

ENHANCEMENTS TO SQL SERVER COLUMN STORES

**Anuhya Mallempati
#2610771**

CONTENTS

Abstract

Introduction

Column store indexes

Batch mode processing

Other Enhancements

Conclusion

ABSTRACT

- SQL server introduced two innovations targeted for data warehousing workloads.
 - 1.Column store indexes
 - 2.Batch processing mode
- Improves performance of data warehouse queries.
- Limitations of initial versions are addressed in upcoming release.

INTRODUCTION

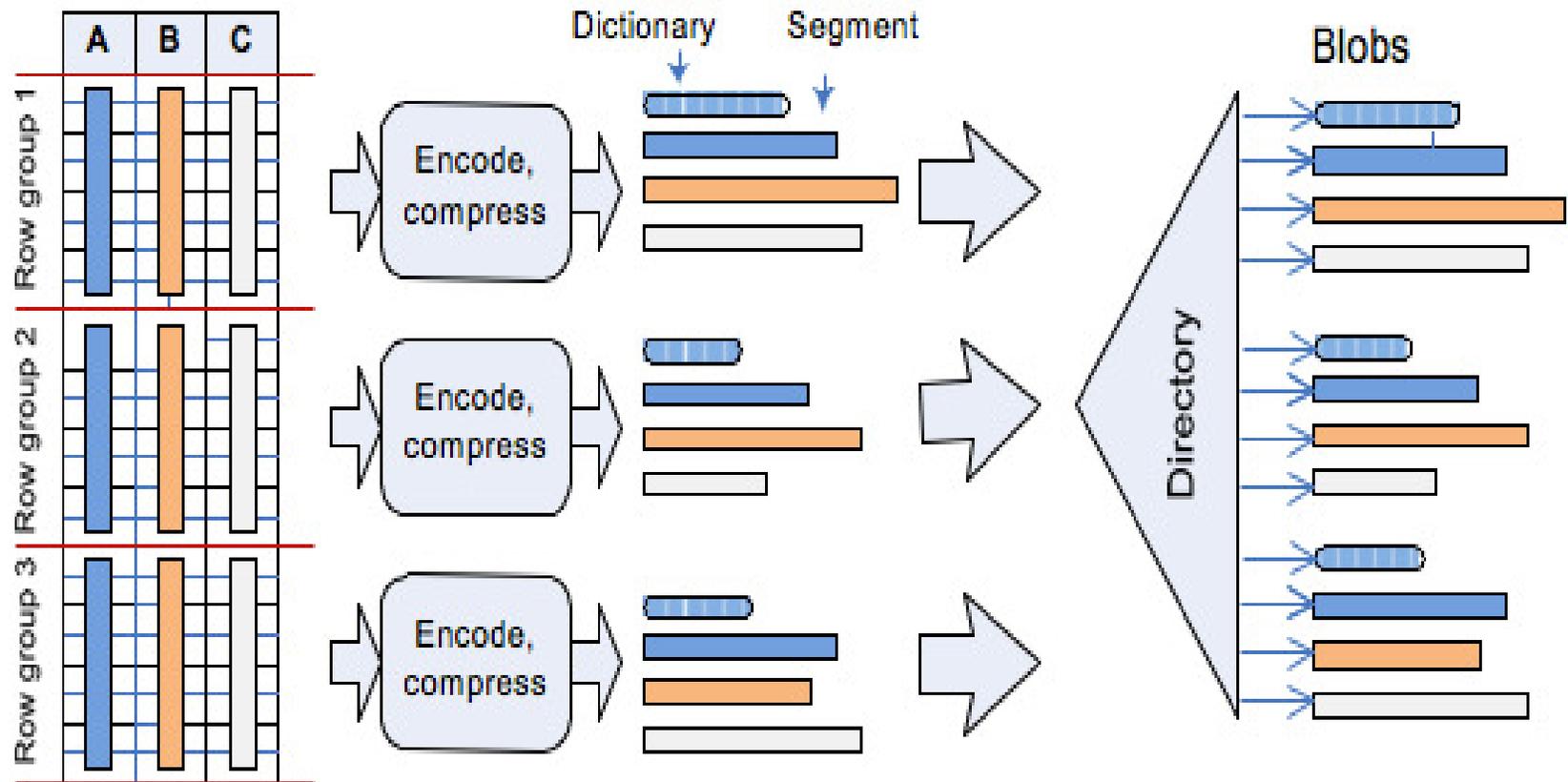
- SQL server long supported row-oriented storage organizations.
- SQL server introduced a new index type, Column Store Index's, where data is stored in column-wise in compressed form.
- It also introduced a new query processing mode, batch processing, where operators process a batch of rows at a time instead of one row at a time.
- Customers reported major performance improvements when using these enhancements.
- Limitations are remedied in the upcoming release.

