

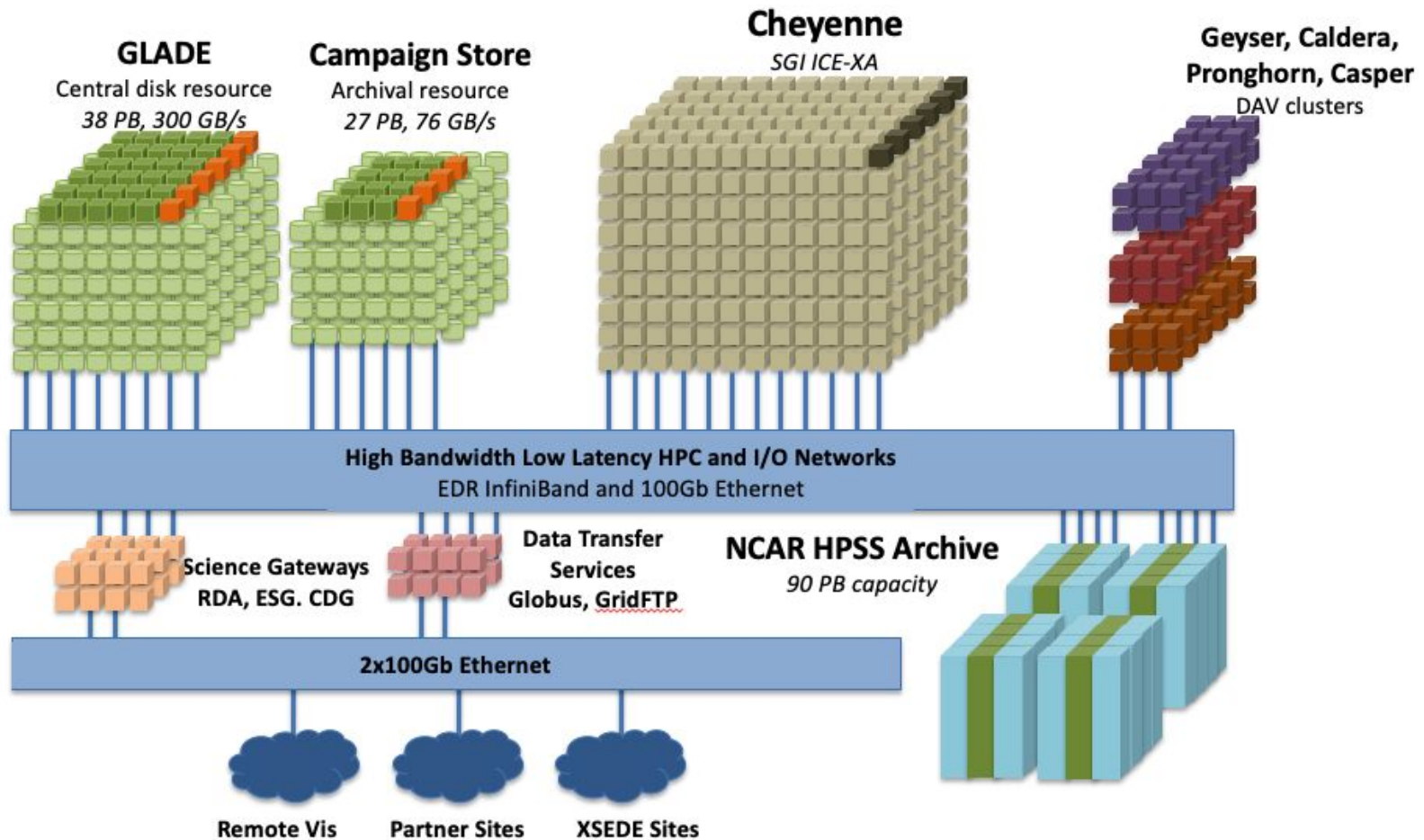
# Using GUF I in Data Management

**Christopher Hoffman**  
**Bill Anderson**

National Center for Atmospheric Research

**HPCSYSPROS**  
2019

# NCAR Computing Environment



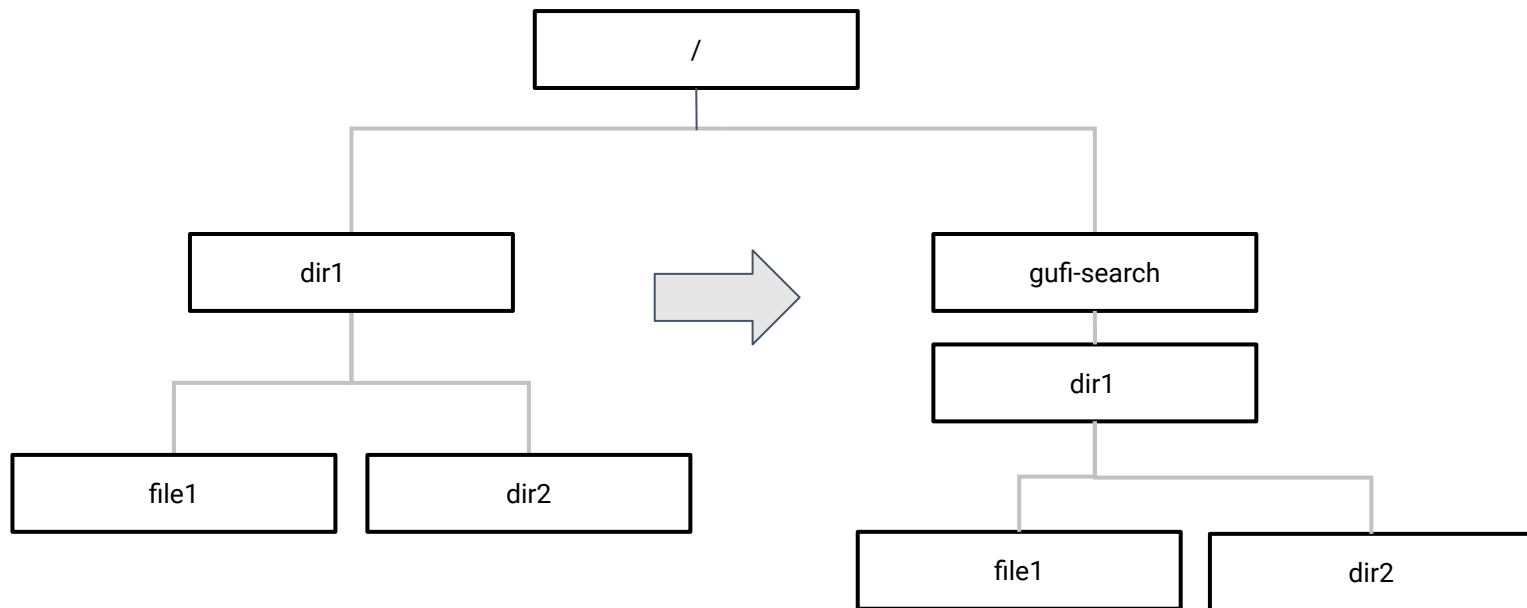
# The Data Management Problem

- Active Projects that involve Data Management
  - Archive Cooling Project
  - Purges
- NCAR has over 2B inodes
  - 1.6B in large file system (GPFS)
  - 120M in home space (GPFS)
  - 100M in Campaign Store (GPFS)
  - 250M in Archive (HPSS)
- Find, pfind are moderately fast for POSIX, not Archive
  - Take resources against POSIX or Archive
- How to uniformly search all spaces that are available
- Pilot Grand Unified File Index from Los Alamos National Laboratory
  - Follow POSIX permissions, open source, ease of use to setup and administer

# How GUFi Works

1. GUFi reads **source** directory tree metadata in a variety of ways (e.g., walk tree, ingest a flat file, call the GPFS API, etc.)
2. It creates a **search** directory tree with the same directory structure, permissions, and ownership; each directory contains a database with metadata about the files in the original directory
3. The GUFi query commands access the **search** tree to quickly provide data (e.g., amount of bytes in a subdirectory, etc.).

