IBM Data and AI

Db2 BLU Improvements

Satya Krishnaswamy, IBM

Nov 12th, 2021

© 2021 IBM Corporation

Agenda

- Db2 Catalog Features
- Db2 BLU Storage and Compression Features
- Db2 Runtime Features

Db2 Catalog Features

New in v11.5.7: SQL statement and DB administration enhancement

- For example, if we have a table 'lapis.t1 (c1 int, c2 int, c3 varchar(10))' and 'create index I1 on T1(c1, c1+c2, upper (c3))' following view will be created by the system:

  'CREATE VIEW “LAPIS “."I1_V"("C1", "K01", "K02") AS SELECT "C1", C1+C2, UPPER (C3) FROM "LAPIS. “."T1"'.
- The RENAME TABLE statement is now extended to support the renaming of tables with indexes on an expression.
- The ADMIN_MOVE_TABLE procedure is now able to move tables having an index with an expression-based key.

Problem addressed: Prior to having this feature, rename of tables which had indexes with expression defined on them involved dropping the indexes, renaming the table, and recreating the indexes back. This was very time consuming operation with impact on performance (statistics needed to be recreated also). Introduction of capability of renaming tables with indexes on expression enabled customers to avoid all the issues related to indexes recreations.

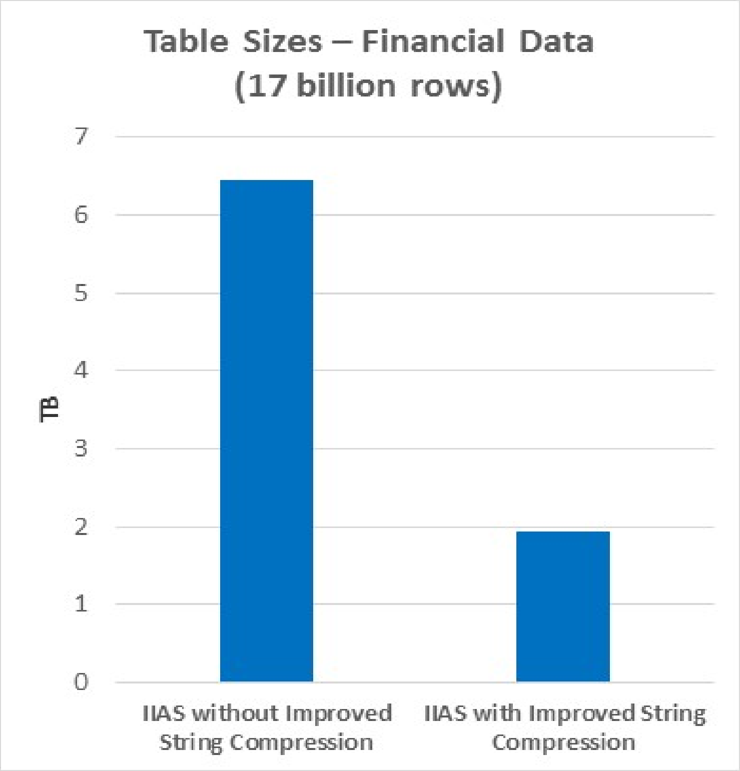
Db2 BLU Storage and Compression Features