What is a COLUMN STORE?

A column store is data that is logically organized as a table with rows and columns and physically stored in a columnar data format.

WHAT IS A COLUMNAR STORE INDEX

- Data is compressed, stored, and managed as a collection of partial columns.
- It is adaptable and can be used as the base storage for a table
- We can use a column store index to answer a query just like data in any other type of index.
- The query optimizer considers the column store index as a data source for accessing data just like it considers other indexes when creating a query plan.



Directory keeps track of

- Location of segments
- Location of dictionaries
- Meta data of each segment such as
 - ✓ Number of rows
 - ✓ Size of each row
 - ✓ How data is encoded
 - ✓ Min and Max values

Dictionary encoding – Frequently occurring values are mapped to a 32 bit id.

2 types of dictionaries – Global and Local

Earlier:

- Fill entries in Global dictionary as it built the index.
- Did not guarantee for most relevant values.

Modified query plan: 2 steps

First step – Samples data and decides to build a global dictionary or not.

Second step – Builds the index using global dictionaries.

SQL Server 2012

Column store indexes can only be used as secondary indexes.

- One copy - primary storage structure(heap or B-tree)
- Another copy - secondary column store index.

Upcoming release

- Column store is the primary and only copy of the data.
- Not really 'clustered' on any key.
- Retain the traditional SQL server convention of referring to the primary index as a clustered index.

Two components are added to make SQL server column indexes updatable.

