Filesystems and Storage on RMACC Summit

Research Computing
UNIVERSITY OF COLORADO **BOULDER**

Be Boulder.

Filesystems and Storage on RMACC Summit

- Mea Trahan
- *Email: Daniel.Trahan@Colorado.edu*

- *RC Homepage: https://www.colorado.edu/rc*
- *RC Email: rc-help@colorado.edu*

- Slides available for download at:
- *https://github.com/ResearchComputing/Filesystems_And_Storage_Fall_2020*

Outline

- Overview of Summit's storage
- Summit architecture and filesystems
- Petalibrary
- Data transfers and tools

Quick note

- Clusters come in all shapes and sizes, so much of the information here may not apply to other systems. HPC is a very diverse landscape so make sure you check with the system administrators of whichever cluster you are using.

Overview of CURC's directories

- 3 major user directories
  - Home – Used for reusable job scripts, setting files, and other important small files.
  - Projects – Used for application and small datasets.
  - Scratch – Work directory. Used with jobs for highspeed access to data or output.
- Table:

	Directory	Capacity	Backup	Purge
Home	/home/$USER	2 GB	2 hours for 7 days	Never
Projects	/projects/$USER	250 GB	6 hours for 7 days	Never
Scratch	/scratch/summit/$USER	10 TB	(none)	90 days

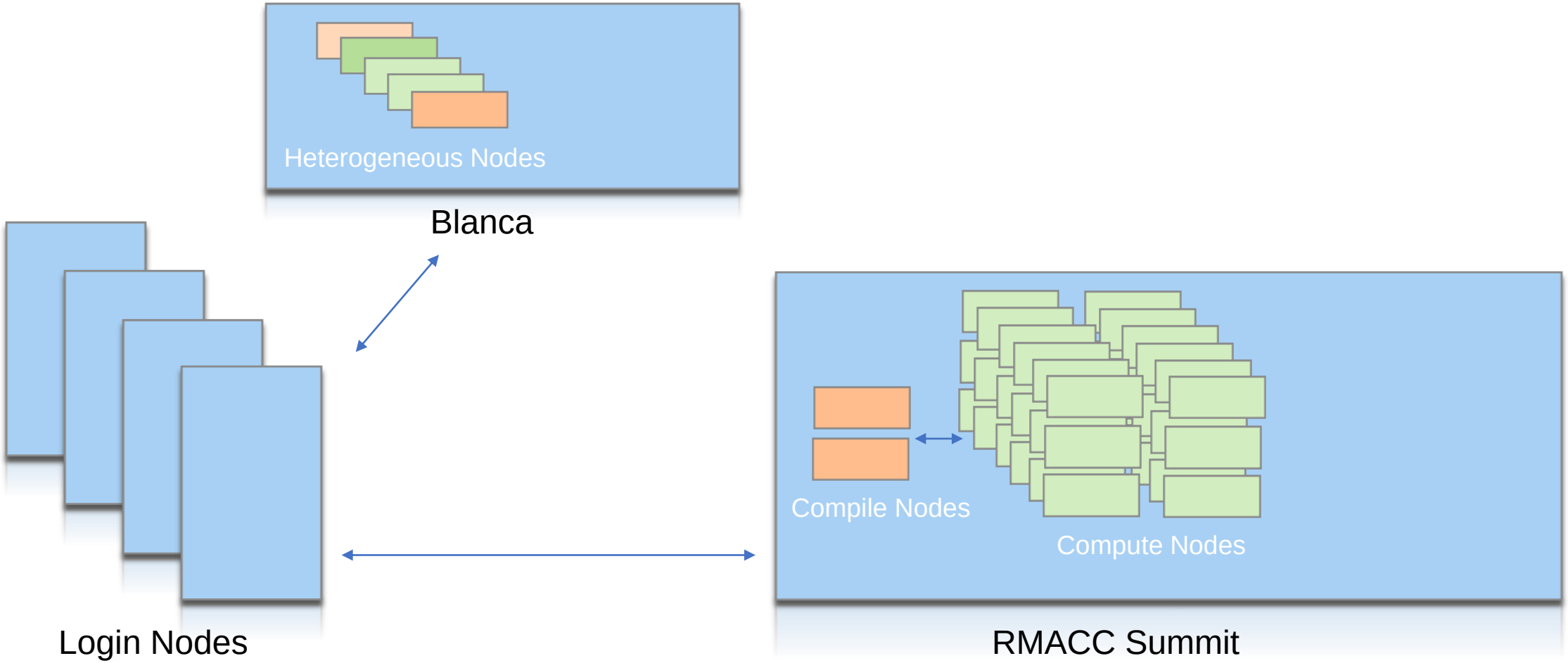
What's makes Summit different?

