Trading CPU Cycles for Gigabytes: Data Reduction Approaches for Archival Storage

Mike Davis

Dell | Ocarina
Collections

Preserve and keep available

Budget for Disk

Preservation only

Throw it away

Time
A new type of HDD with 2-3x the capacity at the same price

The ability to reduce WAN replication traffic by 2-3x

The ability to reduce platform migration time by 2-3x

YMWV
Data Reduction is...

**Dedupe** = Eliminating redundant data, within an object or across objects

**Compression** = Using math to predict data stream patterns
Some Ways to Shrink Data Better

PDF = Deflate \{ text, images, other \}

LZ77 \{ text \} + JPAQ \{ images \} + BBB \{ other \}
5x Better Utilization is Worth A Serious Look
It’s Not Just About Better Algorithms

1. Shrink data well
2. The shrinking needs to happen somewhere appropriate
   - As part of the storage layer
   - A host properly sized for the workload
     › CPU cores for compression, RAM for dedupe
   - Running it: Management, resilience, and policies
3. Transparent to applications and end-users
   - Don’t change file system metadata (2/18/80 rule)
   - Use file mover APIs where available
   - Performance asymmetry favoring read operations
4. Needs to introduce minimal new risk
   - Self describing wrappers
   - Run-anywhere decoding
   - Market reliability: vendor lifecycle, escrow, etc
Applying **Lossy** Compression to Preservation?

An “Object Fidelity Lifecycle”? 

- **Scanned Pages**
  - TAR{TIF}
  - 1x

  ➔ 2 years of inactivity

  ➔ **Annotated OCR**
  - PDF
  - 1/100x

  ➔ 2 years of inactivity

  ➔ **Text**
  - LZ{TXT}
  - 1/1000x
Thank you!