An Oracle White Paper
February 2009

Parallel Capabilities of Oracle Data Pump

ORACLE

Introduction

Oracle Data Pump, available starting in Oracle Database 10g, enables very high-speed movement of data and metadata from one database to another. One of the most useful features of Data Pump is the ability to parallelize the work of export and import jobs for maximum performance.

This paper discusses how the PARALLEL parameter works and what the user should do to optimize this feature.

Parallel Parameter

The Data Pump Export and Import (expdp and impdp) PARALLEL parameter can be set to a value greater than one only in the Enterprise Edition of Oracle Database. A user must be privileged in order to use a value greater than one for this parameter. (In Standard Edition, the PARALLEL parameter is limited to one.) It is most useful for big jobs with a lot of data relative to metadata. Small jobs or jobs with a lot of metadata will not see significant improvements in speed. Unless otherwise noted, the parallelism and the movement of data, not metadata, will be discussed in this paper.

Metadata is never unloaded in parallel, but is sometimes loaded in parallel. There are two situations when metadata is loaded in parallel:

- Multiple workers will load package bodies in parallel.
- One worker will create all the indexes and use Parallel Execution Processes (PX processes) to speed up the work. Parallel Execution Processes can significantly reduce the elapsed time for large indexes.

In Data Pump terminology, a table data object is the basic unit of storage. If a table is not partitioned, the table data object is the table itself. If a table is partitioned but is not subpartitioned, then there is one table data object for each partition. If there are subpartitions, then there is one table data object for each subpartition.

Data Pump uses a multiprocess architecture in which a master control process dispatches job items to one or more worker processes. These worker processes may use PX processes to move data. Parallelism in this context refers to the total number of active worker processes and PX processes that are able to operate in parallel. The default PARALLEL value is one. If Data Pump decides to use PX processes to load or unload a table, then the PX processes are counted against the degree of parallelism used by Data Pump, but the worker process that starts the PX processes is not since it is idle while the PX processes are executing the query. Also, if a worker process has been started but it is idle, it is not counted against the limit specified by the PARALLEL parameter.

Master Control Process / Worker Process

One master control process (MCP) is created for each Data Pump export or import job. The MCP controls the whole job by communicating with the client, starting and controlling the pool of worker processes, performing logging operations, and much more.

The master control process will create a pool of active worker processes to handle work items as needed up to the parallel number. The master control process does not count toward the

parallel total. The degree of parallelism can be increased and decreaseddynamically throughout the life of the job. This is done by the user through the interactive command mode.

Decreasing parallelism does not result in fewer worker processes associated with the job; it merely decreases the number of active worker processes that will be executing at any given time. Any ongoing work must reach an orderly completion point before the decrease takes effect. Therefore, it may take a while to see any effect from decreasing the parallel value. Idle workers are not deleted until the job exits.

Increasing the parallelism takes effect immediately if there is work that can be performed in parallel by a worker or PX process. If there is no work that can be performed in parallel, then no additional worker processes will be created until there is work that requires additional workers.

In a Real application Cluster (RAC) environment, only one instance may run Data Pump jobs at any given time. All worker processes associated with a Data Pump job will be started on the instance where the master control process is running. Note that when workers deploy PX processes, the processes may run on other instances in the RAC transparently.

Dumpfiles

For Data Pump Export, the value that is specified for the parallel parameter should be less than or equal to the number of files in the dump file set. Each worker or Parallel Execution Process requires exclusive access to the dump file, so having fewer dump files than the degree of parallelism will mean that some workers or PX processes will be unable to write the information they are exporting. If this occurs, the worker processes go into an idle state and will not be doing any work until more files are addedto the job. See the explanation of the DUMPFILE parameter in the Database Utilities guide for details on how to specify multiple dump files for a Data Pump export job.

For Data Pump Import, the workers and PX processes can all read from the same files. However, if there are not enough dump files, the performance may not be optimal because multiple threads of execution will be trying to access the same dump file. The performance impact of multiple processes sharing the dump files depends on the I/O subsystem containing the dump files. For this reason, Data Pump Import should not have a value for the PARALLEL parameter that is significantly larger than the number of files in the dump file set.

Access Methods: Direct Path and External Tables

Data Pump supports two access methods to load and unload table row data: direct path and external tables. The access methods play a key role in determining how much parallelism

will be possible. Because both methods support the same external data representation, data that is unloaded with one method can be loaded using the other method. Data Pump automatically chooses the appropriate method for each table data object.

Data Pump provides an external tables access driver that reads and writes files. The format of the files is the same format used with the direct path method. This allows for high-speed loading and unloading of database tables as an alternative to direct path. Although external tables single stream performance is slower than direct path, it exploits the Oracle parallel execution engine for very large tables and partitions, which the Direct Path API cannot do.

Here are some common situations where Data Pump uses external tables as the data access method:

- Loading and unloading very large tables and partitions in situations where parallel SQL can be used and a parallel Data Pump operation was requested
- Loading and unloading tables that contain one or more columns of type BFILE or opaque, or any object type containing opaque columns
- Loading and unloading tables with encrypted columns
- Unloading table and using the QUERY parameter
- Loading tables with active triggers
- Loading clustered tables
- Loading tables with fine-grained access control enabled for inserts
- Loading tables with a global index on a partitioned table

For a complete list of when Data Pump uses external tables, please refer to the technical note on OTN: http://www.oracle.com/technology/pub/notes/technote_pathvsext.html.

How the Parallel Parameter Works for Export Operations

For the export operations described in this section, it is assumed that the user is using Oracle Database Enterprise Edition and the wildcard option with the DUMPFILE parameter.

