SDC
STORAGE DEVELOPER CONFERENCE
SNIA ■ SANTA CLARA, 2015

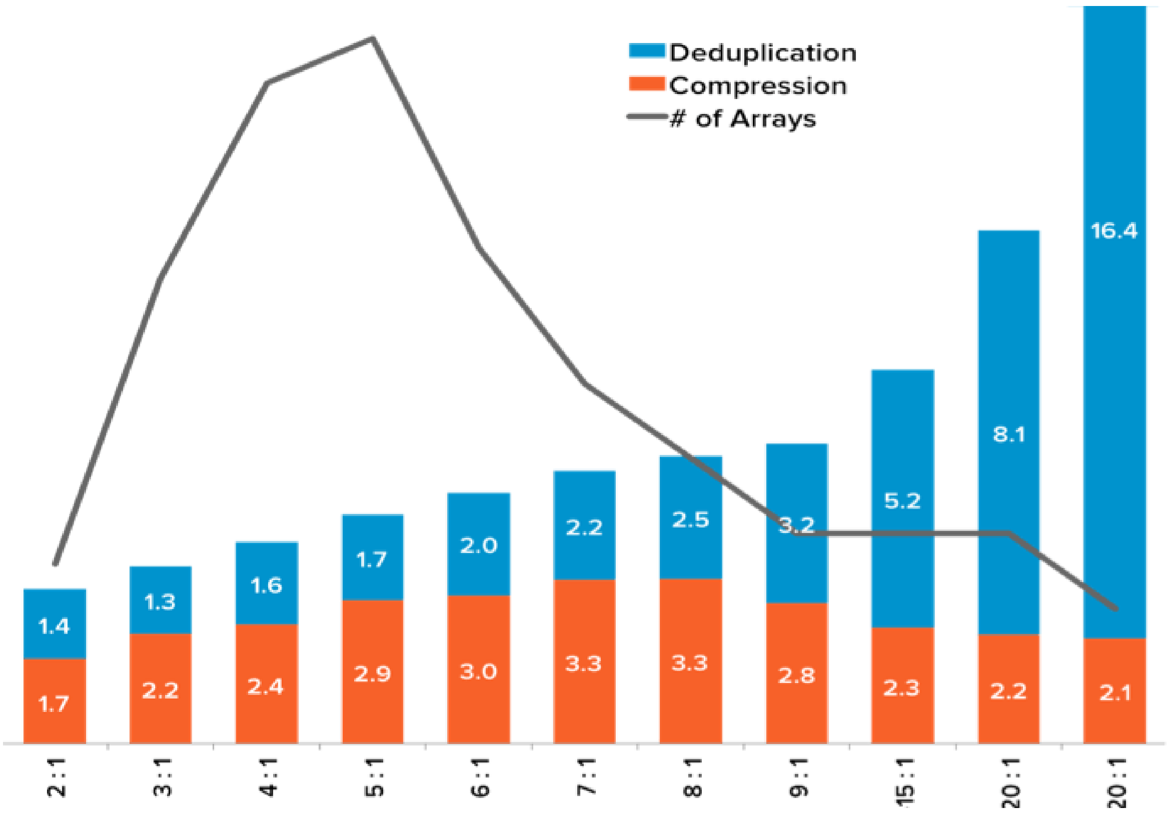
Design Decisions and Repercussions of Compression and Data Reduction in a Storage Array

Chris Golden
Pure Storage

Data Reduction

- Huge performance impact on disk
- Very little performance impact on flash
- Table stakes feature for all-flash storage arrays
- Goals
  - Increase usable capacity
  - Extend media lifetime
- Many different ways to implement
  - These design decisions have ripple effects

Dedup + Compression – Better Together



Actual results from entire Pure Storage install base

2X savings over 4KB dedupe alone

True data reduction, not thin provisioning

Dedup vs Compression

- Compression – A local process reducing the number of bits required to represent data
- Deduplication – A global process ensuring that identical data is only written once

Block Size + Alignment

- All arrays have 3 block sizes and alignments
  - Client, Array Device
- Size mismatches and/or unaligned operations create read / write amplification in the underlying layer
    - Client > Array > Device

Data Protection

- Our choice of data protection matters
  - Mirroring, RAID5, RAID6, etc.
- Should be designed with compression and deduplication in mind

Compression

- ❐ Reduces the number of bits required to represent the original data
- ❐ Final representation is variable sized
- ❐ We can apply more CPU for better results

Compression – How to Choose

- Variables to optimize
  - Compression Speed
  - Decompression Speed
  - Size of representation
- Variables that can be controlled
  - Choice of algorithm, level
- Variables that change with workload
  - Compressibility of the data

