Non-Technical, Technical Aspects of HPC Storage

Dr. Jeffrey Layton
HPC Enterprise Technologist

Agenda

- Three issues facing HPC storage
  - The professor and the video game
  - This island storage
  - Why isn’t storage monitoring easier?
- Nothing really technical (no diagrams, no performance charts)
- More of a discussion and presentation of customer issues

Duh!

- HPC storage is very complex
  - The range of applications is massive!
  - They all think they are the **most** important application
- Applications have different needs:
  - Some applications need really high performance, some don’t
  - Some applications do more streaming IO, some don’t
  - Some applications are serial, some aren’t
- Data is getting colder
  - But long-term data storage is heating up

The Professor and the Video Game

The Professor and the Video Game

- Renewed interest in keeping data accessible for a long time
  - NSF: ~20 years
  - EPSRC in the UK: 20+ years
  - General EMEA requirements – data retention for a very long time
- If you keep data for a very long time several facts become evident:
  - Almost all of the people who produced the data will have left, forgotten, or don’t care about the data
  - Who owns the data?
  - Bit rot becomes a more serious threat
  - How do you search the data?
  - How do you share the data?
  - How do you secure the data?
- From all of this one key aspect becomes: metadata

Metadata, Metadata, Metadata

- The **key** to all of this is metadata
  - Defines, explains, categorizes the data
- Has to be as *accurate* as possible
- Has to be useful to someone else
- It is not the same for every research field, application or even data set
  - Librarians want a consistent set of metadata for everything
  - User defined metadata is probably a necessity
- Can there be a minimum set of metadata for all data?
  - POSIX attributes are a gimme
  - Any others? Application and version?

Metadata – Part II

- Where do you store the metadata?
  - Many people want to store it in a central database
  - This would be the **only** place where metadata is stored
    - Is this wise?
- Have to worry about integrity of the database
  - Backups, copies, bit-rot, recovery of "correct" database
- What happens when someone moves a tree?
  - `mv /data1/user1/tree1 /data1/user1/tree2`
  - How do you update/recover metadata?
  - Note: You could make the data "read-only" to prevent this from casually happening

Alternative Metadata idea

- Store the metadata with the data as the primary location
  - A “scrapper” grabs the metadata from the data files and updates a central database if needed
- Still use a central database for searching
- Pros:
  - Distributed metadata
  - No single point of failure for metadata (not solely dependent on db)
- Cons:
  - Have to update database (but data doesn’t change often)
  - Potential for data corruption of the metadata with the files (bit rot)
  - Worry about metadata when manipulating files (e.g. NFS access)

Motivation

- The key is metadata
- Without it, the data is meaningless
- Therefore, good, accurate, useful metadata is critical
- How to you motivate a researcher to create good, accurate, useful metadata?
  - When the project is done, the motivation is about zero
  - Has to happen while the data is being created
- What motivates users? What motivates people?
- Concepts:
  - Gamification
  - Carrot with no stick

Gamification (according to wikipedia)

- Gamification is the use of game design techniques, game thinking and game mechanics to enhance non-game contexts.
- Typically gamification applies to non-game applications and processes, in order to encourage people to adopt them, or to influence how they are used.
- Gamification works by making technology more engaging, by encouraging users to engage in desired behaviors, by showing a path to mastery and autonomy, by helping to solve problems and not being a distraction, and by taking advantage of humans' psychological predisposition to engage in gaming
- The technique can encourage people to perform chores that they ordinarily consider **boring**, such as completing surveys, shopping, filling out tax forms, or reading web sites.

Gamification and the carrot

- Contests
  - Which research group or researcher can completely tag all of their data?
  - Can use leader boards to signal social status
  - Achievement status
- Reward contest results (the carrot)
- Virtual currency
  - Can exchange for more storage or more compute time
  - Can exchange for real $$ for conferences
  - Can exchange for meals (grad students)
  - Can exchange for gift cards
- Embedding games
  - Have to complete metadata tagging before playing
- Grand challenge: metadata tagging as a game

This Island Storage

This Island Storage

- Let’s assume we have a bunch of data that we need to preserve and share
- How do we do this?
- What are the issues?
  - Bit rot
  - Security (who gets access to the data?)
  - Transmission of the data
  - Processing of the data?
  - Changes in metadata?
- Does it make sense to just set up some sort of simple data server that allows people to search and pull data?
  - The project is over so don’t spend any more money on it
- Create storage islands

