The ttSizeOra utility

Contents

1. Overview
2. Prerequisites
3. Data Type Mapping
4. Usage Example
5. Detailed Syntax and Description
6. Caveats and Limitations
7. Changes in this release
8. Known Issues

1. Overview

ttSizeOra examines a table in an Oracle database and provides an estimate of the memory that would be required for the table, its metadata and its indexes, if the table were to be imported into a TimesTen database.

Options are provided to control various aspects of the sizing estimate and the information that is reported.

The main features provided by ttSizeOra are:

- Generates an accurate memory sizing estimate thereby facilitating capacity planning prior to any actual data movement.
- The sizing estimate is based on an understanding of how certain Oracle data types are mapped to equivalent TimesTen data types.
- The estimate can also optionally take into account memory savings from applying data type mapping of suitable NUMBER columns to TimesTen native integer types.
- Allows control over the boundary that determines if variable length columns are stored INLINE or NOT INLINE in TimesTen, which affects the overall memory usage.
- Allows the user to specify the TimesTen range index type (B-tree or T-tree) to be used when estimating index sizes. The default is B-tree.
- Analyzes the Oracle table data to determine the row count and the average utilization of variable length columns.
- Row count and/or variable length column utilization can be specified on the command line to override the automatically determined values (which may also avoid the need to analyze the data in the Oracle database).

- Two levels of output detail.

- Supported on a wide range of platforms:
  Linux ARM 64 bit
  Linux x86 64 bit

- Supports Oracle database 11g, 12c, 18c, 19c and 21c. Support for 12c and upwards has some caveats (see the section on **Known Issues**).

This utility is not an official production utility and support will be provided on a 'best effort' basis. The most up to date version of this utility can always be found on the TimesTen downloads page here: https://www.oracle.com/downloads/samplecode/otsu-samplecode-downloads.html

Here is a summary of the command syntax.

```
Estimate size of table and indexes when imported into TimesTen
 Usage:

ttSizeOra [-h | -help | -?]

ttSizeOra [-V | -version]

ttSizeOra  -oraConn ouser[/opasswd][@oserver]
           -tbl [owner.]tablename
          [-rows r]
          [-frac f]
          [-typeMap | -noTypeMap]
          [-inlineLimit n]
          [-parallel dop]
          [-btree | -ttree]
          [-verbose]
```

2. Prerequisites

In order to use this utility, you must ensure that the following prerequisites are met:

- The Oracle Database version must be at least 11.2.2. Older versions are not supported.

- The system where you plan to run the utility must be using a supported operating system.

- The system where you plan to run the utility must have a suitable Oracle Client installed (the Instant Client is sufficient) and the client must be configured for connectivity to the Oracle database.

- When running the utility, you should ensure that your shell environment is properly set (PATH, LD_LIBRARY_PATH [or equivalent], etc.)

3. Data Type Mapping

When moving tables from an Oracle database to a TimesTen database, some degree of storage 'inflation' is typically observed due to the differing internal storage architectures of the databases.

Given that memory capacity is typically much more constrained than disk capacity it is very important to try and minimize memory usage especially when data volumes are very large.

When moving tables and their data from an Oracle database to a TimesTen database it is commonplace, and in some cases mandatory, to use different data types for certain columns in TimesTen compared to the Oracle database.

ttSizeOra understands the mandatory type mappings and will apply those automatically when computing the sizing estimate. In addition, it allows you to optionally enable numeric type mapping to achieve further memory savings by mapping NUMBER columns defined with a zero scale to the appropriate TimesTen native integer type. TimesTen native integer types are generally more space efficient than NUMBER types, and also provide better performance. Numeric type mapping is controlled by the **–typeMap** / **-noTypeMap** command line options; **-noTypeMap** is the default.

If a detailed output report is requested (by specifying the –verbose command line option) then details of all type mapped columns, both mandatory and optional, will be included in the sizing report.

For more details on data type mapping, see the documentation for the **ttImportFromOracle** utility.

