MPAS
Model for Prediction Across Scales

MPAS Development: Using git and GitHub

MPAS
Model for Prediction Across Scales

Motivation

The MPAS code – as well as all tools used to set up simulations and to post-process model output – are distributed through GitHub

- You may suspect that GitHub is more than just a means of distributing source code for MPAS

Git and GitHub are not only used for distributing source code, but are central to the *open development* of the MPAS model and supporting tools

MPAS
Model for Prediction Across Scales

Outline

This talk aims to give a *very* brief introduction to our development process, which may be helpful if you should ever want to:

- Report a bug in the code
- Contribute code changes

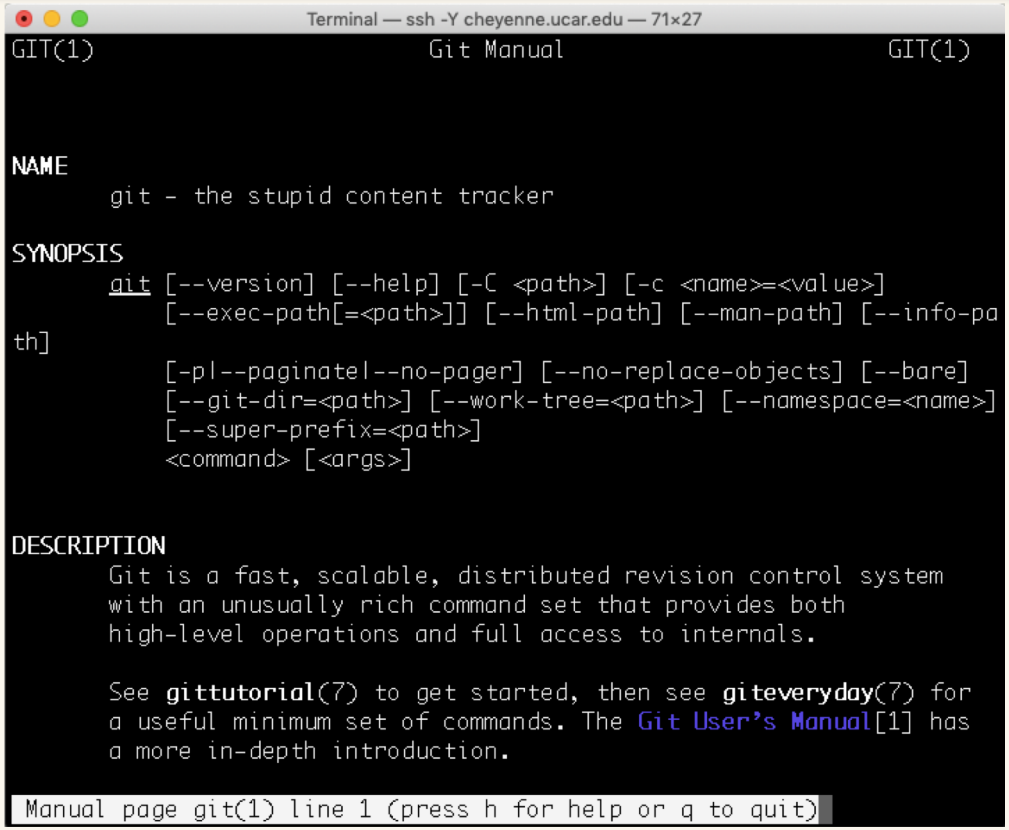
Outline of this talk

1. A quick summary of *git*
2. An introduction to GitHub
3. A high-level view of MPAS development process
4. What to do if you find a bug
5. How to contribute code changes

Git: The stupid content tracker

Git is an open-source, distributed version control system that supports a variety of workflows

- Most of us simply use the git command-line tools



Left: The first page of the git *man* page.

Git: commits and branches

Two of the key concepts in git (and version control in general) are:

