The Vectorization Challenge

The Earth Simulator in Japan and the Cray X1 at Oak Ridge National Laboratory have sparked renewed interest in vector computers among researchers using the Community Climate System Model (CCSM).

To utilize these systems, all the CCSM component models required software engineering efforts to vectorize each code.

The Community Land Model (CLM) was particularly vector hostile and required a complete re-write to provide acceptable performance on vector architectures.

The strategy for CLM vectorization was:

- Develop a single CLM code that runs well on both vector and scalar machines while maintaining the hierarchical nature of the existing data structure.
- Alter data structures to obtain short and predictable data strides.
- Move loops around into science subroutines, and vectorize over these long loops instead of the short loops over nested PFTs and soil/snow levels.
- Create readable and understandable code while avoiding vector-specific versions of code (if
dependent) everywhere possible.
- First, do no harm! Code changes must not reduce the performance on the present target scalar platforms.

New Data Structures in CLM3.0

- In CLM, the horizontal land surface heterogeneity is represented by a nested grid hierarchy of gridcells, landunits, columns, and plant functional types (PFTs).
- The hierarchical grid organization is reflected in the data structures used in the model code.
- The grid hierarchy was previously implemented as arrays of derived data types containing scalars for flux and state variables.
- The new data structures represent the grid hierarchy as derived data types containing arrays for flux and state variables. In Fortran 90, these arrays are implemented as many loops over subgrids.
- The use of pointers in data structures requires compiler directives for loop vectorization.

Arrays of derived data types

- A cache–friendly blocking structure, called clumps, was superimposed on the data structure for improved computational efficiency.
- For optimal load balancing, gridcells are distributed among clumps in cyclic fashion, and clumps are distributed among MPI processes in cyclic fashion.
- Clumps also serve to block data for shared memory parallelism using OpenMP or Cray streaming.
- Another set of super–structures, called filters, were added to better support vectorized processing of columns and PFTs.
- Filters group like columns or PFTs based on their process–specific categorization and are used for indirect addressing into the main data structure hierarchy.

Code Reorganization

- The highest level in the main driver routine run over clumps for each MPI process and provide for OpenMP or Cray Streaming parallelism.
- Science subroutines called within these loops are passed local clump bounds for gridcells, landunits, columns, and PFTs as needed.
- Also passed are relevant filters (counts and vector indices).
- Also passed are relevant filters (counts and vector indices).
- Loops within science subroutines run over grid or subgrid units.
- The use of pointers in data structures requires compiler directives for loop vectorization.

Validated on the Cray X1

- Offline Community Land Model (CLM3.0)
- Offline CLM3.0 with Dynamic Vegetation Model (DGVM)
- Standalone Community Atmosphere Model (CAM3.0) with CLM3.0
- Standalone CAM3.0/CLM3.0 with DGVM

Forrest Hoffman (forrest@climate.ornl.gov)
Mariana Vertenstein (mverten@ucar.edu)
http://climate.ornl.gov/clm/vector_performance/
CLM3.0 Model Release Website
http://www.cgd.ucar.edu/tas/clm/distribution/clm3.0/

Acknowledgements

This research used resources of the Center for Computational Sciences at Oak Ridge National Laboratory which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE–AC05–00OR22725. Additional resources were provided by the National Center for Atmospheric Research which is sponsored by the National Science Foundation (NSF), the Central Research Institute of Electric Power Industry (CRIEPI) in Japan, Cray Inc., and NEC Corporation.