Introduction to Blanca

Andrew Monaghan & Dan Trahan
rc-help@colorado.edu

www.colorado.edu/rc

Slides available for download from:
https://github.com/ResearchComputing/CHANGE_2019

Agenda for Today

- 10:00: Intro to RC, Blanca, Linux (this presentation)
- 11:00: Intro to job scripts and job submission on
- 12:00: Lunch
- 1:00: Using python on Blanca
- 1:30: Using R, Matlab on Blanca
- 2:00: Using X11 and VNC on Blanca (for graphics)
- 2:30: Other topics (data transfer, job arrays, getting help)
- 3:00: Adjourn (we will be on hand to help after this)

Outline for this presentation

- Overview of RC, Blanca
- Logging in
- Basic Linux commands
- File editing
- Linux filesystem
- Environment variables
- Software modules on Blanca
- Other Linux topics (environment variables, modes, wildcards)

What is Research Computing?

- Provide services for researchers that include:
  - Large scale computing
  - Data storage
  - High speed data transfer
  - Data management support
  - Consulting
  - Training

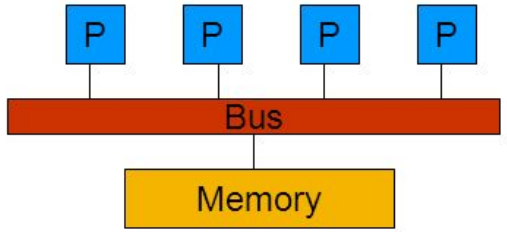
- We are likely best known for:
  - Summit Supercomputer (~12,000 cores)
  - Blanca ”condo” cluster (~2,000 cores)
  - PetaLibrary storage

What Would I Use Blanca For?

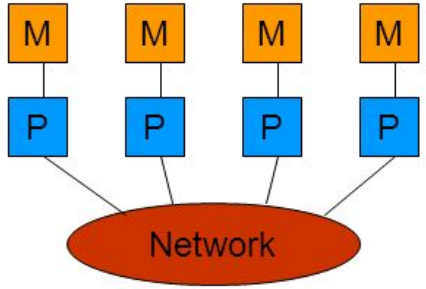
Solving large problems that require more:

- Memory than you have on your personal computer
- Cores/nodes/power than you have on your personal computer

- Blanca is set up for shared memory (single node) parallelization.
  - Use Summit for distributed memory parallelization



