Tips and Tricks to Partition In-Memory Cubes for Faster Performance

Presented by: Trishla Maru
Agenda

Understanding MicroStrategy’s In-Memory Architecture

Tips and Tricks:
• Enabling Parallel Data Fetch
• Enabling the Correct Number of Partitions
• Enabling Partition on the Appropriate Attribute
• Using Cubes vs View Reports

Performance Improvements and Customer Success Stories

Summary and Q&A
Architecture of MicroStrategy’s In-Memory Capability

Web and mobile output
API
Parallel query execution
Data partitioning within cores
Optimized in-memory data structure
Parallel data loading

SOURCE DATA

Tightly coupled for minimal computational distance

Visualization
Application Engines
Analytics Engines
DATA
DATA
DATA
DATA
MicroStrategy’s In-Memory Offering Co-exists With Existing Databases

- Does not replace databases
- Functions as Hot data layer for apps requiring high performance
- Drill through to databases for detail
- Load from databases or directly from files and Hadoop
MicroStrategy’s Next Generation In-Memory Analytics

**PARALLEL**
- Ability to generate parallel queries and fetch it in parallel from the underlying source.
- Improves the speed of cube publication
- Higher Data Throughput

**PARTITIONED**
- Ability to partition the data in the cube
- No 2B row limit per cube. Each cube can be divided into partitions, each partition can contain up to 2B rows
- Higher Capacity/Data Scalability

**IN-MEMORY CUBES**
- Cubes with flexible schema. No pre-joins.
- More Efficient, Optimized, Scalable Cubes for Building Fast Performing Dashboards
Tips and Tricks
Enabling Parallel Queries with MicroStrategy’s In-Memory Analytics

Improve the speed of cube publication by generating maximum parallel queries per report

- Project level setting to configure the “Maximum Parallel Queries Per Report”.

- Network Transfer Rate depends on the theoretical limit between data source and Intelligence Server.

- Each table being imported would be executed over a single thread.

- In order to parallelize a big table, user may want to build multiple views representing slices to be fetched over independent connection.
Enabling Parallel Data Fetch Option

This option is specifically available for OLAP cubes built using the Developer tool

- This is a different option and not to be confused with Parallel Query Execution option.
- Allows for SQL Select Pass for Metrics (typically the last pass) to be fetched over multiple ODBC connections.
- Users are allowed to switch between Permanent Table (for Generic) and Derived Table syntax (optimal for Single Select).
- Number of maximum parallel queries for Parallel Data Fetch would be governed by Number of Partitions.
Parallel Data Fetch Compatibility with VLDB Settings

Parallel Data Fetch currently does not take effect for certain VLDB settings

Parallel Data Fetch for OLAP Cubes would NOT take effect if any of the following are set:
- Insert Mid/Pre/Post Statements
- Table Pre/Post Statements

We do plan to overcome this limitation in the coming releases.
Parallel Data Fetch Compatibility with VLDB Settings

Parallel Data Fetch currently does not take effect for certain VLDB settings

Parallel Data Fetch for OLAP Cubes would NOT take effect if any of the following options are chosen for ‘Data Population for Intelligent Cubes’.
Enabling Partitioning of In-Memory Cubes

Key Facts

• Ability to partition the cubes is one of the biggest advantage of the new in-memory analytics introduced in v10.
• It helps to increase the capacity/data scalability of cubes.
• Although, entirely optional.
• If the user is not partitioning the data, the published cube consists of only one table.

How Many Partitions Can the Cube Have?

• Depends on the number of cores used by Intelligence Server.
• If Intelligence Server is restricted to certain number of cores, through CPU affinity, number of partitions will also be restricted to the limit.
• Currently, can only partition on a single attribute.
Where to Define the Partition?

Using MicroStrategy Developer go to Intelligent Cube editor-> Data menu-> Configure Intelligent Cube-> Options-> Data Partition
Where to Define the Partition?

Using MicroStrategy Web go to Data Import Cube editor -> Edit -> All Objects View on Preview screen
Analytical Functions Support with Partitioning

- Partitioning does limit the types of aggregations that can be performed really fast on the raw data.

- A list of functions that can be handled include distributive functions such as – SUM, MIN, MAX, COUNT, PRODUCT.

- Semi-distributive functions such as STD DEV, VARIANCE that can be re-written using distributive functions.

- Scalar functions such as Add, Greatest, Date/Time Functions, String manipulation functions, etc. are also supported.

- DISTINCT COUNTs on the partition attribute are also supported.

- Derived metrics using any of the MicroStrategy 250+ functions are supported.

- Non-Distributive functions, may witness high CPU and memory consumption.
Recommendations to Select the Partition Key

- Some of the largest fact tables in the application are typically good candidates for partitioning and thus influence the choice of the partition attribute.