Data sharing

- What happens if you put up data for sharing?
  1. People will search the database and perhaps find useful data
  2. They will pull the data from the server to their local storage
     - Eats up network bandwidth
     - Eats up storage at the users end
  3. User will either use data or discard it
  4. Repeat 1 as needed

- Transmitting data eats network bandwidth
  - What happens if you pull the data and discover it’s not useful?
- Wouldn’t it be better to allow the searchers to manipulate or visualize the data before pulling it across the network?
  - If so, you need some computational/visualization resources

Data integrity

- It is desired to keep the data a long time
  - 20+ years is nominal now
- How do we do this?
  - Lots of different ways (hardware and software)
- One way is through checksums and multiple copies
- Treat the checksums as another bit of metadata
  - They are subject to data corruption (bit rot) as well
  - Store it with the data in xattr
- To make doubly sure: use several checksums
  - md5
  - SHA-1
  - SHA-224, SHA-256, SHA-384, SHA-512

Data integrity becomes an HPC problem

- Need to sweep the data, checking checksums and restoring data from copies (also need to update central metadata db)
- Process:
  - Compute checksums for each file (all copies)
  - Compare computed checksums to stored checksums
    - Any file that does not have matching checksums is flagged as bad
  - From good copies, the bad copies are overwritten
  - Update metadata from files to central dp
- This process is repeated continuously
- Computing checksums takes computational resources
- What was a storage problem is also now a computational problem

Computational resources in storage

- Data sharing and data integrity create computational problems
  - Allow users to manipulate/visualize data prior to copying it
  - Data integrity via checksums
- Servers for manipulating data and visualization
  - Cut data sets into pieces
  - Visualize/examine results
  - Remote visualization techniques
- Servers for computing checksums
- Summary at this point:
  - Data storage
  - Networking
  - Servers
  - Remote visualization

This Island Storage

- Classic clusters:
  - 75% compute+networking, 25% storage
- Storage islands:
  - 75% storage, 25% compute+networking
- So you realistically can’t put the data on some inexpensive storage and call it a day
- What solutions exist for this?
  - None
- Each location becomes a “Storage Island” with data and compute resources to support it
  - Standards may need to be developed or agreed upon for interoperability

Why isn’t storage monitoring easier?

Why Isn't Storage Monitoring Easier?

- If you want to monitor storage what do you do?
  - Today you monitor the throughput or performance of the data servers
- Most tools typically just scan the /proc entries
- What do people want?
  - I want to know what my storage is doing?
    - Who's using it?
    - What are the disks doing?
    - What is the file system doing?
    - What are the file servers doing?
    - What is the network doing?
  - I want to know trends
    - Capacity trends
    - Performance trends (network, etc.)
    - Is my storage still performing the same as when I bought it?

Example

- Let’s take NFS as a simple example
- Tools:
  - IOtop
  - IOstat
  - Collectl
  - Lots of others
- Almost all of them just scan the /proc file system
- Only collectl examine other aspects of the NFS server
  - CPUs, memory, network (all from /proc)
- None of them show which clients are using the storage
  - Need details

I want what I want now

- Admins want to know what is happening from a holistic view
  - Dashboards
- They want the ability to drill down
  - Which disks, LUNs are getting hit hardest?
  - **Which clients are hitting the storage the hardest?**
  - How is the file system performing?
  - What is the IO scheduler doing?
  - How is the CPU load on the server?
  - What is the memory doing? (lots of memory pressure?)
  - What is the network doing?
- Keep this data for a long period for trend analysis
- **Big Red Button**

Deeper and deeper

- Couple storage monitoring with job scheduler
  - Which clients are hitting the storage the hardest?
  - Which jobs are associated with these clients?
- Admins/managers want to keep lots of detailed data for trend analysis
  - My storage monitoring problem has now become a storage problem
- Trend analysis:
  - How is capacity growing? How does this coordinate with utilization?
  - How is performance being utilized?
    - Coordinate this with users and jobs (applications)
  - Identification of bottlenecks (hot spots)
  - How is the “data” changing? (age, size, utilization)
- My storage monitoring problem has now become a data processing problem

Summary

Summary

- Lots of issues/problems in HPC storage
- Only presented three today:
  - **The professor and the video game**
    - All about metadata
  - **This Island Storage**
    - How do you store/share data effectively?
  - **Why isn’t storage monitoring easier**
    - Lack of good management/monitoring tools coupled with new additional storage and data processing problems
- The problems aren’t getting easier
- No good solutions

Thanks!