- **Shared memory**



- **Distributed memory**

Source: https://images.slideplayer.com/25/7599921/slides/slide_4.jpg

Blanca

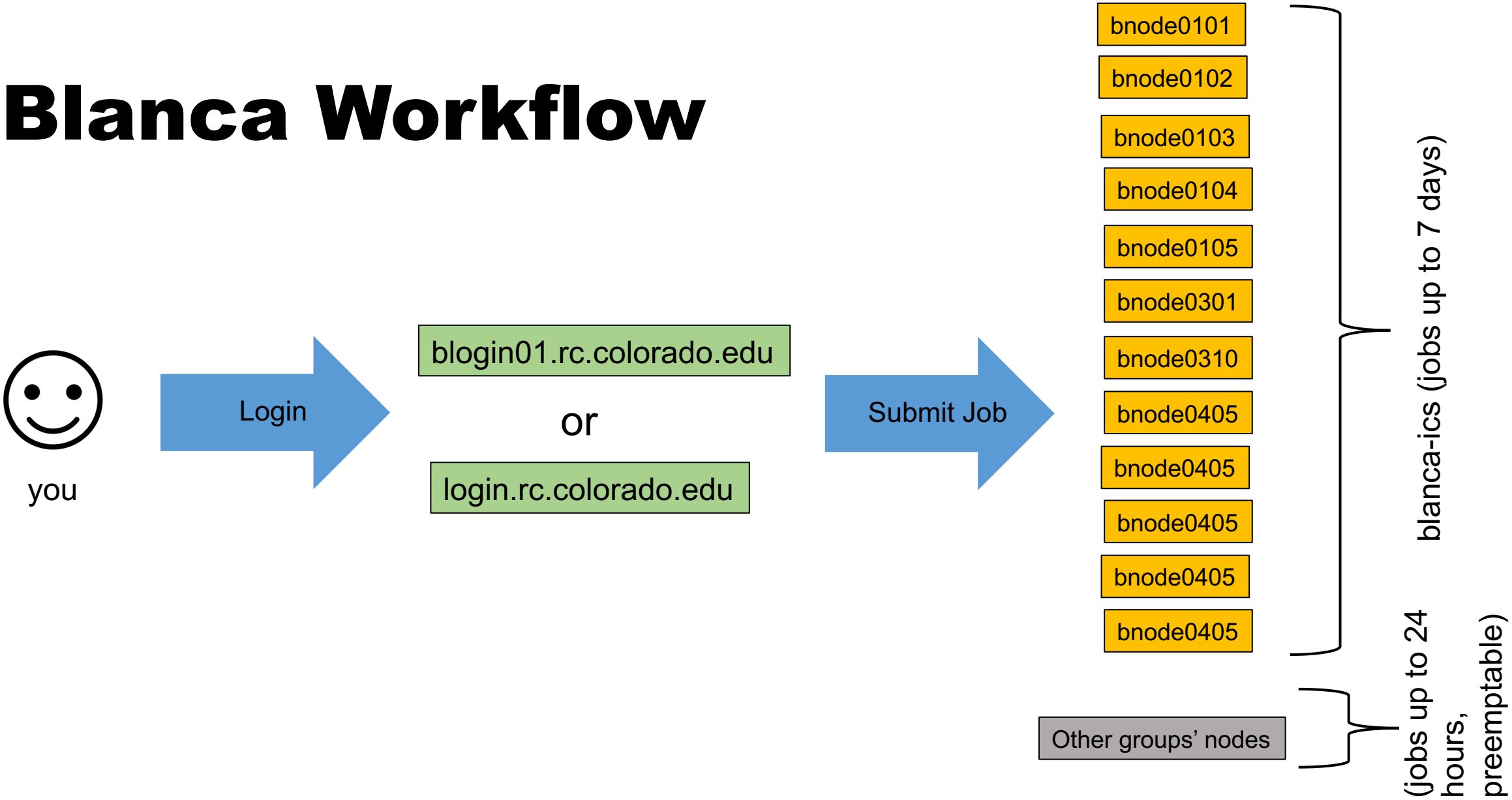
- A “condo” cluster whereby individual research groups own nodes
- List of nodes and groups can be found here
- Users have dedicated access to their group’s nodes (e.g., blanca-ics)
  - Jobs up to 7 days long.
  - Can also run ‘preemptable’ jobs on other groups nodes (jobs up to 24 hours long)

- More documentation on Blanca: https://curc.readthedocs.io/en/latest/access/blanca.html

Blanca ICS nodes

- bnode0101-bnode0105 (5 nodes)
  - 32 cores, avx, sandybridge, 250 GB RAM (7.8 GB/core)
- bnode0301, bnode0310 (2 nodes)
  - 32 cores, avx2, Broadwell, 250 GB RAM (7. GB/core)
- bnode0405-bnode0409 (5 nodes)
  - 28 cores, avx2, Broadwell, 250 GB RAM (8.9 GB/core)

Blanca Workflow



Logging In

- `ssh –X identikey@blogin01.rc.colorado.edu` (Blanca)
  - Can also use `login.rc.colorado.edu` (any RC resource)
- Enter your `identikey_password`
- Authenticate by accepting the Duo push to your smartphone
  - Can also authenticate by text message, phone call, or token
- More info here: https://curc.readthedocs.io/en/latest/access/logging-in.html

Linux

- Part of the Unix-like family of operating systems.
- Started in early ‘90s
- Several distributions are available – from enterprise-grade, like RedHat Linux (RHEL), to more consumer-focused, like Ubuntu.
  - Blanca nodes presently run RHEL7
- Runs on everything from embedded systems to supercomputers.
- Linux is simple, flexible, fast, many potent tools

Anatomy of a Linux command

- command [flags] [flag arguments] [target(s)]
  - ls -l myworkdir/
- Case is important!
- Help on commands is available through the ”man” command (short for manual). E.g.,
  - man ls

File and directory related commands

- **pwd** – prints full path to current directory
- **cd** – changes directory; can use full or relative path as target
- **mkdir** – creates a subdirectory in the current directory
- **rmdir** – removes an empty directory
- **rm** – removes a file (`rm -r` removes a directory and all of its contents)
- **cp** – copies a file
- **mv** – moves (or renames) a file or directory
- **ls** – lists the contents of a directory (`ls -l` gives detailed listing)
- **chmod/chown** – change permissions or ownership
- **df** – displays filesystems and their sizes
- **du** – shows disk usage (`du -skh` shows size of a directory and all of its contents in KB and human readable)

