CSEG Update

Mariana Vertenstein

CCSM Software Engineering Group

NCAR

Brief Overview

- Infrastructure improvements
  - Creation of new CCSM experimental database
- Porting/Performance
  - Now have single executable development CCSM3
  - Update on ports to blueice, XT3/XT4 (jaguar), BG/L
- Development efforts (current and upcoming)
  - Reusability/scaling/portability
  - CAM, CLM, POP2, CICE4, Data Models
  - Single executable sequential CCSM (ESMF/MCT)
  - Update on CCSM3.5
  - Creation of more flexible build for both concurrent and sequential CCSM
  - Creation of next generation long term archiving utility

Infrastructure Improvements

- New experimental run database created
  - Is web browsable - uses Linux/apache/PHP
  - Enables experiments to be tracked and possibly duplicated if needed
    - Simulation case directories are stored in a subversion repository dedicated to CCSM experiments
  - Centralizes metadata associated with a CCSM run
  - All current simulations for ccsm3.5 are now stored in this database

    http://ccsm-rundb.cgd.ucar.edu/case_list.php

Infrastructure Improvements (cont)

- CCSM testing
  - CAM test suite constantly expanding
  - CCSM test suite upgraded to permit “seamless” integration of new science and software
  - CLM test suite upgraded and is currently undergoing rapid expansion
- Subversion development code repository has matured greatly since last year
  - Making extensive use of subversion externals (e.g. almost all non-CAM specific code is now treated via externals)
- Bug tracking via Bugzilla is being utilized extensively - for both problems and enhancements

Porting/Performance

- CCSM3 development code can now be run in either single executable or multiple executable mode
  - single executable will simplify portability and is default
  - build system issues need to be addressed (everything should be built as a unique library)
- CCSM3 development and release code base ported and validated
  - NCAR IBMs (SMT enabled)
  - ONRL Cray X1E and XT3/XT4
- CCSM3 development code ported to BG/L
  - fully active T31_gx3v5 can run on 32 procs
  - all dead T85_gx1v3 can run on 512 procs (in VN or CO mode)
  - memory issues arise at higher resolutions/processor combinations
  - will soon examine regridding/scalability with all-dead sequential system

Porting/Performance (cont)

- Current performance of 1.9x2.5_gx1v4 fully active system - used for CCSM3.5 simulations
  - 18 years/day on NCAR blueice (208 processors)
  - 22 years/day on ORNL jaguar (XT3) (256 processors) - will attempt to scale to 45 years/day on jaguar (XT4) when it comes back in production

Scaling, Scaling, Scaling

- As we try to scale each component out to thousands of processors - performance and memory scaling must be addressed
- Memory scaling bottlenecks are prohibiting the examination of performance scaling on platforms such as BG/L
- Memory scaling goals:
  - Limit number of global non-distributed arrays to 1 or 2
  - Implement parallel I/O throughout CCSM
- Current Status for memory scaling
  - POP2/CICE4 exhibit acceptable memory scaling
  - CLM3 is close - extensive work has been done by Tony Craig to add address this limitation
- Current Status for parallel I/O
  - John Dennis has developed a parallel i/o library
  - Incorporated into POP2 (binary) and HOMME (netCDF)
  - DATM7/CLM are next steps

CLM Development

- Implement memory scaling by removal of all but a handful of global non-distributed arrays
- Implement parallel I/O via us of PIO library
- Extend fine-mesh implementation to permit coupling with atmospheric component on non-regular lat-lon grid (HOMME, FV cubed sphere)
- Replacement of stand-alone CLM forcing code with DATM7/sequential driver.
- Attempt a 1/6 degree offline CLM simulation on BG/L by late spring (require both memory and I/O scaling)
- Incorporation of global urban model in CLM trunk code
- Extension of CLM regression test suite to encompass all new science

CAM Development

- FV dycore
  - Performance enhancements in PILGRIM
  - Implement dynamics import/export states for communication with dynamics/physics coupler.
- HOMME dycore
  - Assumptions of rectangular lat/lon grid removed from standard physics code
  - Running in ideal physics and aqua-planet modes.
- Single Column Atmosphere Model (SCAM)
  - Significant re-factoring implemented to improve maintainability and extensibility of SCAM within CAM
- Enhance build to allow linking to external ESMF library
- WACCM with MOZART chemistry
  - latest WACCM code put on the trunk

CAM Development (cont)

- Near-term planned development
  - Provide external archiving script -- remove archiving functionality from CAM.
  - Convert restart files to NetCDF.
  - Implement parallel I/O via use of PIO library
  - Improve scalability of CAM-FV by allowing physics and dynamics to run on different numbers of processes
  - Continue working on clean dycore interface implementations (FV/HOMME).
  - Transition from a CAM-centric build of the sequential CCSM to a system where each component builds its own library

