FILESYSTEM SHRINK

ALLISON HENDERSON
ORACLE

WHAT IS FILE SYSTEM SHRINK?

- Shrink reduces physical space of the file system on disk.
- Requires a potentially intensive reconstruction of how and where the blocks are allocated.
- Requires significant developer effort to implement

SHRINK IN XFS AND EXT4

- Shrink is currently unsupported in XFS.
- Ext4 has offline shrink, and a proposed lazy online shrink.
- Both efforts still in development.

XFS SHRINK WORKAROUND

- Current workaround is to create a new, smaller files system, copy the data, and remove the larger one.
- Requires enough space for both file systems.
- Could add a feature to mkfs to simplify this process.

FSTRIM ON THIN PROVISIONED VOLUMES

- Fstrim releases unallocated space within the file system
- Does not move or shrink existing allocations.
- Requires thin provisioning to reclaim unused blocks

PROTOFILE, ANOTHER WORKAROUND

- Proto files are used to auto generate a root file system.
- This can be used to generate a small root fs, and avoid needing to shrink.
- Maybe useful for the case of root fs creations.

HOW MUCH SPACE IS NEEDED

- How much do users need to shrink a file system?
- If we need to shrink a lot, might need more complex features
  - rmap
  - parent pointers

ERROR REPORTING AND STATFS

- Shrink can disk usage reporting issues with statfs.
- Need to limit new user block allocations while shrink in progress
- Operations may “fail” as though the disk were full

OTHER FEATURES FOR SHRINK

- Other features could facilitate shrink:
  - Reflink
  - reverse mapping
  - parent pointers.
- Could reduce time/operations/dev effort needed
- Some of the features are still under development.

QUESTIONS TO THINK ABOUT

- What are your use cases for shrink with XFS/EXT4?
- Clone util for mkfs?
- Thin provisioning with fstrim an option?
- Proto files for root fs creation?
- By how much do you need to shrink?
- Care about statfs issues during shrink?