- As you may know Research Computing resources diverge a bit from the average computer.
  - Complex severs architecture connected to two separate clusters
  - Various endpoints and services
  - Several Filesystems for various purposes.
- When you log into Summit all of this seems "seamless…"
- So what gives?

RC's network

- The user facing side of RC's network is composed of 3 major structures:
- Login Nodes
  - VMs with very little compute capability
  - Can access RMACC Summit compile nodes by running `ssh scompile` with the `slurm/summit` module loaded.
  - Can access Blanca by loading `slurm/blanca` and submitting an interactive job.
- Summit
  - CU Boulder high performance computing resource.
  - Composed of forward-facing 483 compute nodes and 2 compile nodes.
- Blanca
  - Buy in condo nodes that compose a heterogeneous compute cluster.
  - Check with your group if you'd like access to this resource.

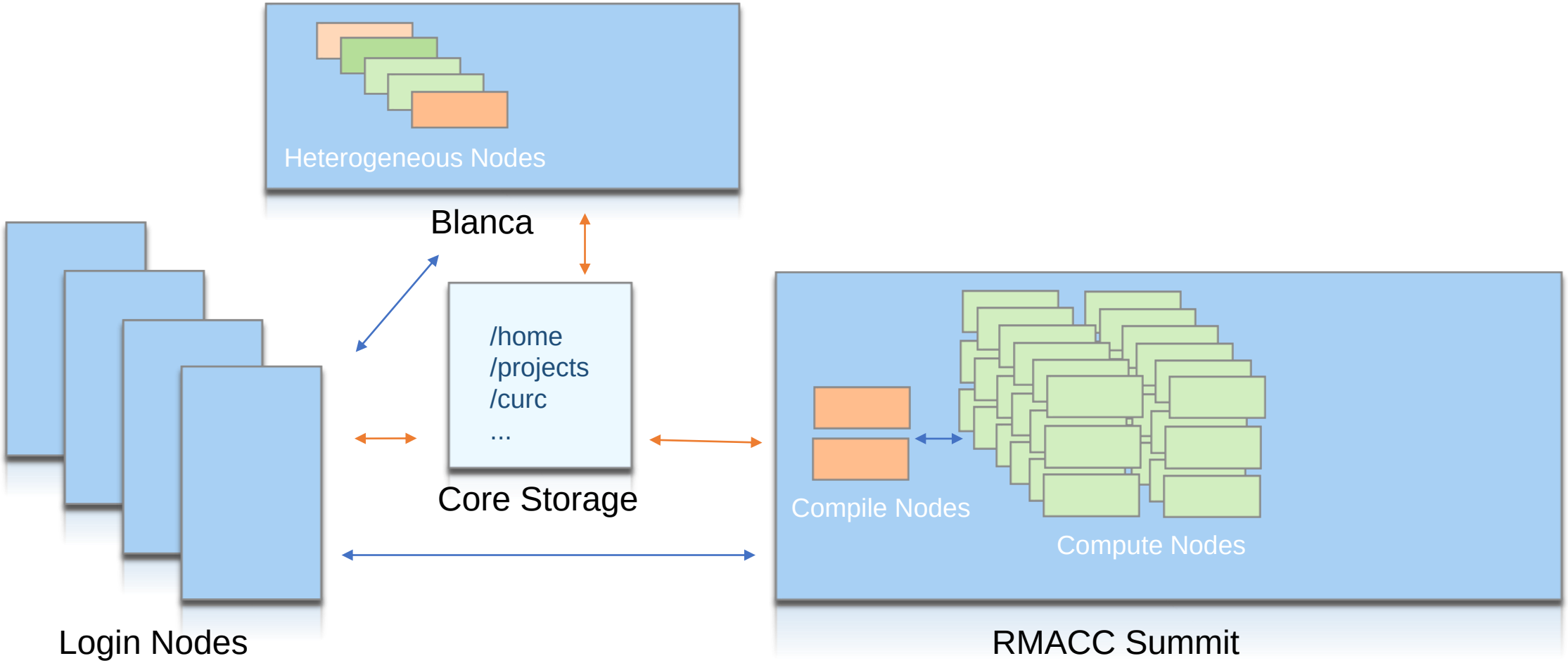
RC's network


Heterogeneous Nodes
Blanca
Compile Nodes
Compute Nodes
Login Nodes
RMACC Summit

RC's filesystems

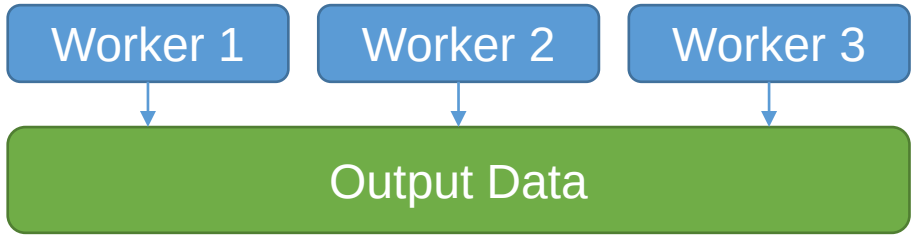
- To reduce the amount of complexity for an end user to manage, RC uses a shared file server to manage user related storage.
- "Core Storage"
  - Contains `/home, /projects, and /rc_scratch`
  - Contains all shared software and the module stack.
- Every node or login VM is connected to this resource allow user to easily manage their files.
- Non-Parallel IO
- Slow reads and writes

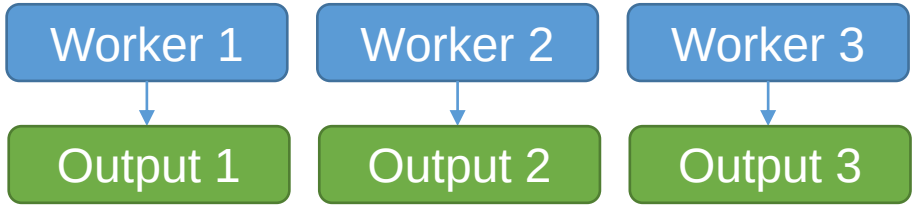