11.5.4, 11.5.5 and 11.5.6 New Features Delivered

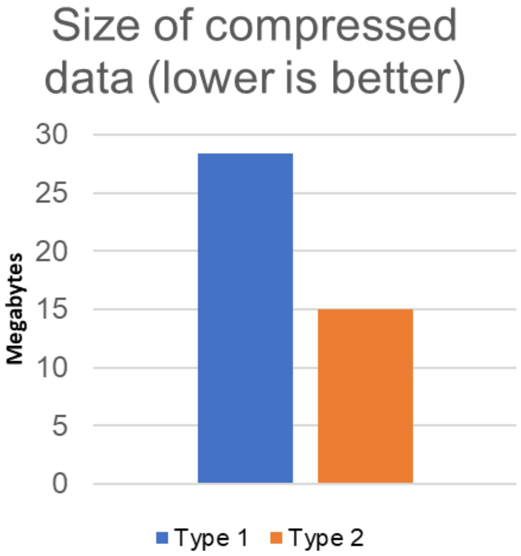
Feature	Problem Addressed	Release
Page-Based String Compression Type 1	Low compression for high cardinality strings that were not encoded by existing compression algorithms. Frequency-based compression not effective for high cardinality string datasets so percentage of values encoded < 10%.	11.5.4
Page-Based String Compression Type 2	Low compression for hex, date, timestamp, numeric data stored in string data types even when page-based string compression Type1 applied.	11.5.4
Deferred Synopsis Tuple Creation for Small Base Tables	Synopsis tables can consume significantly more storage than small base tables due to unused pages in extents.	11.5.4
Reorg Table Recompress Enhancement	Reorg Table Recompress speed and compression ratio could be improved since page-based string compression was not being used.	11.5.5
Trickle Feed Insert	Trickle feed inserts were slow, consumed too much memory, and small tables consumed too much storage space.	11.5.6 (Warehouse Only)

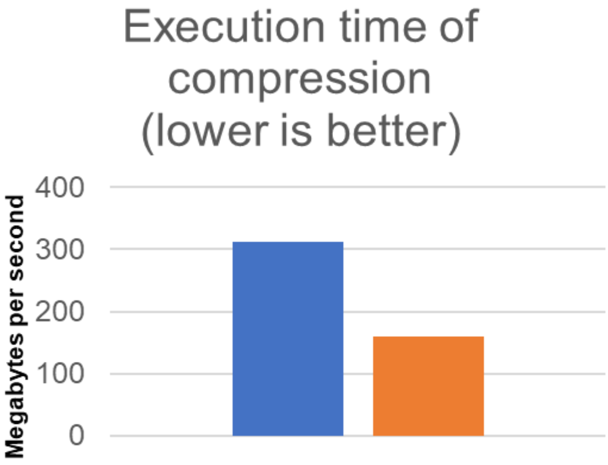
In v11.5.4: Page-Based String Compression Type 1

- New page-based compression algorithm instead of dictionary-based compression algorithm
- Looks for repeating patterns within string data being inserted
- Significantly improves compression of string data types, especially for columns with high cardinality that have many unique values
- Handles long common prefixes well



Page-Based String Compression Type 2



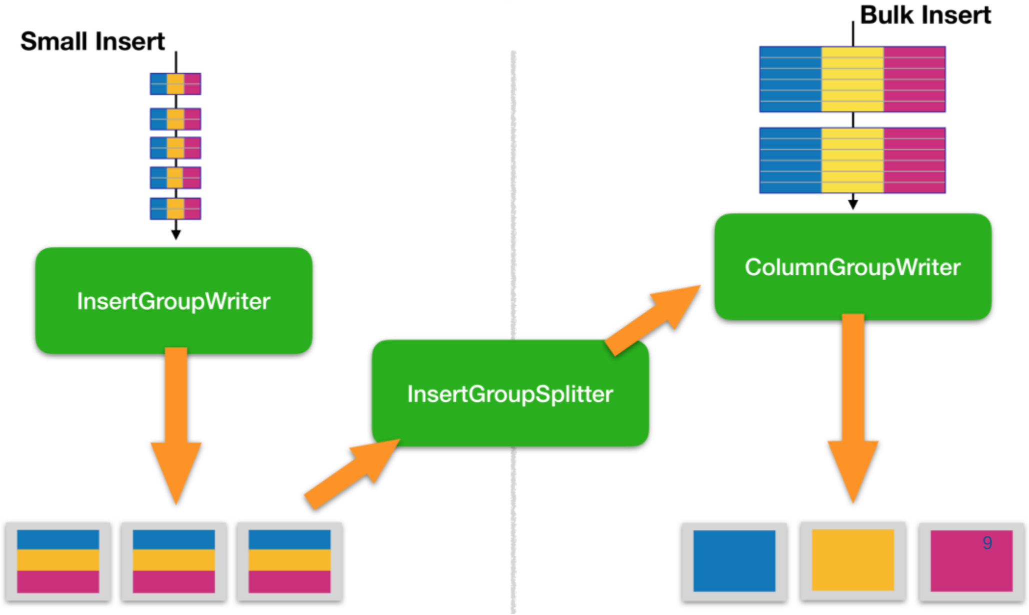


In-house performance measurements compare compression of 3 million randomly generated 10 character phone numbers using page-based string compression types 1 and 2.

