs on the CM-5. The reading of initial conditions consists of filling the array with four time-steps (each an array of shape $9 \times 256 \times 132$) of data. While running, the cyclone model performs checkpointing by writing out the grid array every n iterations. Also during runtime, the program performs its own virtual memory management by storing two thirds of a time step's data ($240 \times 256 \times 132$) to disk while computing with one third of the data ($120 \times 256 \times 132$). This must be done due to the limited RAM available on the OSU CM-5. The computation consists of doing both forward and backward finite differencing up to 1000 times with each time step's data. Again, due to lack of main memory, each of these iterations requires that the starting conditions be read from disk. The usable output from the cyclone modeling program is the array data at the last time step. These data are stored to disk at the end of the run.

3 Tide Model

The tide model program used by Rodney James [2] is written in CM-FORTRAN and runs on the CM-5. The computation consists of multiplying a large (25,000 $\times$ 5,000) matrix of complex double precision values by its hermetian transpose. Because the matrix is 2 gigabytes, it cannot fit into the memory of the CM-5. The matrix is instead broken into a 20 $\times$ 20 array of submatrix files, each containing a 1250 $\times$ 250 block of the matrix and each labeled with coordinates (i, j). The matrix product is found by multiplying each column with the h-transpose of the desired rows. The results are concatenated to a single file which, when the computation completes, contains a 5,000 $\times$ 5,000 element array. This array is then read into main memory, where it barely fits as a single unit. A library routine finds the eigenvector matrix $(5,000 \times 5,000)$ and the eigenvalues (5,000 of them), which are stored to two different files.

4 Ocean Model

The ocean model program by Rodney James [3] is written in C* and runs on the CM-5. The computation consists of repeated nearest-neighbor finite difference operations on a three dimensional grid ranging in size in the (x, y) plane from 128 $\times$ 128 to 512 $\times$ 512 with 2 to 12 vertical (z) levels. The initial condition inputs are trivial, consisting of simple wind stress forces.

File output from the ocean model is for checkpointing or for making movies and takes place after each hour of computation. The output data consist of four double precision values for each grid position for, on a $512 \times 512 \times 12$ model, a total of 25MB per checkpoint. Some runs also output moving averages of the data in the same shape and size.

5 Inverse Ocean Model

The inverse ocean model of Ed Zaron [4] models large scale ocean circulation. The program is written in CM-FORTRAN and runs on the CM-200. The data are laid out in a three-dimensional grid with 180 points in the x direction, 60 in the y direction, and 22 in the z direction. The number of data points is model-specific; finer resolution is not needed.

The inputs of the program consists of files each containing an array of double precision values arranged in shapes $180 \times 60 \times 22$ and 180×60. Typically, three files of the former size and six of the latter are input, although the numbers vary. No file I/O takes place during the computation stage of the program, which consists of a series of conjugate gradient and inner product operations. Output from the program consists of many files, the number varying from run to run, each containing one array of double precision values arranged in shape $180 \times 60 \times 22$.

6 Conclusion

The I/O requirements for four production oceanography programs have been presented. The applications rely exclusively on array-oriented, sequential file operations. Persistent files are used for checkpointing and movie making, while temporary files are used to store out-of-core data.

References

[1] B. Chua. Personal communication, October 1994.

[2] R. James. Personal communication, October 1994.

[3] R. James. Personal communication, October 1994.

[4] E. Zaron. Personal communication, October 1994.