RC's Storage


Heterogeneous Nodes
Blanca
/home
/projects
/curc
...
Core Storage
Compile Nodes
Compute Nodes
Login Nodes
RMACC Summit

Problems with I/O and threads

- Suppose someone is computing with 120 threads and needs to write their data to a file system…
- Single File:
  - Many threads means that applications may idle waiting for free resources.
  - Nonlocking I/O may cause corruption of data.
- Many Files:
  - Separate file writes may lead to issues with the filesystem's metadata service.
- So what do we do?





RC's Parallel filesystems

- RC provides an additional system local parallel file system available on Summit.
- Scratch Space:
  - Spinning disk platters rated at 12 Gb/s
  - GPFS File System for parallel I/O w/ 32 Clints and 4 Servers
    - Distributed metadata to avoid bottlenecking
    - Consistent chunking allows for parallel I/O
  - Locally mounted on Summit nodes.
- Scratch is usually limited to 10 TB of storage but can be expanded upon with request.
  - Purges every 90 days from file creation
  - Technically shared among all users
  - Scratch totals in 1 PB of storage for everyone

Parallel Filesystem

- Normal application I/O is usually lacking the ability to leverage a parallel file system for performance
  - On Summit you will naturally get an I/O performance boost when using scratch.
- Need to utilize specialized software libraries
- MPIIO
  - Middle wear, requires modification of code for efficient usage.
- HDF5
  - High level, use a HDF5 dataset
- NETCDF
  - High level use a Netcdf dataset

