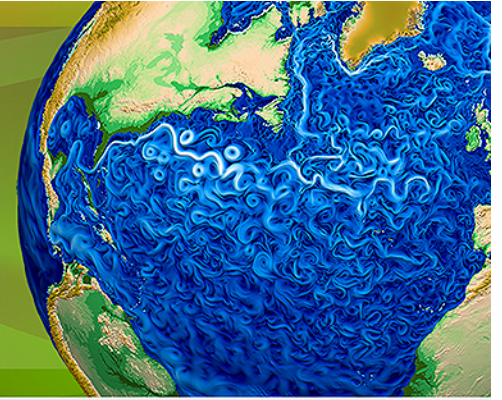




Accelerated Climate Modeling
for Energy



ACME SE Update

Robert Jacob

June 16, 2015

20th CESW Workshop

SEWG Meeting

Breckenridge, CO

ACME in a nutshell...

A new U.S. climate modeling effort led by the U.S. Department of Energy Office of Biological and Environmental Research

or...

“an unprecedented collaboration among eight national laboratories and six partner institutions to develop and apply the most complete, leading-edge climate and Earth system models to challenging and demanding climate-change research imperatives. It is the only major national modeling project designed to address DOE mission needs to efficiently utilize DOE leadership computing resources now and in the future.”

Future DOE leadership computing resources

- Intel/Cray Cori (NERSC)
 - 9,300 Xeon Phi nodes (“Knight’s Corner”)
 - Approx 30PF
 - Production in 2017