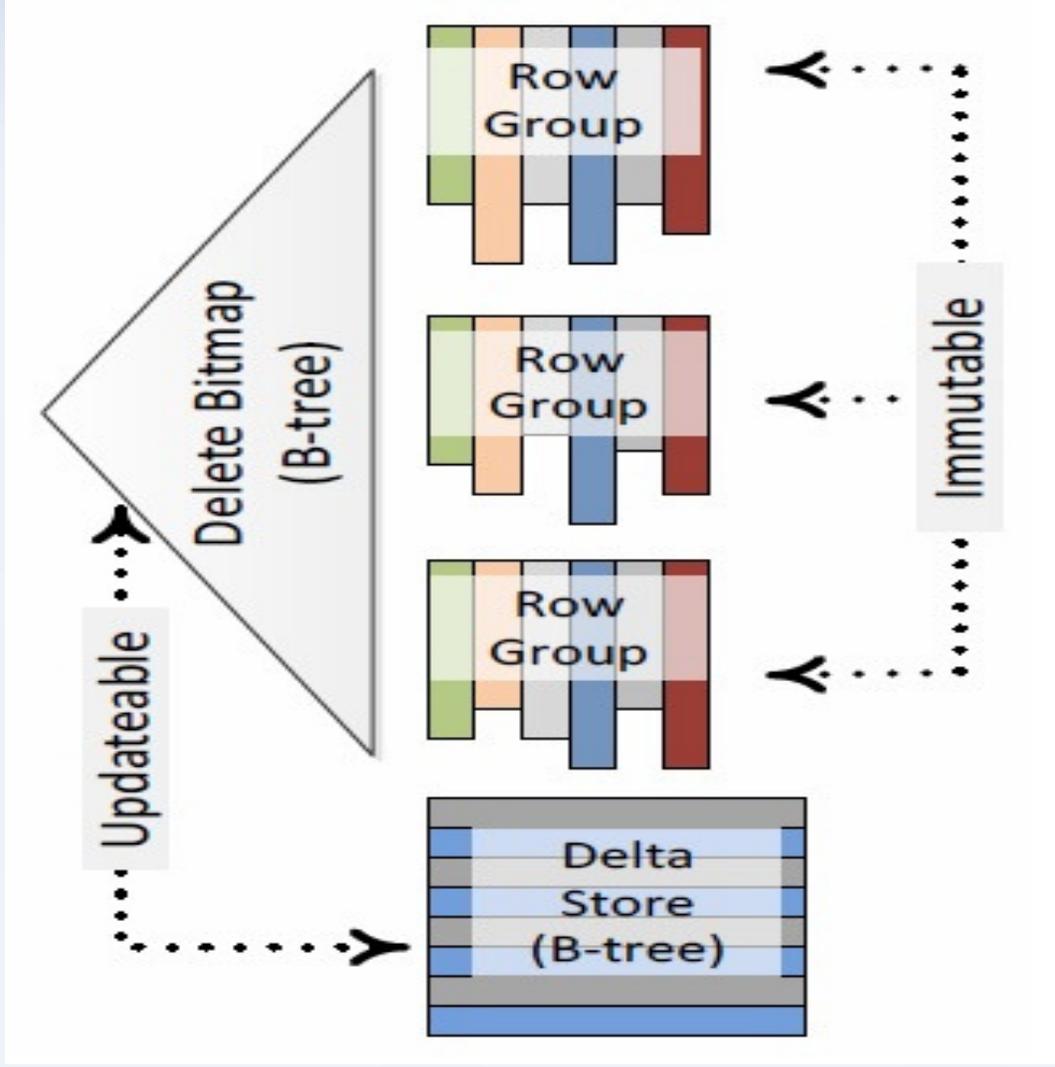
- Delete bitmaps
- Delta stores

Delete bitmaps:

- Every column store index has an associated delete bitmap consulted during scans to disqualify rows that have been deleted.
- It has different in-memory and on disk representations.
In-memory : represented as a bitmap
On disk : represented as a B-tree

Delta store:

- Represented as a B-tree.
- New and updated rows are inserted.
- An index may have multiple delta stores.



Operations on column stores:

Insert: New rows are added into delta store. Efficient because delta stores are traditional B-tree indexes.

Delete: If the row to be deleted is in a delta store, delete it.
If the row to be deleted is in column store, row ID is to be inserted into delete bitmap.

Update: Combination of delete and insert operation.

Tuple Mover:

- Task of converting closed delta stores to columnar storage format is called TupleMover. After the conversion it is deallocated.
- Tuple mover reads a delta store and builds the corresponding *compressed segments*.
- It does not block reads and inserts so it has minimal impact on system concurrency.
- It can be invoked on demand.

Batch mode processing

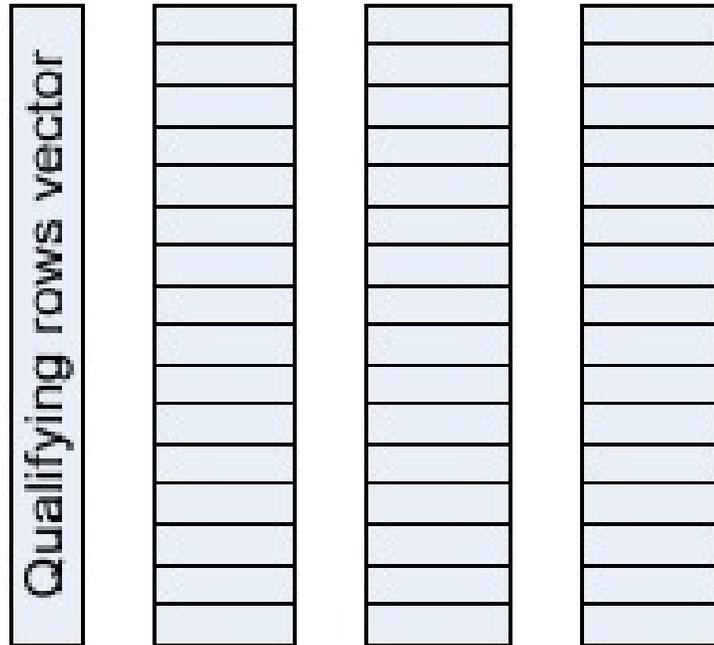
- Introduced for the first time in SQL server 2012.
- Processes a *batch of rows* at a time unlike row-at-a-time execution model. This greatly reduces CPU time.
- A row batch is stored column wise and typically consists of thousands of rows.
- Each column within a batch is stored as a vector in a separate area of memory, so batch mode processing is vector based.
- Only few operators are supported in batch mode: Scan, filter, project, hash(inner)join and hash aggregation.

Batch mode processing(Cont.):

- Batch mode processing spreads meta data access costs and other types of overhead over all the rows in a batch, rather than paying the cost for each row.
- Query optimizer decides whether to use batch-mode or row mode operators.
- Batch mode operators are used for data intensive part of computation ,initial filtering,projection,join and aggregation of inputs.
- Row-mode operators are used on smaller inputs and also for operations not supported by batch-mode operators.

Row batch

Column vectors



Query Processing and Optimization:

SQL server 2012: Batch processing supports only the most heavily used query patterns in data warehousing.

