H2L How-To Library

Informatica

Parallel Processing in Data Archive

Abstract

This article describes the parallel processing capabilities of Data Archive 5.3. To improve archiving performance in Data Archive, you can use parallel processing in SQL statements. You can also process multiple tables simultaneously using java parallelism/threading.

Supported Versions

- Data Archive 5.3

Table of Contents

Introduction

To improve archiving performance in Data Archive, you can use parallel processing in SQL statements. Data Archive parallelism utilizes the Parallel Query and Parallel DML Oracle database features. It invokes them by adding hints to the SQL statements it executes. You can also process multiple tables simultaneously using java parallelism/threading.

Metadata Tuning

To enable parallelism, complete the following tasks:

1. Set the relevant Degree of Parallelism attribute.
2. Add parallel hints to Data Archive metadata.

For example, to archive the PO_DISTRIBUTIONS_ALL table with parallelism, add the following parallel hint to the insert statement metadata for the table:

```
A.po_header_id IN (SELECT /*+ FULL(X) CARDINALITY(X, 1) PARALLEL(X, #) */
X.po_header_id FROM XA_4149_PO_HEADERS_INTERIM X WHERE X.purgeable_flag = 'Y')
```

The pound sign (#) is a placeholder that is replaced at runtime with the corresponding Degree of Parallelism attribute. This variable enables the same metadata to work with different degrees of parallelism, such as the Test and Production environments.

A Data Archive insert statement executes in parallel if you set the Insert Degree of Parallelism and add a parallel hint metadata where clause. Using the where clause above the engine constructs an insert statement as follows:

```
INSERT /*+ APPEND PARALLEL (Z, 8) */ INTO AM_STAGE.AA_3666113 Z
SELECT /*+ PARALLEL (A, 8) */ A.*, A.ROWID FROM PO.PO_DISTRIBUTIONS_ALL A
WHERE
A.po_header_id IN (SELECT /*+ FULL(X) CARDINALITY(X, 1) PARALLEL(X, 8) */
X.po_header_id FROM XA_4149_PO_HEADERS_INTERIM X WHERE X.purgeable_flag = 'Y')
```

If the metadata does not include a PARALLEL(X, #) hint, the Data Archive engine does not process the table in parallel. Also, no parallel hint is added to the insert clause of the statement. This allows you to control which tables the Data Archive engine archives in parallel.

The FULL and CARDINALITY hints ensure that the optimizer uses the proper execution plan when selecting the data to archive. The FULL hint suggests that Oracle does a full table scan on the interim table. The CARDINALITY hint tells the optimizer that the subselect only returns one row, which of course is not accurate, but it ensures that the ERP table is accessed using an index.

It is important to tune the statement properly for the parallel processing to work efficiently. Parallel and other hints might be required in multiple places in the metadata statement text, often combined with other hints, to be effective and improve performance. If you need help tuning one or more statements, contact Informatica Global Customer Support.

The Data Archive engine can also execute business rule update statements in parallel. The implementation is the same. The Update Degree of Parallelism attribute controls the parallelism for these update statements.

Data Archive does not require metadata changes to support delete parallelism. If the Delete Degree of Parallelism attribute is greater than one, the Data Archive engine adds the necessary parallel hints automatically. However, it only processes tables without a primary key metadata constraint in parallel if the Use Oracle Parallel DML For Delete source repository attribute is enabled and the metadata delete statement contains a parallel hint.

Performance-Related Source Repository Attributes

Use Staging

If enabled, the Data Archive engine temporarily copies the data to be archived to staging tables on the ERP instance. Enable this option to increase performance.

Use ROWID for Delete

If enabled, the Data Archive engine stores the ROWID of the source rows in the staging tables and the Delete from Source step deletes rows using the ROWID. This is the fastest way to delete rows in Oracle and the recommended setting.