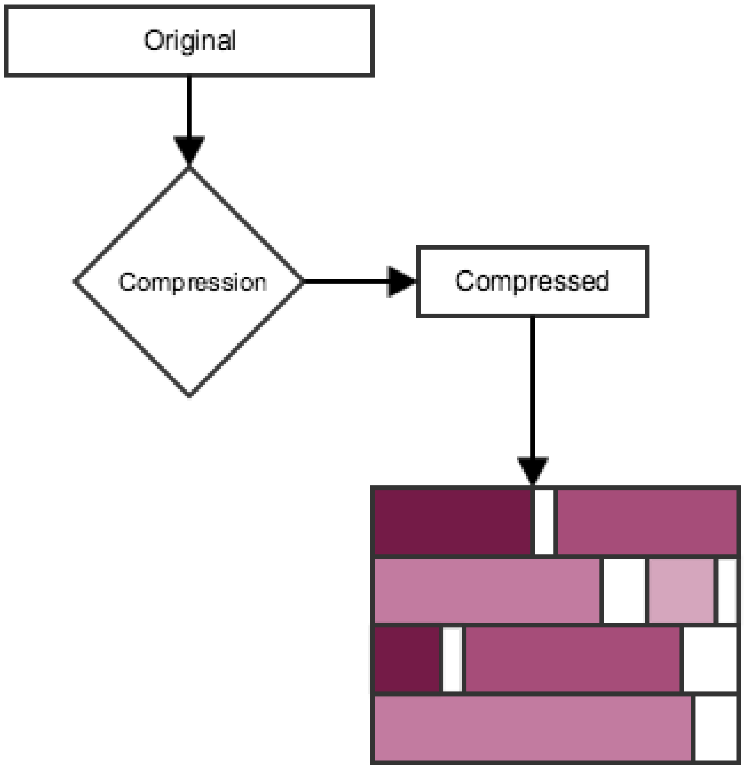
Compression – How to Store

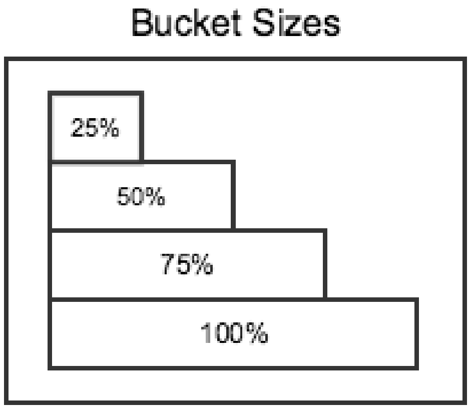
- Full Precision
- Nearest Size Bins
- Large Bins with Many Blocks

Full Precision

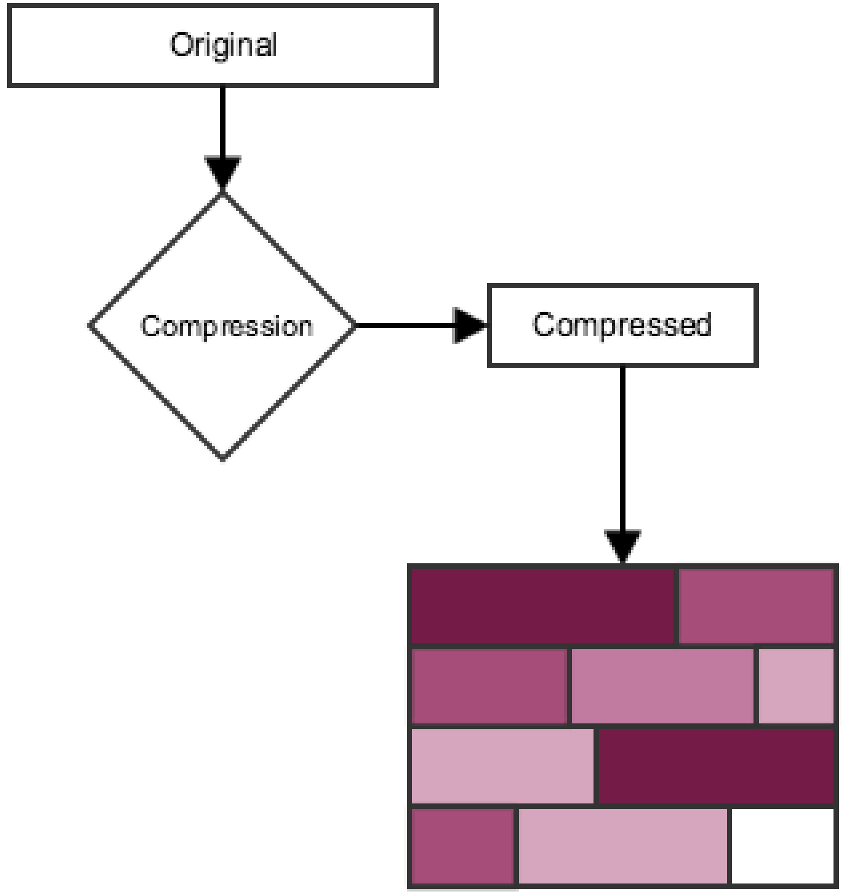
- Find free space on the underlying devices exactly matching the compressed size
- Global fragmentation problem
- Overwrites have 3 outcomes, 2 are bad
  - New data is larger, must find a new place
  - Data is smaller, extra space may be wasted

