Informatica Data Archive

Parallel Processing Benchmarks
Parallel Processing in Data Archive

Objectives

This document presents statistics and results from parallel processing used in the Data Archive product to achieve the archive and purge of a large volume of data in a specific timeframe. Two major objectives to be achieved are as follows:

1. Archive large volume of AP data within a 6-hour window with a preference for complete processing under 3 hours.
2. Validate that the source ERP data remained the "same" after being archived into the HISTORY database.

Outcome Overview

Both objectives outlined by the customer were implemented successfully. The outcome summary is described here, and then the details of the analysis are provided in subsequent sections:

Objective

1. Archive large volume of AP data in less than 3 hours.
2. Validate the source data remained the same after being archived into the History database.
Parallel Processing Overview

Achieved

1. Processed 2.5 million invoices (for a total of 52.7 million records) in 2 hours and 27 minutes.
2. Provided audit of 2 invoices that were archived from 3 different tables (AP_INVOICES_ALL, AP_INVOICE_PAYMENTS_ALL, and AP_INVOICE_DISTRIBUTIONS_ALL).

Data Archive’s parallel processing utilizes Oracle database features ("Parallel Query" and "Parallel DML") and supplements them by adding performance tuning hints to the DML statements it executes. To improve the performance of the archiving process, customers that have application environments containing a lot of data (typically more than a terabyte) and multiple processors on the database server can take advantage of Data Archive’s ability to use parallel processing in its SQL statements.

Parallel processing in Data Archive is enabled via the Instance Configuration page. The relevant attributes are "Degree of Parallelism (DOP) for Archive Statements" and "Degree of Parallelism (DOP) for Delete Statements". The Archive DOP controls the parallelism for insert statements executed during the Generate Candidates and Archive steps of an archiving cycle.

If the DOP for Delete Statements is set to any value greater than 1, Data Archive automatically adds the necessary hints to all archiving delete statements. A typical parallel delete statement looks like this:

```
DELETE /*+ PARALLEL(A,8) */ FROM PO.PO_DISTRIBUTIONS_ALL A
WHERE ROWID IN (SELECT /*+ CARDINALITY(X,1) PARALLEL(X,8) */ ROW_ID
FROM AM_RELOC.AA_000000003666113 X)
```

Enabling Insert/Archive parallelism requires, in addition to setting the Archive DOP attribute, adding parallel hints to Data Archive metadata. For example, archiving the PO_DISTRIBUTIONS_ALL table with parallelism requires a hint to the where clause metadata for this table:

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

The # sign is a placeholder and is replaced at runtime with the specified Archive DOP parameter. The use of this variable enables the same metadata to work with different degrees of parallelism. An Data Archive insert statement is only executed in parallel if both the Archive DOP is set and a parallel hint is found in the where clause metadata. Here is an example:

```
INSERT /*+ APPEND PARALLEL (Z, 8) */ INTO
AM_RELOC.AA_000000003666113 Z
SELECT /*+ PARALLEL (A, 8) */ A.*, A.ROWID
FROM PO.PO_DISTRIBUTIONS_ALL A
WHERE A.po_header_id IN
( SELECT /*+ FULL(X) PARALLEL(X,8) */ X.po_header_id
FROM XA_4149_PO_HEADERS_INTERIM X
WHERE X.purgeable_flag = 'Y' )
```

Parallel hints are required in multiple places in the metadata statement text, often combined with other hints, to be effective and improve performance.

By managing the degree of parallelism for archiving and purging the load on the database server can be controlled. Increasing the degree of parallelism will result in a higher CPU usage. Conversely, decreasing the degree of parallelism will result in a lower CPU usage. This allows the customer to control CPU usage on a per instance basis and to meet the needs of the specific environment. Since Data Archive is a live archive, some customers may want to
allocate less CPU power to the archive/purge process because they might run a 24/7 operation or they may have batch processing occurring at the same time. Other customers may lock all users and processes out while archive/purge is running, and therefore, allocate more or all CPU power to the archive/purge process. These are just some of the factors to be considered when determining the degree of parallelism. Setting the degree of parallelism too high can have the opposite effect on archive/purge processing resulting in inefficient processing.

Environment

This section describes the environment in which the archiving and purging was conducted.

HOME Repository

Instance: APLM
DB Version: 10.2.0.2
Data Archive version: 5.1

The HOME database server was a Sun Sunfire 15k machine with 36 dual-core processors running Solaris 10. Data Archive 5.1 was installed on this server. The web application server used was Tomcat 5.5.20 running Java 1.5_u6.

Oracle Applications Target Database

Oracle ERP Database
Instance: POC2
DB Version: 10.2.0.2 ERP Version: 11.5.10 Database Size: ~ 7 Tb

The Oracle ERP Database was running in a 2-node RAC cluster on two Sun Sunfire 15k machine with 36 dual-core processors running Solaris 10.

History Database

Instance: APLM
DB Version: 10.2.0.2

The HOME database server was a Sun Sunfire 15k machine with 36 dual-core processors running Solaris 10. The History database was the same database instance as the HOME database. The History database is represented by a single schema called AM_HISTORY.