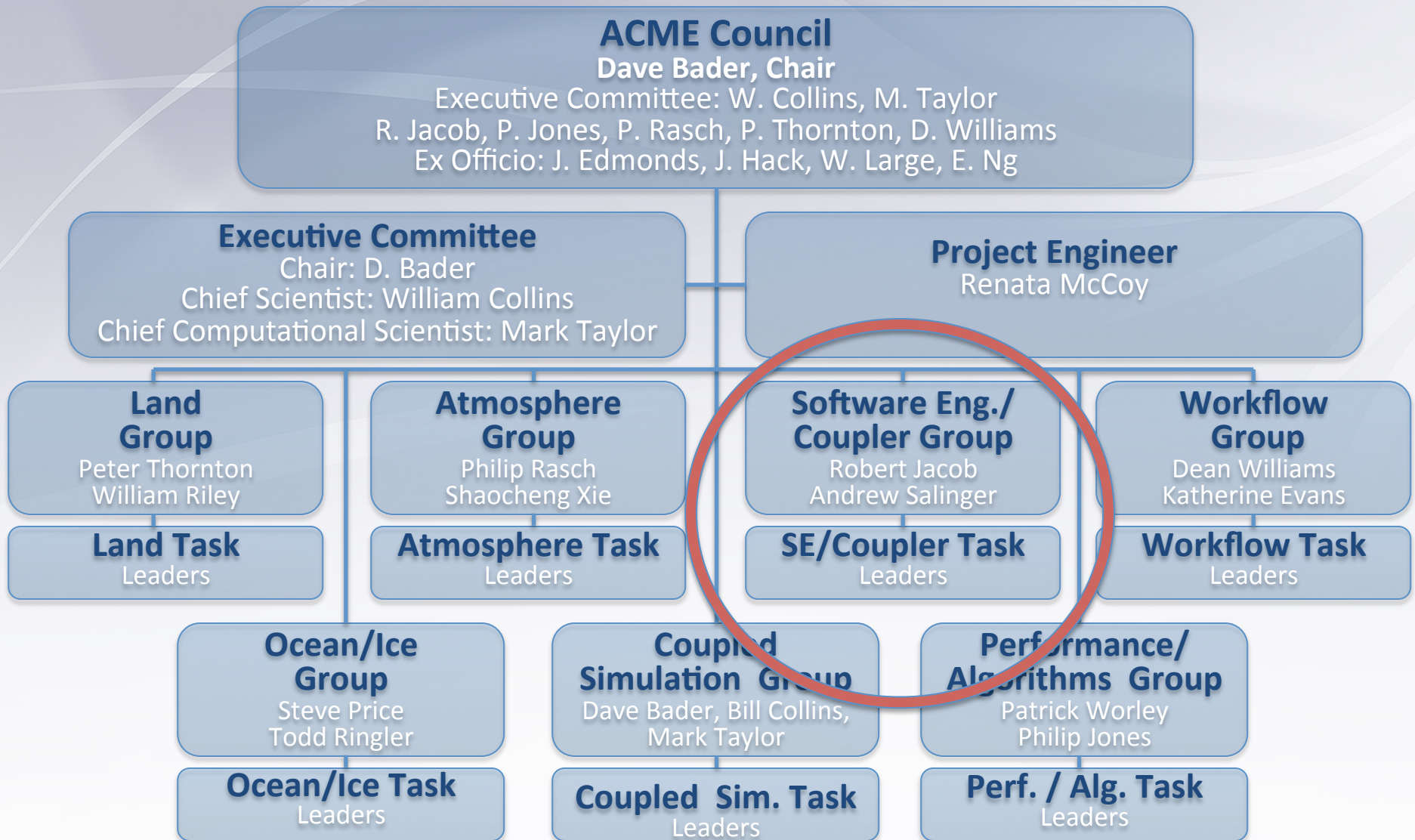


- IBM/NVIDIA Summit (OLCF)
 - 3,400 Power9 nodes
 - Multiple NVIDIA Volta GPUs per node
 - Approx 150PF
 - Production in 2018

- Intel/Cray Aurora (ALCF)
 - 50,000 Xeon Phi nodes (“Knight’s Hill”)
 - Approx 150PF
 - Production in 2019



ACME organization



ACME SE/CPL Group role (1 of 2)

Establishing, maintaining, and improving the software engineering tools, processes and designs used in ACME.

- **Repository**
 - Tools: source code control system, software access.
 - Processes and designs: code development workflow, code review
- **Testing**
 - Tools: automated testing, test suites
 - Processes and designs: continuous integration testing, system and unit testing procedures
- **Productivity**
 - Tools: wiki's and task-tracking tools.
 - Processes and designs: bug tracking, build system, programming standards

ACME SE/CPL Group role (2 of 2)

Develop and maintain inter-model infrastructure and top-level architecture of the ACME model.

- Coupler, main driver, other inter-model architecture
 - Modular interfaces and configurability of model
 - “shr” code
 - Model Coupling Toolkit
- I/O layers
 - Parallel I/O (PIO) sub-system
 - Component-level I/O systems

Work with science groups using “hub and spoke” model.

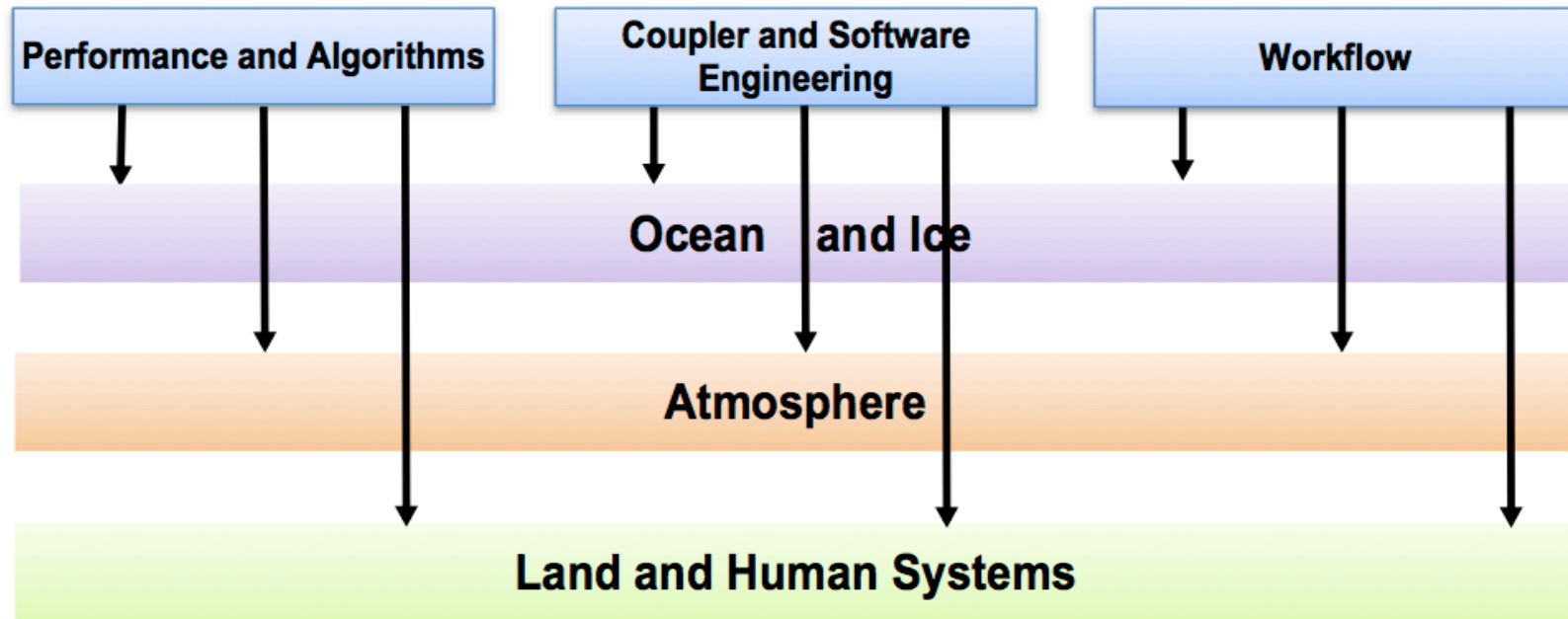


Figure 7.6-2: Computational Science is organized as hubs (the blue boxes) with spokes extending to the three Science areas to enable and facilitate communication and integration.

