NCAR contributions to ACES4BGC: Final report

NCAR contributed to the ACES4BGC project through software engineering work on aerosol model implementation, build system and script changes, coupler enhancements for biogeochemical tracers, improvements to the Community Land Model (CLM) code and testing infrastructure, and coordinating and integrating code changes from the various project participants. These are described in detail below.

Aerosol model

We updated the LLNL sectional aerosol code to a recent version of the CAM trunk, and checked it in to a branch of the CESM Subversion repository. We have reduced the numerous CPP directives used to embed the aerosol model specific code. In addition, code reviews by CAM software engineers identified the need for a cleaner interface between CAM chemistry and aerosol models. This improved interface has been implemented and is now on the CAM trunk.

We have also worked with Xiaohong Liu (formerly at PNNL) to implement a new 4-modal version of the modal aerosol module (MAM4) for CAM. MAM4 adds a new primary carbon mode to the default aerosol scheme in CESM1 to better treat the aging of organic aerosol and black carbon from sources to remote areas. The MAM4 package is now on the CAM trunk, and initial simulations have been performed with it turned on.

CESM Build system and scripts

We enhanced the CESM build system in a number of ways that contributed to this project, as well as providing important long-term capabilities for both ACME and CESM. First, we have made it easier to bring in C++ code. Second, we have made it easier to bring in external libraries, especially those written in C++. And third, we have added capabilities to facilitate performance tuning on LCF platforms.

In addition, we refactored the CESM scripts to be able to address the growing complexity of the model system and thereby facilitate adding new model configurations (component sets) and new model grids (including new statically refined grids).

Finally, together with the Computational Tools and Performance group, we implemented a robust method for setting the project number for a given case. This feature is important for the performance team's ability to collect automated data, as well as for correct job charging. This feature has been incorporated onto the development trunks of both ACME and CESM.

Coupler enhancements for tracers

We finalized enhancements to the CESM coupler that make it easier to add new tracers that are exchanged between components. New tracers can now be added as namelist items, without having to change any coupler code. Previously, coupler code needed to be modified any time a new exchange of fields between components was introduced.

Land model improvements

ACES4BGC has funded numerous land model improvements that will benefit biogeochemical modeling in both ACME and CESM. These have included improved software infrastructure as well as new scientific capabilities:

1. We undertook a major refactoring of the CLM data structures to make the code more modular. Previously, most data in CLM was put in a shared "clmtype" module. We

have moved these data to be local and private to the modules that operate on the data, following a more object-oriented design. For example, one module that has seen a particular benefit from this new design is the module responsible for the emissions of volatile organic compounds (VOCEmissionMod). This refactoring makes it easier to understand data flows between different CLM modules, to add automated unit tests, and to perform scientific development and testing on individual modules.

2. We separated the River Transport Model (RTM) component from CLM and introduced it as a separate component at the driver level. This facilitated the recent introduction of an improved river model developed at PNNL: The MOdel for Scale Adaptive River Transport (MOSART). MOSART has now been incorporated into both ACME and CESM, and will allow for the introduction of new parameterizations of the transport of chemical species.
3. We fixed a number of threading bugs in the CLM code, which had prevented the latest CLM version (CLM4.5) from running multi-threaded. In addition, we developed and applied coding conventions that will help ensure that CLM continues to work properly in a threaded environment. These fixes enable better coupled model performance than was previously possible. We have also prototyped a complete rework of how CLM handles threading, which avoids nearly all of the common threading bugs encountered in new code; this rework is possible thanks to the major CLM refactoring described above.
4. We introduced the first unit tests into the CLM source code. In addition, we have implemented a variety of unit test utility modules, to make it easier to write unit tests for new scientific code. We are now gradually expanding the set of modules covered by unit tests, complementing the existing CESM system tests. This unit test framework will allow bringing in new code with greater confidence and fewer bugs.
5. We greatly enhanced CLM's tool for interpolating initial conditions: First, we introduced the capability to easily regrid CLM initial conditions at runtime (rather than through a separate tool); this capability will also facilitate online regridding capabilities in the future. Second, we added new vertical interpolation capabilities, which allow changing the number of soil layers. Third, we fixed some long-standing bugs in the regridding tool when interpolating from a non-crop case to a crop case, or vice versa. Fourth, we added both unit and system tests of this interpolation code.
6. We implemented "dynamic landunit" capabilities for crops, allowing crop area to change throughout a run. Infrastructure to conserve carbon and nitrogen through these transitions is largely complete, and should be on the CLM trunk within the next few weeks.
7. We have identified reasons for degraded performance of recent CLM versions relative to CLM4, and have put in place plans to address the biggest performance degradations.
8. We made changes to CLM's memory allocation to cut runtime in half, when running with the new crop model.

Coordinating and integrating code changes

We set up a master ACES4BGC branch in the CESM repository for the integration of all changes from this project. In addition, we put in place an automated test suite that tests the main model configurations of interest to this project. This test suite was run whenever major changes were made to the branch.

We integrated the following project developments:

1. New timers and performance-tracking scripts
2. Fixes and enhancements to the atmosphere model's GPU capabilities
3. DMS code in the ocean and atmosphere
4. Machine updates for a number of DOE platforms

Through the integration process, we identified a number of bugs in the original implementations of these developments, and worked with the code's authors to fix these bugs. In addition, we added features to enhance the usability of the new features, such as a general option for turning off prescribed atmospheric chemistry fluxes (replacing these with prognostic fluxes).