SAG Proposal - Campaign Storage Native Compression

- Policy-driven GPFS file compression
  - Target files older than 6 months (based on GPFS creation time) for "z" compression
    - and files larger than 100MB
      - >100MB = 92% of data eligible, ~45M files out of ~710M
      - >1GB = 73%, ~10M files
      - >10GB = 30%, ~500K files
  - Weekly run (TBD)
    - Filesystem scan and file compression
    - Can re-evaluate timing based on usage
  - Compressed files reads are transparent for the user
  - No policy-driven decompression!

- Next steps...
  - SAG Feedback
  - Documentation: ls -l, du/stat behaviors
  - Notify users
  - Test and implement policy on csfs1

Backup - GPFS Compression

•••

Overview and Usage

Joey Mendoza 2022
CISL/HPCD

Goals

- Share a basic overview of GPFS compression
- Decide how best to implement for Campaign Storage
  - User-driven
  - Admin-driven
  - Hybrid
- Docs
  - https://www.ibm.com/docs/en/spectrum-scale/5.0.5?topic=systems-file-compression

Overview

- Compression Types
  - z (default)
    - Cold data. Favors compression efficiency over access speed
  - lz4
    - Active, non-specific data. Favors access speed over compression efficiency.
  - zfast / alphae / alphah
    - Genomic data formats: FASTA/SAM/VCF/FASTQ
  - no
    - Turns compression off
- Usage
  - User-initiated
    - On: `mmchattr [-I defer] --compression {z|lz4} <filename>`
    - Off: `mmchattr [-I defer] --compression no <filename>`
  - Targeted by admin policy
    - policy run to target files
  - Instant vs. Deferred
    - Instant: cmdline returns after operation completes, like gzip. Can take minutes for a multi-GB file.
      - Performed on client node
    - Deferred: file marked to be compressed, will occur during admin mmapplypolicy run
      - Performed on GPFS NSD server
      - Not running today
- Using compressed files
  - Compressed files being read are decompressed by the GPFS client (casper/ch nodes)
  - Compressed data is always sent over the network
  - GPFS NSD servers only do [de]compression during policy run

…Overview

- Compressed files
  - Files compressed in groups of 1-10 blocks (8MB each), if savings is < 10% no compression occurs for that group
  - GPFS does tricks when mmap or Direct I/O is used (performance hit, even more than normal)
    - mmap: decompresses the part of the file being accessed, that part remains illCompressed and will be re-compressed on a deferred policy run.
    - direct I/O: converts into buffered decompressed I/O internally
  - Small files not compressed: < 2 subblocks (32KB for fs1/csfs1)
  - Performance hit seems to be based on compressibility
- Checking Files
  - mmlsattr -L <filename>
    - Relevant flags:
      - COMPRESSION
        - The users intent as to if the file should be compressed
      - IllCompressed
        - Shows if the file is partially or not compressed (compression either in progress or is deferred, or mmap)
      - So a properly compressed file will only have the COMPRESSION attribute
  - File Sizes
    - "ls -l" still shows original file size
    - du/stat can show the compressed size and compressed # of blocks (--apparent-size for orig)
    - gladequotas uses the compressed size (and GPFS quotas in general use the compressed size)

Ways to use it

- User driven
  - mmchattr command
  - Could make a wrapper for compress/decompress, specify a directory, check compression status/ratios
    - mmchattr is the only tool for users provided by GPFS
  - Still need to handle deferred/mmap on admin side
- Admin driven policies (like scratch purge)
  - At the very least needs a policy to handle deferred user [de]compression, or else it won't happen
  - "Automatic" - Possible to set at fileset level, but will only mark new files for deferred compression
  - Compress and decompress? Or just one?
  - Examples
    - Compress files that haven't been accessed in X days
    - Decompress files that have been accessed within X days
    - More: created older than X? belongs in fileset X? low/high file heat? above size X? filename ends in X?
    - Any POSIX attribute...
  - Issues
    - The scheduling of the policy and compression runs: weekly, daily?
    - Users expecting files to be in one state, but the policy hasn't run yet
    - Users jobs running slower due to using compressed files (and not realizing it)
    - Automatic decompression can make projects unexpectedly exceed quota

Some Results

Using default 'z' compression...

- MMM data
  - number of files: 3,170,600
  - uncompressed size: 237 TB
  - compressed size: 208 TB (12% reduction in blocks used)
  - aggregate read rate for uncompressed data: 4.4 GB/sec
  - aggregate read rate for compressed data: 2.9 GB/sec (34% reduction in throughput rate)
- CGD data
  - number of files: 822,265
  - uncompressed size: 235 TB
  - compressed size: 171 TB (28% reduction in blocks used)
  - aggregate read rate for uncompressed data: 7.7 GB/sec
  - aggregate read rate for compressed data: 1.9 GB/sec (75% reduction in throughput rate)
- RAL data
  - number of files: 52,412
  - uncompressed size: 160 TB
  - compressed size: 151 TB (6% reduction in blocks used)
  - aggregate read rate for uncompressed data: 6.3 GB/sec
  - aggregate read rate for compressed data: 5.6 GB/sec (11% reduction in throughput rate)

CS Access Stats (as of Aug 2021)

```
Directory: csfs1

Total Files: 669.56 M

Total Data: 43.8062 PiB

Scan date: 2021-09-15


Last Accessed           Data (%)            # Files (%)

-------------------------------------------------------------

<1 Month                 0 b (0.00%)              0 (0.00%)   (0's because this report was ran on data a month old)

1 Month           14.4486 PiB (32.98%)      158.99 M (23.75%)

6 Months           8.4635 PiB (19.32%)      255.99 M (38.23%)

1 Year            17.2893 PiB (39.47%)      215.52 M (32.19%)

3 Years            1.9665 PiB (4.49%)        28.11 M (4.20%)

5+ Years           1.6383 PiB (3.74%)        10.94 M (1.63%)
```

- 67% of all data not accessed in the last 6 months (29.3 PiB) - would be purged if this was scratch!
- 47% of all data not accessed in the last year (20.79 PiB)