Introduction to I/O in the HPC Environment

Brian Haymore, brian.haymore@utah.edu
Sam Liston, sam.liston@utah.edu

Center for High Performance Computing
Fall 2010
Overview

- Topology
Overview

- SAN storage redundancy – RAID
- RAID = Redundant Array of Inexpensive Disks
Overview

• Types of storage available at CHPC
  – Home Directory (i.e. /uufs/chpc.utah.edu/common/home/uNID)
    • Per department backed up (except CHPC_HPC file system)
    • Intended for critical/volatile data
    • Expected to maintain a high level of responsiveness
  – Group Data Space (i.e. /uufs/chpc.utah.edu/common/home/pi_grp)
    • Optional per department archive
    • Intended for active projects, persistent data, etc.
    • Usage expectations to be set by group
  – Network Mounted Scratch (i.e. /scratch/serial)
    • No expectation of data retention (It’s scratch)
    • Expected to maintain a high level of I/O performance under significant load
  – Local Disk (i.e. /tmp)
    • Most consistent I/O
    • No expectation of data retention
    • Unique per machine
Overview

• Good Citizens
  – Shared Environment Characteristics
    • Many to one relationship (over-subscribed)
    • Globally accessible
    • Global resource are still finite
    • Consider your usage impact when choosing a storage location
    • Be aware of any usage policies
    • Evaluate different I/O methodologies
    • Seek additional assistance from CHPC
Best Practices

- Data Segregation; Where should files be stored?
  - Classify your Data
    - How important is it?
    - Can it be recreated?
    - Is this dataset currently in use?
    - Will this data be used by others?
    - Does this data need to be backed up?
  - Put your data in the appropriate space
    - Home Directory
    - Group Space
    - Scratch
    - /tmp
Considerations

- Backup Impact
  - Performance Characteristics
    - Time (when, duration)
    - Competition/concurrent access
    - Capacity of files backed up
    - Quantity of files backed up
    - Unintended consequences
Considerations

• Network Performance
Considerations

• Data Migration
  – What are we moving; What does it look like?
  – From where to where are we moving the data?
  – What transfer performance expectation do we have?
  – What tool will we use to make the transfer?
    • SSH/SFTP
      – Simple
      – Very portable
    • rsync
      – Restart able
      – File verification
    • tar via SSH
      – More efficient with many small files
    • Compression?
    • Secure
File Operations

• Directory Structure
  – Poor performance when too many files are in the same directory
  – Organizing files in a tree avoids this issue
  – Directory block count significance

• Network vs. Local
  – IOPS vs. Bandwidth
  – Network I/O
    • Overhead
    • Limited by network pipe
    • More efficient for bandwidth vs. IOPS
  – Local I/O
    • Limited size
    • Not globally accessible
    • Depending on hardware offers a fair balance between bandwidth and IOPS
File Operations

• Metadata Operations Performance Considerations
  – Create, Destroy, Stat, etc.
  – IOPS oriented performance

• Application I/O Performance Considerations
  – How often do we open and close files?
  – What I/O granularity do our applications write files?
  – Are we doing anything else silly?
Examples

• Code (user example)
  – Code wrote millions of small files (<10kB) to a single dir
  – Code wrote thousands of larger files (100s of MB)
  – Code uses a compression algorithm to speed up I/O of the larger files.

• Observations
  – Changing default r/w chunk of compression I/O from 16kB to 32kB/64kB improved performance 10%-20%.
  – Changing code to write files in a hierarchical directory structure produced a multiple times speed up.
Examples

- Single Directory vs. Hierarchical Directory Structure
Examples

- Processing Many Small Files over Network vs. Local

![Bar chart showing comparison between Network and Local processing times for 5000 3KB files. The Network has a real time of 125 seconds, with user and system times of 7 and 2 seconds, respectively. The Local has a real time of 16 seconds, with user and system times of 5 and 2 seconds.](http://www.chpc.utah.edu)
Examples

• Bonnie Test

![Bar chart showing performance metrics for different file systems and operations.](chart.png)

- Write with put_c unlocked
- Rewrite
- Write Intelligently
- Read with get_c unlocked
- Read Intelligently

48GB file
Examples

- Tar (Linux Kernel)
Examples

• Compile (Linux Kernel)
Examples

- Fine vs. Coarse I/O (Reads)

![Graph showing I/O performance vs. I/O size](image_url)
Examples

- Fine vs. Coarse I/O (Writes)
Troubleshooting

• Diagnosing Slowness
  – Open a ticket (issues@chpc.utah.edu)
  – File system
  – System load
  – Network load

• Future Monitoring
  – Ganglia

• Additional Information
  – http://www.chpc.utah.edu