Exercise - Improve the execution plan of a query with a materialized view

3 minutes

Improve execution plan of query with materialized view

1. Run the query with the `EXPLAIN` directive (note the `WITH_RECOMMENDATIONS` option as well):

SQL

```
EXPLAIN WITH_RECOMMENDATIONS
SELECT
    T.TransactionItemsCountBucket
    ,count(*) as CustomersCount
FROM
    (
        SELECT
            CustomerId,
            (
                COUNT(*) -
                (
                    SELECT
                        MIN(TransactionItemsCount)
                    FROM
                    (
                        SELECT
                            COUNT(*) as TransactionItemsCount
                        FROM
                            [wwi_perf].[Sale_Hash]
                        GROUP BY
                            CustomerId
                    ) X
                )
            ) / 100 as TransactionItemsCountBucket
        FROM
            [wwi_perf].[Sale_Hash]
        GROUP BY
            CustomerId
    ) T
GROUP BY
    T.TransactionItemsCountBucket
```

```
ORDER BY
    T.TransactionItemsCountBucket
```

2. Analyze the resulting execution plan. Take a close look to the `<materialized_view_candidates>` section which suggests possible materialized views you can create to improve the performance of the query.

XML

```
<?xml version="1.0" encoding="utf-8"?>
<dsql_query number_nodes="5" number_distributions="60" number_distrib-
utions_per_node="12">
<sql>SELECT
    T.TransactionItemsCountBucket
    ,count(*) as CustomersCount
FROM
    (
        SELECT
            CustomerId,
            (
                COUNT(*) -
                (
                    SELECT
                        MIN(TransactionItemsCount)
                    FROM
                    (
                        SELECT
                            COUNT(*) as TransactionItemsCount
                        FROM
                            [wwi_perf].[Sale_Hash]
                        GROUP BY
                            CustomerId
                    ) X
                )
            ) / 100 as TransactionItemsCountBucket
        FROM
            [wwi_perf].[Sale_Hash]
        GROUP BY
            CustomerId
    ) T
GROUP BY
    T.TransactionItemsCountBucket
ORDER BY
    T.TransactionItemsCountBucket</sql>
<materialized_view_candidates>
    <materialized_view_candidates with_constants="False">CREATE MATE-
RIALIZED VIEW View1 WITH (DISTRIBUTION = HASH([Expr0])) AS
SELECT [SQLPool01].[wwi_perf].[Sale_Hash].[CustomerId] AS [Expr0],
    COUNT(*) AS [Expr1]
FROM [wwi_perf].[Sale_Hash]
GROUP BY [SQLPool01].[wwi_perf].[Sale_Hash].[CustomerId]</material-
ized_view_candidates>
```

```
</materialized_view_candidates>
<dsql_operations total_cost="0.0242811172881356" total_number_operations="9">
    <dsql_operation operation_type="RND_ID">
    <identifier>TEMP_ID_99</identifier>
    </dsql_operation>
    <dsql_operation operation_type="ON">
    <location permanent="false" distribution="AllComputeNodes" />
    <sql_operations>
        <sql_operation type="statement">CREATE TABLE [qtabledb].[dbo].[TEMP_ID_99] ([col] INT ) WITH(DISTRIBUTED_MOVE_FILE='');</sql_operation>
    </sql_operations>
    </dsql_operation>
    <dsql_operation operation_type="BROADCAST_MOVE">