- Commits
- Branches

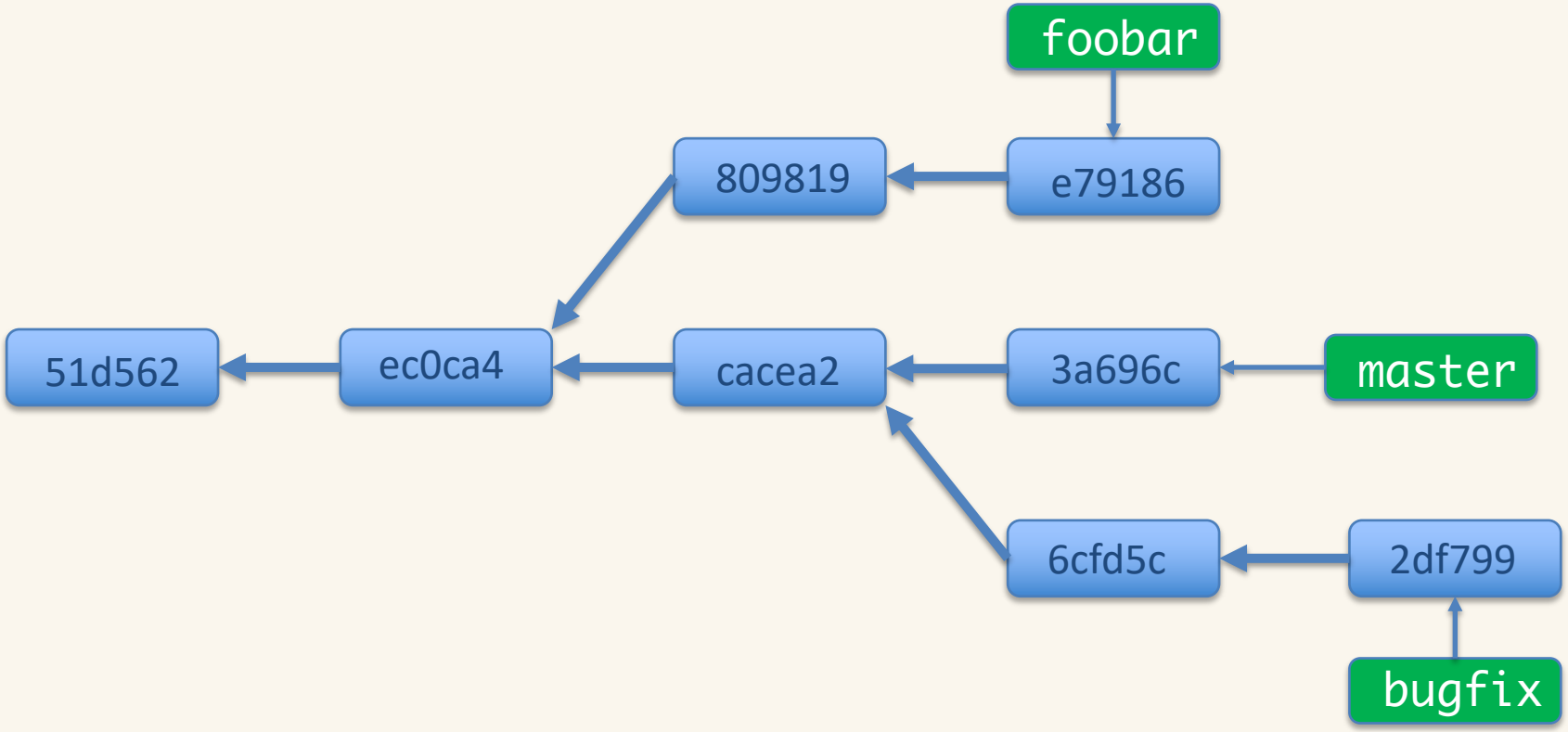
In a git repository, each commit consists of changes to one more files, and each commit has one or more *parent* commits

- Each commit is uniquely identified by a *hash*

A git *branch* is a pointer to a commit that can be updated as new commits are made