Parallel Options Utilized

Insert Degree of Parallelism = 32 Delete Degree of Parallelism = 48

Accomplishments

Following are the accomplishments achieved during the archive/purge processing, which includes 5 steps: Candidate Generation, Freeze Process, Build Staging Area, Archive/Purge and Xpress Merge.
Archive Large Volume of AP Data within 3 Hours

Data Archive relocated 2.5 million invoice transactions from the ERP database to the online archive database APLM. This equated to a total of 52.7 million records. The complete start to finish time of the five standard Data Archive processes was 2 hours and 27 minutes. The breakdown of each process is as follows:

**Process**
- Generate Candidates 0:28:27
- Freeze Process 0:00:01
- Build Staging Area 0:00:08
- Archive/Purge Xpress Merge 1:05:22
- Total Time 0:53:35
- Elapsed Time 2:27:33

CPU usage on the ERP database server during the archive/purge was around 30 - 40 % peak. While the archive/purge process could have utilized more CPU power, the customer and Informatica agreed that this was the optimal outcome since there was no downtime available and other processing was occurring during the archive/purge process.

Job logs of the five processes are provided as an attachment to this document. The poc_job_logs.txt file shows each step performed in the five processes.

Counts were taken prior to the archive/purge, after the archive/purge and after the Xpress Merge. The Data Archive product provides as standard functionality the option to do pre-archive, post-archive, and post-xpress merge counting to provide audit level reports.

The specific table counts are as follows:

<table>
<thead>
<tr>
<th>TABLE_NAME</th>
<th>Before Count ERP</th>
<th>After Count ERP</th>
<th>Relocation Count</th>
<th>History Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP_ACCOUNTING_EVENTS_ALL</td>
<td>1,019,675,297</td>
<td>1,015,225,213</td>
<td>4,450,084</td>
<td>4,450,084</td>
</tr>
<tr>
<td>AP_AE_HEADERS_ALL</td>
<td>1,016,405,485</td>
<td>1,011,955,401</td>
<td>4,450,084</td>
<td>4,450,084</td>
</tr>
<tr>
<td>AP_AE_LINES_ALL</td>
<td>4,368,667,184</td>
<td>4,341,045,668</td>
<td>27,621,516</td>
<td>27,621,516</td>
</tr>
<tr>
<td>AP_BATCHES_ALL</td>
<td>40,726</td>
<td>40,571</td>
<td>155</td>
<td>155</td>
</tr>
<tr>
<td>AP_CHECKS_ALL</td>
<td>468,677,482</td>
<td>466,676,982</td>
<td>2,000,500</td>
<td>2,000,500</td>
</tr>
<tr>
<td>AP_DOC_SEQUENCE_AUDIT</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AP_ENCUMBRANCE_LINES_ALL</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AP_HOLD_ALL</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AP_INV_APRVL_HIST_ALL</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AP_INV_SELECTION_CRITERIA_ALL</td>
<td>37,173</td>
<td>37,173</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AP_INVOICE_DISTRIBUTIONS_ALL</td>
<td>1,111,521,443</td>
<td>1,104,616,064</td>
<td>6,905,379</td>
<td>6,905,379</td>
</tr>
<tr>
<td>AP_INVOICE_PAYMENTS_ALL</td>
<td>525,411,977</td>
<td>522,962,393</td>
<td>2,449,584</td>
<td>2,449,584</td>
</tr>
<tr>
<td>AP_INVOICES_ALL</td>
<td>550,238,340</td>
<td>547,788,756</td>
<td>2,449,584</td>
<td>2,449,584</td>
</tr>
<tr>
<td>AP_MC_CHECKS</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AP_MC_INVOICE_DISTS</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AP_MC_INVOICE_PAYMENTS</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AP_MC_INVOICES</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
### Performance Tuning Hints Used

See the attached file, `ap_arch_purge.tkp`, for details on performance tuning hints used on a per table basis for both archive statements and purge statements.

### Assurance of Data Integrity During Archive/Purge

To ensure the integrity of the data remained intact during the archive/purge process, queries were run for 2 invoices to be archived against 3 tables. These tables were:

- **AP_INVOICES_ALL**
- **AP_INVOICE_DISTRIBUTIONS_ALL**
- **AP_INVOICE_PAYMENTS_ALL**

Queries were run against the source AP tables prior to the archive, against the source AP tables after the archive (expecting 0 rows to be returned), and against the History tables after the archive. A column by column comparison shows that the records were unchanged as they were moved from the Source ERP database to the History database.

The query output shows that 2 records were retrieved from **AP_INVOICES_ALL**, 6 records from **AP_INVOICES_DISTRIBUTIONS_ALL** and 2 records from **AP_INVOICE_PAYMENTS_ALL**.

Because of formatting issues the query output is provided in the original spool files from SQL*Plus which are attached as part of the Data Archive Summary documents. The files are:

- invoices_erp_before.txt: this file contains the query output from the 3 above-mentioned tables from the AP schema in the POC2 database prior to the data being archived.
- invoices_erp_after.txt: this file contains the query output from the 3 above-mentioned tables from the AP schema in the POC2 database after the data was archived. These queries return no rows as expected.
- invoices_am_history.txt: this file contains the query output from the 3 above-mentioned tables from the AM_HISTORy schema in the APLM database after the data was archived.
Authors

ILM Products team