

5

Oracle Data Pump Performance

Learn how Oracle Data Pump Export and Import is better than that of original Export and Import, and how to enhance performance of export and import operations.

The Oracle Data Pump Export and Import utilities are designed especially for very large databases. If you have large quantities of data versus metadata, then you should experience increased data performance compared to the original Export and Import utilities. (Performance of metadata extraction and database object creation in Data Pump Export and Import remains essentially equivalent to that of the original Export and Import utilities.)

- [Data Performance Improvements for Oracle Data Pump Export and Import](#)
Oracle Data Pump Export (`expdp`) and Import (`impdp`) contain many features that improve performance compared to legacy Export (`exp`) and Import (`imp`).
- [Tuning Performance](#)
Oracle Data Pump is designed to fully use all available resources to maximize throughput, and minimize elapsed job time.
- [Initialization Parameters That Affect Oracle Data Pump Performance](#)
Learn what you can do to obtain the best performance from your Oracle Data Pump exports and imports.

5.1 Data Performance Improvements for Oracle Data Pump Export and Import

Oracle Data Pump Export (`expdp`) and Import (`impdp`) contain many features that improve performance compared to legacy Export (`exp`) and Import (`imp`).

The improved performance of the Data Pump Export and Import utilities is attributable to several factors, including the following:

- Multiple worker processes can perform intertable and interpartition parallelism to load and unload tables in multiple, parallel, direct-path streams.
- For very large tables and partitions, single worker processes can choose intrapartition parallelism through multiple parallel queries and parallel DML I/O server processes when the external tables method is used to access data.
- Oracle Data Pump uses parallelism to build indexes and load package bodies.
- Because Dump files are read and written directly by the server, they do not require any data movement to the client.
- The dump file storage format is the internal stream format of the direct path API. This format is very similar to the format stored in Oracle Database data files inside of tablespaces. Therefore, no client-side conversion to `INSERT` statement bind variables is performed.
- The supported data access methods, direct path and external tables, are faster than conventional SQL. The direct path API provides the fastest single-stream performance. The external tables feature makes efficient use of the parallel queries and parallel DML capabilities of Oracle Database.

- Metadata and data extraction can be overlapped during export.

5.2 Tuning Performance

Oracle Data Pump is designed to fully use all available resources to maximize throughput, and minimize elapsed job time.

To maximize available resources, a system must be well-balanced across CPU, memory, and I/O. In addition, standard performance tuning principles apply. For example, for maximum performance, ensure that the files that are members of a dump file set reside on separate disks, because the dump files are written and read in parallel. Also, the disks should not be the same ones on which the source or target tablespaces reside.

Any performance tuning activity involves making trade-offs between performance and resource consumption.

- [How To Manage Oracle Data Pump Resource Consumption](#)
With the `PARALLEL` parameter, you can dynamically increase and decrease Oracle Data Pump Export and Import resource consumption for each job.
- [Effect of Compression and Encryption on Performance](#)
You can improve performance by using Oracle Data Pump parameters related to compression and encryption, particularly in the case of jobs performed in network mode.
- [Memory Considerations When Exporting and Importing Statistics](#)
When you use Oracle Data Pump Export dump files created with a release prior to 12.1, and that contain large amounts of statistics data, this can cause large memory demands during an import operation.

5.2.1 How To Manage Oracle Data Pump Resource Consumption

With the `PARALLEL` parameter, you can dynamically increase and decrease Oracle Data Pump Export and Import resource consumption for each job.

You can manage resource allocations for Oracle Data Pump by using the `PARALLEL` parameter to specify a degree of parallelism for the Oracle Data Pump job. For maximum throughput, do not set `PARALLEL` to much more than twice the number of CPUs (two workers for each CPU).

As you increase the degree of parallelism, CPU usage, memory consumption, and I/O bandwidth usage also increase. You must ensure that adequate amounts of these resources are available. If necessary, to obtain the needed I/O bandwidth, you can distribute files across different disk devices or channels.

To maximize parallelism, you must supply at least one file for each degree of parallelism. The simplest way of doing this is to use substitution variables in your file names (for example, `file%u.dmp`). However, if your disk setup could create contention issues (for example, with simple, non-striped disks), you can prefer not to put all dump files on one device. In this case, Oracle recommends that you specify multiple file names using substitution variables, with each file in a separate directory resolving to a separate disk. Even with fast CPUs and fast disks, the path between the CPU and the disk can be the constraining factor in the degree of parallelism that your system can sustain.

The Oracle Data Pump `PARALLEL` parameter is valid only in Oracle Database Enterprise Edition 11g or later.

5.2.2 Effect of Compression and Encryption on Performance

You can improve performance by using Oracle Data Pump parameters related to compression and encryption, particularly in the case of jobs performed in network mode.

When you attempt to tune performance, keep in mind your resource availability. Performance can be affected negatively with compression and encryption, because of the additional CPU resources required to perform transformations on the raw data. There are trade-offs on both sides.

5.2.3 Memory Considerations When Exporting and Importing Statistics

When you use Oracle Data Pump Export dump files created with a release prior to 12.1, and that contain large amounts of statistics data, this can cause large memory demands during an import operation.