POP2 Development

- Add ecosystem model to CCSM POP2
  - Ecosystem model is currently only in POP1.4
  - Import ecosystem modules into the POP2 framework and add CCSM-specific features
- Scale CCSM POP2 to large numbers of processors
  - eliminate one-to-one correspondence between the decomposed domain sub-blocks and number of ocean processors
  - add support for John Dennis' space-filling curves
- Add HYCOM support to CCSM

CICE Development

- Large effort over the last year to move to CICE4 as the CCSM ice model
- Near term development efforts include:
  - Incorporate John Dennis' space-filling curve code
  - Move to netCDF restarts, grids, etc
  - Implement parallel I/O

Data Model Development

- Docn7, Datm7, Dice7, Dlnd7 successfully implemented as serial code - all data models now have uniform functionality
  - Can re-grid from input to model grid
  - Various time interpolations option supported
- Parallelization of Datm7, Docn7 completed - remaining components will be parallelized by beginning of April
- Examining performance and scaling behavior
  - BGC carbon pool spin ups put unanticipated performance requirements on Datm7 since hourly CAM forcing data must be read in
  - Serial I/O is imposing scaling limitations - parallel I/O needed to truly scale to high processor counts
- Near term - implement parallel I/O via use of PIO library

Single-Executable Sequential CCSM

- Goals
  - Permit plug and play functionality (can easily swap active and data components)
  - Keep full backwards compatibility with current concurrent CCSM
  - Maintain all current stand-alone component functionality
  - Standardize coupling interfaces
- Status
  - Created both MCT and ESMF application code base for drivers and associated mappers and mergers
  - Each component needs to only have one coupling framework specific module - for example

    atm_comp_mct.F90 or atm_comp_ESMF.F90

    to couple to framework dependent top level driver

MCT-CPL7

- MCT top level driver and associated mappers and mergers will be referred to as MCT-CPL7
- MCT-CPL7 code used to communicate fluxes/states between CAM and the CAM surface components and is on the CAM Subversion development trunk
- MCT-CPL7 code is now completely independent of CAM data-structures or “uses” statements
- MCT-CPL7 code has also been used to duplicate offline CLM capability
  - Datm7 (NCEP forcing) coupled to CLM3
  - bfb results with cpl6 Datm7/CLM3 NCEP coupling
  - removes need to hack cpl6 to optimize performance
  - performace is better than when using concurrent cpl6 or current offline CLM3
- Re-gridding has been implemented with dead components to account for different atm/ocn grids (Rob Jacob)
- MCT-CPL7 is now in separate top-level subversion directory and obtained in stand-alone CAM and offline CLM via SVN externals

ESMF-CPL7 and ESMF Stage 1 Evaluation

- ESMF-CPL7
  - ESMF interfaces and driver, mapper and mergers are on a CAM branch
  - High priority placed on migrating this code to CAM development trunk

- ESMF Stage 1 Evaluation consists of following quantitative metrics:
  - Correctness: Successfully met
    - Code is round-off with respect to MCT version on that branch
    - All relevant CAM tests pass with ESMF coupling
  - System requirements (ESMF build requirements): Successfully met
  - Performance: currently being evaluated (bluevista and phoenix)
  - Memory: preliminary results look fine and meet metric

ESMF Stage 1 Evaluation (cont)

- Plan for moving forward
  - ESMF Stage 1 code base will be incorporated into the subversion CAM trunk as soon as quantitative metrics are met. This will permit continued testing and upgrading of ESMF specific code base as the CAM trunk evolves
  - ESMF is currently examining “qualitative” aspects of Stage 1 to ensure that implementation was done optimally
  - ESMF is devoting software support (.5FTE) to successfully meet the quantitative metrics, migrate the code base to the CAM trunk and start on Stage 2 effort
  - ESMF Stage 2 acceptance plan (for fully functional sequential ESMF compliant CCSM) will be drafted by May

Other Development efforts

- POP/ROMS nested one-way coupling as part of NRCM effort
- Creation of CCSM3.5
  - POP2/Ecosystem model
  - CLM3.5
  - CAM4_0 (with some parameterization changes)
  - CICE4
- 1870 spin-up of CCSM3.5
- Creation of more flexible build for both concurrent and sequential CCSM
- Creation of next generation long term archiving utility

Some Time Lines

- Fully functional MCT-CPL7 with all active components by June
- ESMF-CPL7 on CAM trunk by June
- POP/ROMS 1-way coupling run in June
- High resolution offline CLM run in June