

CCSM Software Engineering Working Group Meeting Report
Thursday, 22 June 2006
The Village at Breckenridge

One of the focal points of the June 2006 CCSM Software Engineering Working Group (SEWG) meeting was to examine the impact on CCSM of future petascale plans. Another focal point of the SEWG meeting was to examine the support for new grids in both ESMF and CAM. Finally, a summary of SciDAC and CSEG activities wrapped up the meeting.

Impact of Global Petascale Plans on Geoscience Modeling:

Rich Loft (NCAR, SCD) presented a very broad overview of petascale architectures, NCAR's petascale science plans, and issues of using CCSM as a petascale application. Several significant hardware challenges necessary to achieve petascale architectures were outlined: memory speeds are not keeping up with CPU, microprocessor performance improvement is slowing, and reliability and fault tolerance will pose serious obstacles.

The NSF Track-1 background and goals were summarized. A performance goal of 1 PFLOPS sustained is targeted for "interesting problems" running on a single capability system that is fault tolerant. There is recognition on the part of NSF that meeting this target will be extremely challenging and that applications will need significant modification to run on such a system. The funds for the Track-1 project are projected as \$200M over 4 years starting in FY07. This is a single award for an end-to-end system with facility costs not included. The solicitation was put forth in June 2006. Preliminary proposals are due this September, and three to four preliminary proposers will be invited to submit full proposals that will be due in February 2007. The award will be announced in September 2007.

UCAR will need to define its petascale science goals and the associated system and application development resource requirements needed for the Track-1 proposal. The key question for CCSM is whether it will qualify on the Track-1 system. Running CCSM as an ensemble application on at least 10-25% of the system (where one instance of CCSM would be 10K to 50K processors) could provide a justification for this resource requirement. For CCSM to scale out to this number of processors will require that each component code remove any serialization of memory or I/O and also remove the necessity for "giant" look-up tables. In addition, the current multiple executable implementation of CCSM should be replaced by a single executable model. Finally, dynamic load balancing schemes and communication overlapping should be implemented across components, and particular attention should be paid to addressing algorithms with irregular memory access patterns, since they are expected to perform poorly. CCSM also needs to look at new coupling strategies (single executable concurrent versus single executable sequential) and new ensemble strategies (e.g., stacking instances as multiple threads on a CPU). Finally, CCSM also needs to determine if the current required CCSM software utilities (such as MPI, ESMF, MCT, pNETCDF, and NCL) will scale to the number of projected processors.

Scaling CCSM to a Petascale System:

John Dennis (NCAR, SCD) provided an overview of the current challenges faced in attempting to scale CCSM to a petascale system. The goal is to look at each CCSM component separately

and examine both current scalability limitations and the changes necessary to enable execution on a large number of processors (10,000 to 30,000). Blue Gene/L (BG/L) will be utilized to examine this scalability since it is a prototype for a petascale system, and NCAR currently has access to a large number of processor counts. Currently, POP2.0, CICE4.0, and CLM3.x (the latest development version) have been examined.

John has already modified POP2.0 to achieve significant advances in performance and improve scalability. Three classes of code modifications were made to POP: the introduction of 1D data structures, partitioning via the use of space-filling curves, and the implementation of a 3D boundary exchange (rather than the original 2D boundary exchange that was applied to 3D fields). The resulting modifications only affect nine POP files. Although modification on its own only resulted in a 10-20% speed up, the cumulative impact was huge and resulted in a 2.1x speed up for a 1 degree POP running on 128 processors on a POWER4. More significantly, these modifications enable a 0.1 degree POP2.0 to run at 7.9 years/wallclock day on 30K BG/L processors. The remaining issue in POP2.0 is to implement parallel I/O.

CICE4.0 shares both grid and infrastructure design with POP2.0. It is planned for this model to be the official "CCSM" ice component. John has begun to try to optimize CICE4 performance utilizing the partitioning via the use of weighted space-filling curves. Results were not yet available. As in POP2.0, the remaining issue is to implement parallel I/O.

CLM3.x was also examined for performance and scalability issues. Non-scalable memory problems have been identified and are currently being addressed in CLM development work. Once these issues are resolved, and parallel I/O is implemented, CLM should scale fairly well from a performance perspective, since it is fundamentally scalable code.

The remaining components that must still be examined are CAM and the CCSM coupler, CPL6. CAM has already been ported to BG/L and the generalization to non lat-lon grids is currently being worked on. Although parallel I/O has been implemented for netCDF history files, parallel binary I/O (for restart files) has not yet been addressed. As in the other CCSM components, parallel I/O remains an important outstanding issue that must be resolved for the full system to be run on petascale architectures.

New CAM Grids:

Brian Eaton provided a summary of the changes recently made to CAM to support implementing a dynamical core (dycore) that operates on a non-rectangular lat-lon grid. The talk began by pointing out the lat-lon grid dependencies in CAM's physics and utility packages. Since the design of CAM's physics package uses a data structure based on arbitrary collections of columns, it is not immediately obvious where assumptions about rectangular lat-lon grids might lie. However, a review of the code quickly reveals that the physics-dynamics coupler, the history module, and the code to interpolate boundary datasets all depend on the dycore's commap module, which contains global latitude and longitude arrays.