Details – Nearest Sized Bin





Details – Large Bins



Deduplication

- Identify duplicate bits and store them a single time
- Cryptographic Hash vs Weak Hash and Verify
- Fixed Size vs Variable Sized

Details – Cryptographic vs Weak Hash

- Cryptographic hash
  - More CPU required
  - More metadata required
- Weak Hash
  - Collisions must be resolved via device read

Details – Fixed vs Variable Size

- Fixed size
  - Fewer hashes to calculate
  - Alignment of user writes matters
- Variable size
  - More metadata required
  - Choice of compression becomes important

Variable Length Dedupe + Compression

- Creates read amplification and CPU overhead
  - Must read and decompress entire block
- Overwrites also get more complicated

- TODO: Add diagram

Compression + Dedupe

- Order of operations matters
- Compress first
  - Might require less CPU
  - Works well with fixed sized compression and dedupe
- Dedupe first
  - Can find duplicates in the middle of compressed blocks
  - Works well for variable dedup

Space Reporting

- Compression and dedup make space reporting complicated
- Hard problems
  - Identifying shared space and reporting
  - Predicting future capacity
    - How much space will I reclaim if I remove this data?
    - How much additional data can I store?

Performance

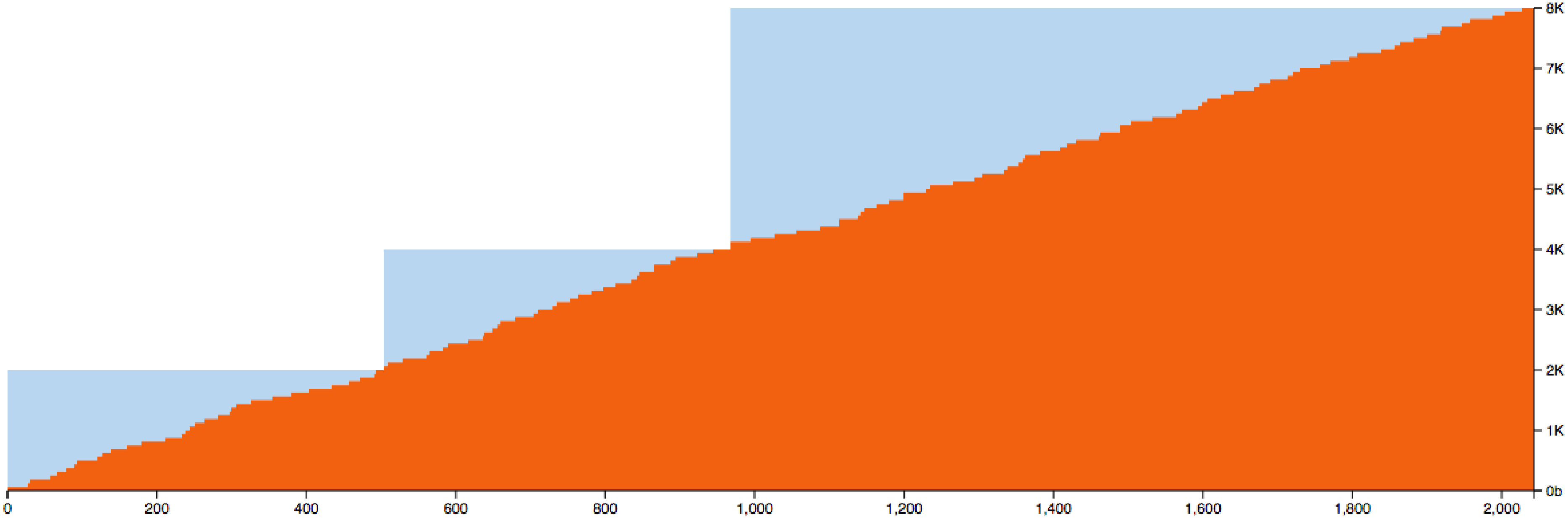
❒ We're skimming some performance off the top to enable greater space utilization

❒ Compression eats CPU cycles

❒ Dedup eats CPU cycles and may create hotspots

❒ Read / Write amplification

PLACEHOLDER - Analysis of a Sample Data Set



Wrap Up

- Different ways to implement data reduction
- Each implementation has implications on the rest of your system design
  - Performance
  - Economics
  - Ease of implementation

Thank You!

Questions?