Process and Program related commands

- **ps** – lists processes (`ps -ef` lists all running processes)
- **top** – shows processes currently using the CPU
- **kill** – sends a signal to a process (kills process by default). Target is Process-ID; found in 2nd column of `ps -ef` output.
- **time** – shows how much wall time and CPU time a process has used
- **free** – memory usage

File-viewing commands

- **more** – displays a file one screen at a time
- **cat** – prints entire file to the screen
- **head** – prints the first few lines of a file
- **tail** – prints the last few lines of a file (with -f shows in real time the end of a file that may be changing)
- **diff** – shows differences between two files
- **grep** – prints lines containing a string or other regular expression (ps –ef | grep XX)
- **tee** – prints the output of a command and also copies the output to a file
- **sort** – sorts lines in a file
- **find** – searches for files that meet specified criteria
- **wc** – count words, lines, or characters in a file

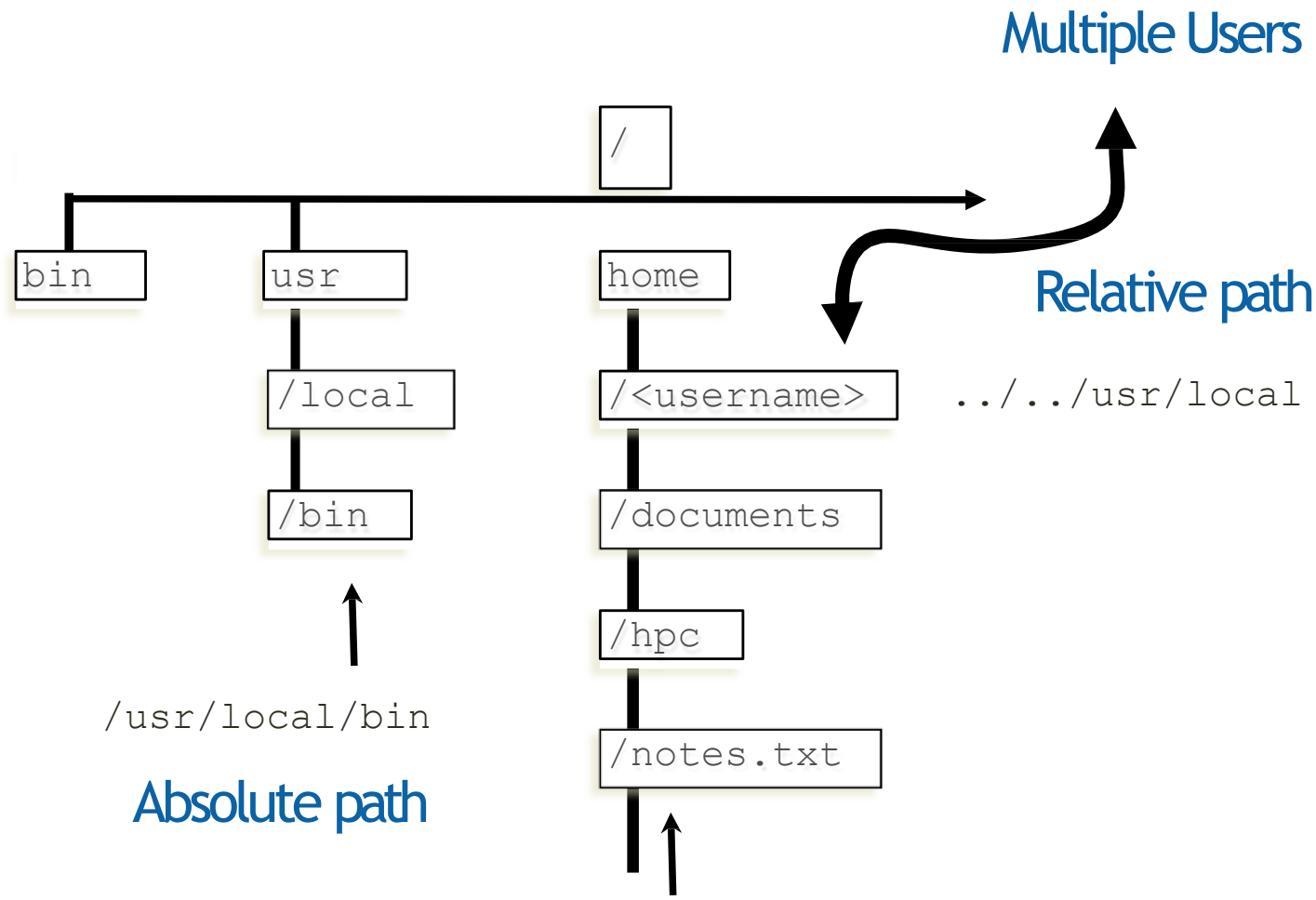
File editing with nano

- To edit a file:
  - nano myfile.txt
- From within Nano:
  - Ctrl+o save (need to confirm filename)
  - Ctrl+x exit
  - Ctrl+k cut
  - Ctrl+u paste
- Other population Linux editors: vi, emacs

The Linux Filesystem

- System of arranging files on disk
- Consists of directories (folders) that can contain files or other directories
- Levels in full paths separated by *forward* slashes, e.g.
- /home/user/scripts/analyze_data.sh
- Case-sensitive; spaces in names discouraged
- Some shorthand:
  - `.` (the current directory)
  - `..` (the directory one level above)
  - `~` (home directory)
  - `-` (previous directory, when used with `cd`)

Filesystem


Multiple Users
/
bin
usr
home
Relative path
/local
/<username>
../../usr/local
/bin
/documents
/hpc
/usr/local/bin
/notes.txt
Absolute path
/home/<username>/documents/hpc/notes.txt

Your personal directories on Blanca

- /home/<username>
  - Very small: 2GB.
  - Backed up daily.
  - Good for ‘can’t lose’ files
- /projects/<username>
  - 250 GB
  - Backed up regularly
  - Good for storing scripts, self-installed software, some data
- /rc_scratch/<username>
  - Group-shared 40 TB partition.
  - Good for jobs with lots of I/O
  - Not backed up
  - Temporary: data deleted 90 days from creation.

Environment variables

- Environment variables store important information needed by Linux users, programs, etc.