Use Oracle Parallel DML for Delete

The way Data Archive processes delete statements during the Delete from Source step is controlled by the Use Oracle Parallel DML for Delete source repository attribute. If this attribute is enabled, Data Archive uses Oracle's parallel DML. Otherwise, the Data Archive engine controls delete parallelism

Oracle Parallel DML

Delete parallelism is achieved by using Oracle's parallel query/DML features. These features are invoked by adding parallel hints to the delete statements that are executed. This method is the fastest way in Oracle to delete data. The following example shows a typical delete statement using Oracle parallelism:

```
DELETE /*+ PARALLEL(A,8) */ FROM PO.PO_DISTRIBUTIONS_ALL A
WHERE (ROWID, PO_DISTRIBUTION_ID) IN (
SELECT /*+ CARDINALITY(X,1) PARALLEL(X,8) */ APPLIMATION_ROW_ID,
PO_DISTRIBUTION_ID FROM AM_STAGE.AA_3666113 X)
```

Parallel Delete Controlled by Data Archive

Data Archive spawns and controls multiple delete processes. Each delete process is a Java thread, where each worker processes a range of rows. This method is slower than Oracle's parallelism, but does not lock the table. Therefore, use this option when you schedule archive cycles during business hours.

The following are sample delete statements executed by parallel worker threads:

```
DELETE FROM PO.PO_DISTRIBUTIONS_ALL A
WHERE (ROWID, PO_DISTRIBUTION_ID) IN (
SELECT /*+ CARDINALITY(X,1) */ APPLIMATION_ROW_ID, PO_DISTRIBUTION_ID FROM
AM_STAGE.AA_3666113_T1 X WHERE APPLIMATION_ROW_NUM BETWEEN 1 AND 100000)

DELETE FROM PO.PO_DISTRIBUTIONS_ALL A
WHERE (ROWID, PO_DISTRIBUTION_ID) IN (
SELECT /*+ CARDINALITY(X,1) */ APPLIMATION_ROW_ID, PO_DISTRIBUTION_ID FROM
AM_STAGE.AA_3666113_T1 X WHERE APPLIMATION_ROW_NUM BETWEEN 1000001 AND 200000)
```

Consider the following factors before you use Oracle parallel DML:

- Oracle parallelism during delete operations requires an exclusive lock on the table that is deleted, preventing other users from updating the same table. Read access on these tables is still possible. Use Oracle parallelism if archive cycles are scheduled outside of business hours when no users need write access to the ERP system.
- The source or ERP tables need to support parallel DML. Tables that you created in an Oracle 8/8i database, which you upgraded to 9i/10g, do not support parallel DML. Use the query below to check the relevant table property:

```
select
 a.property,
 decode(bitand(a.property,536870912), 0, 'DISABLED', 'ENABLED') pdml_enabled
from sys.tab$ a, dba_objects b
where a.obj# = b.object_id
and b.object_name = '&table_name' and b.owner = '&owner';
```

If a table does not support parallel DML, Oracle provides a means to enable it. For more information, contact Informatica Global Customer Support.

Archive Definition Setup

Analyze Interim

This attribute controls when optimizer statistics are gathered for interim tables during Generate Candidates. Analyze the interim for optimal table and index statistics for the Oracle Cost Base Optimizer.

Select one of the following options:

- After Insert. The interim table is analyzed after data is inserted.
- After Insert and Update. The interim table is analyzed after every insert and update statement.
- After Update. The interim table is analyzed after business rule update statements.
- None. No statistics are gathered.

Delete Commit Interval

This attribute controls the commit frequency during delete operations during the Delete from Source step. If the value is 0 or if the table processed does not have a primary key metadata constraint defined, then the delete is executed as a single statement and no commit interval is used.

Insert Commit Interval

This attribute controls the commit frequency during insert operations when you cannot use an "INSERT AS SELECT" type statement.

You cannot use Insert as Select in the following cases:

- The table has a LONG column.
- If the Database Link to Source destination repository attribute is not set during the Copy to Destination step.
- If the table processed has a user-defined type column, such as WF_EVENT_T, and the database version is Oracle 9i or lower during the Copy to Destination step.

The Generate Candidates and Copy to Staging steps use Insert as Select type statements except when the table contains a LONG column. Undo usage is minimal during these steps, because the Data Archive engine adds an APPEND hint to the insert statement. In this case, Oracle does a direct path data load above the table high water mark.

