The examination of Data compression as a solution for data storage and capacity optimization

BY

AMINE LASFAR

1ST BEXIS2 USER AND DEVELOPER CONFERENCE IN JENA (JULY 9-10, 2015)

Data in the science research space

- High cost of maintaining the Data
- High rate of Data growth
- High scalability

Results

- Big data bubbling up
- High price of I/O operations

Data movement

- I/O Bottlenecks
- Memory access speed, “The memory wall”

Results

Users will be waiting longer than they wanted when they queried the data

Available solutions - Hardware

- Acquisition of more servers, more CPU and RAM, less waiting time for users
- It’s not addressing the root problem (Database)
- Data would outgrow the hardware, again!

Available solutions - Software

- **NoSQL** (**B**asically **A**vailable, **S**oft-state, **E**ventually consistent)

  Will have to rewrite the main application

- **Upgrade Storage engine (MyISAM for Postgres)** (Compressing tables)

  Will result in read-only tables and few writes if any

- ***Compression***

Data Compression

- **What ?** Efficient storage
- **How ?** Eliminating Redundancy
- **Why ?** Optimization in terms of space

Results

1. **Efficient means of storing huge volumes of data**
2. **Less I/O to transfer**
3. **Enabling the use of complex queries**
4. **Performance gains**

Lossless Compression

- **Definition:** "*is used to avoid loosing any content between original data and compressed output; therefore, uncompressing it restores the data to its native state.*" [1]
- **Achieve high compression at low procession cost**
- **Significant savings when properly integrated with the DBMS**

[1] G. Held and T. R. Marshall, Data and Image Compression: Tools and Techniques.

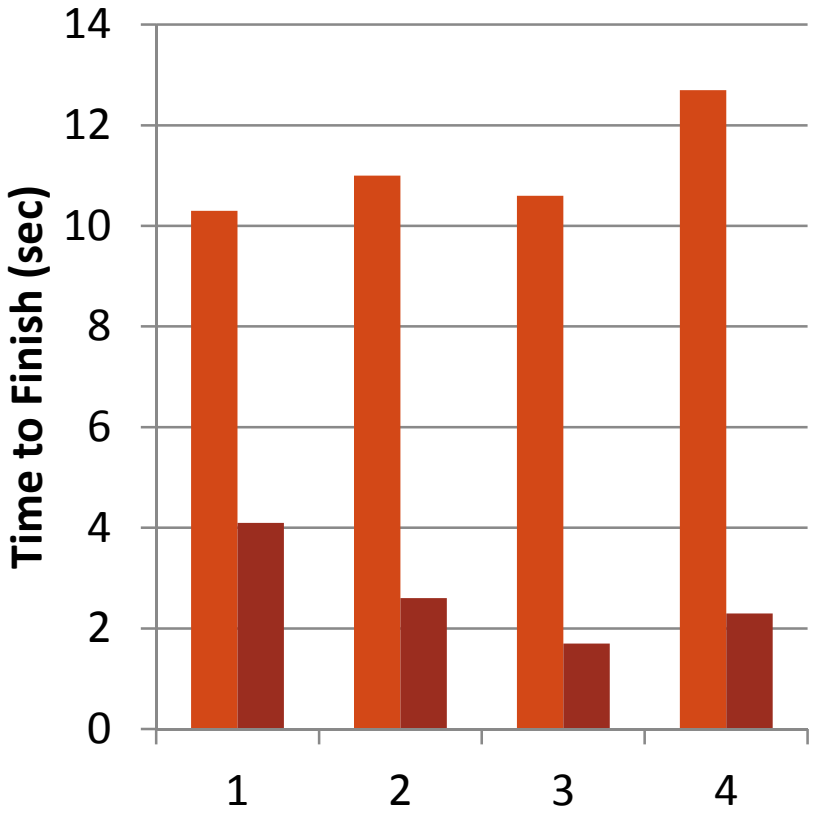
Row Compression

1. **Rows per page ratio**
    1. Queries need fewer I/Os for same amount of data
    2. Performance Improvements

