УДК 004.62
Гарасівка А. - асп. гр. СПс-31
Тернопільський національний технічний університет імені Івана Пулюя

ПОРІВНЯННЯ МЕТОДІВ ЗМЕНШЕННЯ ВИКОРИСТАННЯ ДИСКОВОГО ПРОСТОРУ СЕРВЕРНОГО СЕРЕДОВИЩА

Науковий керівник: д. т. н., доцент, Лупенко А. М.

Harasivka A. V.
Ternopil Ivan Puluj National Technical University

COMPARISON OF METHODS FOR REDUCING STORAGE UTILIZATION IN SERVER ENVIRONMENT

Supervisor: Lupenko A. M., Dr.

Ключові слова: машинне навчання
Keywords: machine learning

The rapid increase in digital data generation has led to significant challenges in storage management. Many information-technology-related companies are seeking effective and secure methods to reduce storage use without compromising data accessibility and integrity. Various methods of reducing storage utilization include deduplication, partitioning, and compression.

Data deduplication [1] is a highly effective method to reduce storage space consumption by eliminating the need for storing identical files or data blocks multiple times. Instead, only one copy of each unique data block is stored, and references are used to the location of the original copy. This method has low efficiency in cloud environments where vast amounts of unique data are typically stored. However, in the storage of data backups or backup applications, deduplication can reduce storage needs by up to 90 − 95% [2], while in standard file systems, it can lead to a reduction of up to 68% [3]. There are 3 main types of data deduplication methods based on granularity [4]: file, fixed-size block, and variable-sized block. File-level deduplication finds and removes entire duplicate files. Fixed-size block deduplication divides a file into fixed-size blocks and eliminates duplicate blocks. Variable-sized block deduplication utilizes different sizes of chunks to identify redundant data, but it will create more metadata and lead to hash collisions. Block-level deduplication is typically more efficient as it can detect duplicates even if they are stored across different files or portions of the storage system.

Data compression reduces the size of data by encoding it with a specific algorithm. Compression methods can be applied to almost all types of data, including text, images, and videos, but not effective for already compressed or encrypted data. Archiving provides immediate storage savings, but is quite CPU-intensive, especially with high-compression algorithms. Data saving will depend on the entropy of the data, and could reach 47% for data tables [**Ошибка! Источник ссылки не найден.**], 75% for video files[5], and 90% or more storage saving for the text data [7].

Data sharding is a process of reducing extra data in sets such as databases. Sharding refers to breaking up a database into smaller, distributed databases or nodes (shards). It helps distribute data across multiple servers, which can prevent individual servers from becoming overloaded with data. This method often results in more efficient use of resources, especially

when dealing with extremely large datasets. In scenarios where data is replicated across nodes (for redundancy), sharding can help reduce the replication overhead by allowing a more granular distribution of data across multiple storage systems. For instance, databases with redundant or similar data (like logs or time-series data) can achieve significant storage reductions using compression, Cassandra (NoSQL Database) can reduce storage requirements by 30-50% across multiple nodes due to better distribution and replication control [8]. Additionally, time-series databases often see better storage efficiency after implementing partitioning and sharding because older data can be archived or compressed without impacting performance.

The most optimal way to implement one of the storage-saving methods mentioned above – use a specific filesystem: ZFS and Btrfs for Linux OS (optional block-level deduplication; transparent LZ4, GZIP, ZSTD compression), SDFS (OpenDedup) for Linux offers block-level deduplication for large-scale storage systems [3], ReFS in Windows Server OS, APFS in Mac operating system (native compression for system files, copy-on-write to avoid duplicate writes).

In the particular case of backups, there are a few storage saving solutions: Veeam Backup & Replication offers block-level deduplication and integrated compression; Veritas NetBackup & Backup Exec could perform deduplication across multiple storage locations: cloud, tape, and disk-based storage; Commvault has source-side and target-side deduplication.

Aside from operating systems – cloud solutions also provide the feature of deduplication: AWS S3, offers intelligent automated storage tiering with deduplication in object storage. Google Cloud Storage with Nearline & Coldline offers deduplication at the object level.

In conclusion, there are multiple ways to reduce storage utilization. Deduplication is ideal for environments with high data redundancy (backups, virtualization). Compression is useful when data size needs to be reduced immediately. Partitioning is more about optimizing data access speed and organization rather than reducing total storage space.

ЛІТЕРАТУРА

1. Xia W., Jiang H., Feng D., Douglis F., Shilane P., Hua Y., Fu M., Zhang Y., Zhou Y., A comprehensive study of the past, present, and future of data deduplication, Proc. IEEE, vol. 104, no. 9, pp. 1681–1710, Sep. 2016, http://dx.doi.org/10.1109/JPROC.2016.2571298.
2. Meyer D. T., Bolosky W. J., A study of practical deduplication, ACM Trans. Storage, vol. 7, no. 4, pp. 1–20, Jan. 2012, http://dx.doi.org/10.1145/2078861.2078864.
3. OpenDedup. Accessed: Aug. 6, 2023. Available: http://opendedup.org.
4. Patra S. S., Jena S., Mohanty J. R., Gourisaria M. K.. DedupCloud: An optimized efficient virtual machine deduplication algorithm in a cloud computing environment. Data Deduplication Approaches, 2021. https://doi.org/10.1016/B978-0-12-823395-5.00009-4.
5. Quan L., Ziling H., Kun C. and Jianmin X. Efficient and Real-Time Compression Schemes of Multi-Dimensional Data from Ocean Buoys Using Golomb-Rice Coding. MDPI, Mathematics, 2025. https://doi.org/10.3390/math13030366.
6. Bell C., Klinton B.. Data compression in high-resolution video streaming, ICTACT Journal on Image and Video Processing, 2024
7. Liu H., Chuang C., Lin C., Chang R., Wang C., Hsieh C. Data Compression for Energy Efficient Communication on Ubiquitous Sensor Networks. Journal of Science and Engineering, Vol. 14, No. 3, pp. 245-254 (2011).
8. Reducing Apache Cassandra® Disk Usage via Schema Optimization. Accessed: Feb. 22, 2025. Available: https://medium.com/open-source-journal/reducing-apache-cassandra-disk-usage-via-schema-optimization-49deb6ed0f7d