Data Warehouse Optimisation Report

Car Insurance DW
Piotr Bereznowski
Marek Drwal

Basic information

Size of the DW: 1000000 rows

Testing environments: MS Windows *10 19045.3031*, MS SQL Server 2022; MS Windows *11 22621.1702*, MS SQL Server 2022

Testing procedure

This test concerns three types of analytical servers: MOLAP, ROLAP and HOLAP. The test is performed for each kind of server in two variants: with aggregations and without aggregations defined. The test examines cube processing time and execution time of three MDX queries. As a result of execution time for each query an arithmetic mean of ten results is considered. Test validity is preserved by repeating the test procedure on two different machines.

Aggregations

The aggregations have been designed to consider dimension attributes *DT Assessor.Specialisation, DT Client.Sex, Submission date.Year, Assessment date.Year* and *DT Car.Class* as fully aggregated. Such a setup has been chosen, since these attributes are frequently used, have very few members and are positioned at the top of dimensions' hierarchies. Therefore, one aggregation has been obtained through the design, with 99% level of optimisation and 3.7 MB extra storage utilisation (data according to MS VS).

Procedure

Repeat the following steps for all 3 types of servers (MOLAP, ROLAP, HOLAP), first without aggregation, second with aggregations:

1. Clear cache
2. Deploy all
3. Repeat for 3 queries
    a. Repeat 10 times
        i. Clear cache
        ii. Execute
        iii. Write down time
    b. Calculate mean time, write to the results

Test queries

1. Aggregated on dates

As a query with aggregation on dates the one realising Question 1 in the Analytical Problem 1 has been chosen. The query presents *Value of claim, Number of claim facts, Value diff* and *Number diff* measures for members of the dimension attribute *Size* from *DT Car* dimension aggregated on dates with *Submission Date Year* and *Month* members.

The query

```
WITH
	MEMBER [Measures].[Value diff]
	AS
		'([ID Submission Date].[Month].CurrentMember, [Measures].[Value of claim]) -
		 ([ID Submission Date].[Month].PrevMember, [Measures].[Value of claim])'
	MEMBER [Measures].[Number diff]
	AS
		'([ID Submission Date].[Month].CurrentMember, [Measures].[Number of claim
facts]) -
		 ([ID Submission Date].[Month].PrevMember, [Measures].[Number of claim facts])'
SELECT
	{ [Measures].[Value of claim],
	  [Measures].[Value diff],
	  [Measures].[Number of claim facts],
	  [Measures].[Number diff]} ON COLUMNS,
	{ NonEmptyCrossJoin ([DT Car].[Size].[Size].Members,
				[ID Submission Date].[Year].[Year].Members,
				[ID Submission Date].[Month].[Month].Members) } ON ROWS
FROM [Car Insurance];
```

2. For a particular dimension attribute

As a query using a particular dimension attribute the one realising Question 5 in the Analytical Problem 1 has been chosen. The query presents *Value of claim* measure for members of the dimension attribute *Specialisation* from *DT Assessor* dimension.

The query

```
SELECT
	{ [Measures].[Value of claim] } ON COLUMNS,
	{ ([DT Assessor].[Specialisation].[Specialisation].Members ) } ON ROWS
FROM [Car Insurance];
```

3. General one

As the third query, the one realising Question 5 in the Analytical Problem 2 has been chosen. The query presents *MIN indemnity*, *MAX indemnity*, and *AVG indemnity* for each product of members of attributes *Class* and *Size* from DT Car dimension.