To avoid running out of memory during the import operation, be sure to allocate enough memory before beginning the import. The exact amount of memory needed depends on how much data you are importing, the platform you are using, and other variables unique to your configuration.

One way to avoid this problem altogether is to set the Data Pump `EXCLUDE=STATISTICS` parameter on either the export or import operation. To regenerate the statistics on the target database, you can use the `DBMS_STATS` PL/SQL package after the import has completed.

Related Topics

- [EXCLUDE](#)
- [EXCLUDE](#)
- *Oracle Database SQL Tuning Guide*

5.3 Initialization Parameters That Affect Oracle Data Pump Performance

Learn what you can do to obtain the best performance from your Oracle Data Pump exports and imports.

- [Performance Guidelines for Oracle Data Pump Parameters](#)
To obtain optimal performance with exports and imports, review and test initialization parameter settings that can improve performance.
- [Setting the Size Of the Buffer Cache In a GoldenGate Replication Environment](#)
Oracle Data Pump uses GoldenGate Replication functionality to communicate between processes.
- [Managing Resource Usage for Multiple User Oracle Data Pump Jobs](#)
To obtain more control over resource use when you have multiple users performing data pump jobs in the same database environment, use the `MAX_DATAPUMP_JOBS_PER_PDB` and `MAX_DATAPUMP_PARALLEL_PER_JOB` initialization parameters.

5.3.1 Performance Guidelines for Oracle Data Pump Parameters

To obtain optimal performance with exports and imports, review and test initialization parameter settings that can improve performance.

The settings for certain Oracle Database initialization parameters can affect the performance of Data Pump Export and Import.

In particular, you can try using the following settings to improve performance, although the effect may not be the same on all platforms.

- `DISK_ASYNC_IO=TRUE`
- `DB_BLOCK_CHECKING=FALSE`
- `DB_BLOCK_CHECKSUM=FALSE`

The following initialization parameters must have values set high enough to allow for maximum parallelism:

- `PROCESSES`
- `SESSIONS`
- `PARALLEL_MAX_SERVERS`

Additionally, the `SHARED_POOL_SIZE` and `UNDO_TABLESPACE` initialization parameters should be generously sized. The exact values depend upon the size of your database.

5.3.2 Setting the Size Of the Buffer Cache In a GoldenGate Replication Environment

Oracle Data Pump uses GoldenGate Replication functionality to communicate between processes.

If the `SGA_TARGET` initialization parameter is set, then the `STREAMS_POOL_SIZE` initialization parameter is automatically set to a reasonable value.

If the `SGA_TARGET` initialization parameter is not set and the `STREAMS_POOL_SIZE` initialization parameter is not defined, then the size of the streams pool automatically defaults to 10% of the size of the shared pool.

When the streams pool is created, the required SGA memory is taken from memory allocated to the buffer cache, reducing the size of the cache to less than what was specified by the `DB_CACHE_SIZE` initialization parameter. This means that if the buffer cache was configured with only the minimal required SGA, then Data Pump operations may not work properly. A minimum size of 10 MB is recommended for `STREAMS_POOL_SIZE` to ensure successful Data Pump operations.

5.3.3 Managing Resource Usage for Multiple User Oracle Data Pump Jobs

To obtain more control over resource use when you have multiple users performing data pump jobs in the same database environment, use the `MAX_DATAPUMP_JOBS_PER_PDB` and `MAX_DATAPUMP_PARALLEL_PER_JOB` initialization parameters.

The initialization parameter `MAX_DATAPUMP_JOBS_PER_PDB` determines the maximum number of concurrent Oracle Data Pump jobs for each pluggable database (PDB). With Oracle Database 19c and later releases, you can set the parameter to `AUTO`. Beginning with Oracle Database

23ai and later releases, the default setting is `AUTO`. This setting means that Oracle Data Pump derives the actual value of `MAX_DATAPUMP_JOBS_PER_PDB` to be 50 percent (50%) of the value of the `SESSIONS` initialization parameter. If you do not set the value to `AUTO`, then the default value is 100. You can set the value from 0 to 250.

Oracle Database Release 19c and later releases contain the initialization parameter `MAX_DATAPUMP_PARALLEL_PER_JOB`. When you have multiple users performing data pump jobs at the same time in a given database environment, you can use this parameter to obtain more control over resource utilization. The parameter `MAX_DATAPUMP_PARALLEL_PER_JOB` specifies the maximum number of parallel processes that are made available for each Oracle Data Pump job. You can specify a specific maximum number of processes, or you can select `AUTO`. Beginning with Oracle Database 23ai and later releases, the default setting is `AUTO`. If you choose to specify a set value, then this maximum number can be from 1 to 1024 (the default is 1024). If you choose to specify `AUTO`, then Oracle Data Pump derives the actual value of the parameter `MAX_DATAPUMP_PARALLEL_PER_JOB` to be 25 percent (25%) of the value of the `SESSIONS` initialization parameter.

Related Topics

- [MAX_DATAPUMP_JOBS_PER_PDB Oracle Database Reference](#)
- [MAX_DATAPUMP_PARALLEL_PER_JOB Oracle Database Reference](#)