Science Computing Carpentry

Shelley Knuth
shelley.knuth@colorado.edu
http://goo.gl/forms/8VidcwOhRT

www.rc.colorado.edu

Outline

- Basis for rest of the workshop
- Topics
  - Working with shared file systems (supercomputers)
    - Job submission
    - Differing nodes
    - What not to do
  - Version Control/Git
    - What is version control?
    - What is git?
    - Let's manage some code!

What Is a Supercomputer?

- A supercomputer is one large computer made up of many smaller computers and processors
- Each different computer is called a **node**
- Each node has processors/cores
  - Carry out the instructions of the computer
- With a supercomputer, all these different computers talk to each other through a communications network
  - Example - InfiniBand

Different Node Types

- Login nodes
  - This is where you are when you log in
  - No heavy computation, interactive jobs, or long running processes
  - Script or code editing, minor compiling
  - Job submission
- Compute/batch nodes
  - This is where jobs that are submitted through the scheduler run
  - Intended for heavy computation

What is Job Scheduling

- Supercomputers usually consist of many nodes
- Users submit jobs that may run on one or multiple nodes
- Sometimes these jobs are very large; sometimes there are many small jobs
- Need software that will distribute the jobs appropriately
  - Make sure the job requirements are met
    - Reserve nodes until enough are available to run a job
    - Account for offline nodes
- Also need software to manage the resources
- Integrated with scheduler

Job Scheduling

- On a supercomputer, jobs are scheduled rather than just run instantly at the command line
  - People “buy” time to use the resources
  - Shared system
  - Request the amount of resources needed and for how long
  - Jobs are put in a queue until resources are available
  - Once the job is run they are “charged” for the time they used

Job Scheduling - Priority

- What jobs receive priority?
  - Can depend on the center
  - Can arrange for certain people who “pay more” receive priority
  - Generally though based on job size and time of entry
- Might have different queues based on different job needs
- Can receive priority on a job by creating a reservation

Job Schedulers

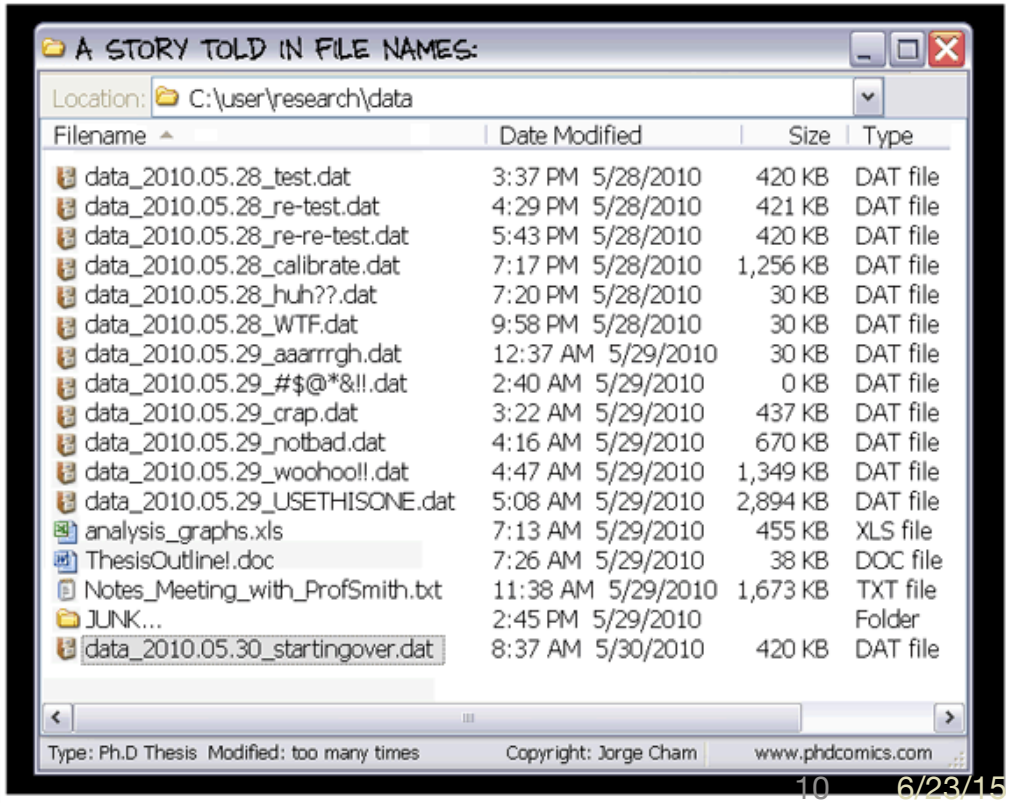
- Jobs on supercomputers are managed and run by different software
  - Some systems have jobs submitted using Torque and scheduled with Moab
  - Some use Slurm
  - SLURM = Simple Linux Utility for Resource Management
    - Open source
    - Increasingly popular

What is Version Control?