# How GUFi Works: Source and Search Trees



# Challenges

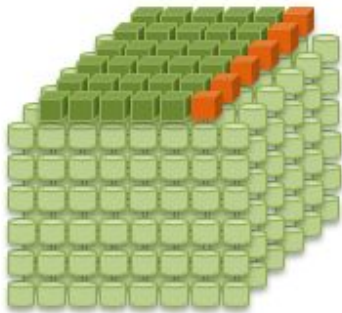
- When starting out GUFi has not been in production anywhere yet
- NCAR has billions of files across many storage systems
- NCAR purges folder and directories on scratch
  - Some processes must be reworked to do this at scale
- Created a few processes
  - MPI based GPFS ILM Scanner
  - GPFS ILM Scan treewalk
  - GUFi flat file generator

# Integration Details

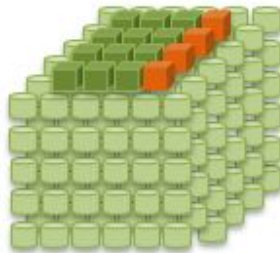
- Supermicro NVMe box with 10 NVMe slots
- 2x 22 core Xeon Processor
- 768GB of RAM
- Storage components
  - RAM for queries and index process, 768GB
  - NV SSD capacity to store GUFIs trees, 4x750GB
  - RAID-0 for NVMe devices
- After initial ingest and post processing db is fixed backup GT
- Weekly index of storage system