- Attributes that are frequently used for filtering or selections don’t make for good partition attributes, as they tend to push the analysis towards specific sets of partitions thus minimizing the benefits of parallel processing.

- Partition attribute should also allow for near uniform distribution of data across the partitions, so that the workload on each partition is more evenly distributed.

- Columns on which some of the larger tables in the application are joined also make for good partition attributes.

- To support best dashboard execution and concurrency performance, we have chosen to limit the number of logical CPUs engaged for any single grid evaluation to 4.
Recommendations to Make How Many Number of Partitions

- Each partition can hold a maximum of 2 billion rows, so the number of partitions should be picked accordingly.

- Typically, the number of partitions should be set to be equal to half the number of logical cores available to Intelligence server. This maximizes CPU usage to offer the best possible performance during cube publishing.

- Lower cap on the number of partitions would be dictated by the number of rows in the largest table divided by 2 billion, since each partition can hold up to 2 billion records.

- Higher cap would be dictated by the number of cores on the box. The number of partitions should typically be in-between these two numbers and closer to Half the number of logical cores.
General Cube Sizing

• Ensure that the Intelligence Server has the capacity to support all cubes in memory. In estimation of memory consumption, the RAM can consume up to 3 times the table size.

• In the case where the cube has multiple tables, each table size can be added up to estimate the peak memory requirement. In-memory Partitioning generally results in more memory requirement as compared to No Partitioning.

• For understanding PRIME Cube structure better, one can enable Engine -> CSI logs before publishing the cubes.
Cube as Dataset Vs View Report as Dataset

Working Set/View/Normal Reports, used as datasets in documents, currently cannot support multi pass analytics for document grids.

View reports allow customers to drag and drop more out of box derived metrics into different dashboards that can be built and saved in view reports. We are looking to reduce that gap for Cubes.

View reports are less optimal than cubes when using as datasets.
Performance Improvements and Case Studies
Benefits with Next Generation In-Memory Analytics

Access the Database With Higher Throughout

- OLAP Services
  - Data: 5M rows
  - Fetch Rate: 5074kb/sec

Prime 8Partitions
- Data: 5M rows
- Fetch Rate: 22454kb/sec

Upload data 4 times faster

Create and Publish the Cube With Higher Data Scalability

- OLAP Services
  - Data: 2.35B rows
  - Failed due to 2 billion row limit

Prime 8Partitions
- Data: 2.35B rows
- Publish time: 5:14:23
- Cube size: 265GB

Increase the data scalability up to 80 times

Analyze the Data With Faster Response Time

- OLAP Services
  - Data: 8M rows
  - Response Time: 0:06:33

Prime 8Partitions
- Data: 8M rows
- Response Time: 0:04:25

50% Faster Data Interactions
Significant Performance Improvements with New In-Memory Engine

![Graphs showing response times for Online Travel Industry and Largest eCommerce Industry with OLAP Services, Prime Non Partition, and Prime with Partition.](image)

- **OLAP Services**
- **Prime Non Partition**
- **Prime with Partition**
Key BI Characteristics:
INDUSTRY: Finance
BI COMPONENTS: Reports, Dashboards, VI
DATABASE: Oracle, Essbase
HARDWARE: 40-core CPU, 2TB RAM
CUBE DETAILS: 850 million row fact table
APPLICATIONS: Financial Data Analysis
PRODUCT VERSION: 10.2.0

Business Use and Benefits
• Ability to provide an ad-hoc controlled dashboard to their end users, that are able to analyze their $ metrics across a variety of dimensions.

• Dashboard performance improved from 6 minutes (on 9.4.1) to 12 seconds with v10.2.0.

• This is also helping the company to move away from a cube database and leverage MicroStrategy’s in-memory to give users the ad-hoc query performance within seconds.
Key BI Characteristics:

INDUSTRY: Technology
BI COMPONENTS: Reports, Dashboards, VI
DATABASE: Teradata
APPLICATIONS: CEO Dashboard
PRODUCT VERSION: 10.2.0

Business Use and Benefits

- Ability to provide a sophisticated high performant dashboard used by the CEO
- Dashboard previously took 40 hours to run and now is published every day.
Summary

BI Applications Need to be Highly Performant and Scalable
MicroStrategy is ready to support these with its next generation in-memory analytics

Data Loads Need to be Faster
MicroStrategy offers it with parallel data fetch capability

Increasing Data Volumes That Need to be Analyzed
MicroStrategy’s data partitioning achieves higher data scalability

Need for Interactive Faster Running Dashboards
MicroStrategy’s visualization capabilities are tightly coupled with the underlying data fetching layer

Customers Say it All
Numerous success stories proves the quality and functionality of MicroStrategy’s robust BI platform
Questions?

Q&A