    <operation_cost cost="0.00096" accumulative_cost="0.00096" average_rowsize="4" output_rows="1" GroupNumber="69" />
    <source_statement>SELECT [T1_1].[col] AS [col] FROM (SELECT
MIN([T2_1].[col]) AS [col] FROM (SELECT COUNT(CAST ((0) AS INT)) AS
[col], 0 AS [col1] FROM [SQLPool01].[wwi_perf].[Sale_Hash] AS T3_1
GROUP BY [T3_1].[CustomerId]) AS T2_1 GROUP BY [T2_1].[col1]) AS T1_1
OPTION (MAXDOP 6, MIN_GRANT_PERCENT = [MIN_GRANT],
DISTRIBUTED_MOVE(N''))</source_statement>
    <destination_table>[TEMP_ID_99]</destination_table>
    </dsql_operation>
    <dsql_operation operation_type="RND_ID">
    <identifier>TEMP_ID_100</identifier>
    </dsql_operation>
    <dsql_operation operation_type="ON">
    <location permanent="false" distribution="AllDistributions" />
    <sql_operations>
        <sql_operation type="statement">CREATE TABLE [qtabledb].[dbo].[TEMP_ID_100] ([col] INT, [col1] BIGINT )
WITH(DISTRIBUTED_MOVE_FILE='');</sql_operation>
    </sql_operations>
    </dsql_operation>
    <dsql_operation operation_type="SHUFFLE_MOVE">
    <operation_cost cost="0.0233211172881356"
accumulative_cost="0.0242811172881356" average_rowsize="12" output_rows="95.5518" GroupNumber="75" />
    <source_statement>SELECT [T1_1].[col1] AS [col], [T1_1].[col] AS
[col1] FROM (SELECT COUNT_BIG(CAST ((0) AS INT)) AS [col], [T2_1].
[col] AS [col1] FROM (SELECT (([T3_2].[col] - [T3_1].[col]) / CAST
((100) AS INT)) AS [col] FROM (SELECT MIN([T4_1].[col]) AS [col] FROM
[qtabledb].[dbo].[TEMP_ID_99] AS T4_1) AS T3_1 INNER JOIN
(SELECT COUNT(CAST ((0) AS INT)) AS [col] FROM [SQLPool01].[wwi_perf].
[Sale_Hash] AS T4_1 GROUP BY [T4_1].[CustomerId]) AS T3_2
ON (0 = 0)) AS T2_1 GROUP BY [T2_1].[col]) AS T1_1
OPTION (MAXDOP 6, MIN_GRANT_PERCENT = [MIN_GRANT],
DISTRIBUTED_MOVE(N''))</source_statement>
    <destination_table>[TEMP_ID_100]</destination_table>
    <shuffle_columns>col;</shuffle_columns>
    </dsql_operation>
    <dsql_operation operation_type="RETURN">
    <location distribution="AllDistributions" />
```

```
    <select>SELECT [T1_1].[col1] AS [col], [T1_1].[col] AS [col1] FROM
(SELECT CONVERT (INT, [T2_1].[col], 0) AS [col], [T2_1].[col1] AS
[col1] FROM (SELECT ISNULL([T3_1].[col], CONVERT (BIGINT, 0, 0)) AS
[col], [T3_1].[col1] AS [col1] FROM (SELECT SUM([T4_1].[col1]) AS
[col], [T4_1].[col] AS [col1] FROM [qtabledb].[dbo].[TEMP_ID_100] AS
T4_1 GROUP BY [T4_1].[col]) AS T3_1) AS T2_1) AS T1_1 ORDER BY [T1_1].
[col1] ASC
OPTION (MAXDOP 6, MIN_GRANT_PERCENT = [MIN_GRANT])</select>
    </dsql_operation>
    <dsql_operation operation_type="ON">
    <location permanent="false" distribution="AllDistributions" />
    <sql_operations>
        <sql_operation type="statement">DROP TABLE [qtabledb].[dbo].
[TEMP_ID_100]</sql_operation>
    </sql_operations>
    </dsql_operation>
    <dsql_operation operation_type="ON">
    <location permanent="false" distribution="AllComputeNodes" />
    <sql_operations>
        <sql_operation type="statement">DROP TABLE [qtabledb].[dbo].
[TEMP_ID_99]</sql_operation>
    </sql_operations>
    </dsql_operation>
</dsql_operations>
</dsql_query>
```

3. Create the suggested materialized view:

SQL

```
CREATE MATERIALIZED VIEW
    mvTransactionItemsCounts
WITH
(
    DISTRIBUTION = HASH([CustomerId])
)
AS
SELECT
    CustomerId
    ,COUNT(*) AS ItemsCount
FROM
    [wwi_perf].[Sale_Hash]
GROUP BY
    CustomerId
```