# NCAR GUFU Workflow

**GLADE**



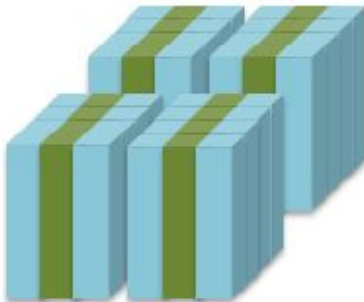
**Campaign Store**



**Data Management**



**NCAR HPSS Archive**

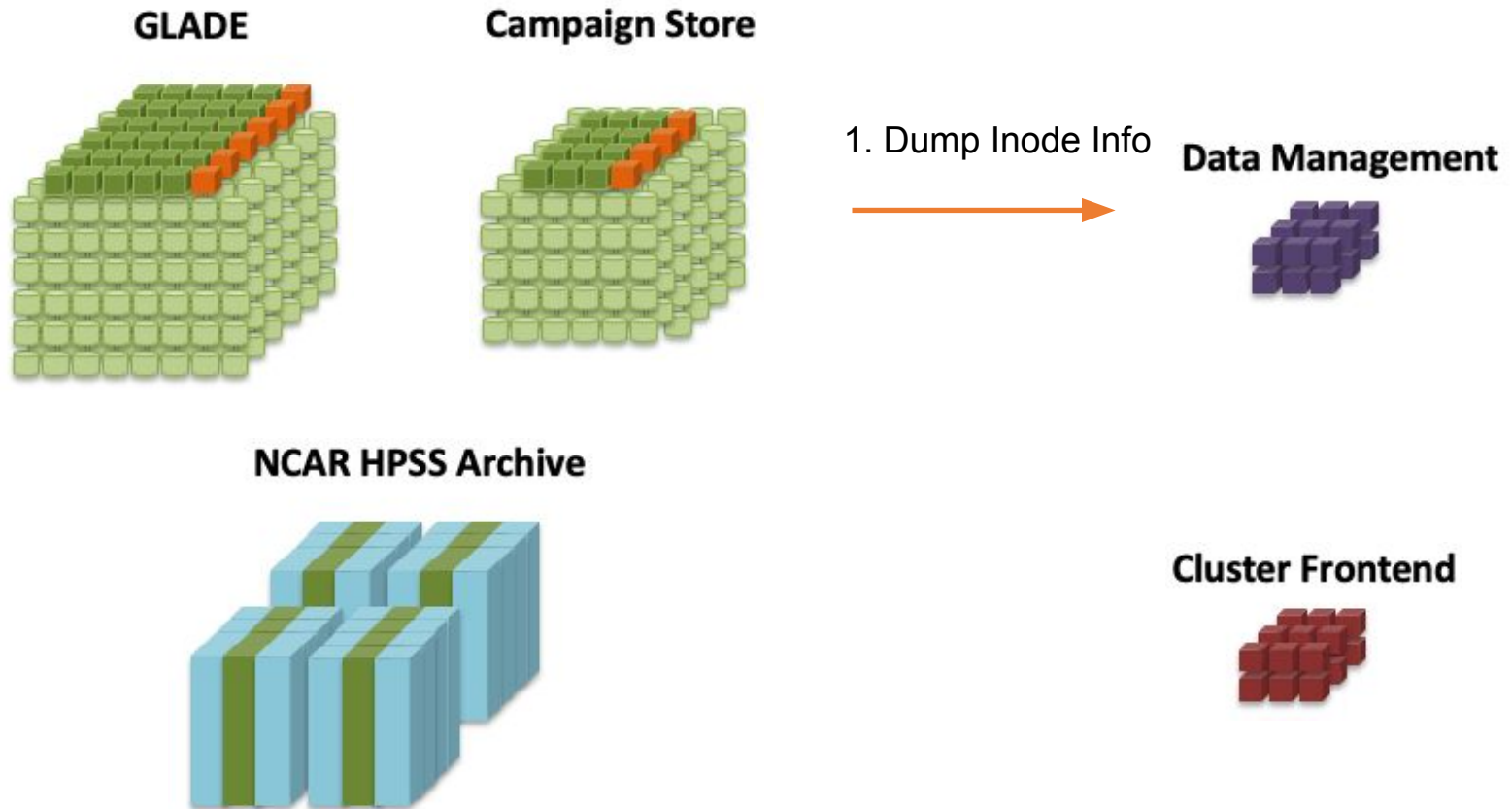


**Cluster Frontend**

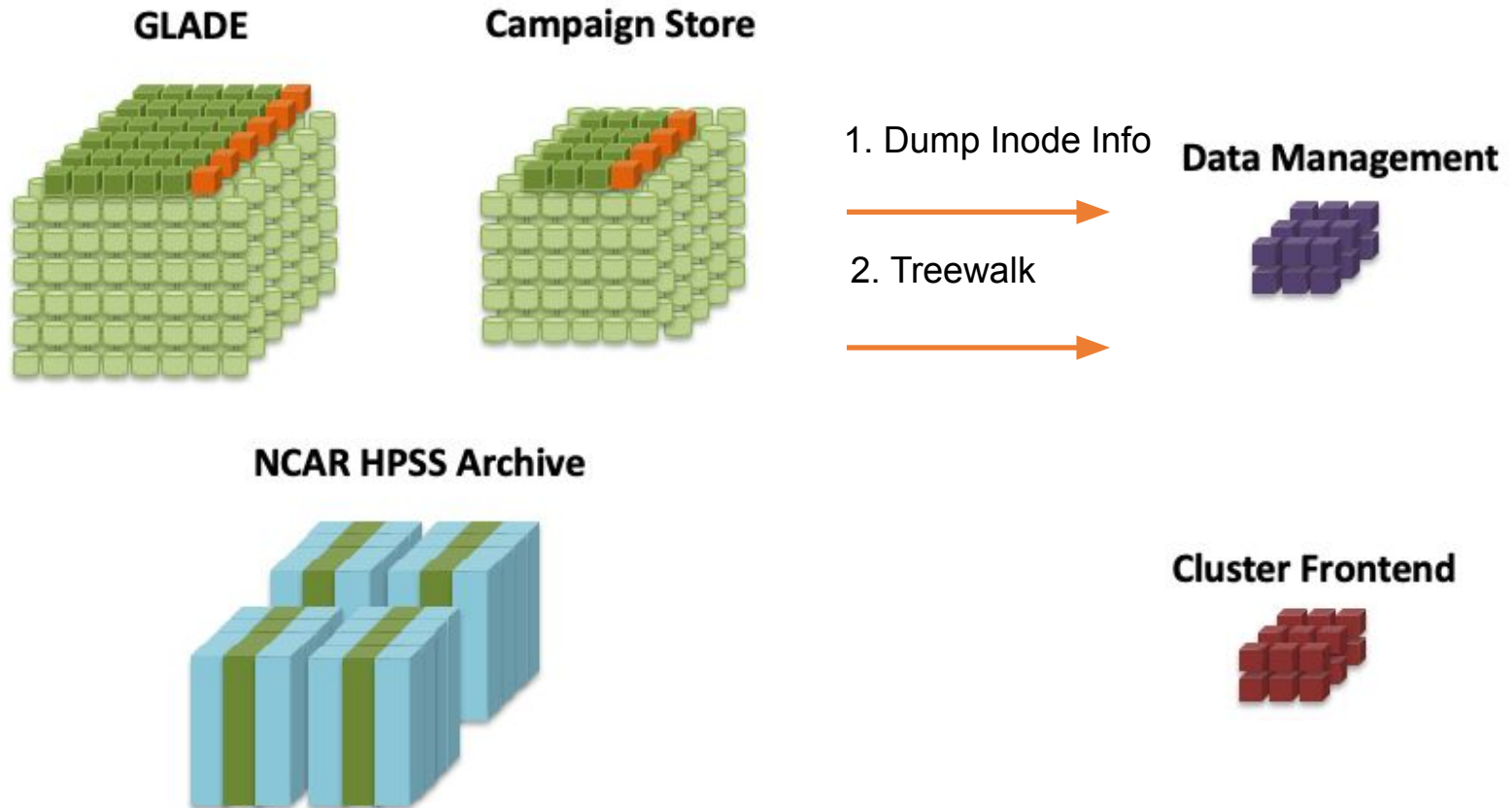




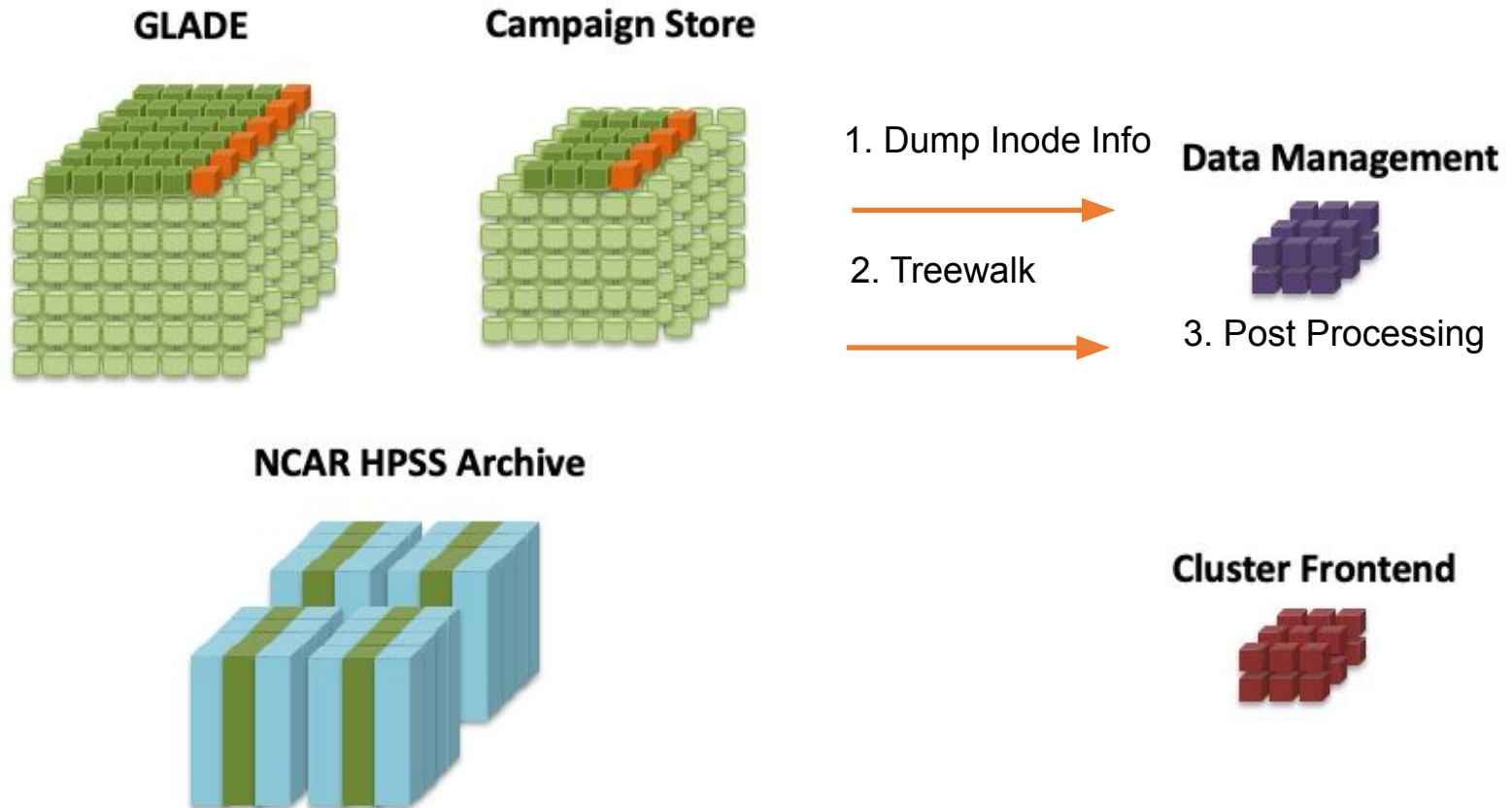
# NCAR GUFU Workflow



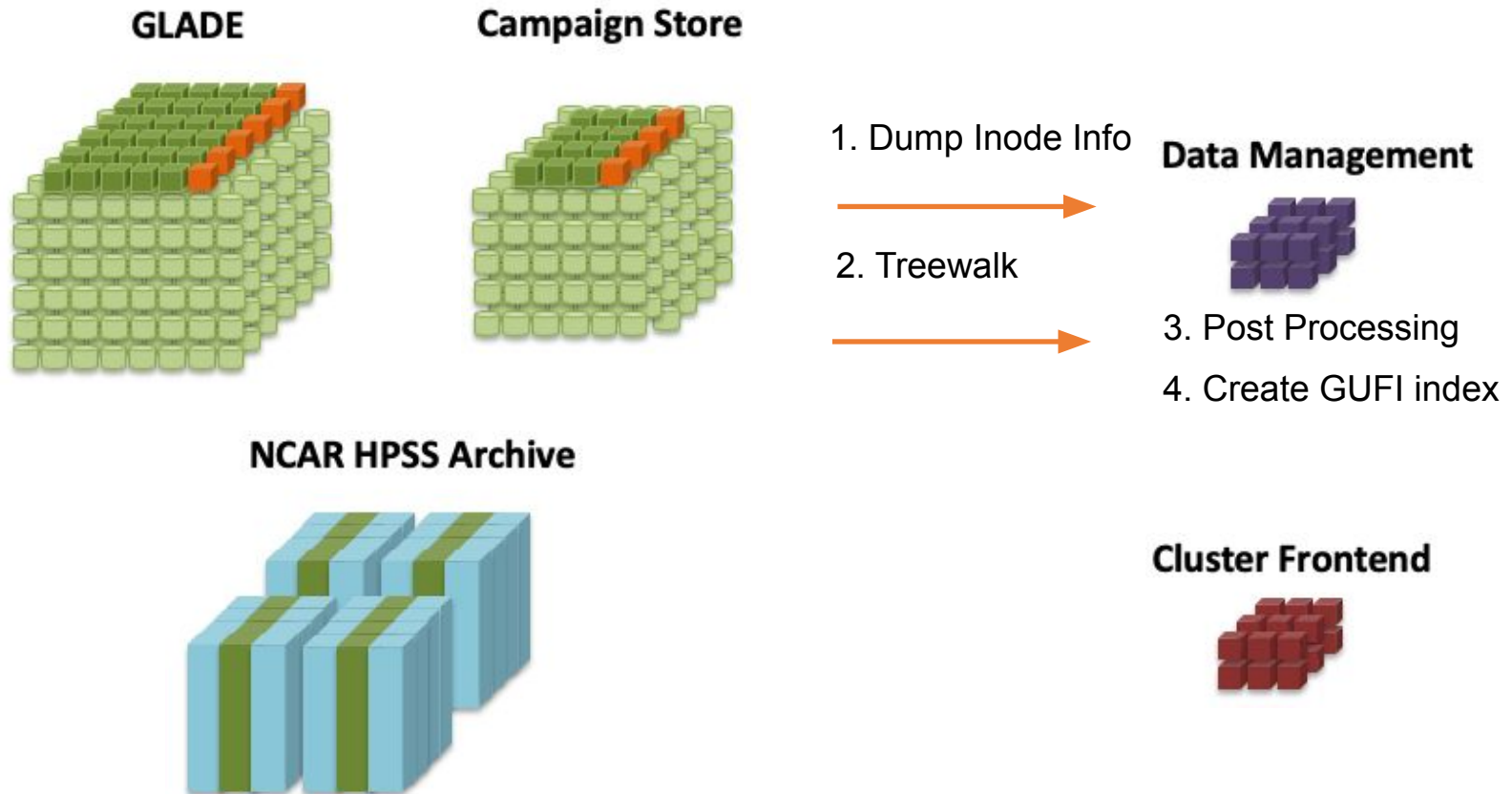
# NCAR GUFU Workflow



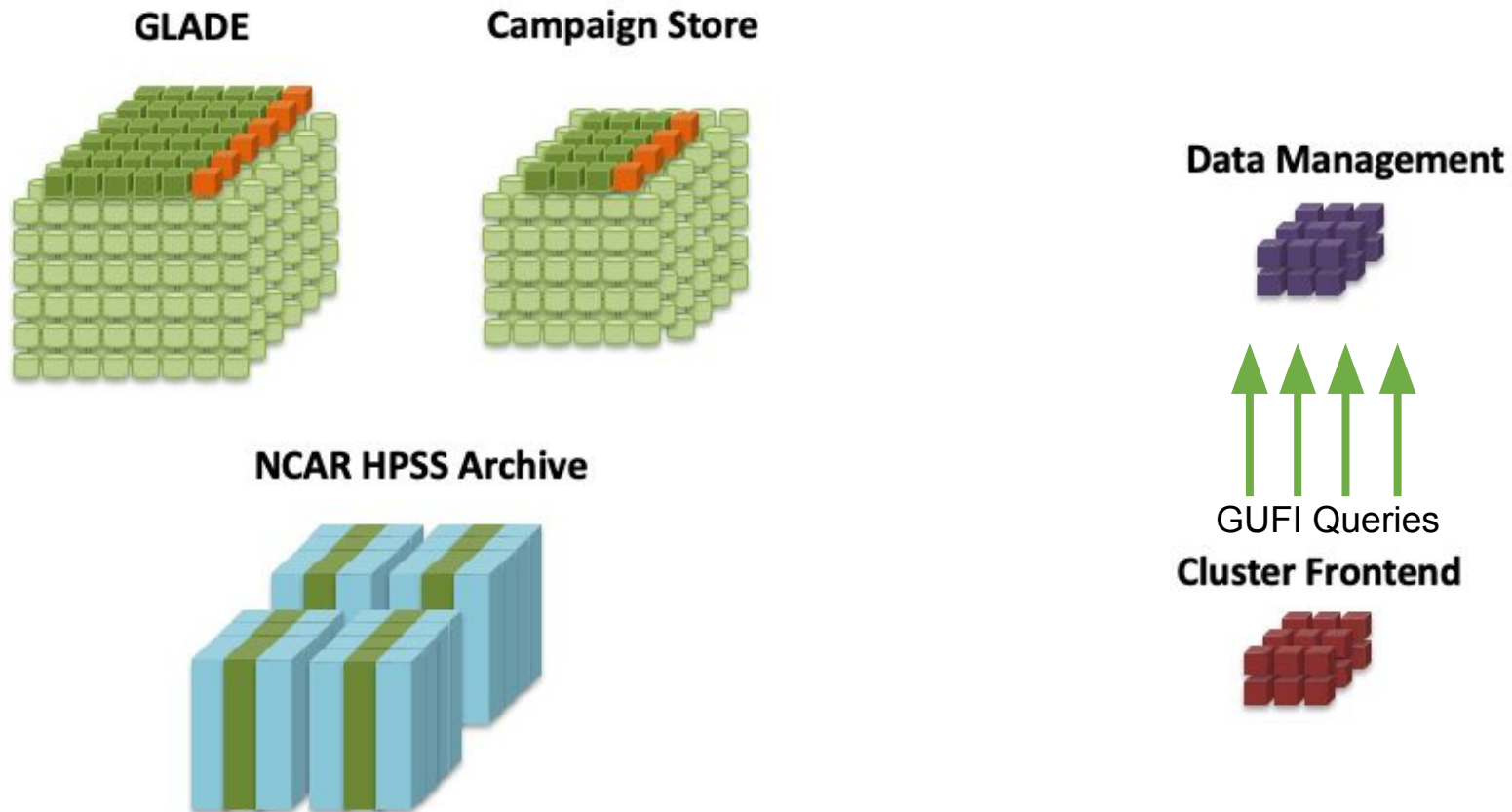
