Clustered Columnstore Indexes (SQL Server 2014)

Dave Fackler
Business Intelligence Architect
davef@rollinghillsky.com

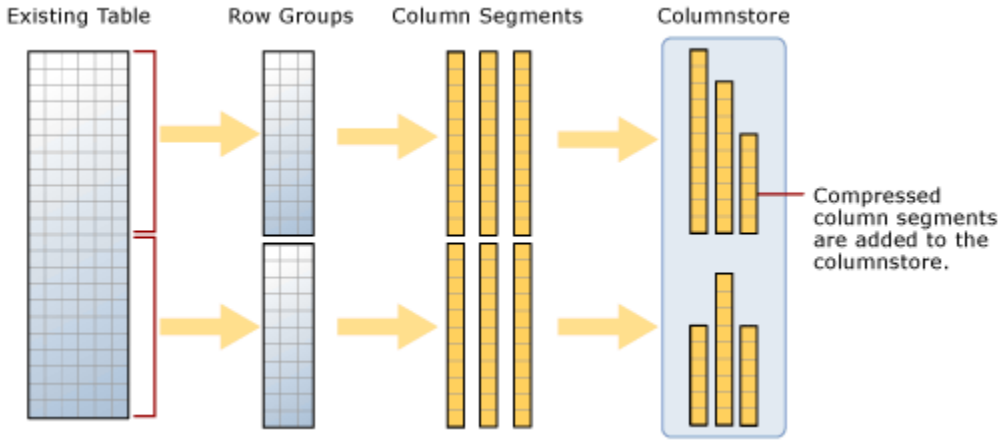
PASS SQL saturday
#402 | INDIANAPOLIS 2015

Agenda

- What are (clustered) columnstore indexes?
- Some limitations
- How do inserts/updates/deletes work?
- Using clustered columnstore indexes

Columnstore Indexes

- Use new storage and compression based on columns (the Vertipaq engine)
  - Data split into row groups of 102,400 to 1,048,576 rows
  - Column segment created for each column in the table
  - Column segments (each column) compressed and stored



Types of Columnstore Indexes

- Non-clustered columnstore indexes
  - Introduced with SQL Server 2012
  - Can be built on table with clustered index or on a heap
  - Only includes those columns that are included when built
  - Makes the table read-only ☹
- Clustered columnstore indexes
  - Introduced with SQL Server 2014
  - Is the only index on a table when created (no other indexes can be created)
  - Includes all columns of the table
  - Is updateable ☺

Limitations of Clustered Columnstore Indexes

- Cannot have any other indexes
- Table cannot have
  - PK constraints, FK constraints, unique constraints
  - Computed columns or sparse columns
  - Unsupported data types (text, uniqueidentifier, geography, xml, timestamp, hierarchyid, varchar(max), nvarchar(max))
- Cannot be combined with replication, change tracking, change data capture
- Cursors are not supported on CCI-indexed tables

Demo

(Scripts 1-12)

How do Inserts/Updates/Deletes Work?

- Clustered columnstore indexes use delta stores
  - Delta store is an “open” row group that can accept new rows
  - If delta store exists (less than 102,400 rows), it can be used
  - Otherwise, new delta stores are created as needed
- Inserts (unless using bulk insert) get added to delta store, but not immediately compressed
- Deletes get marked as deleted in columnstore segments
- Updates behave like deletes + inserts
- Delta stores get marked as “closed” once they are full

How do Inserts/Updates/Deletes Work?

- Closed delta stores compressed automatically over time
  - Tuple Mover background process looks for closed delta stores
  - Wakes up every few minutes, does work if needed, then sleeps
  - Is single-threaded, so it can take a long time to handle lots of closed delta stores…
- ALTER INDEX … REORGANIZE
  - Works like Tuple Mover, but is multi-threaded
- ALTER INDEX … REBUILD
  - Closes open delta stores (even if small)
  - Compresses all closed delta stores

Demo

(Scripts 13-18)

Using Clustered Columnstore Indexes

- Initially loading a large amount of data (100M+ rows)
  - Create table with regular clustered index, load, then create CCI
  - Or create table with CCI and bulk load data
- Monitor rowgroups and minimize open/closed delta stores
  - Queries must scan delta stores that are not compressed
  - Data there is not sorted, not indexed, and acts like a (slow) heap
  - Large number of rows in open/closed delta stores kills performance
- Use partitioning and reorganize/rebuild index by partition
  - After large sets of changes, determine partitions affected
  - Use REORGANIZE (faster) or REBUILD (more complete)

Using Clustered Columnstore Indexes

- Metadata information
  - sys.column_store_segments
    - Returns information about each column in each columnstore segment
  - sys.column_store_dictionaries
    - Returns information about dictionaries used (for columns needing them)
  - sys.column_store_row_groups
    - Returns information about rowgroups (likely the most useful of the three)
  - sp_spaceused
    - Warning: can return incorrect information for a table with CCI

Questions?