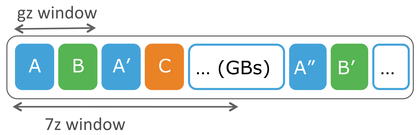
Migratory Compression

Coarse-grained Data Reordering to Improve Compressibility

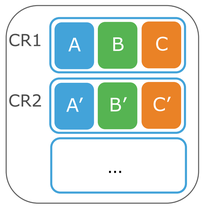
Xing Lin*, Guanlin Lu, Fred Douglis, Philip Shilane, Grant Wallace
*University of Utah, EMC - Data Protection & Availability Division

MOTIVATION

Compress a single, large file: traditional compressors use small windows and can't find similarity across a large range



Migrate data for long-term retention: similar data may be in different regions.

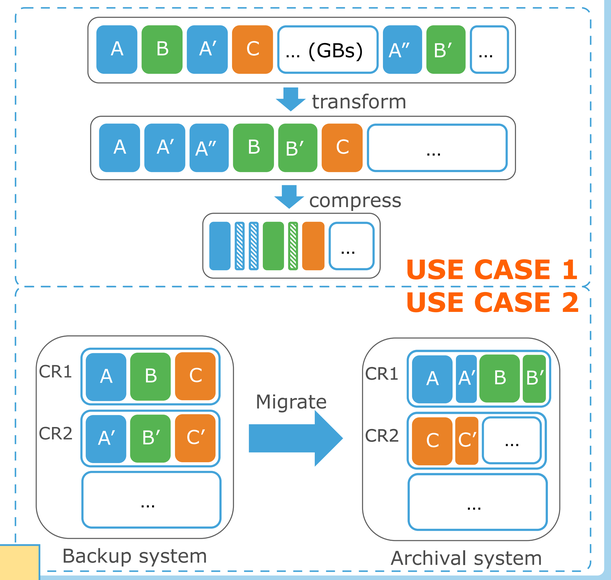


MIGRATORY COMPRESSION (MC)

Idea:
coarse-grained reorganization to group similar blocks to improve compressibility

Benefits:
- A generic pre-processing stage for any standard compressors
- Improve compressibility and sometimes throughput

Challenges:
- *Similarity detection:* similarity feature (based on [Broder 97])
  - A strong hash for duplication detection
  - Weak hashes provide hints about similarity among blocks
- *Data Reorganization:* re-arrange the input data, to group similar blocks



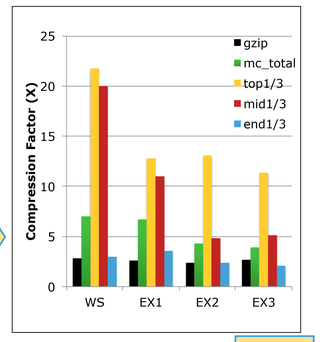
USE CASE 2
MC FOR ARCHIVE STORAGE

Tradeoff: price over performance

DDFS system uncompresses LZ, then recompresses as GZ for archival: 25-44% better CF

With MC: identify and compress similar blocks together
- Identify and sort by cluster sizes of similar blocks
- Migrate in K passes: K is determinted by storage to buffer ratio; largest clusters in the first phases, then progressively smaller clusters

Datasets
WORKSTATIONS; Exchange[123]



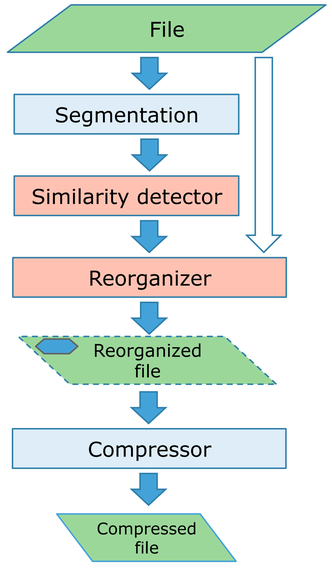
✓ MC improves CF
- [44% (EX3), 157% (WS)]
- Top 2/3 compresses very well

✓ Migrate Runtime
- EX1: 3X compared with gzip

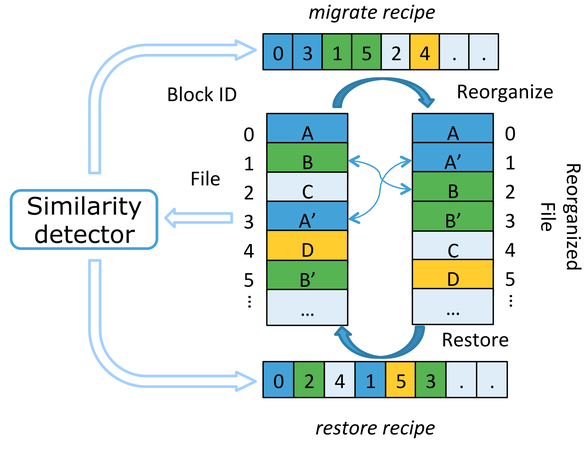
✓ Read Performance
- Entire EX1: 1.3 X
- Final backup of EX1: 7 X (1.24 X if only reorganize the top third

USE CASE 1
mzip: MC FOR COMPRESSING A SINGLE, LARGE FILE

Workflow



Example



- *Segmentation*: partition into blocks, calculate similarity features
- *Similarity detector*: identify duplicate and similar blocks; output migrate/restore recipe
- *Reorganizer*: rearrange the input file

Evaluations (in-memory)

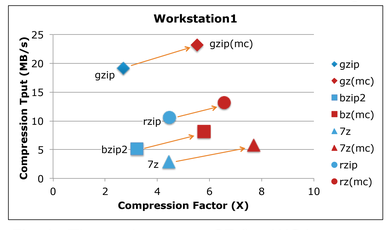


Fig 1. Throughput vs. CF for WS1

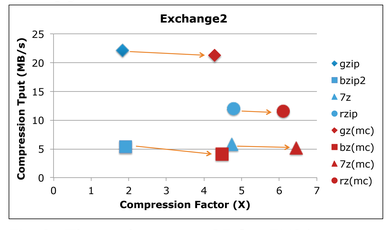


Fig 2. Throughput vs. CF for EX2*

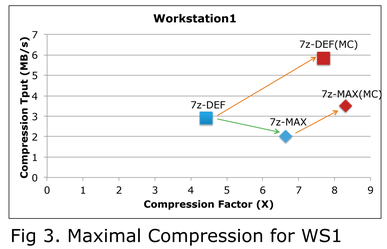


Fig 3. Maximal Compression for WS1 with/without MC

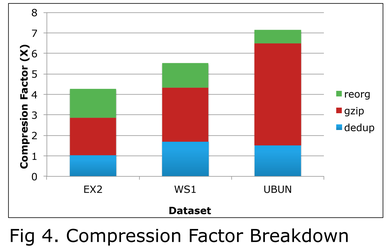


Fig 4. Compression Factor Breakdown

* Deduplication factor for exchange2 is very low, thus the overhead in doing MC becomes evident.

SUMMARY

Migratory Compression preprocesses data to make it more compressible
- Identify and cluster similar data

mzip
- Improves existing compressors, in both **compressibility** and frequently **runtime**
- ***Redraw the performance-compression curve!***

Archival storage
- MC reduces $/GB further