Deduplication

Leonhard Reichenbach

Arbeitsbereich Wissenschaftliches Rechnen
Fachbereich Informatik
Fakultät für Mathematik, Informatik und Naturwissenschaften
Universität Hamburg

2019–01–24

Agenda

The Problem: Data

There is a lot of it

- Processing power grows faster than storage/network capacity and speed
- Storage space is limited and expensive in acquisition and maintenance
- Experiments produce rapidly increasing amounts of data
- Idea: exploit redundancies, only store unique blocks

Deduplication

- Data is partitioned into chunks
- Each chunk receives a fingerprint (hashing)
- Only chunks with a unique fingerprint are stored
- Original input is reconstructed if read

Why do we not just use compression?

- Compression only uses redundancies inside a file
- Deduplication can exploit redundancies across multiple files
  - Even when they are incompressible on their own
- Both can be used together

Difference to Compression

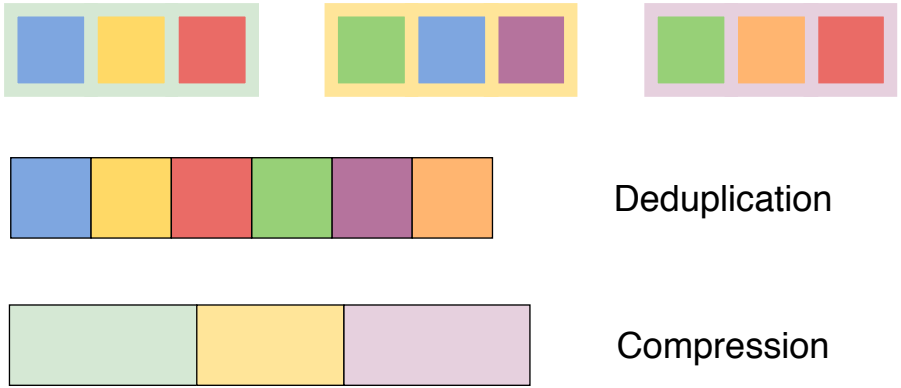

Figure: Comparison between deduplication and compression for three stored files

Essential Steps

- Chunking
  - Static or dynamic
  - Content-defined, boundary-aware
  - Size typically between 4 and 16 KB
  - Smaller chunks are more likely to be duplicates but cause more overhead
- Fingerprinting
  - Cryptographic hashes e.g. SHA-256
  - Optional byte-wise comparison

Writing I

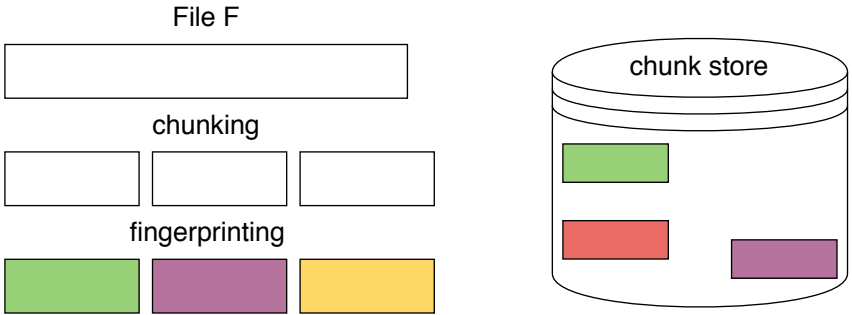

Figure: Static chunking followed by fingerprinting

Writing II

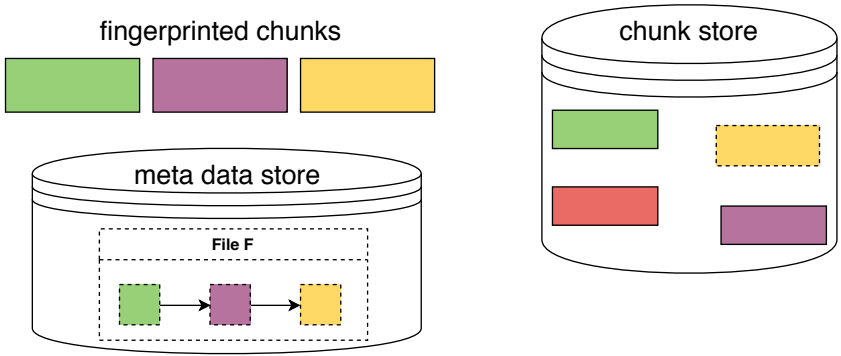

Figure: Storing of new blocks and file meta information

Reading, Modifications and Deletions

- Reading:
  - The meta data store is checked for the corresponding chunks
  - The chunks are gathered from the chunk store and the file is reconstructed
- Modification:
  - Chunks belonging to more than one file are copied first
  - This is called copy-on-write
- Deletion:
  - Chunks only belonging to the deleted file are deleted from the chunk store