Spokes “improve SE consistency and accelerate adoption of new tools across science groups, as well as remove barriers that hinder sharing of technical advances across science model component groups.”

The ACME SE/CPL Group

Hub

- Jed Brown (ANL)
- Jim Foucar (SNL)
- Robert Jacob (ANL)
- Jeff Johnson (LBNL)
- Jayesh Krishna (ANL)
- Andy Salinger (SNL)
- Mark Taylor (SNL)

Spokes

- ATM: Balwinder Singh
- LND: Dali Wang, Gautum Bisht
- OCN/ICE: Doug Jacobsen*
- COUPLED: TBD

* = honorary Hub member

Notable help from: Pat Worley, Susannah Burrows, Renata McCoy

ACME Software Engineering Vision

- Testing, Testing, Testing
 - Code always runs on targeted platforms
 - System AND unit testing.
- Collaborative code development.
 - Everyone responsible for maintaining code base.
 - Code is a readable archive of what was done.
- Effective Communication / Management Tools.
- Leverage algorithms and software from ASCR (SciDAC) and other sources whenever possible.

Productivity tools: Selected Atlassian Confluence for project wiki

- <https://acme-climate.atlassian.net>
- Hosted solution so no need to worry about any lab's network access rules (but do have to pay).
- Good organization and commenting abilities. Every ACME group has a Confluence "space". Anyone can add and comment on pages.
- Rapidly and naturally replaced email lists as primary means of group communication.
- Maximizes visibility in to the project: any ACME group member can read/comment on material in any other group's "space".
- Helps geographically distributed team stay in touch.

Example page from ACME Confluence



Software Engineering/Coupler Group

Pages

Blog

SPACE SHORTCUTS

SE/CPL Meeting notes

SE/CPL Decision log

ACME Documentation

SE How-to articles

Shared links

Retrospectives

PAGE TREE

- Highlights
- Planning and Reporting for SE/CPL Group
- Configuration Management
- Help: Git and GitHub
- Help: Testing**
- SE/CPL Decision log
- Discussion Pages
- Shared links
- SE/CPL Meeting notes

Pages / Software Engineering/Coupler Group

Edit Watching Share ...

Help: Testing

Created by Robert Jacob, last modified on Apr 06, 2015

Use this page to ask questions about how to run or interpret output from the ACME tests.

DO NOT paste 100's of lines of output in to the discussion. Make a [gist](#) on github and include the link.

Like Be the first to like this

help

123 Comments



Robert Jacob

`.cs.status*` is reporting one PASS and multiple FAIL's in what look like identical cases:

```
PASS ERS.f19_g16_rx1.A.blues_intel.generate.ACMEv0.2-106-g85d996f-intel-mpich
FAIL ERS.f19_g16_rx1.A.blues_intel.generate.ACMEv0.2-106-g85d996f-intel-mpich
FAIL ERS.f19_g16_rx1.A.blues_intel.generate.ACMEv0.2-106-g85d996f-intel-mpich
FAIL ERS.f19_g16_rx1.A.blues_intel.generate.ACMEv0.2-106-g85d996f-intel-mpich
```

Turns out the "FAIL"s are all in the IOP versions of that case:

```
blogin3[95]: cat ERS_IOP.f19_g16_rx1.A.blues_intel.G.mpich/TestStatus
```

```
PASS ERS_IOP.f19_g16_rx1.A.blues_intel
PASS ERS_IOP.f19_g16_rx1.A.blues_intel.memleak
PASS ERS_IOP.f19_g16_rx1.A.blues_intel.generate.ACMEv0.2-106-g85d996f-intel-mpich
PASS ERS.f19_g16_rx1.A.blues_intel
```

Productivity tools: Selected Atlassian JIRA for day-to-day project tracking.

- All work in ACME should be associated with a task in JIRA.
- Simple life cycle: open, in-progress, done.
- As in Confluence, everyone can see and comment on any task. And also create and assign anyone a task.
- Tasks can have dependencies on other tasks, subtasks.
- ACME staff can log work on a task and comment on what was done.
- Using elements of Agile project management: sprints, retrospectives.

Repository: Chose git and github for source code control and repository

- ACME has a private organization on github with a mix of private and public repositories.
- <http://github.com/ACME-Climate/>
- Again, a hosted solution.
- Scalable to a large number of users.
- Comes with many features for tracking development
- Github “Issues” used for bug tracking.
- Repository initialized with cesm1_3_beta10.
- Entire code base in a single repository.