It is the responsibility of the dycore to define the global grid, and CAM's current dycores all use a rectangular lat-lon grid, which is contained in the commap module. To break the assumption by the dynamics-physics coupler (in the phys_grid module) that the dycore uses a lat-lon grid,

the use of `commap` was removed from `phys_grid` and instead the `phys_grid_init` method must now ask the dycore via the `get_horiz_grid_d` method of `dyn_grid` to return the global column indices along with the lat and lon coordinate of each column in each dynamics block. So the dycore is now responsible for defining a global column numbering. If the grid is a rectangular lat-lon grid that ordering must start on the southernmost latitude at the Greenwich meridian, and increase to the east, then north.

The global column numbering is used to define an unstructured grid that can be stored in netCDF datasets that conform to CF metadata conventions. `phys_grid` provides `gather/scatter` methods to do transforms between chunks and global fields. The global field can be unstructured or can be a rectangular lat-lon grid. The dycore should provide the method that writes the global grid information to the history file. A `dyn_io` module was suggested as a container for this method and the `gather/scatter` methods needed to transform between the global field and the dynamics decomposition. These methods have not yet been implemented.

The boundary data interpolation supports two classes of data. If the data is zonally averaged, then both time and spatial interpolation to the dynamics grid latitudes is done. In the case of data with three spatial dimensions, it is assumed that spatial interpolation to the dynamics grid has been done outside of CAM and only time interpolation is allowed.

The existing code for doing zonal interpolations was using the global latitude grid from `commap` and doing the spatial interpolation once on `masterproc`, then broadcasting. This was replaced by a broadcast of the input data (optionally read on every process) and then interpolate just to the set of latitudes that actually occur in the set of chunks in each process. This parallelizes the spatial interpolation. The new code is in the boundary data module and is used in all places where zonally averaged boundary data is used.

For boundary data that is spatially 3D, new methods have been added to support the unstructured grid to the `boundarydata` module. These methods read the needed data from the unstructured grid in each process and put the required data into the chunked data structures. The code for dealing with the 3D rectangular lat-lon grids has been left in place. That code typically reads the 3D grid on `masterproc` and scatters to the chunks in each process that are subsequently time interpolated. The code that does this still needs to be consolidated in the `boundarydata` module.

New Grids and Directions in ESMF:

Don Stark (ESMF) summarized the support for new grids and outlined new directions for ESMF. New directions include extending ESMF into new domains, such as space weather and hydrology, as well as the introduction into ESMF of new coupling paradigms, such as multiple executable and direct inter-component data transfers (via “put” and “get” methods), 3D grids, and observational data streams.

New support for non-rectilinear grids in ESMF was included in the internal ESMF release (Version 3.0.0) put out May 2006. This internal release introduced several new features that support regridding for structured and unstructured grids and also introduced new data structures that will be utilized as the building blocks for ESMF support of new grids such as cubed sphere, tripole and geodesic. The ESMF 3.0.0 internal release requires that the user provide the grid and

remapping weights required for remapping. This is the same functionality that is currently assumed in CCSM (utilizing MCT). Unlike MCT, however, ESMF does not assume a 1D data structure and uses multidimensional ESMF_Array objects to reference native arrays.

A summary was also provided on the three new classes utilized by ESMF to support general structured grids: the DELayout class, the DistGrid class and the Array class. The DELayout class provides an additional layer of abstraction to the Virtual Machine (VM). The DistGrid object describes the index space topology and its decomposition in terms of Decomposition Elements (DEs). In particular, combined with a DELayout and a VM, the DistGrid object defines the data distribution of a domain decomposition across a component's computational resources. Finally, the new Array class describes decomposed array data via a distributed data class.

Future ESMF support for regridding will allow the option of internally generated grids and remapping weights.

SciDAC Update:

Pat Worley (SciDAC) gave an update on software engineering activities in the SciDAC CCSM Consortium Project, including a description of activities since the last CCSM workshop, a retrospective on SciDAC-1 performance engineering, and an overview of SciDAC-2 software engineering plans. Recent software engineering activities included support for the addition of interactive carbon and sulfur cycles to CCSM, the implementation of ocean ecosystem trace gases through the coupler, introduction of MCT into the surface model interfaces, support for non-lon/lat grids in the CAM physics/dynamics interfaces, and work on single-executable CCSM. Performance engineering activities dealt primarily with porting, performance evaluation, and performance optimization on the Cray X1E and XT3. Algorithmic studies were also undertaken, investigating FVCAM problem and processor scalability and the performance impact of increases in the number of tracers.