4. Usage Example

In this section I present a step by step usage example. It is a very simple one but hopefully it serves to illustrate how the utility is used. The typical sequence of steps is as follows:

1. Determine access information (user, password, hostname, port, service name etc.) for the Oracle database.

2. Make sure that the system where you intent to run ttSizeOra has a functioning Oracle client installation (the Oracle InstantClient is sufficient). Configure the client installation so that it can connect to the target Oracle database and verify this using SQL*Plus.

3. Obtain the correct ttSizeOra binary for your platform by downloading it from oracle.com (see the Overview section above). Place the binary somewhere where you can easily run it from and be sure to set the permissions to make it executable (Unix or Linux platforms only).

4. Identify the tables for which you wish to obtain sizing estimates.

5. Decide which sizing and reporting options you wish to use (may differ from table to table).

6. Run ttSizeOra for each table, saving the results in a file.

7. Examine the file and use the estimates therein to size the TimesTen database.

In the rest of this section, I take a more detailed look at each of these steps and provide examples.

In this example, the Oracle database server is accessed by TNS name ‘MYDB’. The TNS entry, as configured in the client TNSNAMES.ORA file, looks like this:

```
MYDB =
  (DESCRIPTION =
```

```
    (ADDRESS =
      (PROTOCOL = TCP)(HOST = ol8-odb1.oracle.net)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl)
    )
  )
```

As for any Oracle application, the TNS_ADMIN environment variable should be set to point to the directory containing the TNSNAMES.ORA file before running the utility (if TNS name resolution is the desired method for accessing the Oracle Database server). ttSizeOra also supports the use of Oracle 'Easy Connect' syntax (serverhostname:portno/dbservicename) for accessing the Oracle database.

In this example, the credentials to access the Oracle database are username ‘scott’ and password ‘tiger’. User scott has the following tables, indexes and data:

```
CREATE TABLE PARENTTAB
(
    P_PK          NUMBER(10,0) NOT NULL,
    P_C1          NUMBER,
    P_C2          DATE NOT NULL,
    P_C3          VARCHAR2(200),
    P_C4          NVARCHAR2(1000),
    CONSTRAINT CP_PK PRIMARY KEY ( P_PK )
);

CREATE UNIQUE INDEX P_IX1 ON PARENTTAB( P_C1 );

CREATE INDEX P_IX2 ON PARENTTAB( P_C2 );

INSERT INTO PARENTTAB VALUES(1, 10, '2012-12-01', 'ABCDE', 'ABCDE');
INSERT INTO PARENTTAB VALUES(2, 20, '2012-12-02', NULL, 'A');
INSERT INTO PARENTTAB VALUES(3, 30, '2012-12-03', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ', NULL);
INSERT INTO PARENTTAB VALUES(4, 40, '2012-12-04', 'ABC', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ');
INSERT INTO PARENTTAB VALUES(5, 50, '2012-12-05', 'ABCDE', NULL);
CREATE TABLE CHILDTAB
(
    C_PK          NUMBER(6,0) NOT NULL,
    C_FK          NUMBER(10,0),
    C_C1          VARCHAR2(255),
    CONSTRAINT CC_PK PRIMARY KEY ( C_PK ),
    CONSTRAINT CC_FK FOREIGN KEY ( C_FK)
        REFERENCES PARENTTAB( P_PK )
);

INSERT INTO CHILDTAB VALUES ( 1, 3, 'ZZZ');
INSERT INTO CHILDTAB VALUES ( 2, 2, NULL );
INSERT INTO CHILDTAB VALUES ( 3, 5, 'ABCDEFGHIJKLMNOPQRST' );
```