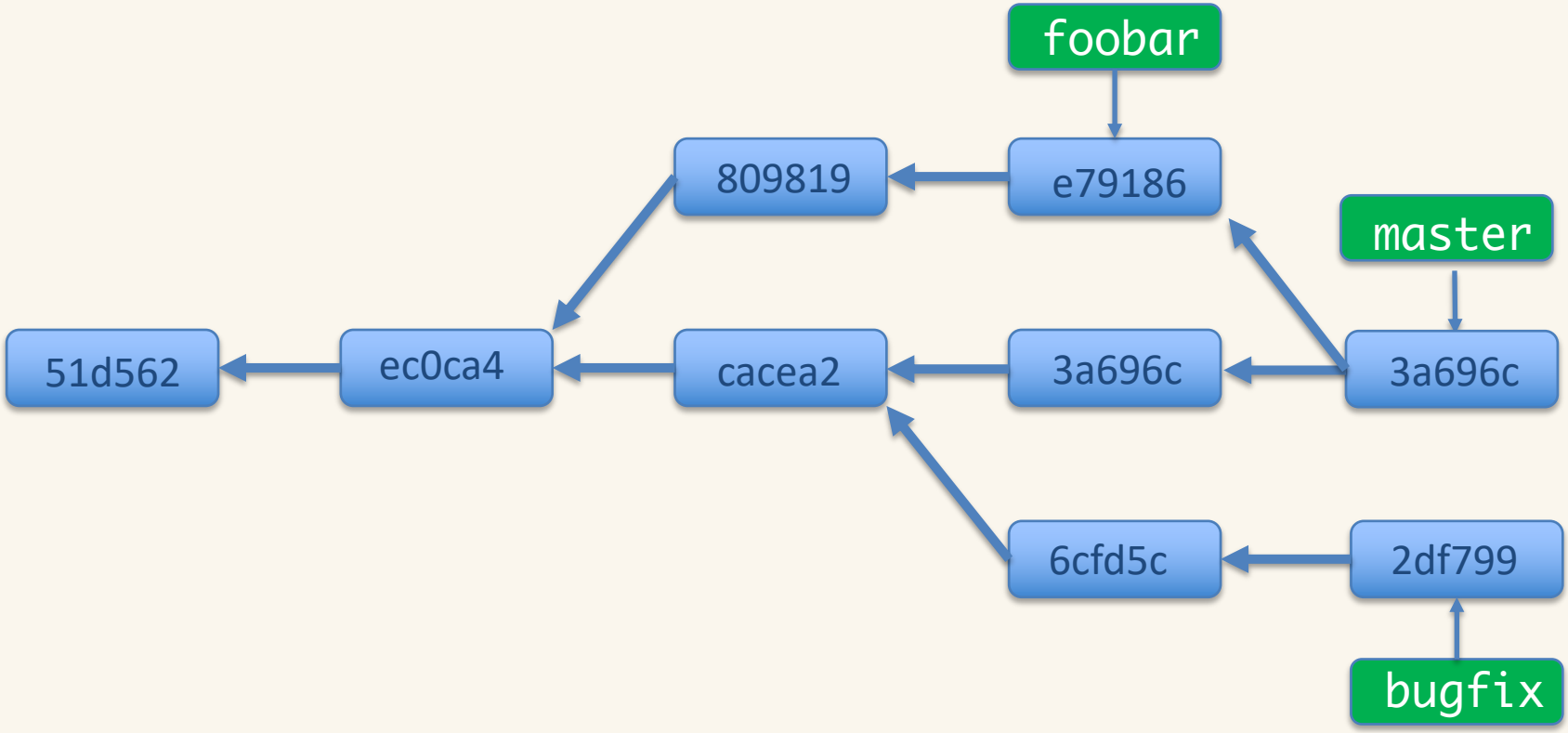
It can be helpful to visualize git repositories as directed, acyclic graphs!

Git: commits and branches



Above: A cartoon depiction of a git repository with three branches (foobar, master, and bugfix) and eight commits (f1d562, ec0ca4, etc.)

Git: merging branches



Above: The changes from the “foobar” branch have been *merged* onto the “master” branch.

MPAS
Model for Prediction Across Scales

GitHub: Hosting for git repositories, plus…

GitHub provides hosting for git repositories, plus helpful features:

- Issue tracking
- Pull Requests
- Forks
- Wikis, task management, feature requests, and a few other things that MPAS-Atmosphere developers don't tend to use much…

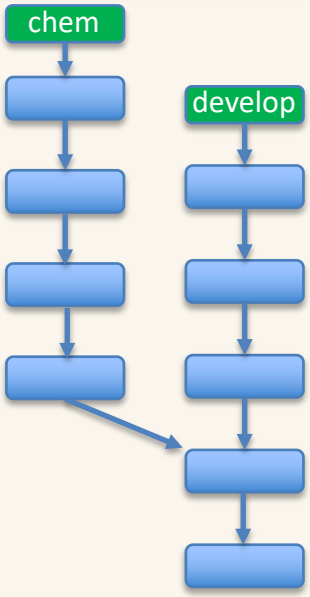
Pull Requests provide a web-based mechanism to request that one branch be merged into another

- Reviewers can be assigned
- Discussion of the changes can take place
- Approval of the changes is made explicit

GitHub: Hosting for git repositories, plus…

Forks are *clones* of a repository on GitHub in which an individual can create their own branches, and independently do development work

- Importantly, *Pull Requests* may be used to merge branches across forks of a repository



