An Oracle White Paper
September 2009

Oracle Data Pump On Oracle Real Application Clusters

Disclaimer

Introduction

Oracle Data Pump, first available in Oracle Database 10*g*, enables very high-speed movement of data and metadata between databases. One of the most useful features of Data Pump is the ability to parallelize the work of export and import jobs for maximum performance. However, prior to Oracle Database 11*g* release 2 (11.2), the use of parallelism was generally confined to a single instance, even when that instance was part of an Oracle Real Application Clusters (RAC) configuration. Data Pump did not take advantage of cluster resources to provide more parallelism and higher availability.

When you are planning to use parallelism, it is important to first consider various factors to determine whether a Data Pump job is a good candidate for it. The following are some of the factors that determine the amount of paralellism that is possible in a Data Pump job (regardless of whether the job is running in an Oracle RAC environement or not):

- The number of files in a dump file set relative to the parallelism requested
- Whether the wildcard option is used with the DUMPFILE parameter
- The access method used - direct path or external tables
- Whether Oracle Database Enterprise Edition is being used (this is the only edition on which Data Pump supports the use of parallelism)

This paper discusses how you can control parallelism of Data Pump jobs in an Oracle RAC environment and what is happening behind the scenes in parallel mode.

The Oracle Data Pump PARALLEL Parameter

The Data Pump Export and Import utilities (expdp and impdp) provide a PARALLEL parameter that can be used to set the degree of parallelism for a Data Pump job. It is valid only in the Enterprise Edition of Oracle Database 10*g* and later. The degree of parallelism controls the number of worker processes and parallel slaves that can operate on data simultaneously when the master control process decides there is some advantage to doing so. The PARALLEL parameter is most useful for big jobs with a lot of data relative to metadata. Small jobs or jobs with a lot of metadata relative to data do not typically show significant improvements in speed.

Data Pump is a multiprocess architecture. When a job is started by the impdp or expdp clients or with the Data Pump API, a master control process for the job is created first. The master control process divides the job into work items and decides which work items can be done in parallel. Depending on the specified degree of parallelism for the job, the master control process can create multiple worker processes to operate on the data. Worker processes, in turn, may employ parallel slaves to do some of the work depending on the type and size of the data and on whether direct path or external tables is being used to move the data.

Prior to Oracle Database 11*g* release 2 (11.2), the master control process created worker processes on the same instance on which it was running, ignoring whether or not the instance was part of an Oracle RAC environment. By ignoring the fact that the instance on which it was running was part of an Oracle RAC, the master control process did not capitalize on the ability to use other potentially idle instances. Those idle instances are valuable resources not being used to their best advantage. (Note that it has always been possible for parallel slaves to run on other instances in an Oracle RAC.)

What s New for Data Pump in an Oracle RAC Environment in Oracle Database 11*g* Release 2 (11.2)?

There are two noticeable changes for Data Pump in an Oracle RAC environment in Oracle Database release 11.2. First, there is no longer a restriction on running multiple Data Pump jobs simultaneously on different Oracle RAC instances. Prior to Oracle Database release 11.2, it was only possible to start a Data Pump job on one instance if no Data Pump jobs were running on other instances within the Oracle RAC configuration. You had to either run all jobs on the same instance or wait for jobs to finish on other instances first. This restriction has been lifted in Oracle Database release 11.2, and jobs can be started and run simultaneously on different instances of the Oracle RAC configuration.

The other change in 11.2 is that worker processes are no longer confined to running on the same Oracle RAC instance as the client or master control process. Worker processes can be distributed across Oracle RAC instances to better utilize Oracle RAC resources and provide higher levels of availability. The only constraint is that the directory object specified for dump files must point to shared storage that can be seen by all Oracle RAC instances where worker processes are allowed to run. (Note that this has always been true when it is possible for parallel slaves to run on other Oracle RAC instances as part of a Data Pump job.)

There are two new client parameters: CLUSTER and SERVICE_NAME. The information provided with these new parameters is passed along through the DBMS_DATAPUMP.START_JOB() API. The CLUSTER parameter informs Data Pump whether it is okay to use cluster resources. The default is CLUSTER=Y. You can specify CLUSTER=N to force Data Pump processes (the worker processes, in particular) to run only on the instance where the job is started. When CLUSTER=Y or the default setting is used, the SERVICE_NAME parameter can be specified to constrain the job to run on a subset of Oracle RAC instances, rather than on any instance in the cluster. The SERVICE_NAME must be a valid known service that identifies a resource group associated with that service. The resource group defines the Oracle RAC instances that belong to that group. If SERVICE_NAME is not specified to constrain which Oracle RAC instances can be used, Data Pump attempts to use all active Oracle RAC instances.

If the Oracle RAC instance where the client and master control process are running dies or leaves the cluster, the job aborts and can probably be restarted at some future time. If some other Oracle RAC instance dies where a worker process is running, Data Pump attempts to start a new worker on a different Oracle RAC instance to finish the work that worker was doing. If there are multiple failures, the job will probably abort but should be restartable at some later time. In most cases of single-point failures, the worker process failover capability should lead to higher overall availability.

