

Fortress

Or.
Just what is a tar file, anyway?

Steve Kelley, 21 August 2020

With thanks to Lev Gorenstein, who actually made most of these slides

Cluster Home directories

\$HOME
/home/myusername

Features:

- Small (25 GB) and mildly performant
- Redundant hardware, never purged, protected by snapshots
- Everyone with RCAC account has it, but probably of little use outside of clusters
- “*My home is my castle*” – belongs to you. A known pain point for PIs (if student graduates and leaves important data behind).

Good for:

- **Your personal codes, programs, scripts, configuration files**
- **Your data and results** (if small)
- OK to run jobs off of it, but only if mostly number-crunching (little I/O)
- **Medium and long-term** storage

Cluster Scratch directories

`$RCAC_SCRATCH`
`/scratch/mycluster/myusername`

Features:

- huge (100+ TB) and **very performant**
- Engineered for large sustained I/O (not so much for myriad of small files though)
- Belongs to you (pain point for PIs)

Good for:

- Your massive generated **intermediate data**
- Perfect to run **data-intensive jobs** off of it.
- **NOT** for valuable codes, programs, scripts, configuration files
- **NOT** for long-term precious final results
- **Beware of weekly purges!**

Use the Fortress!

Fortress tape archive

Features:

- Huge (25 **PB**), free and practically unlimited
- Tape library with a robotic arm and a disk cache in front (i.e. fast uptake, slow egress)
- Redundant hardware, never purged, protected by multiple physical copies on separate tapes.
- Accessible from all RCAC clusters as well as on- and off-campus with additional tools
- Everyone with RCAC account gets personal Fortress space. Additionally, labs with Depot also get lab Fortress space (access controlled by `mylab-data` Unix group membership)
- *Don't have to own cluster nodes!*
- Your space belongs to you, group spaces belongs to the PIs!

Personal and group spaces
/home/myusername and /group/mylab

Good for:

- **BACKUP of your precious personal or lab-wide codes, data and results**
 - *“Vita brevis, ars longa”* – ancient wisdom that snapshots are transient and only true backups last
- **NOT for running jobs off of it!** But perfectly ok to generate/process massive data in scratch, and then send to Fortress in one job
- Native tools `hsi` and `htar` available for Linux, SFTP for other command lines, and Globus endpoint makes life great for everyone
- **Tape prefers few large files over ton of small ones – bundle ‘em!**
- **Long-term archival storage**

Migrate between storage systems throughout data lifecycle

- Amount of data never goes down
- Old data is precious
 - Do you need 10 years worth old data in hot immediate vicinity in your Depot space?
 - Risky (what if you accidentally rm in the wrong place?) And expensive.
 - Maybe copy the all 10 years out to Fortress, and delete the first 7 from Depot.
 - You get peace of mind (full backup!) and savings (fewer TB needed!)
 - If you need to review old data five years from now, you'll get it from tape into your scratch
 - Yes, it may take an hour to get it from tape. Negligible compared to 5 years.
- Current data is precious, too!
 - You finished a run, generated some raw data? Send them to Fortress first, then analyze.
 - Why not put an ht ar in almost every job script?
- Also true for large data-generating scientific instruments (sequencers, microscopes, etc.)

Fortress access by hsi and htar

- All supported methods: www.rcac.purdue.edu/knowledge/fortress/storage/transfer
 - hsi and htar are special command-line utilities for HPSS tape archive
 - Installed on all clusters, available for download for other Linux computers
 - Other OS can use SFTP (via fortress-sftp.rcac.purdue.edu) or Globus
- hsi is remote shell-like interface to Fortress
 - navigate, list, manipulate files, transfer things in or out
 - ls, cd, mkdir, cp, mv, put, get – great for interactive work (but can also be batched)
 - www.rcac.purdue.edu/knowledge/fortress/storage/transfer/hsi
- htar closely mimics regular tar
 - create, list, extract tarballs stored on Fortress (also makes a matching index file for faster searches)