4. Check the execution plan again:

SQL

```
EXPLAIN WITH_RECOMMENDATIONS
SELECT
    T.TransactionItemsCountBucket
```

```
    ,count(*) as CustomersCount
FROM
    (
        SELECT
            CustomerId,
            (
                COUNT(*) -
                (
                    SELECT
                        MIN(TransactionItemsCount)
                    FROM
                    (
                        SELECT
                            COUNT(*) as TransactionItemsCount
                        FROM
                            [wwi_perf].[Sale_Hash]
                        GROUP BY
                            CustomerId
                    ) X
                )
            ) / 100 as TransactionItemsCountBucket
        FROM
            [wwi_perf].[Sale_Hash]
        GROUP BY
            CustomerId
    ) T
GROUP BY
    T.TransactionItemsCountBucket
ORDER BY
    T.TransactionItemsCountBucket
```

The resulting execution plan indicates now the use of the `mvTransactionItemsCounts` (the `BROADCAST_MOVE` distributed SQL operation) materialized view which provides improvements to the query execution time:

XML

```
<?xml version="1.0" encoding="utf-8"?>
<dsql_query number_nodes="5" number_distributions="60" number_distrib-
utions_per_node="12">
<sql>SELECT
    T.TransactionItemsCountBucket
    ,count(*) as CustomersCount
FROM
    (
        SELECT
            CustomerId,
            (
                COUNT(*) -
                (
                    SELECT
                        MIN(TransactionItemsCount)
```

```
                    FROM
                    (
                        SELECT
                            COUNT(*) as TransactionItemsCount
                        FROM
                            [wwi_perf].[Sale_Hash]
                        GROUP BY
                            CustomerId
                    ) X
                )
            ) / 100 as TransactionItemsCountBucket
        FROM
            [wwi_perf].[Sale_Hash]
        GROUP BY
            CustomerId
    ) T
GROUP BY
    T.TransactionItemsCountBucket
ORDER BY
    T.TransactionItemsCountBucket</sql>
<materialized_view_candidates>
    <materialized_view_candidates with_constants="False">CREATE MATE-
RIALIZED VIEW View1 WITH (DISTRIBUTION = HASH([Expr0])) AS
SELECT [SQLPool01].[wwi_perf].[Sale_Hash].[CustomerId] AS [Expr0],
    COUNT(*) AS [Expr1]
FROM [wwi_perf].[Sale_Hash]
GROUP BY [SQLPool01].[wwi_perf].[Sale_Hash].[CustomerId]</material-
ized_view_candidates>
</materialized_view_candidates>
<dsql_operations total_cost="0.0242811172881356" total_number_opera-
tions="9">
    <dsql_operation operation_type="RND_ID">
    <identifier>TEMP_ID_111</identifier>
    </dsql_operation>
    <dsql_operation operation_type="ON">
    <location permanent="false" distribution="AllComputeNodes" />
    <sql_operations>
        <sql_operation type="statement">CREATE TABLE [qtabledb].[dbo].
[TEMP_ID_111] ([col] INT ) WITH(DISTRIBUTED_MOVE_FILE='');</sql_opera-
tion>
    </sql_operations>
    </dsql_operation>
    <dsql_operation operation_type="BROADCAST_MOVE">
    <operation_cost cost="0.00096" accumulative_cost="0.00096" aver-
age_rowsize="4" output_rows="1" GroupNumber="134" />
    <source_statement>SELECT [T1_1].[col] AS [col] FROM (SELECT
MIN([T2_1].[col]) AS [col] FROM (SELECT CONVERT (INT, [T3_1].[col], 0)
AS [col], 0 AS [col1] FROM (SELECT ISNULL([T4_1].[col], CONVERT (BIG-