First let us run ttSizeOra with default options to analyze these two tables and generate sizing information.

```
$ ttSizeOra  -oraConn  scott/tiger@mydb -tbl parenttab
Rows = 5
```

```
Total in-line row bytes = 23595

Out-of-line columns:
  Column P_C3                           total      200    avg size     40
  Column P_C4                           total      110    avg size     22
  Total out-of-line column bytes = 310

Indexes:
  B-tree range index SCOTT.CP_PK adds 5170 bytes
  B-tree range index SCOTT.P_IX1 adds 6194 bytes
  B-tree range index SCOTT.P_IX2 adds 5170 bytes
  Total index bytes = 16534

Total = 40439

$ ttSizeOra  -oraConn  scott/tiger@mydb -tbl childtab
Rows = 3

Total in-line row bytes = 14773

Out-of-line columns:
  Column C_C1                           total      144    avg size     48
  Total out-of-line column bytes = 144

Indexes:
  B-tree range index SCOTT.CC_PK adds 5170 bytes
  B-tree range index SCOTT.CC_FK adds 5170 bytes
  Total index bytes = 10340

Total = 25257
```

Now let's repeat the exercise specifying numeric type mapping, the use of T-tree indexes and requesting detailed output:

```
$ ttSizeOra  -oraConn  scott/tiger@mydb -tbl parenttab -typemap -ttree -verbose
Column type mappings:
  P_PK: NUMBER(10,0) -> TT_BIGINT

Rows = 5

Total in-line row bytes = 23595

Out-of-line columns:
  Column P_C3                           total      200    avg size     40
  Column P_C4                           total      110    avg size     22
  Total out-of-line column bytes = 310

Indexes:
  T-tree range index SCOTT.CP_PK adds 4074 bytes
      (as a B-tree index would be 5170 bytes)
      (as a Hash index would be 2874 bytes)
  T-tree range index SCOTT.P_IX1 adds 4074 bytes
      (as a B-tree index would be 6194 bytes)
```

```
      (as a Hash index would be 3114 bytes)
  T-tree range index SCOTT.P_IX2 adds 4074 bytes
      (as a B-tree index would be 5170 bytes)
      (as a Hash index would be 2922 bytes)
  Total index bytes = 12222

Total = 36127

$ ttSizeOra  -oraConn  scott/tiger@mydb -tbl childtab -typemap -ttree -verbose
Column type mappings:
  C_PK: NUMBER(6,0) -> TT_INTEGER
  C_FK: NUMBER(10,0) -> TT_BIGINT

Rows = 3

Total in-line row bytes = 14773

Out-of-line columns:
  Column C_C1                           total      144    avg size     48
  Total out-of-line column bytes = 144

Indexes:
  T-tree range index SCOTT.CC_PK adds 4074 bytes
      (as a B-tree index would be 5170 bytes)
      (as a Hash index would be 2874 bytes)
  T-tree range index SCOTT.CC_FK adds 4074 bytes
      (as a B-tree index would be 5170 bytes)
      (as a Hash index would be 2922 bytes)
  Total index bytes = 8148

Total = 23065
```

And we are done!

5. Detailed Syntax and Description

```
Estimate size of table and indexes when imported into TimesTen
 Usage:

ttSizeOra [-h | -help | -?]

ttSizeOra [-V | -version]

ttSizeOra  -oraConn ouser[/opasswd][@oserver]
           -tbl [owner.]tablename
          [-rows r]
          [-frac f]
          [-typeMap | -noTypeMap]
          [-inlineLimit n]
          [-parallel dop]
          [-btree | -ttree]
          [-verbose]
```

Options:

-h or -help or -?

Display this help information.

-V or -version

Display program version information.

-oraConn ouser[/opasswd][@oserver]

Specifies connection credentials for the source Oracle database.
ouser - The Oracle username.
opasswd - The password for 'ouser'. If this is omitted
the user will be prompted for the password.
oserver - Identifies the source Oracle database server. This
value may be a TNS name or an Easy Connect string in
the format 'hostname:portno/dbservice'. 'oserver' may
be omitted if you are connecting locally to the
default database.

-tbl [owner.]tablename

Specifies the Oracle database table for which sizing is to be
estimated.

-rows r

Specifies the expected number of rows in the table. 'r' must be > 0.
If this option is omitted, the utility uses the number of rows in the
existing table to estimate the size, or 1 row if the table is empty.