Ex:

✓ Inner join

✗ Outer join

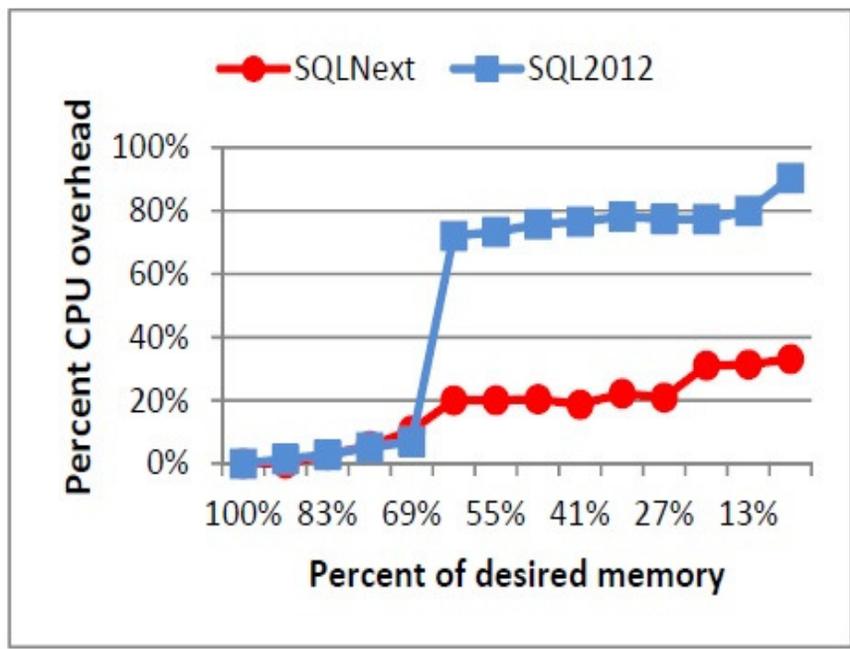
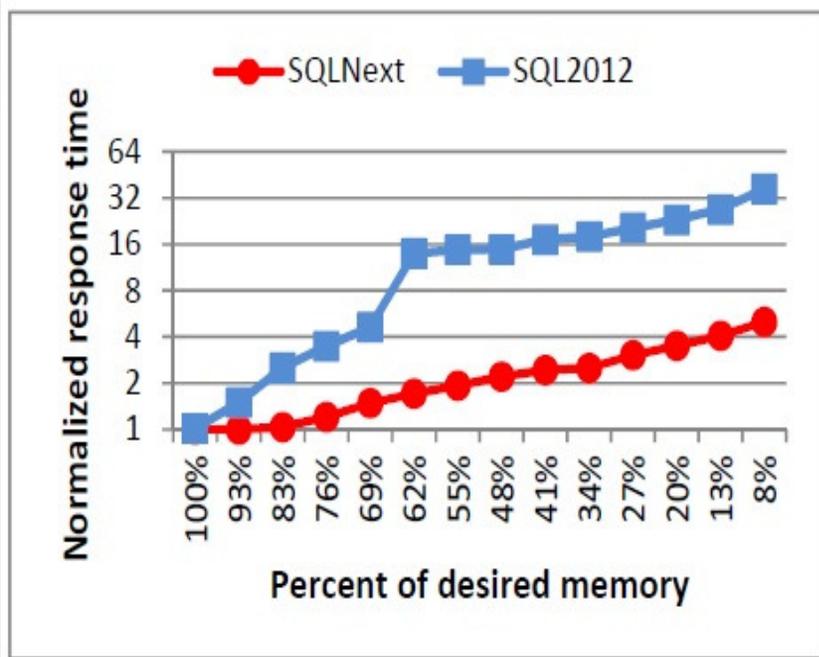
✓ Group-by-aggregate

✗ Scalar aggregates

Upcoming release: Extends batch processing capabilities. Considers batch execution for iterators anywhere in the query plan. Supports all SQL server join types, union all, and scalar aggregation.

Hash Join:

- When there is not enough memory for the query to build a hash table, join processing switches from batch-mode to row-mode.
- Row-mode hash join is able to execute in low memory conditions by *spilling* some of the input data temporarily on to disk. But query performance degrades when the switch happened.
- So batch-mode hash join is enhanced by adding splitting functionality.



Other Enhancements

- Supports all the data types.
- Supports storing of short strings by value instead of converting.
 - ✓ Removes extra overhead on dictionary.
 - ✓ Improves column store compression.
- Improve feature parity between column store index and row store indexes.

Example: Possible to add, drop, modify columns of a clustered column store.
- Data check functionality

Conclusion:

Two innovations targeted for data warehousing namely column store indexes and batch mode processing were introduced in SQL server 2012 which reduces total I/O and CPU usage and also improves query performance. The limitations in initial version were addressed in the upcoming release. This presentation mainly focus on the enhancements of column store indexes and batch processing in upcoming version of SQL server.

THANK YOU