ACME uses “simplified gitworkflows” for development workflow

- 'master' is always stable and ready to release
- New features are complete and tested before appearing in 'master'
- Commits are minimal logically coherent, reviewable, and testable units
- Related commits go together so as to be reviewable and debuggable by specialist (by an “Integrator”)
- New development is not disrupted by others' features and bugs
- Rapid collaboration between developers possible
- `git log --first-parent` reads like a ChangeLog

Typical ACME development using gitworkflows:

- “git clone” the ACME repository.
- Create a branch with a specified naming convention:
 - <username>/<model>/<feature>
 - jgfouca/scripts/testing_enhancements
- Do work on your branch. Ignore changes to master while you’re working. Run tests on your branch.
- When done, submit a “Pull Request” in github.
- Integrator reviews code and moves it to a special branch called “next”
 - Many features stay in next while being tested together.
- When tests pass, branch “graduates” to master. Done!
- Used for all development.

ACME testing: Using the CESM testing system

- Added machine files for many DOE lab clusters.
- Defined new test suites: `acme_developer`, `acme_integration`
 - `acme_developer`:
 - should run on any platform used for development
 - used by developers before issuing a Pull Request on their new code.
 - Should run “fast”. Less than 1 hour.
 - `acme_integration`:
 - runs overnight on “next” branch.
- Continuous Integration: Run your test suite every time something is added to “next”. Expensive, so just run nightly on the 1-4 things that were added to next that day.

ACME testing with Jenkins



Jenkins


“The leading open-source continuous integration server. Built with Java, it provides 1054 plugins to support building and testing virtually any project.”

- Jenkins server (at Sandia) runs the test suites automatically at specific times on specified machines.
- Required creating numerous helper scripts (written in python) to interface with CESM test system.
- Test results are formatted and exported to Cdash.org for viewing.

ACME testing results at cdash.org

http://my.cdash.org/index.php?project=ACME_Climate (world readable)

My CDash All Dashboards Plans & Pricing Support Log Out Tuesday, May 05 2015 00:04:15 EDT









 **ACME_Climate**
board Calendar Previous Current Next Project Settings



No file changed as of **Wednesday, April 29 2015 - 21:00 EDT**

- 4 days ago: 10 tests not run on acme_integration_master_intel
- 5 days ago: 10 tests not run on acme_integration_master_intel
- 6 days ago: 1 test failed on regression_test
- 7 days ago: 1 test failed on regression_test
- 7 days ago: 1 test failed on regression_test

[See full feed](#)

Nightly

Site	Build Name	Update		Configure		Build		Test			Build Time
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass		
redsky	 acme_integration_master_intel 						10	0	21	Apr 30, 2015 - 04:26 EDT	
melvin	 acme_developer_master_gnu 						0	0	15	Apr 29, 2015 - 22:39 EDT	
melvin	 acme_developer_next_gnu 						0	0	15	Apr 30, 2015 - 05:05 EDT	
skybridge	 acme_integration_next_intel 						0	0	31	Apr 30, 2015 - 03:56 EDT	

  CDashPro 2.3.0 © Kitware | [Report problems](#) | [Privacy Policy](#) | 0.059s

Other ACME SE/CPL Progress: Parallel I/O

- Introduced new run-time configuration options in PIO1.
- Will re-implement in PIO2 and add PIO2 to ACME (assuming no loss in performance).
- Added test cases that reflect targeted ACME resolutions, number of tracers.
- Collaboration with IBM: Modifications to MPI-IO for performance based on test cases.

Other ACME SE/CPL Progress: New MCT Release!

- Support for NAG 6.0 and Mac
- Bug fixes including ones found by valgrind (many thanks to NCAR's Sean Santos for above)
- New features to output Router and GSMap info from full model to build communication test programs.
- mpi-serial 2.0
 - a small single-node MPI replacement library.
 - Not a stub-library MPI_Send/Recv really copies data.
 - In 2.0
 - Many more MPI datatypes/functions added.
 - Self-contained build system (autoconf)
 - Developed by ALCF's Raymond Loy
- MCT 2.9 out tomorrow!



Plans for the rest CY15

- Adopt CIME
- Adopt PIO2
- Expand test coverage
 - Run nightly on LCF's
 - More ACME-specific compsets
- New coupler capabilities
 - Ocean – land-ice coupling (Jeremy Fyke)
 - Support for sub-grid orography (many people)
- Improve coupler performance (for initialization and communication)

More information

<http://climatemodeling.science.energy.gov/projects/accelerated-climate-modeling-energy>



Accelerated Climate Modeling
for Energy



<http://www.mcs.anl.gov/mct/>