2. **Tradeoff**

    Extra CPU cycle usage, but on modern systems no CPU bottlenecks in favor of eliminating I/O bottlenecks

Performance when compressed



User experience

- Original Data size: 21 MB
- Using R to access
- 4 tries (cache on the hard drive and in the operating system)

Results

~ 75% disk space saving and finishes

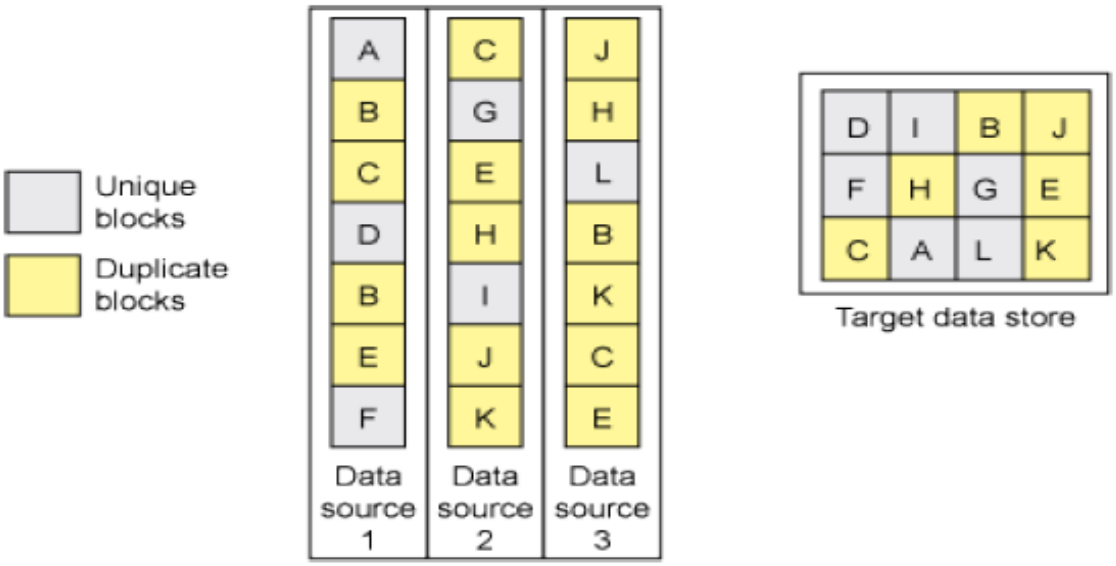
Deduplication

- **Definition:** "*Data deduplication is a specialized data compression technique for eliminating coarse-grained redundant data*" [2]
- **Efficient when applied to inactive data for backup**
- **Limit number of copies to gain more space capacity**
- **How ?**

By removing duplicated data blocks and leaving a pointer to the original data

[2] Dale McInnis, 14 February 2012, Integrated support for data deduplication devices inDB2 for Linux, UNIX, and Windows,

Deduplication



Integrated support for data deduplication devices in DB2 for Linux, UNIX, and Windows

Backup deduplication using DB2

Tivoli Storage Manager (TSM): native data deduplication client in DB2

- **Two options:**

  - TSM Server: start with transferring backup data to the storage pool, then reduction process starts, at the end the deduplicated data is removed

  - TSM Client: started during the backup process, and can be combined with compression to provide even more storage saving

  - TSM Hybrid: combination of both (Server and Client) by splitting deduplication subsets between both server and client at the same time

Best Practices

- Consider frequently accessed large tables
- Consider the cost of compression / decompression

Compression Steps

1. Identify potential tables for compression

2. Pre-estimate the compression ratio

3. Enable compression settings

   On DB2: *alter table TABLE_NAME compress yes;*

4. Monitor compression using statistics


Thank you!