New in v11.5.6: Trickle Feed Insert Performance Enhancement

Goals:

- Speed up “trickle” inserts and reduce memory footprint.
- Decrease size of very small tables.



New in v11.5.6: Trickle Feed Insert Performance Results

- **Dramatic reduction in table size for small tables**
  - Helps customers who are sensitive to storage consumption
  - Example: TPCDS STORE table reduced from 1.5GB to 130MB in a 1/3 Rack IIAS
- **Results in customer insert scenarios**
  - Insert performance enhancement
    - 7% improvement for in-house massive insertion IBM Data Analytics Accelerator workload
    - 4% improvement for large trickle workload (thousands of rows) for a healthcare provider
    - 16-23% improvement for Insert from Subselect
  - Dramatic reduction in log volume and dirty page writes
    - A large bank workload showed 78% log reduction and 44% less page writes
    - A healthcare provider workload showed 70% log reduction
  - Page-based string compression for trickle inserts
    - Improved from zero compression to 2.5x compression for unencoded string data for TPCH CUSTOMER table
- **Industrial query benchmark results**
  - TPCH – query performance not impacted with data inserted through trickle feed
  - TPCDS – same results as TPCH

Db2 Runtime Features

11.5.4, 11.5.5 and 11.5.6 New Features Delivered

Feature	Problem Addressed	Release
Compact Varchar: Vectors and Workunits*	Improves efficiency of processing VARCHAR values that are significantly shorter than the schema defined width. Adds efficient storage infrastructure and uses that infrastructure in the Work Units and Vectors used to stored data flowing through evaluator chains during query processing.	11.5.4
Compact Varchar: Group by and Hash Join Keys*	Improves memory usage, spilling, and performance for Group By and Hash Join keys that involve VARCHARs that are significantly shorter than schema defined width. Builds on the Compact Varchar infrastructure added in 11.5.4 to efficiently handle VARCHARs.	11.5.5
Compact Varchar: Aggregation Distinct*	Improves memory usage, spilling, and performance for Aggregation DIstinct queries that involve VARCHARs that are significantly shorter than schema defined width. Builds on the Compact Varchar infrastructure added in 11.5.4 to efficiently handle VARCHARs.	11.5.6

**No external changes from runtime features. Added value is in improved performance and memory utilization primarily for queries involving VARCHAR that have data with actual width significantly shorter than schema defined width.*

Compact Varchar: GroupBy and HashJoin Keys

- **Available in v11.5.5**
- **PQA Tests on “targeted” Group By and Join queries involving…**
  - Large schema width VARCHAR columns
  - Relatively short data values
- **Improvements seen across all performance metrics**
  - **Stability:** No -901, -955, -968 errors with feature enabled: **Higher stability**
  - **Performance:** Execution times for targeted individual queries: **Up to 12x faster**
  - **Spilling buffer pool reduced: Up to 1200x less spilling**
  - **Spilling to disk improved: No spilling to disk for many queries with feature on**
  - **Customer example: A query ran in 35 sec that was 1-2 hours before feature**

Compact Varchar: AGG DISTINCT Performance

- **Available in v11.5.6**
- **PQA Tests on “targeted” AGG DISTINCT queries involving…**
  - Large schema width VARCHAR columns
  - Relatively short data values

- **Improvements seen across all performance metrics**
  - **Stability:** Zero -901, -955, -968 errors with feature enabled: **Higher stability**
  - **Performance:** Query execution time speed up for targeted queries: **2x – 28x faster**
  - **Spilling buffer pool reduced: Up to 64x less spilling**
  - **Spilling to disk improved: No spilling to disk when feature enabled**