# NCAR GUFU Workflow



# NCAR GUFi Workflow



# NCAR GUFi Workflow



# Archive Cooldown Project

- Project goals: slow ingest to archive, move data to appropriate location and clean up unneeded data.
- Utilizing GUFi for single queries and complex queries
- Created precomputed summaries for common complex queries
  - These can be stored in a summary table in GUFi tree
- Example summarized query:

`<directory> contains 12345 files distributed in 123 directories,  
owned by 4 users (list of users) and 6 projects for a grand total  
of 200TB`

`user u1 owns nnn files for a total of 10TB`

`(etc for the other users)`

`projects p1 has mmm files for a total of 20TB`

`(etc for the other projects)`

# Purge

- Purge timeframes
  - scratch, 4 months
  - projects, 1 year
  - Campaign Store, 5 years
- Users can query files that may be candidates for deletion/removal
- `# gufi_find -ctime -atime -mtime`
- This will give users a list of files to move to other spaces for longer retention
- Yes, we check for touches

# POSIX/HPSS vs GUFi Equivalent

Task	POSIX	HPSS	GUFi
get space	du <path>	hsi du <path>	gufi_stats total-space <path>
find file	find <path> -name <file>	hsi find <path> -name <file>	gufi_find -name <file> <path>



# Performance

	# files	du (native)	du (GUFI)	speedup	find (native)	find (GUFI)	speedup
Campaign Store	3.1M	9m20s	3.6s	<b>155x</b>	3m20s	9.8s	<b>20.4x</b>
GPFS	100K	2m49s	8.5s	<b>19x</b>	29.6s	6.9s	<b>4.2x</b>
HPSS	29M	174m8s	50s	<b>208x</b>	214m32s	5m12s	<b>41x</b>

# How to get started

- Download source code and browse documentation:

<https://github.com/mar-file-system/GUFI>

- Contact us for any NCAR implementation specific questions at:

[choffman@ucar.edu](mailto:choffman@ucar.edu)