The query

```
WITH
	MEMBER [Measures].[min]
		AS MIN([DT Car].[Car type Hierarchy].Children, [Measures].[Value of claim])
		, FORMAT_STRING='CURRENCY'
```

```
        MEMBER [Measures].[max]
                AS MAX([DT Car].[Car type Hierarchy].Children, [Measures].[Value of claim])
                , FORMAT_STRING='CURRENCY'
        MEMBER [Measures].[avg]
                AS 'AVG([DT Car].[Car type Hierarchy].Children, [Measures].[Value of claim])'
                , FORMAT_STRING='CURRENCY'
SELECT
        { [Measures].[min], [Measures].[max], [Measures].[avg] } ON COLUMNS,
        { ( [DT Car].[Class].[Class].Members,
                [DT Car].[Car type Hierarchy].[Size] ) } ON ROWS
FROM [Car Insurance]
;
```

Machine 1: Windows 10

	MOLAP		ROLAP		HOLAP	
	No Aggregations	Aggregations	No Aggregations	Aggregations	No Aggregations	Aggregations
Delay	436	324	417	1444	428	342
Processing time	7579	8709	126	750	6284	4643
Total size	93.28 MB	93.32 MB	38.27 MB	38.23 MB	38.27 MB	38.30 MB
Q1 time	163.8	174.7	1114.9 Without outliers: 468.7	1212.5 Without outliers: 484.1	743.4 Without outliers: 541.5	718.0 Without outliers: 493.4
Q2 time	104.4	4.3	1899.0 Without outliers: 339	2049.4 Without outliers: 465.8	371.2	7.8
Q3 time	129.5	134.0	2228.2 Without outliers: 552.3	771.2	572.7	1020.8 Without outliers: 590.3

Machine 2: Windows 11

	MOLAP		ROLAP		HOLAP	
	No Aggregations	Aggregations	No Aggregations	Aggregations	No Aggregations	Aggregations
Delay	1241	710	2369	2335	2122	2364
Processing time	13124	8160	5990	5744	6657	6273

Total size	79.04 MB	79.07 MB	30.92 MB	30.92 MB	30.92 MB	30.95 MB
Q1 time	134.2	136.5	1283.5 Without outliers: 290.0	1416.7 Without outliers: 279.7	1255.3 Without outliers: 279.2	1775.7 Without outliers: 287.6
Q2 time	73.8	18.7	979.6 Without outliers: 143.4	1146.4 Without outliers: 149.4	177.9	7
Q3 time	95	102.3	273.3	298.2	783.3 Without outliers: 287.9	585.6 Without outliers: 269.3

Summary

The results of the investigation were in line with the theory. The MOLAP model stores data in the optimised multi-dimensional storage. This multidimensional storage includes the fact table and the aggregations calculated during the processing of the cube, which allows fast access to them during execution of the queries (queries are executed the fastest for the MOLAP model). However, having the optimised multi-dimensional storage of data has a drawback of longer processing of the cube and storage of duplicate data (processing time and total size are the highest for the MOLAP model).

The ROLAP model stores data in relational database (more precisely in relational data warehouse). Consequently, it does not create duplicates of data and the cube processing is much faster. Nonetheless, execution of each query is associated with a need to perform all of the related calculations (in the MOLAP model substantial part of these calculations is performed during the cube processing). As a result, the queries take more time but processing time is shorter.

The HOLAP model implements the MOLAP model for more-aggregate and less-detailed data and the ROLAP model for detailed data. This means that queries with aggregation are supposed to be executed faster, as it has been observed for Q2, but all other queries are executed as slowly as in the case of the ROLAP model.

The latter above-mentioned issue indicates the crucial importance of proper design of aggregations. Well-designed aggregations, matching warehouse's characteristics, allow significant querying speed increase both in hybrid and multidimensional OLAPS, as it is the case of Q2. Interestingly, in terms of this HOLAP, excess memory load does not exceed any unacceptable level, providing considerable benefit in overall performance.

Notwithstanding, the formerly mentioned strengths are not observable for Q1 and Q3. Although dimension attributes regarding dates have been aggregated (*Year)*, the influence on Q1 execution is negligible. It is presumably due to slicing by another dimension attribute, not included in the aggregation, namely *DT Car.Szie*. In terms of Q3, aggregation's impact is not significant, since the queried measures relate to aggregation functions different than those specified in the cube.

It is also worth pointing out the substantial outliers observed for the ROLAP and the HOLAP models. Their presence seems to indicate that the calculations performed by the relational database may in some situations take significantly longer than expected, most likely when there is not enough computing power available to the relational database management system or when the relational database management system is performing other tasks simultaneously.