Introduction to Archival Storage at NERSC

Nick Balthaser
Storage Systems Group

February 15, 2013
Agenda

• Objectives
  – Describe the role of archival storage in a tiered storage strategy
  – Log into the NERSC archive
  – Store and retrieve files from the archive
  – Avoid common problems

• Archive Basics
  – What is an archive?
  – Why should I use one?
  – Features of the NERSC archive

• Using the NERSC Archive
  Note: Unix/Linux command-line familiarity required
  – How to log in
  – Storing and retrieving files with HSI
  – Storing and retrieving directories with HTAR
  – Avoiding common mistakes

• Questions, Problems, Further Reading
• Hands-on Examples
Archive Basics
What is an archive?

• **Long-term storage of permanent records and information**
  – Often data that is no longer modified or regularly accessed
  – Storage time frame is indefinite or as long as possible
  – Archive data typically has, or may have, long-term value to the organization

• **An archive is not a backup**
  – A backup is a copy of production data
  – Value and retention of backup data is short-term

• **A backup is a copy of data. An archive *is* the data.**
Why should I use an archive?

• **Data growth is exponential**

  ![](cumulative_storage.png)

  *Cumulative Storage by Month and System*

  - **File system space is finite**
    - 80% of stored data is never accessed after 90 days
    - The cost of storing infrequently accessed data on spinning disk is prohibitive
    - Important, but less frequently accessed data should be stored in an archive to free faster disk for processing workload
Why should I use an archive (continued)?

- **Archives are an important component of a tiered data management strategy**
  - Align value and access patterns of data with media on which it is stored:
    - Flash: IO intensive workloads
    - Disk: primary storage
    - Tape: backup, long-term storage (archive)

- **Tape still the lowest cost/GB**
  - 30 year shelf life
  - Energy savings over disk
  - Lower admin costs
  - Lower bit error rate (BER)

- **Typical use cases at NERSC include:**
  - Long-term storage of very large raw data sets
    - Good for incremental processing
  - Long-term storage of result/processed data
  - Backups (e.g. global scratch purges)
Features of the NERSC archive

• **NERSC implements an “active archive”**
  – NERSC archive supports parallel high-speed transfer and fast data access
    • Data is transferred over parallel connections to the NERSC internal 10Gb network
    • Access to first byte in seconds or minutes as opposed to hours or days
    • The system is architected and optimized for ingest

• **The archive uses tiered storage internally to facilitate high speed data access**
  – Initial data ingest to high-performance FC disk cache
  – Data migrated to enterprise tape system and managed by HSM software (HPSS) based on age and usage

• **The NERSC archive is a shared multi-user system**
  – Shared resource, no batch system. Inefficient use affects others.
  – Session limits are enforced
The NERSC archive is a Hierarchical Storage Management system (HSM)

- Highest performance requirements and access characteristics at top level
- Lowest cost, greatest capacity at lower levels
- Migration between levels is automatic, based on policies
Using the NERSC Archive
How to Log In

- **The NERSC archive uses an encrypted key for authentication**
  - Key placed in `~/.netrc` file at the top level of the user’s home directory on the compute platform
  - All NERSC HPSS clients use the same `.netrc` file
  - The key is IP specific. Must generate a new key for use outside the NERSC network.

- **Archive keys can be generated in two ways**
  - Automatic: NERSC auth service
    - Log into any NERSC compute platform using ssh
    - Type “hsi”
    - Enter NERSC password
  - Manual: [https://nim.nersc.gov/](https://nim.nersc.gov/) web site
    - Under “Actions” drop down, select “Generate HPSS Token”
    - Copy/paste content into `~/.netrc`
    - `chmod 600 ~/.netrc`
Storing and Retrieving Files with HSI

- HSI provides a Unix-like command line interface for navigating archive files and directories
  - Standard Unix commands such as `ls`, `mkdir`, `mv`, `rm`, `chown`, `chmod`, `find`, etc. are supported

- FTP-like interface for storing and retrieving files from the archive (put/get)
  - Store from file system to archive:
    - `bash-3.2$ hsi
      A:/home/n/nickb-> put myfile
      put 'myfile' : '/home/n/nickb/myfile' ( 2097152 bytes, 31445.8 KBS (cos=4))`
  - Retrieve file from archive to file system:
    - `A:/home/n/nickb-> get myfile
      get 'myfile' : '/home/n/nickb/myfile' (2010/12/19 10:26:49 2097152 bytes, 46436.2 KBS )`
  - Full pathname or rename file during transfer:
    - `A:/home/n/nickb-> put local_file : hpss_file`
    - `A:/home/n/nickb-> get local_file : hpss_file`
Storing and Retrieving Directories with HTAR

• HTAR stores a Unix tar-compatible bundle of files (aggregate) in the archive
  – Traverses subdirectories like tar
  – No local staging space required--aggregate stored directly into the archive