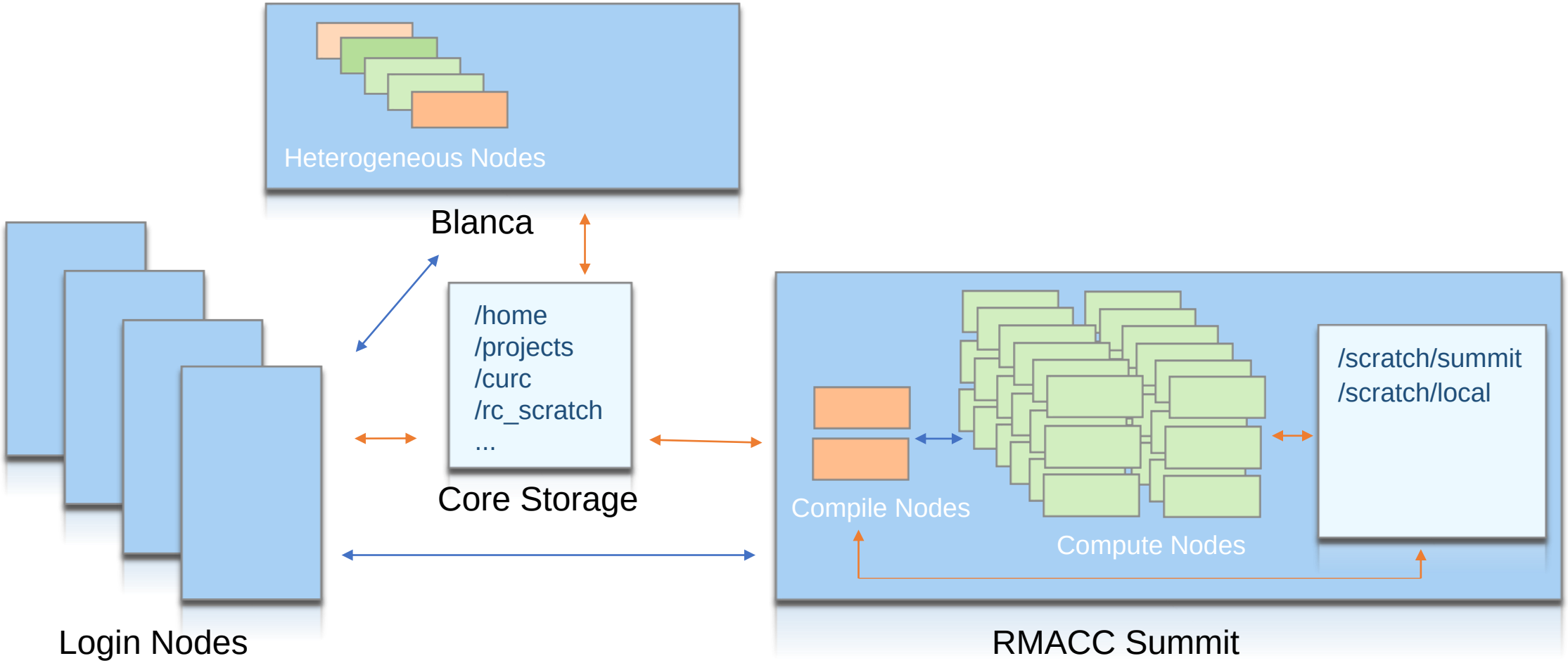
Local Node SSDs

- Summit Nodes also hold onto 100 GB of local node SSD storage.
- These SSDs are not shared among nodes so must move files over within job.
- No Cooperative Parallel I/O
- Located at `/scratch/local`

Some more filesystems…

- High Performance I/O is less available with Blanca
- Petalibrary Active
  - High Speed performance
  - Allows for parallel reads and writes
  - Currently utilizes BeeGFS as its parallel filesystem, will be switching to GPFS
  - More on this in a minute…
- RC_Scratch
  - Not the best storage on RC…
  - Older scratch space past its service dates
  - Large but slow
  - "*I'm sometimes wonder if /projects or /home would be faster than rc_scratch…"*