For Data Pump jobs running in a non-Oracle-RAC environment, it may not make sense to specify high degrees of parallelism. But in an Oracle RAC environment, high degrees of parallelism allow distribution of worker processes across the Oracle RAC environment, which can be beneficial when there are large amounts of data, relative to metadata, to move. Because there is some overhead involved with distributing Data Pump processes across an Oracle RAC environment, there will be cases where there is no performance benefit. In those cases, you can specify CLUSTER=N to force all Data Pump processes to run on a single instance. Note that this has no bearing on parallel slaves and whether they run on other Oracle RAC instances.

Dump Files

It is worth reemphasizing the importance of specifying dump files properly in the Data Pump job to achieve the desired levels of parallelism, especially in an Oracle RAC environment. First, make sure the directory object points to shared storage that can be seen by any Oracle RAC instances designated to run Data Pump worker processes. Also, 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. If there are not enough dump files, the performance will not be optimal because multiple threads of execution will be trying to access the same dump file. If this occurs, the worker processes go into an idle state.

Rather than listing specific file names with the DUMPFILE parameter, the preferred approach is to use the wildcard option with the DUMPFILE parameter during export operations. The wildcard option allows you to specify multiple dump files by using a substitution variable (%U) in the filename. This is called a dump file template. The new dump files are created as they are needed, beginning with 01 for %U, then using 02, 03, and so on. Enough dump files are created to allow all processes specified by the current setting of the PARALLEL parameter to be active. If one of the dump files becomes full because it has reached the maximum size specified by the FILESIZE parameter, it is closed and a new dump file (with a new generated name) is created.

If %U is used, then during an import each file that matches the template is examined (until no match is found) in order to locate all the files that are part of the dump file set. Sufficient information is contained within the files for Import to locate the entire set, provided the file specifications in the DUMPFILE parameter encompass the entire set. The files are not required to have the same names, locations, or order that they had at export time.

How Can You Know Where Data Pump Processes Are Running?

Instance information is now provided in some new ways. Worker status information returned by the DBMS_DATAPUMP.GET_STATUS() API includes the instance where that worker process is running. Additionally, Data Pump views (for example, DBA_DATAPUMP_SESSIONS) also indicate the instance where each Data Pump session is running.

There are no user options that control precisely where a worker process will run. The first worker process is always created on the instance where the master control process is running. If the master control process decides that one worker process is adequate for the job, or if parallelism is set to one, then no other Oracle RAC instances are used for that job. In cases where Data Pump does use other Oracle RAC instances, the active instances are used in a round-robin fashion.

What is Data Pump Doing Behind the Scenes?

Data Pump has its own process service layer (kupp) that has been enhanced for Oracle Database 11*g* release 2 (11.2) to provide new options for worker process creation. In general, the changes are transparent to other layers of Data Pump except that the master control process now calls a worker initialization routine to provide some initial context for worker process creation (such as whether or not cluster resources can be used), and the main loop in the master process now polls for worker error and termination messages with a call into the process layer.

The process layer introduces a new process model for worker processes in Oracle Database release 11.2. Worker processes are created as ksv slaves. As ksv slaves, they can be created on other Oracle RAC instances and use ksv messages for communication with each other or with the master control process. There are other useful services provided by ksv such as the ability to kill all slaves at once.

Prior to Oracle Database release 11.2, Data Pump used ksb to create a background process and SGA to convey the initialization context for the worker process. Now, as of Oracle Database release 11.2, Data Pump allows the creation of worker processes as ksv slaves on any specified Oracle RAC instance and ksv messages to establish the initial worker process context. There is additional handshaking between the master control process and worker processes during startup to make sure the process is started successfully. The handshaking uses new events as well as ksv messages. When a worker terminates, instead of loading termination context in SGA, a ksv message is sent to the master control process with the information. Ksv can also determine when one end of the communication link has disappeared, and this provides a heartbeat detection mechanism for the master and worker processes, to know when the other is still alive and well. The worker process code runs inside this ksv bubble and is completely transparent to the worker.

It should be noted that the primary method used by Data Pump processes to communicate with each other is still Advanced Queuing (AQ); this has not changed with the new worker process model. The ksv messages are only used to establish initialization context for the process and to convey termination status information. During worker process execution, communication between the master control process and worker processes still uses AQ. When AQ is used for communication between two processes on the same instance, buffered messaging is used. If the processes are on different instances, then persistent queue messages are used, which involve slightly more overhead.

Creating a worker process on another instance takes slightly longer than creating one locally, and there is slightly more overhead with communication between master and worker processes when multiple instances are involved. However, for large Data Pump jobs with lots of data well suited for parallelism, this overhead can be insignificant compared to the performance gains from using the additional resources available in an Oracle RAC environment.

A Data Pump process model that is distributed across an Oracle RAC environment introduces additional complexity and more difficulty when troubleshooting problems because situations become possible that could never occur in single-instance configurations. The new Oracle RAC support provided by Data Pump attempts to handle all such cases and the existing mechanisms should be adequate to troubleshoot most problems. Customer feedback will be very helpful since it is not possible to anticipate or test every possible Oracle RAC configuration.

Conclusion

It is important to optimize the parallel capabilities of Oracle Data Pump so that export and import jobs run as efficiently as possible. If the PARALLEL parameter and the wildcard dump file template are used, and if the job has a large amount of data (not metadata), then performance over original Export and original Import will be much improved. When the concept of parallelism is expanded to include the distribution of work across Oracle RAC instances, the performance gains can be significant and the Data Pump jobs themselves are less susceptible to the single points of failure that are possible in single-instance configurations.

Oracle Data Pump On Real Application Clusters
September 2009

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