-frac f

Specifies the estimated average fraction of VARCHAR or VARBINARY
column sizes that will be used. 0<= f <= 1.0. If this option is
omitted and the table contains variable sized columns, a table scan is
performed to determine the average sizes. If the table is empty, the
fraction is estimated to be 0.5 (50%) filled for all variable length
columns.

-typeMap | -noTypeMap

Controls whether numeric type mapping is applied when sizing. If type
mapping is selected then columns with NUMBER type are mapped, where
possible, to the appropriate TimesTen native integer type. The
default is –noTypeMap.

-inlineLimit n

Specifies the value to be used to decide if variable length columns
will be stored inline or out of line.

```
         n   - VARCHAR2 and NVARCHAR2 columns whose maximum
defined size exceeds 'n' bytes will be stored               NOT
INLINE otherwise they will be stored INLINE.
              The minimum allowed value is 0 (always store NOT
INLINE), the maximum value is 32768 and the default               is
128.

  -parallel dop

     Specifies the degree of parallelism used when querying the
source database in the analysis phase.

       dop        - The degree of parallelism used for source table
analysis scans. Values can be in the range 0 to 128                    (0
or 1 means do not use parallel scan). The default                    is
to use the default degree of parallelism defined                    in
the source database.

  -btree | -ttree

    Specifies whether range indexes should be considered to be B-trees
or T-trees. The default is B-trees.

  -verbose

    If not specified then only basic sizing information is displayed.
    When specified, details of any type mapped columns are output as
well as sizing information for alternate index types.
```

6. Caveats and Limitations

There are many significant differences between the Oracle database and the TimesTen database, thus there are many DDL related features available in the Oracle database that cannot be exactly mapped to a TimesTen database. When ttSizeOra encounters something that cannot be sensibly mapped it will make some assumptions, output a warning message detailing those, and proceed with the sizing estimate.

The list below highlights some of the more important caveats and limitations but it is not guaranteed to be a complete or definitive list.

1. The Oracle Database Extended Types feature (12c and later) is not supported. The utility will refuse to run against an Oracle database configured with MAX_STRING_SIZE = 'EXTENDED'.

2. The only database objects that can be sized are tables, materialized views, unique constraints, primary key constraints, foreign key constraints and indexes. No other object types are supported.

3. Materialized views are sized as if they are tables.

4. Columns in Oracle database tables with data types not supported by TimesTen (user defined types and XML types for example) will not be included in the sizing estimate. A warning will be emitted, and they will be ignored.

5. TimesTen does not support the data type TIMESTAMP WITH TIMEZONE. When this type of column is imported it is not possible to import the timezone information which is therefore lost (a warning is emitted); the column is treated as a regular TIMESTAMP type.

6. Only regular and bitmap indexes are considered. All other types of index are ignored.

7. Only constraints in an ENABLED, VALIDATED state will be considered.

8. Only indexes in VALID, VISIBLE state will be considered.

9. Oracle database allows 'duplicate' indexes (same set of columns in the same order) to be created on a table while TimesTen does not. ttSizeOra will ignore all except one of any duplicate indexes.

7. Changes in this release

Changes for release 22.1.1.32.0 from release 1.2.0

- Revised versioning to 22.1.1.32.0 to be more aligned with TimesTen versions
- Added Linux ARM 64bit
- Retired Windows and masOS platforms
- Fixed a buffer size issue.

Changes for release 1.2.0 from release 1.1.1

- Updated build OS and compiler versions to address compatibility issues with newer OS versions.
- Reduced the set of supported platforms.

Changes for release 1.1.1 from release 1.1.0

- Changed the default TimesTen index type to B-tree.
- Added support for Oracle database 12c and later (non-extended type mode only).
- Reduced the list of supported platforms (dropped all 32-bit platforms).

Changes for release 1.1.0 from release 1.0.0

- Added support for Mac OS X 64-bit. This requires the use of the TimesTen 11.2.2.7.8 (or later) 64bit client. Mac OS X 32-bit is still supported using the TimesTen 11.2.1 32-bit client. **Release 1.0.0**

- Initial release

8. Known Issues

- None at this time.