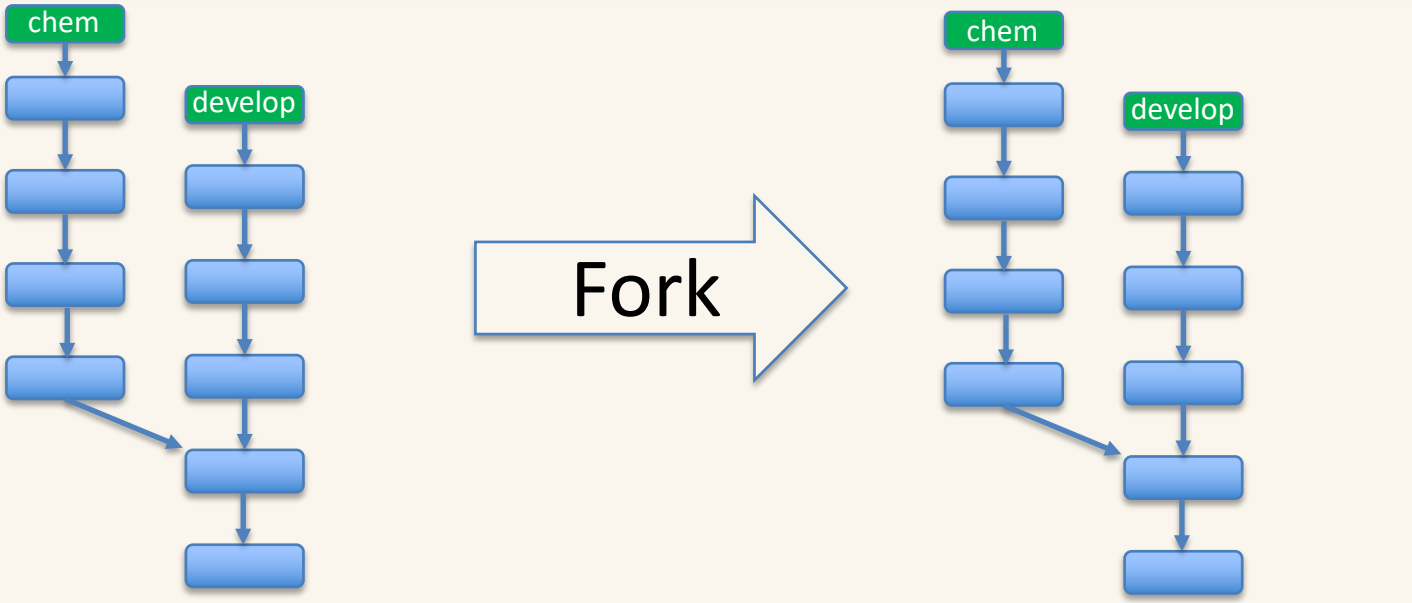
MPAS-Dev/MPAS-Model

mgduda/MPAS-Model

GitHub: Hosting for git repositories, plus…

Forks are *clones* of a repository on GitHub in which an individual can create their own branches, and independently do development work

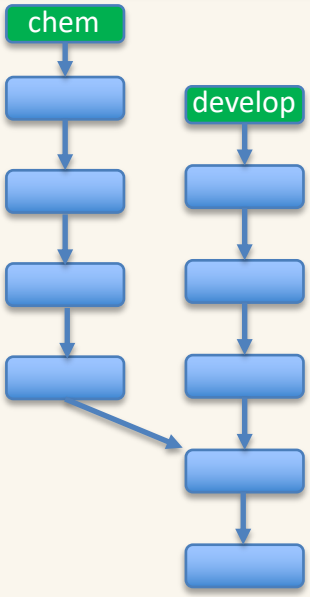
- Importantly, *Pull Requests* may be used to merge branches across forks of a repository



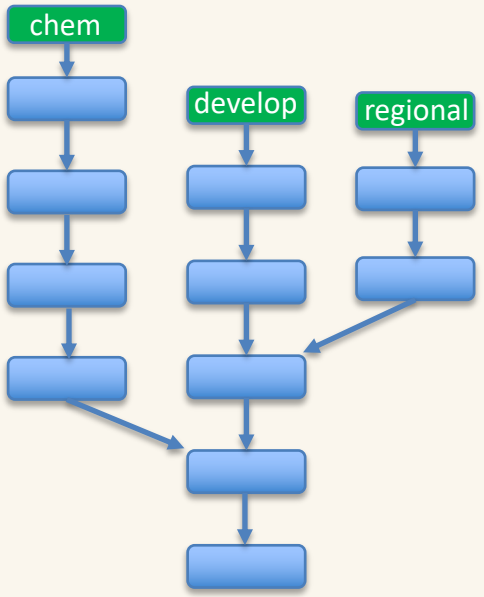
GitHub: Hosting for git repositories, plus…

Forks are *clones* of a repository on GitHub in which an individual can create their own branches, and independently do development work

- Importantly, *Pull Requests* may be used to merge branches across forks of a repository