The performance engineering retrospective listed the large number of performance tuning options implemented in CAM, and summarized the performance improvement in CAM over the past 5 years, from parallel algorithm improvements (more than a factor of 4) and optimizations on new architectures (more than a factor of 6). The new SciDAC-2 project will focus on accelerating the development of a first generation Earth System model based on CCSM. This will require extending CCSM to include representations of biological, ecological, chemical, and aerosol processes, while continuing to improve the performance, portability, and scalability of the model on current and future computing architectures.

CSEG Update:

Mariana Vertenstein (CSEG) gave an update on software engineering work in the CSEG group. The summary spanned five basic areas: improvements to the CCSM scripts, status of the data model rewrite project, porting and performance, progress in both the single executable concurrent CCSM and single executable sequential CCSM, and software engineering progress for the various CCSM components.

Significant work was done to improve the CCSM scripts infrastructure, including the addition of a new timing tool that provides automated information to help determine load balance,

throughput, and cost of a run. These changes make it easier to run and test new CCSM science with optimal load balance. The CCSM test framework was also rewritten using this new functionality, resulting in a new test bed that will make it more straightforward to both add new tests and to perform a wide variety of current regression tests from a single command line invocation.

Significant performance and porting work was accomplished over the last several months. Both release and development versions of the CCSM were ported to ORNL Cray X1 (phoenix), and work is underway to port the CCSM to the ORNL Cray XT3 (jaguar).

The serial version rewrite of all CCSM data models has been completed, resulting in data components that can perform spatial interpolation from input data resolution to model resolution and that are able to data cycle over a subset of years. In addition, DATM7 can now duplicate stand-alone CLM functionality, and DOCN7 has both data-ocean model and slab-ocean model functionality. The next step is to parallelize the serial versions of all data models and begin using these parallel versions in the prototype sequential CCSM that is being developed.

Progress has also been made in creating both single executable concurrent and sequential CCSM versions. The existence of a single executable concurrent system should improve both CCSM portability and code debugging. CSEG is leveraging the work that Helen He and Chris Ding (SciDAC) have done in the creation of a single executable concurrent system. Progress has also been made in the creation of a single executable sequential system. Starting from stand-alone CAM, a “pseudo-sequential CCSM” has been created that is independent of CAM data structures and where the time evolution is based upon ESMF general time management utilities. Both ESMF and MCT will be examined as coupling frameworks. The new “coupling” layer that has been introduced in the design of this system will make it straightforward to add new CCSM components and permit the atmosphere and land to run on a different grid from the ocean and ice components.

Component updates included the implementation in CAM of non lat-lon grids, as well as significant FV dycore interface refactoring. Finemesh grids have been implemented in CLM, providing the new capability to have CLM run on its own independent grid. The incorporation of major CCSM features into the LANL POP2.1 code base is now complete, and the resultant POP2 code has been incorporated into the CCSM code repository and CCSM scripts. Finally, the Polar Climate Working Group has decided to incorporate CICE as the new standard CCSM ice model (with the name “Community Ice Code”). Testing has been done using CICE 3.1 to establish main differences with CSIM 5.0. CICE 3.1 changes answers significantly but is still within the realm of the same climate. The goal is to incorporate CICE4 (with new data structures) into CCSM and into pseudo-sequential CCSM

Attendees:

Mariana Vertenstein, NCAR
Pat Worley, Oak Ridge National Lab
Cecelia DeLuca, NCAR
Adrienne Middleton, NCAR
Bert Semtner, Naval Postgraduate School

Charles Hakkarinen
Christiane Jablonowski, University of Michigan
David Bailey, NCAR
John Dennis, NCAR
Diane Feddema, NCAR
Dmitry Shkurko, Intel Corporation
John Drake, Oak Ridge National Lab
Brian Eaton, NCAR
Erik Kluzek, NCAR
Michael Ham, Oak Ridge National Lab
Ilana Stern, NCAR
Ilene Carpenter, SGI
Robert Jacob, Argonne National Lab
James Edwards, IBM
Jeff Lee, NCAR
Juliana Rew, NCAR
Jon Wolfe, NCAR
Brian Kauffman, NCAR
Keith Lindsay, NCAR
William Lipscomb, Los Alamos National Lab
Mathew Maltrud, Los Alamos National Lab
Mark Taylor, Sandia National Lab
Arthur Mirin, Lawrence Livermore National Lab
Mathew Rothstein, NCAR
Nancy Norton, NCAR
Jerry Olson, NCAR
Peter Lauritzen, NCAR
Philip Jones, Los Alamos National Lab
Kevin Raeder, NCAR
Ram Nair, NCAR
Rory Kelly, NCAR
Sumner Dean, Los Alamos National Lab
Donald Stark, NCAR
Gary Strand, NCAR
Silverio Vasquez, NCAR
Joe Tribbia, NCAR
William Spotz, Sandia National Lab
Yoshikatsu Yoshida, CRIEPI
Trey White, Oak Ridge National Lab
Mark Petersen, Los Alamos National Lab
Minghua Zhang, Stony Brook University