In a typical export that includes both data and metadata, the first worker process will unload the metadata: tablespaces, schemas, grants, roles, tables, indexes, and so on. This single worker unloads the metadata, and all the rest unload the data, all at the same time. If the metadata worker finishes and there are still data objects to unload, it will start unloading the data too. The examples in this document assume that there is always one worker busy unloading metadata while the rest of the workers are busy unloading table data objects.

For every export operation, Data Pump estimates how much disk space each table data object in the export job will consume (in bytes). This is done whether or not the user uses the ESTIMATE parameter. The estimate is printed in the log file and displayed on the client's standard output device. The estimate is for table row data only; it does not include metadata. This estimate is used to determine how many PX processes will be applied to the table data object, if any.

The columns of the tables are examined to determine if direct path, external tables, or both methods can be used. For direct path, the parallel number for the table data object is always one since direct path does not support parallel unload of a table data object. PX processes are only used with external tables.

If the external tables method is chosen, Data Pump will determine the maximum number of PX processes that can work on a table data object. It does this by dividing the estimated size of the table data object by 250 MB and rounding the result down. If the result is zero or one, then PX processes are not used to unload the table.

For example, if the table size is 600 MB, there will be two Parallel Execution Processes determined by dividing the estimated size by 250 MB (parallel threshold) and rounding down. In this case there will also be two worker processes: one for the metadata and one for the data. The worker process for the data acts as the coordinator for the PX processes and does not count toward the parallel total. Thus, in this case, the degree of parallelism used by the export job is three: one for the metadata worker and two for the PX processes. If the user looks at the log file or uses the interactive STATUS command to see what the workers are doing, only two worker processes will be visible as the PX processes are not visible in expdp and impdp.

Consider another example where the table size is 400 MB. In this case, there will be no Parallel Execution Process. When 400 MB is divided by 250 MB, the resulting parallel value is one. The worker will unload the data using either direct path or external tables without any parallelism.

If a job is not big enough to make use of the maximum parallel number, then the user will not see the maximum number of active workers and Parallel Execution Processes. For example, if there is one 800 MB table, and it has been determined that external tables will be used, there will be one worker for the metadata, one worker for the data, and three PX processes. As mentioned above, the worker process for the data acts as the coordinator for the PX processes and does not count toward the parallel total. So, if a user specifies PARALLEL = 10, the degree of parallelism is actually four. The user will only see one active worker in the STATUS display. Data Pump is working optimally; the job is too small for the specified degree of parallelism.

What happens if there is a large job and PARALLEL = 4? Can the Master Control Process automatically add more workers if they are needed? No, the PARALLEL parameter can only go up to the maximum number that was specified by the user. If the degree of parallelism for

a table would cause the job to exceed the limit specified by the PARALLEL parameter, the number of Parallel Execution Processes will be scaled down to fit within the limit. The user can, however, increase the parallelism of the job via the interactive command line.

In a RAC environment, Parallel Execution Processes can run on other instances. The user might not see them running on the same instance where the Data Pump job is running. This might make it even less obvious that Data Pump is running optimally.

How the Parallel Parameter Works for Import Operations

The PARALLEL parameter works a bit differently in Import than Export. Because there are various dependencies that exist when creating objects during import, everything must be done in order. For Import, no data loading can occur until the tables are created because data cannot be loaded into tables that do not yet exist.

Data Pump Import processes the database objects in the following order:

- The first worker begins to load all the metadata – the tablespaces, schemas, etc., until all the tables are created.
- Once the tables are created, the first worker starts loading data instead of metadata and the rest of the workers start loading data too.
- Once the table data is loaded, the first worker returns to loading metadata again. The rest of the workers are idle until the first worker loads all the metadata up to package bodies.
- Multiple workers load package bodies in parallel.
- One worker loads metadata up to and including secondary tables.
- Multiple workers load secondary table data.
- One worker loads the remaining metadata.

Note: One worker creates all the indexes but uses PX processes up to the PARALLEL value so indexes get created faster.

Thus, an import job can be started with a PARALLEL = 10, and the user will only see one worker being utilized at certain points during job execution. No other workers or Parallel Execution Processes will be working until all the tables are created. When the tables are created, a burst of workers and possibly PX processes will execute in parallel until the data is loaded, then the worker processes will become idle.

When loading table data objects, Data Pump divides the size of the data in the dump file by 250 MB to estimate the number of potential PX processes that can be used to load the data. If the number is greater than one, Data Pump can use PX processes to load the data, assuming

there is enough parallelism available. Note that table data objects that are large enough can be loaded in parallel even if they were unloaded by one worker.

Getting the Most From the Parallel Parameter

Here is a general guideline for what should be considered when using the PARALLEL parameter:

- Set the degree of parallelism to two times the number of CPUs, then tune from there.
- For Data Pump Export, the PARALLEL parameter value should be less than or equal to the number of dump files.
- For Data Pump Import, the PARALLEL parameter value should not be much larger than the number of files in the dump file set.
- A PARALLEL greater than one is only available in Enterprise Edition

Conclusion

Users want to optimize the use of Data Pump's parallel capabilities so that export and import jobs run as efficiently as possible. If the PARALLEL parameter and the wildcard dump file template are used, and the job has a large amount of data (not metadata), then performance over Original Export and Original Import will be much improved.

Data Pump uses the PARALLEL parameter as the maximum allowed parallelism for the job, with the maximum number of active workers and Parallel Execution Processes at any given time. If a user is monitoring the workers, there will be times when workers will be busy and other times when some workers may be idle. This is expected and means that Data Pump is behaving properly.

Parallel Capabilities of Oracle Data Pump
February 2009
Author: Carol Palmer

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Oracle is committed to developing practices and products that help protect the environment

Copyright © 2009, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

0109