MPAS-Dev/MPAS-Model

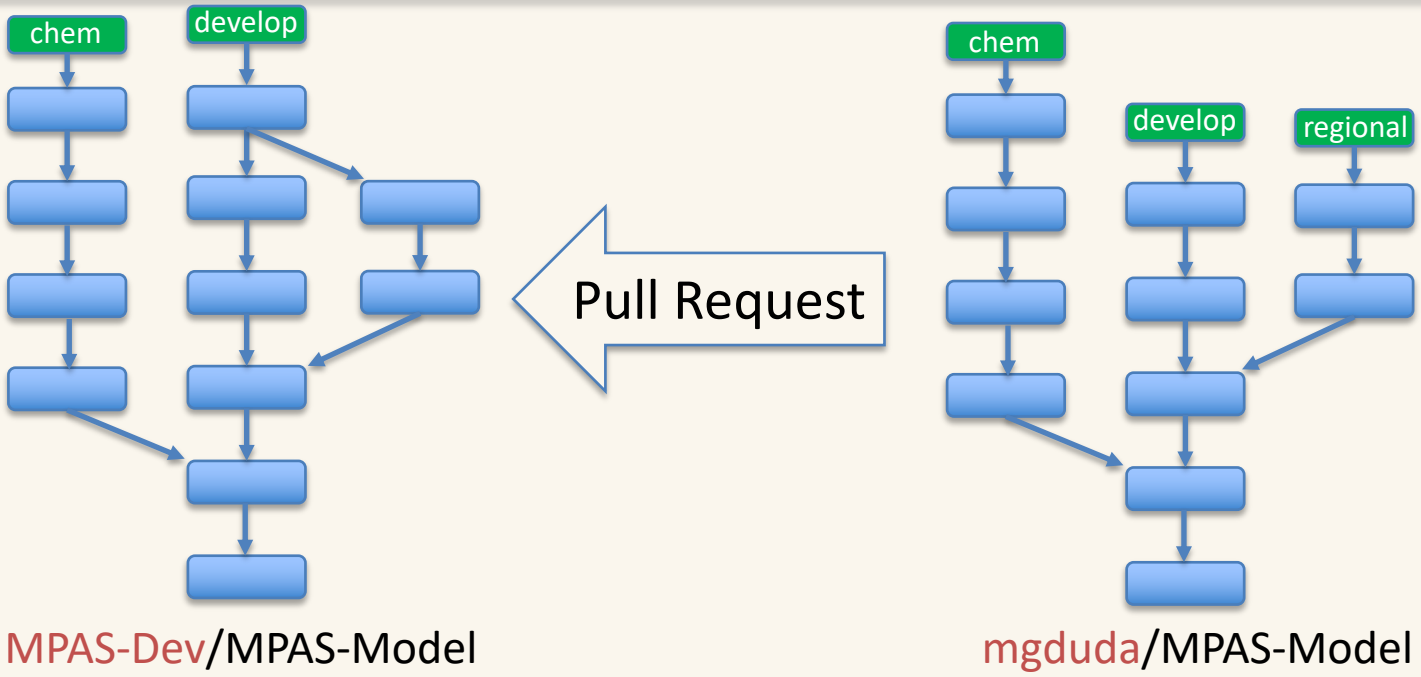


mgduda/MPAS-Model

GitHub: Hosting for git repositories, plus…

Forks are *clones* of a repository on GitHub in which an individual can create their own branches, and independently do development work

- Importantly, *Pull Requests* may be used to merge branches across forks of a repository



MPAS
Model for Prediction Across Scales

GitHub: Hosting for git repositories, plus…

Let's look at some Pull Requests, a fork, and some branches in a web browser…

MPAS
Model for Prediction Across Scales

Reporting bugs in MPAS-Atmosphere

First, make a new post to the forum… often times, so-called "bugs" are actually not bugs

MPAS
Model for Prediction Across Scales

Contributing code changes

Talk to us first to get some guidance…

Summary

1. The development of all MPAS model and supporting tools/utilities uses *git* repositories hosted on GitHub
2. All code changes begin as *branches*, which are proposed for inclusion in the main development branch (and later, to a released version of the code) through *pull requests*
3. Each developer has their own *fork* of the MPAS repository, in which they do their individual development
4. If you suspect you've found a bug, begin by posting a question to the MPAS-Atmosphere support forum; if it is a bug, a GitHub *Issue* will be opened
   - http://forum.mmm.ucar.edu/
5. If you'd like to contribute to MPAS-Atmosphere, begin by creating a *fork* of the MPAS-Model repository, and contact us!