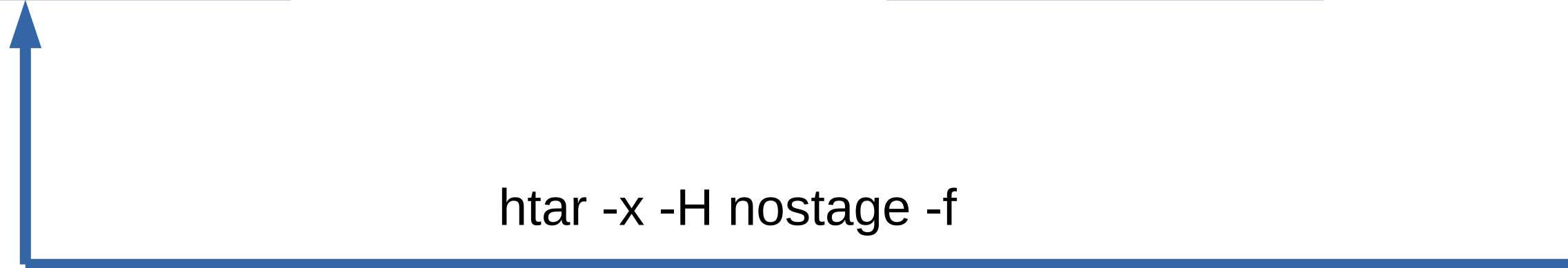
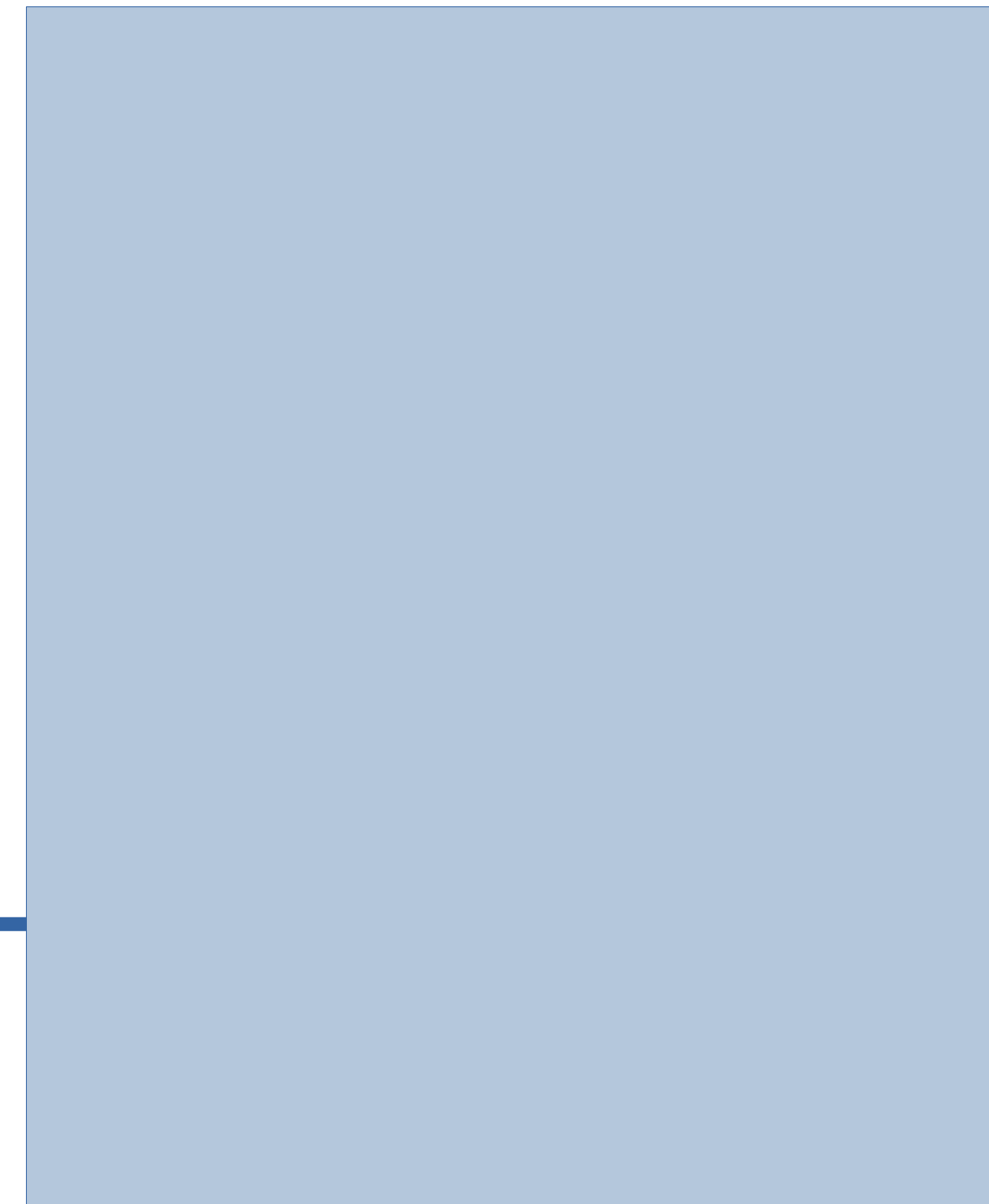
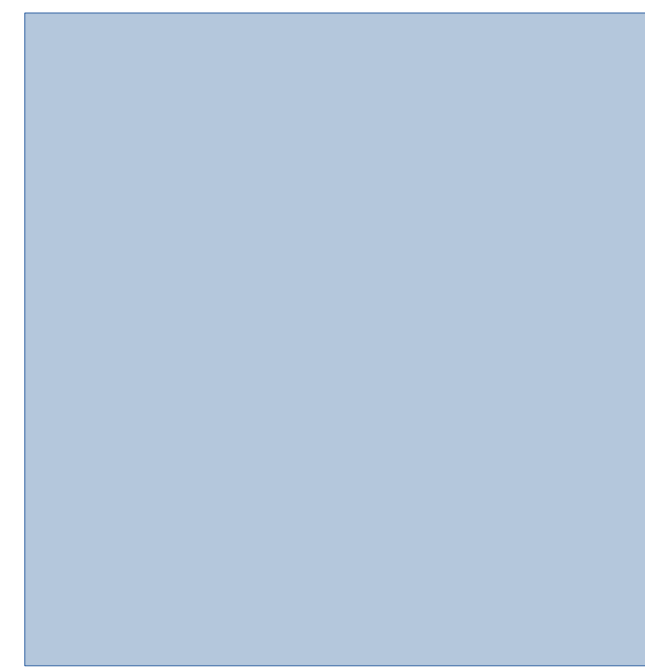
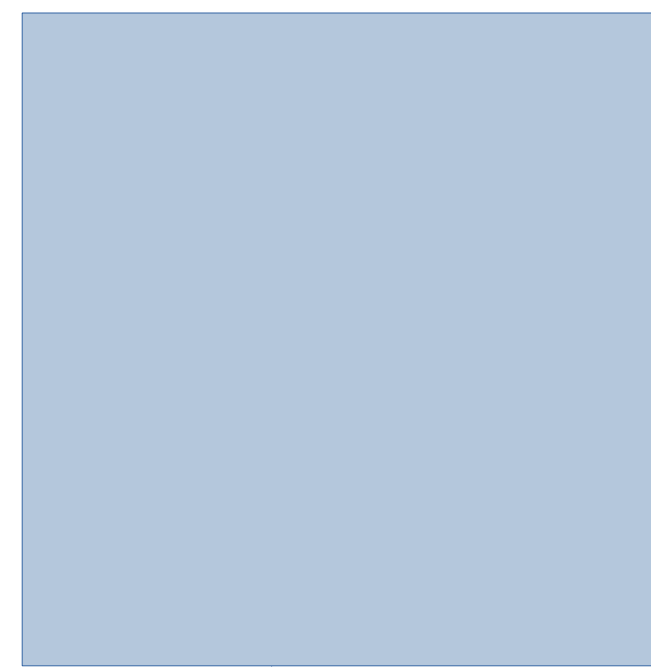
```
htar -cPvf /path/on/fortress/archive.tar datadir(s) # store
htar -xvf /path/on/fortress/archive.tar [file(s)] # extract
htar -tvf /path/on/fortress/archive.tar # list
```
 - www.rcac.purdue.edu/knowledge/fortress/storage/transfer/htar