- Version control records changes to a file or to code
- Keeps track of:
  - What changes are made
  - Who makes them
- By keeping track of changes, you can recall previous versions later

Version Control and You

- Probably already using version control
- Rename files
- Copy into another directory
- Easy to overwrite
- Which version is newest?

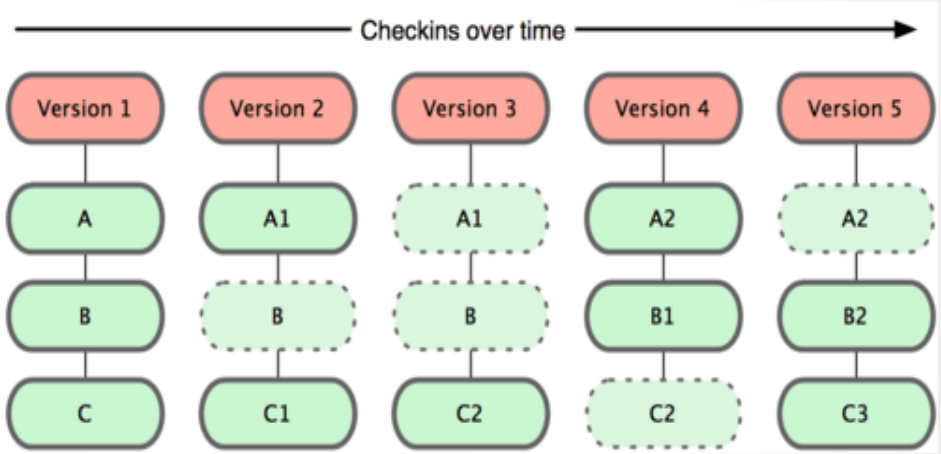


Version Control Advantages

- Revert files back to a previous state
- Easy recovery
- Multiple backups
- See who might have changed a file that is causing issues
  - History of changes by who and when
  - More efficient tracking of documents, code

Distributed Version Control Systems

- Git is a distributed version control system
- Instead of checking out the latest version of the file, checks out a snapshot of all files at that time
- Easier to recover from loss
- Only needs local resources to operate, generally



git-scm.com

How Does Git Operate? – Typical Work Flow

- General work flow of a project:
  - Create a file
  - Make changes to the file
  - Save the changes to the file
  - Make more changes
  - …
- How does this fit into git?

How Does Git Operate? – Typical Work Flow in Git

- **Start tracking a project**
- Create a file
- **Track the file**
- Make changes to the file
- Save the changes to the file
- **Track those changes**
- Make more changes
- …
- **Commit changes to a local repository**
- **Push to a remote repository**

Git Lingo

- **Repository**
  - A folder that houses all the files that are part of a project
  - Files are tracked or untracked
- **Tracked**
  - Git is monitoring a file as part of a project
- **Commit**
  - A snapshot of the project at a single point in time

Let's Work on Some Code

- git_tutorial_usgs2015.ipynb

Remote Repositories

- One of the great things about version control systems is that you can push/pull changes to/from remote locations
- This allows many things:
  - Making public something really great you've done!
  - Grabbing something really great that's been done!
  - Group collaboration on a specific project

Getting Another Git Repository

- Get a repository from either copying one that exists or importing an existing project to git
- You import the existing project based on what we've already done
- To copy an existing git repository, type

git clone https://github.com/ResearchComputing/USGS_2015_06_23-25.git

- Receive a copy of all the data the server has
- Not a checkout of the latest version of code

Remote Git Repositories

- You never write directly to a remote repository
- You need to create a corresponding local branch which tracks the remote branch
- You can push or pull changes to or from the remote branch

git pull

git push

- Usually you have a public repository where everyone pulls changes from
- You push your changes there

Questions?

- git-scm.com (Lots of great information in this talk was found here!)
- try.github.io/levels/1/challenges/1 (Learn git in 15 minutes!)
- Git.or.cz/course/svn.html

- Shelley.Knuth@Colorado.edu
- Survey: http://goo.gl/forms/8VidcwOhRT