• Recommended utility for storing small files

• Some limitations
  – 5M member files
  – 64GB max member file size
  – 155/100 path/filename character limitation
  – Max archive file size* currently 10TB

• Syntax: `htar [options] <archive file> <local file|dir>`
  – Store
    - `bash-3.2$ htar -cvf /home/n/nickb/mydir.tar ./mydir`
  – List
    - `bash-3.2$ htar -tvf /home/n/nickb/mydir.tar`
  – Retrieve
    - `bash-3.2$ htar -xvf /home/n/nickb/mydir.tar [file...]`

* By configuration, not an HPSS limitation
Avoiding Common Mistakes
Small Files

• Tape storage systems do not work well with large numbers of small files
  – Tape is sequential media—tapes must be mounted in drives and positioned to specific locations for IO to occur
• Mounting and positioning tapes are the slowest system activities
  – Small file retrieval incurs delays due to high volume of tape mounts and tape positioning
  – Small files stored periodically over long periods of time can be written to hundreds of tapes—especially problematic for retrieval
• Use HTAR when possible to optimize small file storage and retrieval
• Recommend file sizes in the 10s – 100s of GB
Large Directories

- Each HPSS system is backed by a single metadata server
  - Metadata is stored in a single SQL database instance
  - Every user interaction causes database activity

- Metadata-intensive operations incur delays
  - Recursive operations such as "chown –R ./*" may take longer than expected
  - Directories containing more than a few thousand files may become difficult to work with interactively

-bash-3.2$ time hsi -q ‘ls -l /home/n/nickb/tmp/testing/80k-files/’ > /dev/null 2>&1

real 20m59.374s
user 0m7.156s
sys 0m7.548s
Large Directories, continued

- hsi “ls –l” exponential delay:
Long-running Transfers

• Failure prone for a variety of reasons
  – Transient network issues, planned/unplanned maintenance, etc.
• Many clients do not have capability to resume interrupted transfers
• Can affect archive internal data management (migration) performance
• Recommend keeping transfers to 24hrs or less if possible
Session Limits

- 15 concurrent session/user enforced
- Can be administratively reduced if a user is negatively affecting system usability for others
Questions, Problems, Further Reading
Asking Questions, Reporting Problems

• Contact NERSC Consulting
  – Toll-free 800-666-3772
  – 510-486-8611, #3
  – Email consult@nersc.gov.
Further Reading

- NERSC Website

- HSI and HTAR man pages are installed on NERSC compute platforms

- Gleicher Enterprises Online Documentation (HSI, HTAR)
  - http://www.mgleicher.us/index.html/htar/

Hands-on Examples
Logging into archive: Hands-on

• Using ssh, log into any NERSC compute platform
  
  -bash-3.2$ ssh dtn01.nersc.gov

• Start HPSS storage client “hsi”
  
  -bash-3.2$ hsi

• Enter NERSC password at prompt (first time only)
  
  Generating .netrc entry...
  nickb@auth2.nersc.gov's password:

• You should now be logged into your archive home directory
  
  Username: nickb  UID: 33065  Acct: 33065(33065) Copies: 1 Firewall: off [hsi.3.4.5 Wed Jul 6 16:14:55 PDT 2011][V3.4.5_2010_01_27.01]
  A:/home/n/nickb-> quit

• Subsequent logins are now automated
Using HSI: Hands-on

- Using ssh, log into any NERSC compute platform
  -bash-3.2$ ssh dtn01.nersc.gov

- Create a file in your home directory
  -bash-3.2$ echo foo > abc.txt

- Start HPSS storage client “hsi”
  -bash-3.2$ hsi

- Store file in archive
  A:/home/n/nickb-> put abc.txt

- Retrieve file and rename
  A:/home/n/nickb-> get abc_1.txt : abc.txt
  A:/home/n/nickb-> quit

- Compare files*
  -bash-3.2$ sha1sum abc.txt abc_1.txt
  f1d2d2f924e986ac86fd736b36c94bdf32beec15 abc.txt
  f1d2d2f924e986ac86fd736c94bdf32beec15 abc_1.txt

* Note: checksums supported in the next HSI release with: ‘hsi ‘put –c on local_file : remote_file’
Using HTAR: Hands-on

• Using ssh, log into any NERSC compute platform
  -bash-3.2$ ssh dtn01.nersc.gov

• Create a subdirectory in your home directory
  -bash-3.2$ mkdir mydir

• Create a few files in the subdirectory
  -bash-3.2$ echo foo > ./mydir/a.txt
  -bash-3.2$ echo bar > ./mydir/b.txt

• Store subdirectory in archive as “mydir.tar” with HTAR
  -bash-3.2$ htar –cvf mydir.tar ./mydir

• List newly created aggregate in archive
  -bash-3.2$ htar –tvf mydir.tar

• Remove local directory and contents
  -bash-3.2$ rm –rf ./mydir

• Extract directory and files from archive
  -bash-3.2$ htar –xvf mydir.tar
Section Title