Fortress

Cluster Scratch

Disk

Tape



hsi subcommands

ls -U to show storage type – Disk or Tape

```
-rw----- 1 kelley itap 11 66817 TAPE 6366394880 Apr 9 14:17 satsuma.tar
-rw----- 1 kelley itap 10 66817 DISK 419616 Apr 9 14:17 satsuma.tar.idx
```

Otherwise similar to normal ls.

- **syntax for get and put**

```
get <local filename> : <remote filename>
```

```
put <local filename> : <remote filename>
```

- if you want to share, make sure to check (and correct) permissions for any directories

```
hsi ls -l
```

```
hsi chmod or
```

```
hsi mkdir -m 2770 dirname
```


htar (so what is a tar file anyway?)

tar is an archive format – many files in one ‘container’ file

File operations (open, close) are slow. Finding a file on tape is **very** slow.

tar lets us bundle many thousands of small files into one large file. And now our overhead becomes negligible.

We only need to find one file instead of many, we only need to open one file, and we can stream the file from tape continuously, rather than bouncing back and forth.

important tar options:

- -c create a tar file
- -t read the table of contents of a tar file
- -x extract the contents of the tar file (all or some)
- -f Always needed – the name of the tar file to operate on.

Things to note about htar

Unlike hsi, htar uses the same syntax for filenames as the normal Unix tar

Unlike tar, htar **always** operates on files on Fortress. The -f filename is relative to the Fortress filesystem, NOT the filesystem you are logged into. (but you don't need to remember the pesky ':', which is nice)

```
htar -c -v -f /group/groupname/fortress_file_name.tar localdirectory
```

will create a tar file named 'fortress_file_name.tar' in the 'groupname' group directory, and it will fill the tar file with all the files and subdirectories under 'localdirectory'

Extracting individual files needs the **exact** name as reported by
htar -tf

Scripting

Anything you can type, you can put into a script. And anything you put into a script, you can put into a scheduled job. (but there's a problem)

```
#!/bin/bash
#SBATCH -t 4:00:00

cd project1
make_some_data

htar -c -v -f project1.tar .

# Note that space and dot at the end of the last line.
# Do you remember what that means?
# (it tells htar what directory to pour into the container)
```

This pattern Works!

The problem

```
#!/bin/bash  
#SBATCH -t 4:00:00  
  
cd project1  
htar -x -v -f project1.tar .  
  
reanalyze_the_data
```

This pattern may NOT work, because reading from tape may take a long time.

If your work depends on data currently stored on Fortress, you need to think about timing – you don't want to schedule a job that runs out of walltime because the tape drives were busy

Consider **hsi stage**

Consider **htar -x -v -H nostage -f**

Bottom Line – tl;dr

Don't clog up /scratch – bring Fortress into your science life

But remember that a tape system has quirks and characteristics you won't see with other tools. (it's worth it, though)

Thank You

Questions?

Slides: rcac.purdue.edu/training

Email Help: rcac-help@purdue.edu

Drop-in Coffee Hours: Monday–Thursday, 2:00 – 3:30pm

WebEx

rcac.purdue.edu/coffee

Friday Talks: Fridays, 2:00pm

WebEx

rcac.purdue.edu/news/events