INT, 0, 0)) AS [col] FROM (SELECT SUM([T5_1].[ItemsCount]) AS [col]
FROM (SELECT [T6_1].[CustomerId] AS [CustomerId], [T6_1].[ItemsCount]
AS [ItemsCount] FROM [SQLPool01].[dbo].[mvTransactionItemsCounts] AS
T6_1) AS T5_1 GROUP BY [T5_1].[CustomerId]) AS T4_1) AS T3_1 WHERE
([T3_1].[col] != CAST ((0) AS BIGINT))) AS T2_1 GROUP BY [T2_1].
[col1]) AS T1_1
OPTION (MAXDOP 6, MIN_GRANT_PERCENT = [MIN_GRANT],
```

```
DISTRIBUTED_MOVE(N''))</source_statement>
    <destination_table>[TEMP_ID_111]</destination_table>
    </dsql_operation>
    <dsql_operation operation_type="RND_ID">
    <identifier>TEMP_ID_112</identifier>
    </dsql_operation>
    <dsql_operation operation_type="ON">
    <location permanent="false" distribution="AllDistributions" />
    <sql_operations>
        <sql_operation type="statement">CREATE TABLE [qtabledb].[dbo].
[TEMP_ID_112] ([col] INT, [col1] BIGINT )
WITH(DISTRIBUTED_MOVE_FILE='');</sql_operation>
    </sql_operations>
    </dsql_operation>
    <dsql_operation operation_type="SHUFFLE_MOVE">
    <operation_cost cost="0.0233211172881356"
accumulative_cost="0.0242811172881356" average_rowsize="12" out-
put_rows="95.5518" GroupNumber="140" />
    <source_statement>SELECT [T1_1].[col1] AS [col], [T1_1].[col] AS
[col1] FROM (SELECT COUNT_BIG(CAST ((0) AS INT)) AS [col], [T2_1].
[col] AS [col1] FROM (SELECT (([T3_2].[col] - [T3_1].[col]) / CAST
((100) AS INT)) AS [col] FROM (SELECT MIN([T4_1].[col]) AS [col] FROM
[qtabledb].[dbo].[TEMP_ID_111] AS T4_1) AS T3_1 INNER JOIN
(SELECT CONVERT (INT, [T4_1].[col], 0) AS [col] FROM (SELECT
ISNULL([T5_1].[col], CONVERT (BIGINT, 0, 0)) AS [col] FROM (SELECT
SUM([T6_1].[ItemsCount]) AS [col] FROM (SELECT [T7_1].[CustomerId] AS
[CustomerId], [T7_1].[ItemsCount] AS [ItemsCount] FROM [SQLPool01].
[dbo].[mvTransactionItemsCounts] AS T7_1) AS T6_1 GROUP BY [T6_1].
[CustomerId]) AS T5_1) AS T4_1 WHERE ([T4_1].[col] != CAST ((0) AS
BIGINT))) AS T3_2
ON (0 = 0)) AS T2_1 GROUP BY [T2_1].[col]) AS T1_1
OPTION (MAXDOP 6, MIN_GRANT_PERCENT = [MIN_GRANT],
DISTRIBUTED_MOVE(N''))</source_statement>
    <destination_table>[TEMP_ID_112]</destination_table>
    <shuffle_columns>col;</shuffle_columns>
    </dsql_operation>
    <dsql_operation operation_type="RETURN">
    <location distribution="AllDistributions" />
    <select>SELECT [T1_1].[col1] AS [col], [T1_1].[col] AS [col1] FROM
(SELECT CONVERT (INT, [T2_1].[col], 0) AS [col], [T2_1].[col1] AS
[col1] FROM (SELECT ISNULL([T3_1].[col], CONVERT (BIGINT, 0, 0)) AS
[col], [T3_1].[col1] AS [col1] FROM (SELECT SUM([T4_1].[col1]) AS
[col], [T4_1].[col] AS [col1] FROM [qtabledb].[dbo].[TEMP_ID_112] AS
T4_1 GROUP BY [T4_1].[col]) AS T3_1) AS T2_1) AS T1_1 ORDER BY [T1_1].
[col1] ASC
OPTION (MAXDOP 6, MIN_GRANT_PERCENT = [MIN_GRANT])</select>
    </dsql_operation>
    <dsql_operation operation_type="ON">
    <location permanent="false" distribution="AllDistributions" />
    <sql_operations>
        <sql_operation type="statement">DROP TABLE [qtabledb].[dbo].
[TEMP_ID_112]</sql_operation>
    </sql_operations>
    </dsql_operation>
    <dsql_operation operation_type="ON">
```

```
    <location permanent="false" distribution="AllComputeNodes" />
    <sql_operations>
        <sql_operation type="statement">DROP TABLE [qtabledb].[dbo].
[TEMP_ID_111]</sql_operation>
    </sql_operations>
    </dsql_operation>
</dsql_operations>
</dsql_query>
```