

UsingOracle ' sParallelExecutionFeatures PDF转换可能丢失图片
或格式，建议阅读原文

https://www.100test.com/kao_ti2020/246/2021_2022_UsingOracle_c102_246106.htm The full list of Oracle parallel execution features currently includes the following Parallel Query Parallel DML Parallel DDL Parallel Data Loading Parallel Recovery Parallel Replication How Parallel Execution Works Operations That Can Be Parallelized Oracle can parallelize operations that involve processing an entire table or an entire partition. These operations include: SQL queries requiring at least one full table scan or queries involving an index range scan spanning multiple partitions. Operations such as creating or rebuilding an index or rebuilding one or more partitions of an index. Partition operations such as moving or splitting partitions CREATE TABLE AS SELECT operations, if the SELECT involves a full table or partition scan. INSERT INTO . . . SELECT operations, if the SELECT involves a full table or partition scan. Update and delete operations on partitioned tables Parallel Query Parallel query is the most commonly used of Oracles parallel execution features. It was the first parallel execution feature to be developed by Oracle and was introduced in Oracle Release 7.1 as the Oracle Parallel Query Option (PQO). Parallel execution can significantly reduce the elapsed time for large queries, but it doesn't apply to every query. To parallelize a SELECT statement, the following conditions must be met: At least one of the tables is accessed through a full table scan, or an index is accessed through a range scan involving multiple partitions. If the execution involves a full table scan, the statement

must contain a PARALLEL hint specifying the corresponding table, or the corresponding table must have a parallel declaration in its definition. If the execution involves an index range scan spanning multiple partitions, the statement must contain a PARALLEL_INDEX hint specifying the corresponding index, or the corresponding index must have a parallel declaration in its definition. The following two sections explain how the degree of parallelism is chosen for a SELECT statement and discuss restrictions on the use of the parallel query feature.

Setting the Degree of Parallelism

Once Oracle decides to execute a SELECT statement in parallel, the degree of parallelism is determined by following precedence rules: Oracle retrieves the DEGREE and INSTANCES specifications from the definition of all tables and indexes involved in the query and chooses the highest values found for those settings. Oracle checks the statement for a parallel hint. If such a hint is found, the hint overrides the degree of parallelism obtained as a result of the previous step. You can use the PARALLEL and PARALLEL_INDEX hints to specify the degree of parallelism for a SELECT statement. You can use the NOPARALLEL and NOPARALLEL_INDEX hints to ensure that parallel execution is not performed.

Example

```
alter table emp parallel (degree 4);
select degree from user_tables where table_name = EMP;
select count(*) from emp;
alter table emp noparallel;
SELECT /* PARALLEL(emp,4) */ COUNT(*) FROM emp;
```

Parallel DML Data Manipulation Language (DML) operations such as INSERT, UPDATE, and DELETE can be parallelized by Oracle. Parallel execution can speed

up large DML operations and is particularly advantageous in data warehousing environments where it's necessary to maintain large summary or historical tables. In OLTP systems, parallel DML sometimes can be used to improve the performance of long-running batch jobs.

Deciding to Parallelize a DML Statement

When you issue a DML statement such as an INSERT, UPDATE, or DELETE, Oracle applies a set of rules to determine whether that statement can be parallelized. For UPDATE and DELETE statements, the rules are identical. INSERT statements, however, have their own set of rules.

Rules for UPDATE and DELETE statements

Oracle can parallelize UPDATE and DELETE statements on partitioned tables, but only when multiple partitions are involved. You cannot parallelize UPDATE or DELETE operations on a nonpartitioned table or when such operations affect only a single partition.

Rules for INSERT statements

Standard INSERT statements using a VALUES clause cannot be parallelized. Oracle can parallelize only INSERT ... SELECT ... FROM statements.

Examples:

```
alter session enable parallel dml;
insert /* parallel (emp_big,4,1) */ into emp_big
0select *
from emp;
commit;
alter session disable parallel dml;
```

Parallel DDL

Parallel DDL works for both tables and indexes, whether partitioned or nonpartitioned. For nonpartitioned tables and indexes, only the following types of DDL statements can be parallelized: CREATE TABLE...AS SELECT, CREATE INDEX, ALTER INDEX...REBUILD.

If you're working with partitioned tables and indexes, the scope of Oracle's parallel DDL support broadens. The following statements can be parallelized for partitioned tables and indexes: CREATE

TABLE...AS SELECT ALTER TABLE...MOVE
PARTITION ALTER TABLE...SPLIT PARTITION CREATE
INDEX ALTER INDEX...REBUILD PARTITION ALTER
INDEX...SPLIT PARTITION Not all tables allow these operations to
be executed in parallel. Tables with object columns or LOB columns
don't allow parallel DDL. 100Test 下载频道开通，各类考试题目
直接下载。详细请访问 www.100test.com