Degree of Parallelism

In general, the possible degree of parallelism depends on the following factors:

- The number of CPUs available on the database server. Use the following query to find out about the number of CPUs available to an Oracle instance:

```
select value from v$parameter where name = 'cpu_count';
```

- The setting of the parallel_max_servers initialization parameter. Use the following query to check the current setting:

```
select value from v$parameter where name = 'parallel_max_servers';
```

- Set this parameter to at least twice the value of the degree of parallelism chosen. For example, to execute a delete statement with a degree of 8, Oracle needs to spawn 16 parallel threads, where 8 threads work on the necessary select to retrieve the rows and 8 perform the delete.

Delete Degree of Parallelism

The degree of parallelism used for delete operations during the Delete from Source step.

Insert Degree of Parallelism

The degree of parallelism used for insert operations during the Generate Candidates, Copy to Staging, and Copy to Destination steps.

Update Degree of Parallelism

The degree of parallelism used for update operations during the Generate Candidates step.

Restore Definition Setup

Delete Commit Interval

Unused. Restore cycles issue a single delete statement. Because single commit delete statements possibly delete large volumes of data, size the UNDO tablespace on the history database accordingly.

Insert Commit Interval

This attribute controls the commit frequency during insert operations when you cannot use an Insert as Select statement.

You cannot use Insert as Select in the following cases:

- The table has a LONG column.
- The Database Link to Source destination repository attribute is not set during the Copy to Destination step.
- The table processed has a user defined type column, such as WF_EVENT_T, and the database version is Oracle 9i or lower during the Copy to Destination step.

Delete Degree of Parallelism

Currently unused.

Insert Degree of Parallelism

The degree of parallelism used for insert operations during the Generate Candidates, Copy to Staging, and Copy to Destination steps.

Java Parallelism - Processing Multiple Tables Simultaneously

The Data Archive engine can process multiple tables simultaneously depending on the step executed and the value of the corresponding Degree of Parallelism attribute.

The maximum number of concurrent processes or threads depends on the value of the informia.maxActiveAMThread property. Set this property in the conf.properties file located in the web server root directory. Default is 10. This property is a system-wide setting. It does not apply to a single archive cycle. It limits the total number of java threads the Data Archive engine executes in parallel.

Use the following URL to see all active and pending threads in the Data Archive thread queue manager:

```
http://<hostname>:port/jsp/tqm.jsp
```

Generate Candidates

This step executes in step order.

Build Staging

This step processes multiple tables simultaneously up to the maximum number of threads allowed.

Copy to Staging

This step processes the tables sequentially in step order and uses Oracle parallel DML if the insert degree of parallelism is greater than one. However, you can set the insert degree to one to enable java parallelism. Data Archive processes multiple tables simultaneously up to the maximum number of threads allowed.

Validate Destination

This step processes multiple tables simultaneously up to the maximum number of threads allowed.

Copy to Destination

This step processes the tables sequentially in step order and uses Oracle parallel DML if the insert degree of parallelism is greater than one. However, you can enable Java parallelism by setting the insert degree to one. Data Archive processes multiple tables simultaneously up to the maximum number of threads allowed.

Restore cycles process the Copy to Destination step sequentially in step order.

Purge Staging

This step processes multiple tables simultaneously up to the maximum number of threads allowed.

Database Link Setup for Restore Cycles

To increase performance, Data Archive can use database links during Restore cycles. The Data Archive engine copies data using Insert as Select SQL statements. However, since the Copy to Destination step during Restore possibly inserts into heavily indexed ERP tables, you need to properly size the UNDO tablespace on the ERP instance.

To set up the restore database links:

1. Connect to the ERP source instance as the application user.
2. Create a private database link that points to the staging user on your history instance.
3. Edit the Restore Destination Repository and enter the name of the database link created in the Database Link to Source attribute.
4. Connect to the history instance as the history agent user and create a private database link that points to the application user on the ERP instance.
5. Edit the Restore Source Repository and enter the name of the database link created in step 4 in the Database Link to Production (Restore only) attribute.

Author

Thomas Wagner
Senior Product Specialist