Parallel Filesystems



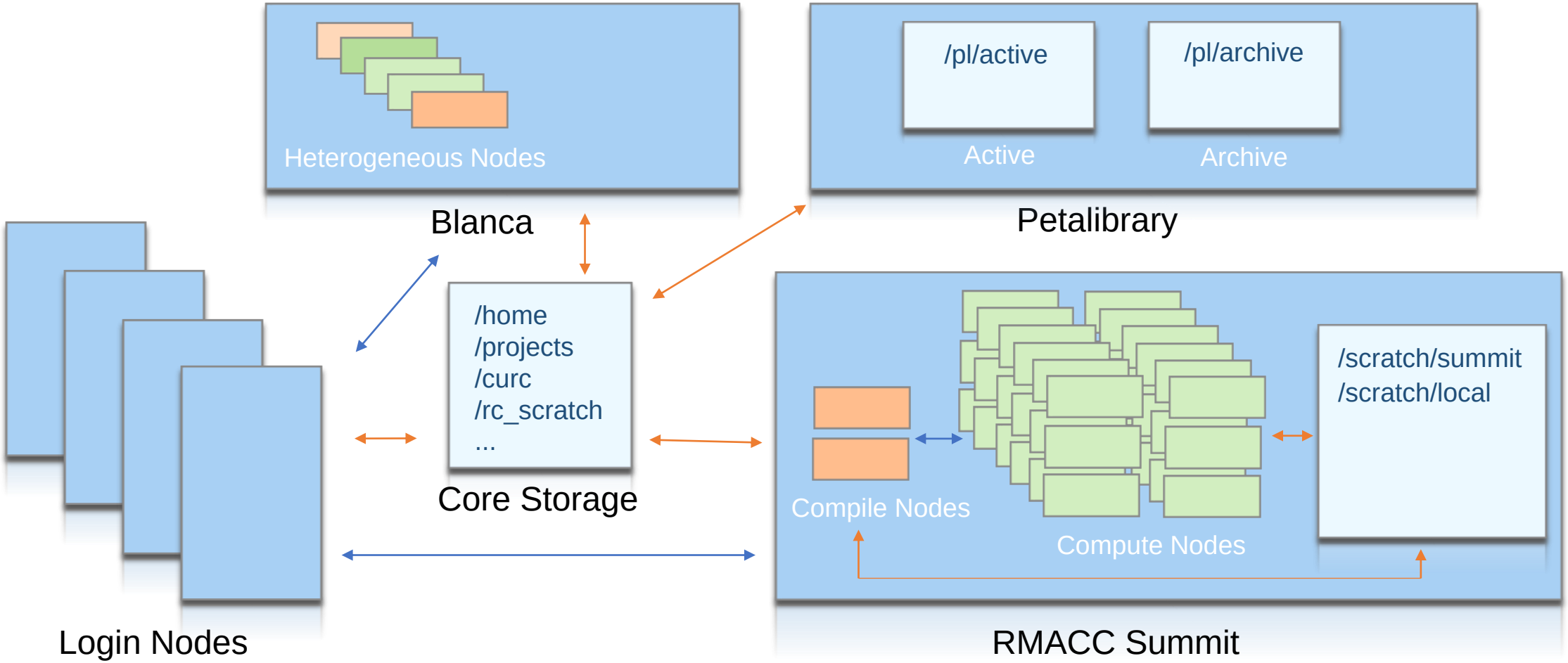
What is Petalibrary?

- Research Computing offers a subsidized but **paid**, long-term storage solution closely coupled with RC resources.
- Petalibrary
  - Large scale subsidized storage solution
  - Enterprise Grade
  - RC Staff supported with assistance on transfer strategies
  - Available in several flavors:
    - Active – Disk
    - Archival – Tape
    - Active Storage with Archive copy

Hardware Specifications

- Active Storage
  - Spinning disk platters for frequent reads and writes
  - GPFS filesystem
  - Parallel file I/O capable
  - RAID-6 file protection
  - Allocations located at: `/pl/active/`
- Archive Storage
  - Tape storage for infrequent reads and writes
  - iRods backed with StrongBox
  - Redundant copies of Data on separate tapes
  - Allocations located at: `/pl/archive/`

Petalibrary



Checking your storage limits:

- *curc-quota* – Research computing tool to monitor disk usage.
  - Provides detailed summary of your core storage
  - Provides detailed summary of scratch space on compile and compute nodes
  - Also lists current capacity of all Petalibrary allocations you have access to

```
[userXXXX@login12 ~]$ curc-quota
```

Data Transfers

- Data transfers are usually handled by one of 2 methods:
- Globus
  - By far the most stable and recommended way for data transfers
  - Fast transfers
  - Transfers continue if a user disconnects
  - Web GUI option or Globus Connect Personal
- SCP/SFTP
  - Secure Copy and Secure File Transfer Protocol
  - Straightforward method of transferring data
  - Generally recommend only to move small files less than a Gigabyte.

More on Data transfers

- Less common methods of transferring data…
- sshfs
  - Mounting the RC filesystem to your drive remotely!
  - Single sign in for multiple data transfers
  - Great when needing to repeatedly access files on RC Resources
- rsync/rclone
  - Another utility to transfer files
  - Particularly useful in repeated file transfers and synchronization of file sets
  - Snapshot like backups
  - https://github.com/ResearchComputing/Documentation/blob/dev/docs/compute/rclone.md

Some additional notes…

- Do not run your job against `/home` or `/projects`
  - Slows down your jobs.
  - Slows down everyone else's jobs.
  - We monitor for this and will kindly ask you to stop.
  - Use Scratch instead for your data!
- Always recover your data from scratch after your job completes!
  - No backup and we can't get your data back!
- Always have a safe backed up location for your data!

Questions?

Thank you!

- Please fill out the survey: http://tinyurl.com/curc-survey18
- Contact information: rc-help@Colorado.edu
- Slides: *https://github.com/ResearchComputing/Filesystems_And_Storage_Fall_2020*