- Type ‘env’ to see your currently set environment variables

- Useful Environment variables:
  - `PATH`: directories to search for commands
  - `HOME`: home directory
  - `PWD`: current working directory
  - `USER`: username
  - `LD_LIBRARY_PATH`: directories to search for shared objects (dynamically-loaded libs)

Software

- Common software is available to everyone on the systems
- Can install your own software
  - But you are responsible for support
  - We are happy to assist
- RC uses modules to manage software
  - You can load modules to prepare your environment for using software
    - Loading sets any environment variables
    - Enables application to find appropriate libraries, etc.

Using Modules

- Some modules might require a specific hierarchy to load
  - For some modules, you may need to specify a specific version
    - For example, `module load R/3.3.0`
  - For other modules, you may be able to be more generic
    - For example, `module load matlab`
- Some modules may require you to first load other modules that they depend on
- To find dependencies for a module, type `module spider <package>`
- To find out what software is available, you can type `module avail`
- To set up your environment to use a software package, type `module load <package>/<version>`

Modes (aka permissions)

- Three classes of users:
  - User (u) aka “owner”
  - Group (g)
  - Other (o)
- Three types of permissions
  - Read (r)
  - Write (w)
  - Execute (x)

.. own grp oth

-|---|---|---

`drwxr-xr--`

Modes (continued)

- `chmod` changes modes:
- To add write and execute permission for your group:

```
chmod g+wx filename
```

- To remove execute permission for others:

```
chmod o-x filename
```

- To set only read and execute for your group and others:

```
chmod go=rx filename
```

Shell Wildcards and Special Characters

- `*` - matches zero or more characters
- `?` - matches a single character
- `#` - comment; rest of the line is ignored
- `\` - escape; don't interpret the next character

End of first module

- We'd love your feedback: http://tinyurl.com/curc-survey18
- Course materials for today:
  https://github.com/ResearchComputing/CHANGE_2019
- Blanca (and other) documentation
  https://curc.readthedocs.io/en/latest/access/blanca.html
- More detailed tutorials from our "HPC Fundamentals" course:
  https://github.com/ResearchComputing/Fundamentals_HPC_Spring_2019