Online and Offline Deduplication

- Online deduplication:
  - The deduplication step is done directly on write
  - Can be done before or after the data is sent over the network of a distributed file system
  - Online client deduplication would be optimal to avoid network and storage bottlenecks
  - Used by e.g. FUJITSI ETERNUS AF series
- Offline deduplication:
  - The file system is scanned in regular intervals for files eligible for deduplication
  - Used by e.g. Windows Server

Content-Defined Chunking I

- Dynamically sized chunks
- Border defined by a (rolling) hash function
- Minimum and maximum chunk size can be supplied optionally

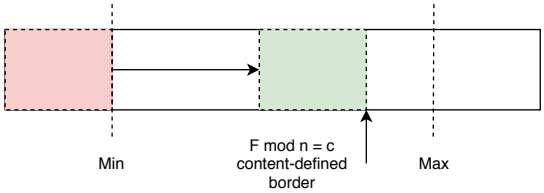

Figure: Sliding-window approach

Content-Defined Chunking II

- If the content is moved, the chunk moves

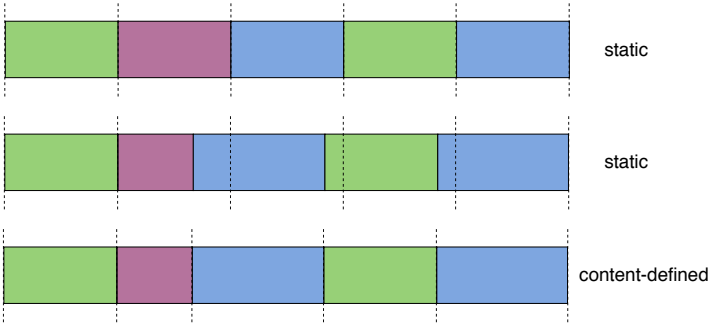

Figure: Comparison between static and content-defined chunking

Further modifications

- Boundary-aware chunking for e.g. tar files
  - Utilizes information about the file format (tar header)
  - Can increase the deduplication ratio further, depending on the data
  - Increased the deduplication by a factor of 3 over content-defined chunking in [Sun+]
- Usage of simpler hash functions to reduce the lookup structure overhead
  - Might require additional collision handling or byte-wise comparison

Complications in Practice

- The meta data store needs to be kept in RAM for reasonable access times
- With SHA-256 only storing the hashes takes 4 GB per TB of data (8 KB chunks)
- For 54 PB of data 221 TB of RAM are necessary
- The Mistral supercomputer would need its whole RAM to hold the meta data store
- Deduplication increases the power consumption

Analysis for HPC Datasets [Mei+12]

- Analyzed over 1212 TB of data from HPC environments
- HPC datasets contain approximately 15–30% redundancies
- The majority of the redundancy is found in the local scope of the projects
- Static chunking detects 6–8% less redundancies than content-defined chunking

Summary

- Deduplication can utilize redundancies across multiple files
- In theory, a possibility to circumvent network and storage bottlenecks
- HPC datasets would be viable candidates for deduplication
- Online deduplication requires vast amounts of additional RAM
- Offline deduplication is successfully used in cloud and backup environments

References I

[Mei+12] Dirk Meister et al. “A study on data deduplication in HPC storage systems”. In: *SC Conference on High Performance Computing Networking, Storage and Analysis, SC '12, Salt Lake City, UT, USA - November 11 - 15, 2012*. Ed. by Jeffrey K. Hollingsworth. IEEE/ACM, 2012, p. 7. ISBN: 978-1-4673-0804-5. DOI: 10.1109/SC.2012.14. URL: https://doi.org/10.1109/SC.2012.14.

[Sun+] Baegjae Sung et al. “An efficient data deduplication based on tar-format awareness in backup applications”. In: