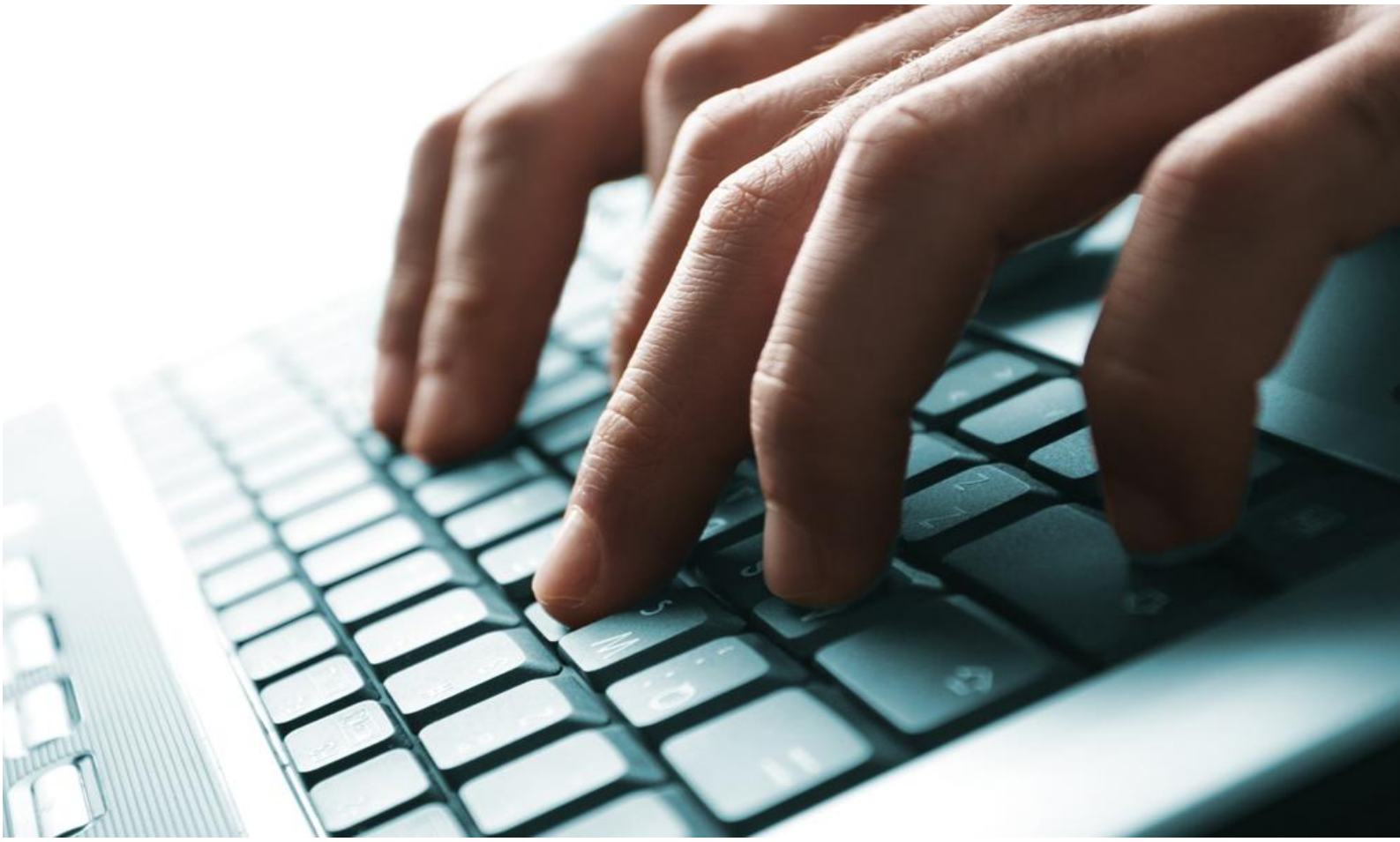
Oracle 系列电子书

Oracle 并行 SQL 指南

并行 SQL 使得 SQL 语句可以被多线程或进程同时处理。并行处理可以把合适的 SQL 语句的性能提升到一定程度，这种提升程度通常是其它任何方法都做不到的……

- *理解并行 SQL 的概念*
- *利用并行 SQL 改善 Oracle 数据库性能*
- *判断什么时候使用并行 SQL*
- *如何配置 Oracle 并行处理*

TechTarget 中国



Oracle 并行 SQL 指南

没有并行技术的时候，一个会话只能利用 CPU 或者磁盘设备其中之一。结果，串行执行 SQL 语句不能利用整个计算机的处理能力。并行执行使得单个会话和 SQL 语句能利用多个 CPU 和磁盘设备的处理能力——Guy Harrison

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

并行 SQL 使得 SQL 语句可以被多线程或进程同时处理。今天，双核和四核处理器的广泛应用，意味着运行 Oracle 数据库的即便是最廉价的现代计算机也会包含一个以上的 CPU。尽管 PC 和笔记本可能只有一个磁盘设备，但数据库服务系统通常会跨多个独立磁盘设备分布保存数据库文件。

没有并行技术的时候，也就是 SQL 语句被顺序处理的时候，一个会话只能利用这些 CPU 或者磁盘设备其中之一。结果，串行执行 SQL 语句不能利用整个计算机的处理能力。并行执行使得单个会话和 SQL 语句能利用多个 CPU 和磁盘设备的处理能力。

并行处理可以把合适的 SQL 语句的性能提升到一定程度，这种提升程度通常是其它任何方法都做不到的。并行处理只有在 Oracle 企业版中才支持。

理解并行 SQL 的概念

在串行的(非并行的)执行环境中，单独的进程或者线程承担需要处理你 SQL 语句的操作，而且在后续活动开始之前，前面的每个动作必须完成。单个 Oracle 进程可能只能利用单个 CPU 的处理能力，而且在任何给定时刻只能从一块磁盘执行读操作。因为大部分现代硬件平台都包括一块以上的 CPU，而且因为 Oracle 数据通常会跨多块磁盘分布，所以串行 SQL 执行不能利用所有可用的处理能力。

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

例如，请看下面的 SQL 语句：

```
SELECT *
FROM sh.customers
ORDER BY cust_first_name, cust_last_name, cust_year_of_birth
```

如果不用并行查询选项来执行，单个进程将会负责提取“CUSTOMERS”表的所有行。也就是这同一个进程，还要负责对这些行排序，以满足“ORDER BY”从句的要求。图 1 描绘了工作流程。

我们可以请求 Oracle 并行执行这句 SQL，需要使用“PARALLEL”关键字声明：

```
SELECT /*+ parallel(c,2) */ *
FROM sh.customers c
ORDER BY cust_first_name, cust_last_name, cust_year_of_birth
```

如果并行处理可用，“CUSTOMERS ”表将被并行的两个进程进行扫描。另外会有两个进程将被分配对结果行进行排序。最后一个进程(也就是一开始发起 SQL 的会话)会整理这些行并返回结果集。那个请求和协调这些并行进程流的进程叫查询协调器(Query coordinator)。图 2 描绘了事件执行顺序。

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

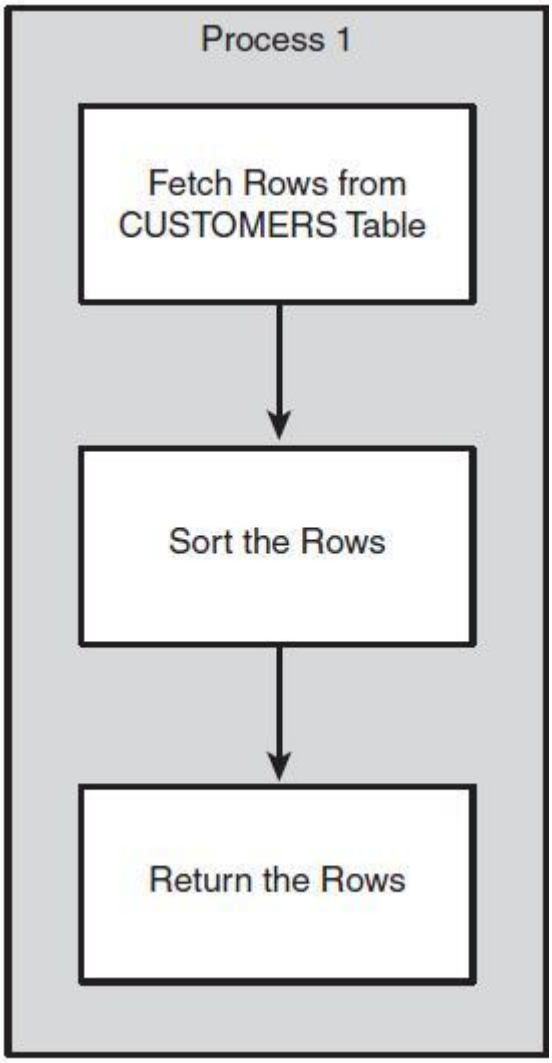


图 1 SQL 语句的串行执行

Oracle 支持并行处理的操作范围很广，包括查询，DDL 和 DML：

- 涉及表或者索引范围扫描的查询。
- 批量插入，更新或者删除操作。
- 表和索引的创建。
- 利用“DBMS_STATS”收集对象统计。

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

- 利用恢复管理器(RMAN)备份和恢复操作。

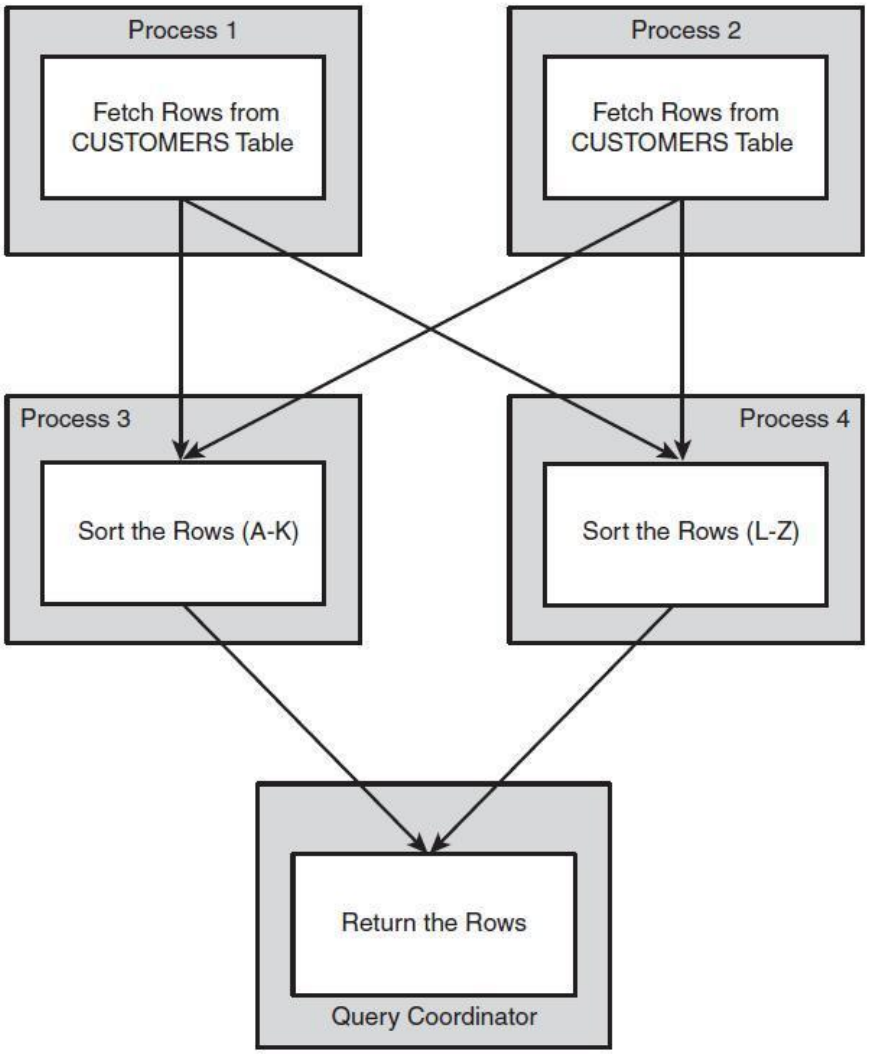


图 2 并行执行

并行处理和并行度

并行度(DOP)定义了将被创建的执行并行流的数量。最简单的情况，它可以

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

理解为分配为支持你 SQL 执行的并行伺服进程的数量。然而，并行进程的数量更多的时候是 DOP 的两倍。这是因为重要的执行计划中的每一步都需要把数据供给后续步骤，因此要分配两个进程组维护并行处理流。

例如，如果语句包含全表扫描，一个“ORDER BY”和一个“GROUP BY”，需要三组并行进程：一个用来扫描，一个用来排序，一个用来分组。因为 Oracle 会重用第一组并行进程(就是执行扫描的那个进程)来执行第三个操作(Group By 操作)，所以实际上只需要两组进程。这种方法的结果是，分配的并行伺服永远不会超过 DOP 的两倍。

图 3 展示了并行伺服如何被为 2 个 DOP 进行分配。

并行伺服池

Oracle 服务器维护一个并行伺服进程池，这些进程对于并行操作可用。

数据库配置参数是“PARALLEL_MIN_ SERVERS”和“PARALLEL_MAX_SERVERS”决定了该池的初始尺寸和最大尺寸。如果当前没有足够的伺服处于激活状态，而且该池也没有达到它的最大值，Oracle 将创建更多伺服进程。在经历过可配置的迟钝期之后，伺服进程将关闭，直到该池再次达到它的最大尺寸。

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

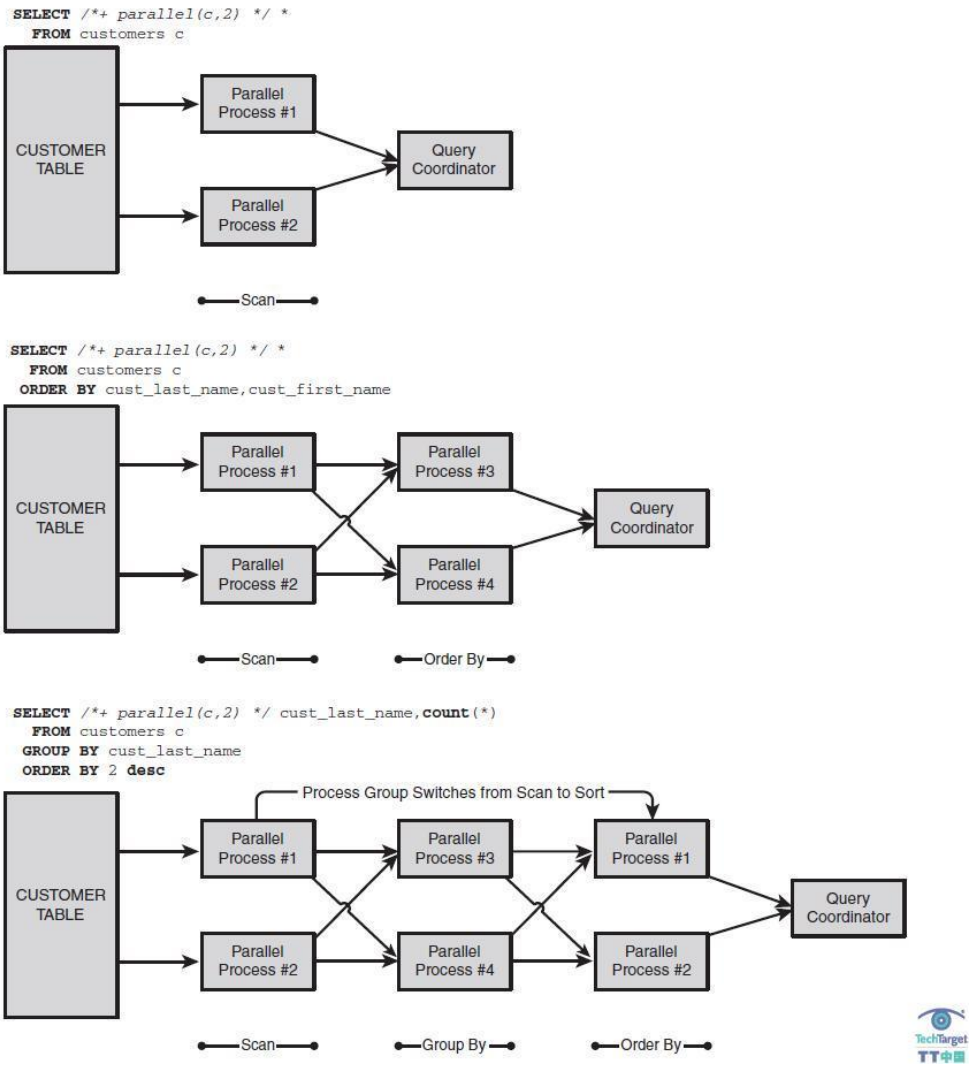


图 3 2 个 DOP 的并行进程分配

如果没有足够的查询进程来满足你的语句发起的 DOP 请求，就会出现如下结果之一：

- 如果有一些并行查询伺服可用，但是数量少于你的 SQL 语句要求的数量，你的语句可能以减少的 DOP 运行。
- 如果没有并行查询伺服可用，你的语句可能会串行运行。

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

- 在特定情况下，你可能会遇上一个错误。这种情况只会在数据库参数“PARALLEL_MIN_PERCENT”被设置为比请求的可用伺服百分比高的情况下。
- 在 Oracle 11g R2 以及之前的版本，你的 SQL 执行可能被延迟，直到有充足的并行服务器可用。

理解 Oracle 中的并行查询 IO

Oracle 缓存区如何通过频繁地缓存在共享内存中访问数据来帮助降低磁盘 IO 瓶颈。Oracle 有一种轮换 IO 机制，叫做“直接路径 IO”，如果它判断到绕过缓存区直接执行 IO 会更快速的话，它就会启用。例如，Oracle 在读写临时段进行排序或者整理中间结果集时就会使用直接 IO。从 Oracle 11g 开始，Oracle 有时也优先利用直接路径 IO 来处理串行表访问，而不是正常的缓存 IO。

在执行并行查询操作时，Oracle 通常会使用直接路径 IO。通过使用直接路径 IO，Oracle 可以避免创建高速缓存竞争，并可以使 IO 更加优化地在伺服之间分配。此外，对于执行全表扫描的并行操作，在高速缓存找到匹配数据的机会相当低，因此高速缓存几乎没有增加什么价值。

在 Oracle 10g 以及更早的版本，并行查询总是使用直接路径 IO，而串行查询将总是使用缓存 IO。在 11g 中，Oracle 可以对并行查询利用缓存 IO(从 11g

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

R2 以后的版本支持)，而且串行查询也可能利用直接路径 IO。然而，并行查询仍然不太可能利用缓存 IO，因此，可能比串行查询需要更高的 IO 成本。当然，更高的 IO 成本将在所有并行进程之间共享，这样整体性能仍可能更胜一筹。

利用并行 SQL 改善 Oracle 数据库性能

你可以预期从部署并行 SQL 获得的性能提升多少，取决于你的托管计算机，Oracle 配置和 SQL 语句配合程度是否良好。如果并行处理的所有条件都满足了，你可以期望可观的性能提升效果与采用的 DOP 是成比例的。

在许多系统上，有效并行的限制是由段的范围决定的，而不是由硬件配置决定的。例如，如果你拥有 32 位 CPU 和 64 位独立磁盘设备，你可能希望达到至少 32 个 DOP 或者可能甚至是 64DOP 的有效并行。然而，如果你查询的表只是跨六块磁盘分布，如果你把 DOP 增加超过 6 左右，你可能会发现性能提升效果会下降。

图 4 描绘了在为执行全表扫描和单表“GROUP BY”的 SQL 语句增加 DOP 时获得的性能提升曲线。

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

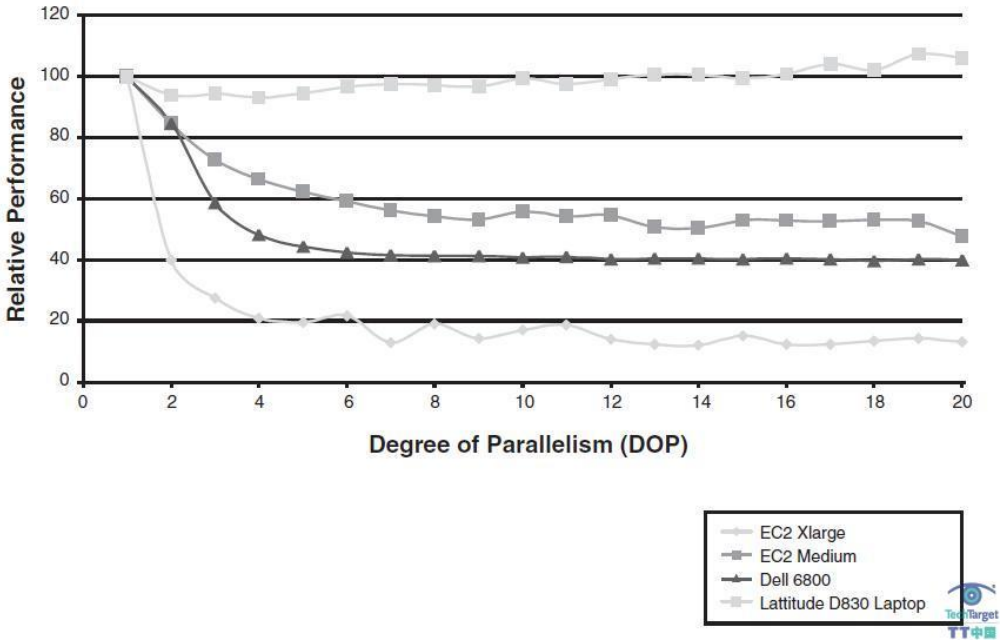


图 4 在各种主机配置上的为各种 DOP 获得提升

主机配置展示如下：

- Amazon 密集型 CPU 扩展大型 EC2 镜像。这是运行在 Amazon AWS 云(装备有 8 块 2.5GHz 的 CPU，在大量条带式 SAN 上做了存储)上的虚拟服务器。
- Amazon 密集型 CPU 中型 EC2 镜像。这与扩展大型镜像类似，但是只有两个 CPU。
- 戴尔 6800 4 个 CPU 的服务器，磁盘存储在采用了 ASM 的大量条带式 SAN 上。
- 戴尔 Latitude D830 笔记本电脑。它是双核的，但是所有的数据文件都

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

在一块磁盘上。

在每种情况下，并行 SQL 是唯一运行着的 SQL。

这些例子的结果显示，对于配置适当的系统，可用的 CPU 数量越多，性能提升越高。然而，尝试在一台不合适的主机(比如：我的笔记本电脑)上使用并行机制充其量是徒劳无功的，最坏的情况可能适得其反。

通过并行处理获得的性能提升，最主要依赖于主机的硬件配置。要从并行处理获得益处，主机应该配备多块 CPU，而且数据应该跨多块磁盘设备分布。

判断什么时候该使用并行处理

为了避免使用过度，你需要知道哪些情况该使用并行处理。本文会简单介绍什么时间该使用 Oracle 数据库的并行 SQL，以及如何监视并行 SQL。

一位开发人员曾经看见我使用并行 SQL 关键字来对临时查询获得快速响应。此后不久，该开发者的每一句 SQL 都写上了并行关键字，因为过度的并行处理，系统性能遭受了数据库服务器负载过度的问题。

教训很明显：如果系统中每一个并发 SQL 都试图使用系统的所有资源，并行会使性能变得更糟，而不是更好。因此，我们应该只在不影响其它并发数据库请求的情况下使用并行 SQL 提高性能。

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

下面的内容讨论了一些情况，你可以在这些情况下有效地使用并行 SQL。

● 你的服务器计算机有多块 CPU？

如果托管你 Oracle 数据库的计算机有多块 CPU，并行处理通常会更有效。这是因为 Oracle 服务器执行的大部分操作(访问 Oracle 共享内存，执行排序，磁盘访问)需要 CPU 资源。如果托管计算机只有一个 CPU，并行处理可能会竞争该 CPU，因而实际性能可能会降低。

几乎任何一台现代计算机都有一个以上的 CPU;双核(一个处理器插槽中有两个 CPU)配置可能是能找到的运行 Oracle 服务器的最低配置了，包括运行部署数据库的桌面计算机和笔记本电脑。然而，运行在虚拟机中的数据库可能被配置为唯一(虚拟化的)CPU。

● 要访问的数据存储在多个磁盘驱动器上

如果查询必须的数据可以在 Oracle 缓存区找到的话，许多 SQL 语句只需要很少的磁盘访问(甚至不需要访问磁盘)就可以解决。然而，对于大表的全表扫描(经常采用并行处理的一种操作)往往需要大量的物理磁盘读操作。如果要访问的数据在一块磁盘上，并行处理会对该磁盘排队处理，并行处理的优势可能不会得到实现。

如果数据均匀地跨多块采用了某种形条带形式的磁盘设备分布，并行的优势就最大化地发挥了。我们在第 21 章会讨论条带式存储原则。

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

● 长期运行或者资源密集的情况可以采用并行 SQL

并行 SQL 适合长期运行或者资源密集型语句。在激活和协调多个并行查询处理方面和处理这些进程之间信息流的协调方面是存在开销的。对于使用期较短的 SQL 语句，这种开销可能比整个 SQL 的响应时间都要长。

并行处理通常用于以下情况：

- 长期运行的报表。
- 批量更新大表。
- 对大表构建或者重建索引。
- 为了分析处理创建临时表。
- 为了提升性能或者为了清除不需要的表而重建表。

并行处理通常不适合事务处理环境。在这些环境中，多个会话并行处理事务。充分利用可用 CPU 资源的目的已经实现，因为每个并发事务可以利用不同的 CPU。如果允许单个用户独占多个 CPU 的话，实施并行处理可能实际上会降低整体性能。

并行处理适合在低并发的环境中长期运行操作。并行处理不太适合 OLTP 风格的数据库。

● 执行至少一个全表，索引或者分区扫描的 SQL

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

并行处理对于那些包含对表，索引或者分区进行扫描的操作，一般是受限制的。然而，该 SQL 可能包含混合操作，只有其中一些会涉及扫描。例如，利用索引做嵌套循环连接来连接两个表可以完全并行，即使驱动表是被全表扫描访问的。

尽管被索引查找驱动的查询一般不会并行化处理，但是如果基于本地分区索引查询分区表，每个索引扫描可以在针对与索引分区相对应的表分区上执行。

● 你的主机处理能力有盈余

如果你的服务器是满负荷运行的话，你不可能实现并行处理的全部收益。并行处理对于在未充分利用的，多 CPU 的计算机上执行单个任务时效果最好。如果计算机上的所有 CPU 都比较忙，你的并行处理将会遇到 CPU 瓶颈，性能也会下降。

要记住，当某个会话采用并行查询时，它对计算机资源请求了更大的份额。如果许多进程同时尝试并行运行，结果通常会是不能达到请求的并行度，从而会去申请比它们该分配比例更多的资源。

● 调整良好的 SQL

对于调优比较差的 SQL 做并行化调整会降低它的执行时间。而且，你还会放大 SQL 对数据库服务器的影响，并增加它对其它会话的影响。你应该在试图给它授权

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

访问更多数据库服务器资源之前，确保该 SQL 是高效率的。SQL 并行化不是调优 SQL 的替代品。

如何配置 Oracle 并行处理

Oracle 试图自动化系统配置来使并行操作的性能最大化。然而，仍然有许多手工调整的空间，我们可以调整数据库，优化 SQL 并行性能。

● 判断并行度

合适的并行度 DOP 对于良好的并行性能很关键。Oracle 会按如下方式设定 DOP：

1、如果指定或请求了并行执行，但是没有指定 DOP，默认 DOP 会设置为该系统上 CPU 内核数量的两倍。对于 RAC 系统，DOP 值会是整个集群内核数量的两倍。默认值是由配置参数“PARALLEL_ THREADS_PER_CPU”控制的。

2、对于 Oracle 11g R2 之后的版本，如果“ PARALLEL_DEGREE_POLICY”被设置为“AUTO”，Oracle 将根据被执行的运行性质和涉及对象的大小自动调整 DOP 值。

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

3、如果“PARALLEL_ADAPTIVE_MULTI_USER”被设置为“TRUE”，Oracle 将基于该系统的整个负载调整 DOP。当系统承受更重的负载时，DOP 值将会减少。

4、在 Oracle 11g 或者更高版本中，如果“PARALLEL_IO_CAP”被设置为 TRUE，Oracle 将把 DOP 限制为 IO 子系统可以支持的值。这些 IO 子系统限制可以通过“DBMS_RESOURCE_ MANAGER.CALIBRATE_IO”存储过程计算。

5、DOP 可以被指定到表或者索引一级，可以通过在“CREATE TABLE”，“CREATE INDEX”，“ALTER TABLE”或者“ALTER INDEX”中使用“PARALLEL”从句来实现。

6、“PARALLEL”关键字可以被用来指定某个查询中指定表的 DOP。

7、不管任何其它设置，DOP 不能超过“PARALLEL_MAX_SERVERS”可以支持的数量。对于大部分 SQL 语句，服务器数量需要是请求 DOP 的两倍。

正如我们可以在图 13-4 中看到的，超过优化点增减 DOP 会导致进一步性能提升的失败。然而，超过最佳优化限制强行增加 DOP 值会对整个系统性能带来重大的负面影响。尽管被并行化的 SQL 可能不会随着 DOP 的增加而降低，但是该系统的负载会持续增加，而且会引起其它并发运行的 SQL 要遭受响应时间降低的影响。

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

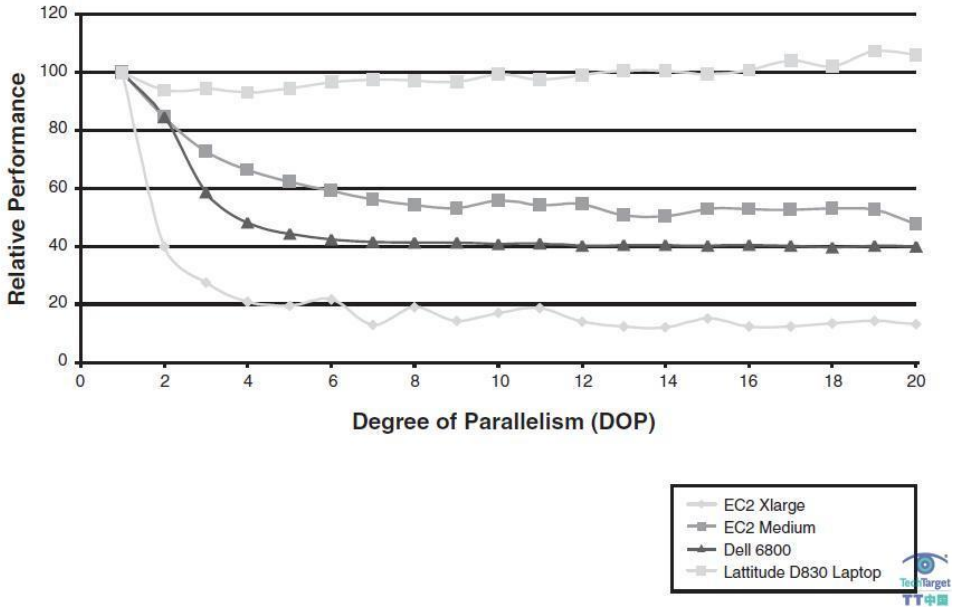
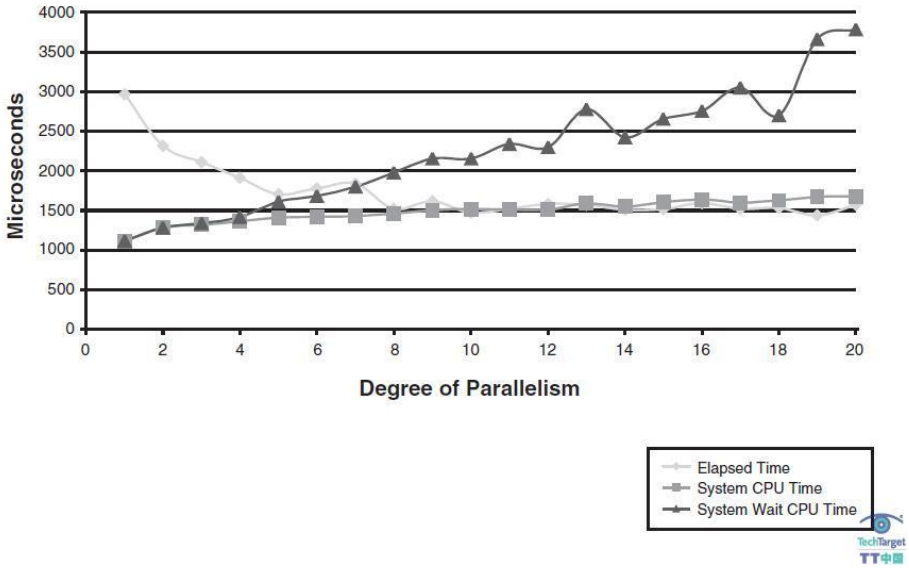


图 13-5 展示了增加 DOP 会如何影响 CPU 利用率。当我们达到最优 DOP 点时(该系统中的值大约是 8)，查询时间的减少变的平缓了。然而，其它会话在等待 CPU 可用上花费的时间会持续增长。其它等待访问 CPU 的会话将需要继续等待，这会导致响应时间变慢。



超过最优水平增加 DOP 可能给主机造成过载，降低其它 SQL 的性能。

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

● “Parallel”关键字

“PARALLEL ”关键字可以调用并行处理。在最简单的格式中，该关键字不需要参数，请看下面的例子：

这种写法是合法的，但是并非总是必须在关键字中指定表名或别名：

SELECT /*+ parallel(s) */ * FROM sh.sales s

该关键字可以请求指定的 DOP 值：

SELECT /*+ parallel(s,8) */ * FROM sh.sales s;

“NOPARALLEL ”关键字可以被用来禁用并行处理：

SELECT /*+ noparallel */ COUNT (*) FROM sales;

在 11g R2 中，“AUTO ”选项允许你请求“AUTO ”设置“PARALLEL_DEGREE_POLICY”用来计算 DOP：

SELECT /*+ parallel(auto) */ COUNT (*) FROM sales;

对于临时查询的执行，你可能想设置明确的 DOP。然而，对于内嵌到应用程序中的 SQL，这可能不是个好主意，因为该 SQL 不太可能适应计算机配置(实例有更多可用 CPU)，负载(更多并发会话)或者配置(并行伺服数量或者默认 DOP)的变更。

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

对于嵌入的 SQL，最好的做法可能是省略明确的 DOP 或者利用“AUTO”关键字(在 Oracle 11g R2 和更高版本中支持)。

● 并行配置参数

确定最佳 DOP 值是一项艰巨的任务，在考虑到并发系统的时候尤其如此。幸运的是，Oracle 为使这项工作自动化投入了巨大的努力。每次 Oracle 发布都增加了智能层面的自动化并行配置。在一般情况下，在尝试手工配置自动处理之前，你应该试试 Oracle 自动分配功能。

不过，进行重要的调整是可能的;下面列出了重要的配置参数，你可以利用它们调整优化并行 SQL：

parallel_adaptive_multi_user	当设置为“TRUE”时，Oracle 将根据该系统的负载调整 DOP。在负载较重的系统中，Oracle 将降低请求的或者默认的 DOP。
parallel_degree_limit	在 Oracle 11g 和更高版本中，（该参数）设置对 DOP 能达到的绝对值限制。CPU 阻止 DOP 超过的值是由“parallel_threads_per_cpu”参数指定的。IO 值设置了最大的 IO 限制，是通过运行存储过程“DBMS_RESOURCE_MANAGER.CALIBRATE_IO”得到的。“AUTO”允许 Oracle 自己选择一个值。对应于具体 DOP 值的整数值也可能被指定。
parallel_degree_policy	在 11g R2 和以后的版本中，这个参数控制 DOP 计算的方式。设置为“MANUAL”值等价于 11.1 以及更早版本中的处理

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

	方式。如果设置为“AUTO”，DOP 将基于 SQL 语句的操作类型和表大小进行计算。“AUTO”还可以使并行查询从缓存中提取数据，而不是利用直接路径 IO，而且如果请求的 DOP 执行不是立即可用的话将会把并行处理排入队列等待。
parallel_execution_message_size	在参与并行处理的进程之间设置用于交互的缓存区大小。
parallel_force_local	对于 Oracle 11g R2 之后的版本，这个参数如果设置为“TRUE”，会阻碍 RAC 集群上的多实例并行机制。
parallel_io_cap_enabled	这个 11g 的参数如果设置为“TRUE”的话，将把 DOP 限制为 Oracle 认为 IO 子系统可以支持的值。要使用这个参数，你需要首先用存储过程“DBMS_RESOURCE_MANAGER.CALIBRATE_IO”算出 IO 限制。
parallel_max_servers	可以启动并行服务器的最大数量。这提供了对于可以执行并发并行操作数量的绝对限制。
parallel_min_percent	如果设置为非零值，这个参数决定了对于查询 DOP 接受的最小值。由于系统负载或者其它也使用并行服务器池的并行处理的原因，如果请求的 DOP 或者设定的 DOP 不能被提供出来，DOP 将只会降低“PARALLEL_MIN_ PERCENT.”的值。例如，如果你的查询请求 DOP 值为 8，而只有 5 个可用（5 / 8 = 62%），那么如果“PARALLEL_ MIN_PERCENT”值小于 62，你的查询将会并行执行。如果“PARALLEL_ MIN_PERCENT”值大于 62，你的语句将出错终止，或者如果“PARALLEL_DEGREE_ POLICY”被设置为“AUTO”，那将会把该任务放到队列后续执行。
parallel_min_servers	并行服务器的最小数量，该数值会在数据库第一次启动时被初始化。
parallel_min_time_threshold	指定某个 SQL 语句需要的运行时间超过多少（单位是秒）就自动并行化处理。如果对 SQL 语句运行时间的估算值超过了这个门槛，Oracle 就自动并行化处理该 SQL。默认值“AUTO”会让 Oracle 来自动计算一个值。
parallel_threads_per_cpu	设置每个 CPU 可以应用的并行线程数量。Oracle 通常会限制 DOP 以便不会超过这个限制。

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

● 理解并行解释计划

解释计划反映了对于并行 SQL 语句的额外步骤，反映了并行执行涉及的额外的并行操作。

例如，请看下面这个简单的 SQL 语句和它的执行计划：

SQL> EXPLAIN PLAN FOR

2 SELECT * FROM customers

3 ORDER BY cust_last_name;

Id	Operation	Name
0	SELECT STATEMENT	
1	SORT ORDER BY	
2	TABLE ACCESS FULL	CUSTOMERS

“CUSTOMERS ” 表被扫描，行被扫描进行排序。当该语句并行化执行的时候，执行计划中会增加额外的操作。

SQL> EXPLAIN PLAN FOR

2 SELECT /*+ parallel */ *

3 FROM customers

4 ORDER BY cust_last_name;

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

```
SQL> SELECT * FROM table (DBMS_XPLAN.display

2 (null,null,'BASIC +PARALLEL'));
```

Id	Operation	Name	TQ	IN-OUT	PQ Distrib
0	SELECT STATEMENT				
1	PX COORDINATOR				
2	PX SEND QC (ORDER)	:TQ10001	Q1,01	P->S	QC (ORDER)
3	SORT ORDER BY		Q1,01	PCWP	
4	PX RECEIVE		Q1,01	PCWP	
5	PX SEND RANGE	:TQ10000	Q1,00	P->P	RANGE
6	PX BLOCK ITERATOR		Q1,00	PCWC	
7	TABLE ACCESS FULL	CUSTOMERS	Q1,00	PCWP	

新计划中包含了各种“PX”步骤，描述了涉及到的并行操作。我们分别看看每个步骤：

PX BLOCK ITERATOR	这个操作通常是并行处理步骤中的第一步。“BLOCK ITERATOR ”把表分成多块，可以由参与并行处理的每一台服务器并行处理。
PX SEND	“PX SEND”操作只是指明数据是从一个并行进程传送到另一个并行进程。
PX RECEIVE	“PX RECEIVE”操作说明一个并行进程接收了另一个并行进程传递的数据。
PX SEND QC	这是给并行查询协调进程的一个发送操作。
PX COORDINATOR	该步骤简单地展示了并行查询协调者从并行流接收数据并返回给 SQL 语句。

图 6 展示了这些步骤如何与 DOP 值为 2 的并行处理相关联。

“PX SEND”和“PX RECEIVE”操作与分配方案(在“DBMS_XPLAN”的“PQ Distrib”列有显示)有关，描述了数据如何被从一台伺服送到另一台伺服。

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

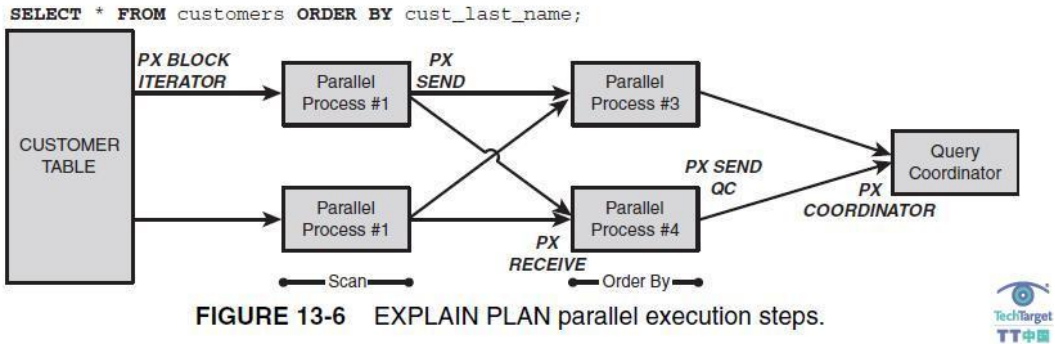


FIGURE 13-6 EXPLAIN PLAN parallel execution steps.

图 6 执行计划并行执行步骤

在排序操作中，通常会看到“RANGE”选项，因为被排序的行是基于排序列进行分配的。例如，在前面的查询中，当按“CUST_FIRST_NAME”做排序时，Oracle 可能把名称“A-K”开头的送到一台伺服器上，名称“L-Z”开头的送到另一台伺服器上。下面是通常会遇到的分配选项：

“DBMS_XPLAN”输出“IN-OUT”列描述了数据如何在并行进程之间和内部进行流转。该列对应于表“PLAN_TABLE”中的列“OTHER_TAG”。这些列可以包含表 13-1 中所示的值之一。

在“PLAN_TABLE”或者“DBMS_XPLAN”输出中，“PARALLEL_FROM_SERIAL”或者“S->P”的出现可能代表着不同的并行执行流出现了一个串行瓶颈。

● 如何跟踪并行执行

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

在并行执行 SQL 的情况下，利用 SQL 跟踪来调优我们的查询变得多少更困难了些。这是因为参与并行执行的每个进程都有它自己的跟踪文件。此外，因为这些进程是在所有 SQL 和会话中共享的，所以跟踪文件中除了我们感兴趣的数据，还包含有其它 SQL 和会话的跟踪数据。

表 13-1 并行数据流标识

IN-OUT VALUE	OTHER_TAG VALUE	DESCRIPTION
P->P	PARALLEL_TO_PARALLEL	该标记表示传递结果给并行进程结果集的并行处理。例如，并行表扫描可能传递结果来并行排序。
P->S	PARALLEL_TO_SERIAL	这通常是并行查询的极限。结果被输入到并行给查询协调器。
P->S PCWC	PARALLEL_COMBINED_ PCWC WITH_PARENT PARALLEL_COMBINED_ WITH_CHILD	该步骤是并行执行的。父任务和子任务也都是被该进程并行执行的。例如，在并行嵌套循环连接中，并行查询进程扫描驱动表，同时还发起对连接表的索引查找。
S->P	PARALLEL_FROM_SERIAL	传递结果给并行进程组的串行操作。使用这个标签意味着在并行语句中存在串行瓶颈，因为它表明并行处理可能在等待串行处理。

然而，通过有点令人费解的过程，我们是可以跟踪并行执行的。下面是操作步骤：

利用“DBMS_SESSION.SET_ IDENTIFIER”在你的会话中设定唯一的客户端标识。

利用“DBMS_MONITOR.CLIENT_ ID_TRACE_ENABLE”对该客户端标识启用跟踪。

运行你的并行 SQL。

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

利用“trcsess ”工具创建新的跟踪文件，其中只包含你客户端标识的条目。

像往常一样分析新的跟踪文件。

下面我们展示了步骤 1，步骤 2 和步骤 3：

```
BEGIN

DBMS_SESSION.set_identifier ('gh pqo test 27');

DBMS_MONITOR.client_id_trace_enable

(client_id => 'gh pqo test 27',

waits => TRUE);

END;

/

SELECT /*+ parallel */ prod_name, SUM (amount_sold)

FROM products JOIN sales

USING (prod_id)

GROUP BY prod_name

ORDER BY 2 DESC;
```


所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

下面我们执行步骤 4 和步骤 5：

$ trcsess clientid='gh pqo test 27' output=pqo_test_27.trc *

$ tkprof pqo_test_27.trc pqo_test_27.prf sort='(prsela,fchela,exeela)'

TKPROF: Release 11.1.0.6.0 - Production on Mon Dec 29 19:40:38 2008

Copyright 1982, 2007, Oracle. All rights reserved.

合并的跟踪文件现在不仅准确地反映了我们调用会话的活动，而且反映了执行该查询涉及的所有并行服务器进程。

要跟踪并行执行，请设置一个客户端标识，然后利用“trcsess ”工具来把该客户端标识对应的跟踪记录提取到单个文件中。

我们还可以利用“_px_trace”实现对并行服务器活动的高级跟踪。例如：

ALTER SESSION SET "_px_trace"="compilation","execution","messaging";

10391 事件还可以被用来转储关于并行服务器分配的信息：

ALTER SESSION SET EVENTS '10391 trace name context forever, level 128';

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

这些事件的产生都非常隐性，有时会有一大堆输出，可能应该只在所有其它技术都不能清楚分析并行执行时才使用。

● 并行 SQL 中的“V$PQ_TQSTAT”视图

即便有了“EXPLAIN PLAN”和 SQL 跟踪输出，要想知道并行查询是怎么执行的也仍然是很困难的。例如，DOP 值到底是多少？每个并行服务器进程做了多少工作？

“V$PQ_TQSTAT”视图包含有在每个并行查询服务器组之间传递数据的信息，包括传输和接收的行数。不幸的是，该视图只在发起并行查询的会话中可见，而且只保留最近执行的查询。这就限制了它在生产环境中的可用性，但是它在调试并行查询时仍然是有价值的。

例如，请考虑下面这个并行查询：

```
SQL> SELECT /*+ parallel */
2 prod_id, SUM (amount_sold)
3 FROM sales
4 GROUP BY prod_id
5 ORDER BY 2 DESC;
```


所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

Id	Operation	Name	TQ	IN-OUT
0	SELECT STATEMENT			
1	PX COORDINATOR			
2	PX SEND QC (ORDER)	:TQ10002	Q1,02	P->S
3	SORT ORDER BY		Q1,02	PCWP
4	PX RECEIVE		Q1,02	PCWP
5	PX SEND RANGE	:TQ10001	Q1,01	P->P
6	HASH GROUP BY		Q1,01	PCWP
7	PX RECEIVE		Q1,01	PCWP
8	PX SEND HASH	:TQ10000	Q1,00	P->P
9	HASH GROUP BY		Q1,00	PCWP
10	PX BLOCK ITERATOR		Q1,00	PCWC
11	TABLE ACCESS FULL	SALES	Q1,00	PCWC

如果我们在查询执行后直接查询“V$PQ_TQSTAT”视图，我们可以看到在每个并行服务器组之间传递的行数。每个唯一的“TQ_ID”对应于服务器组之间的一次交互，这一点可以从执行计划中的“IN-OUT”列的“P->P”或者“P->S”看出来。你可以把“TQ_ID”的值与“EXPLAIN PLAN”输出中的“TQ”列进行关联。

```
SQL> SELECT dfo_number, tq_id, server_Type, MIN (num_rows),

MAX (num_rows),count(*) dop

2 FROM v$pq_tqstat

3 GROUP BY dfo_number, tq_id, server_Type

4 ORDER BY dfo_number, tq_id, server_type DESC;
```


所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

对于复杂的并行 SQL，可能有在“DFO_NUMBER”列中由不同值表示的多个并行通道。

利用“V$PQ_TQSTAT”视图来度量真正的 DOP 值和并行服务器之间传输的数据量。

● 充分利用并行 SQL 中的统计数据

我们可以通过检查“V$PX_SESSION”视图的内容，来实时获得该系统上并行执行出现的情况，该视图展示了哪个并行伺服进程当前正在执行 SQL。把“V$PX_SESSION”视图与“V$SESSION”和“V$SQL”连接可以使我们发现采用并行处理的会话和 SQL，并进一步查看期望 DOP 和实际 DOP：

```
SQL> WITH px_session AS (SELECT qcsid, qcserial#, MAX (degree) degree,
  2 MAX (req_degree) req_degree,
  3 COUNT ( * ) no_of_processes
  4 FROM v$px_session p
  5 GROUP BY qcsid, qcserial#)
  6 SELECT s.sid, s.username, degree, req_degree, no_of_processes,
```


所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

```
7 sql_text

8 FROM v$session s JOIN px_session p

9 ON (s.sid = p.qcsid AND s.serial# = p.qcserial#)

10 JOIN v$sql sql

11 ON (sql.sql_id = s.sql_id

12 AND sql.child_number = s.sql_child_number)

13 /

SID USERNAME DEGREE REQ_DEGREE NO_OF_PROCESSES

---------- -------- ---------- ---------- ---------------

SQL_TEXT

------------------------------------------------------------

144 OPSG 18 18 36

select /*+ parallel(sa,18) */ prod_id,sum(quantity_sold)

, sum(amount_sold) from sales_archive sa group by prod

_id order by 3 desc
```


所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

“V$SYSSTAT”包含有一些与并行查询降级有关的统计数据，可以帮助我们了解并行查询的从请求 DOP 降级的频率：

```
SQL> SELECT name,value, round(value*100/sum(value) over(),2) pct
2 FROM v$sysstat
3 WHERE name LIKE 'Parallel operations%downgraded%';
```

NAME	VALUE	PCT
Parallel operations not downgraded	109	93.97
Parallel operations downgraded to serial	0	0
Parallel operations downgraded 75 to 99 pct	0	0
Parallel operations downgraded 50 to 75 pct	3	2.59
Parallel operations downgraded 25 to 50 pct	2	1.72
Parallel operations downgraded 1 to 25 pct	2	1.72

如何优化并行 SQL 性能

要在你客户的 SQL 环境实现并行执行，你需要知道 SQL 是否准备好并行执行了。在判断 SQL 适合程度之后，你需要确保你有合理的实施计划，以及知道如何优化并行执行性能。

● 优化并行执行的性能

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

现在，我们已经掌握了并行执行理论的基础，现在可以指定一些准则来优化并行执行。下面是从大部分并行执行中提取出来的一些准则：

- 从优化串行执行的 SQL 开始。
- 确保该 SQL 是合适的并行执行 SQL。
- 确保数据库服务器主机适合配置为并行执行。
- 确保执行计划的所有部分都并行化了。
- 确保请求的 DOP 是可实现的。
- 监视请求的 DOP 与实际情况比较。
- 检查数据问题和进程间负载问题。

我们现在来详细看看这些准则。

● 从优化串行执行的 SQL 开始

一个最理想的并行计划与最佳的串行计划可能是有区别的。例如，并行处理通常从表或索引的扫描开始，而最佳串行计划可能是基于索引查找开始。然而，你应该确保你的查询在进行并行优化之前先对串行执行进行优化，原因如下：

串行调试的结构和方法主要针对索引和统计集合，而这些经常对好的并行调试也非常关键。

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

如果请求并行执行的资源不可用，你的查询可能被串行化(这取决于“PARALLEL_DEGREE_ POLICY”和“PARALLEL_MIN_PERCENT”的设置)。在这种情况下，你要确保你并行查询的串行计划足够好。

缺少调优的 SQL 甚至可能变成更差的 SQL，至少考虑到对其他用户的影响时是这样，这使它被允许消耗数据库服务器更多的 CPU 和 IO 资源。

在为并行执行优化 SQL 语句时，要从未串行执行 SQL 优化开始。

● 确保该 SQL 是合适的并行执行 SQL

不是每个 SQL 都能从并行执行中获益的。下面是一些例子，这些情况的 SQL 语句可能不应该被并行化。

- 串行执行时，执行时间很短的 SQL 语句。
- 可能在多个会话中高并发率运行的 SQL 语句。
- 基于索引查找的 SQL 语句。非并行的索引查找或者范围扫描不能被并行化。然而，索引全扫描可以被并行化。在分区索引上的索引查找也可以被并行化。

确保要并行化的 SQL 是适合并行执行的。OLTP 类型的查询通常不适合并行化处理。

● 确保系统适合配置为并行执行

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

不是所有的 SQL 都适合并行执行，也不是所有的数据库服务器主机适合配置并行处理。在当今世界，大部分物理服务器主机都满足如下最小需求：多块 CPU 和跨多个物理驱动器的数据带。然而，一些虚拟主机可能不满足这些最小需求，而桌面计算机通常只有唯一的磁盘设备，因此通常不适合调整为并行执行。

不要尝试在那些不满足最小需求(多块 CPU 和跨多个磁盘驱动器的数据带)的计算机系统上使用并行执行。

● 确保执行计划的所有部分都被并行化了

在复杂的并行 SQL 语句中，很重要的一点是要确保该查询执行的所有重要步骤都实现了并行。如果某复杂查询的其中一个步骤是串行执行的，其他并行步骤可能也不得不等待该串行步骤完成，这样并行机制的优势就完全丧失了。“PLAN_TABLE”表中的“OTHER_TAG”列用“PARALLEL_FROM_SERIAL”标记指定了这样一个步骤，“DBMS_XPLAN”在“IN-OUT”列中记录了“S->P”。

例如：在下面的例子中表“CUSTOMERS ”是并行化的，但是表“SALES ”不是。对两个表的连接和“GROUP BY”包括许多并行操作，但是对“SALES ”表的全表扫描不是并行化的，而且登记机(tell-tale)串到并(S->P)标记展示了“SALES”行被串行提取到后续并行操作中：

SQL> ALTER TABLE customers PARALLEL(DEGREE 4);

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

```
SQL> ALTER TABLE sales NOPARALLEL ;

SQL> EXPLAIN PLAN FOR

SELECT /*+ ordered use_hash(c) */

cust_last_name, SUM (amount_sold)

FROM sales s JOIN customers c

USING (cust_id)

GROUP BY cust_last_name;

SQL> SELECT * FROM table (DBMS_XPLAN.display

(NULL, NULL, 'BASIC +PARALLEL'));
```


所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

Id	Operation	Name	TQ	IN-OUT	PQ Distrib
0	SELECT STATEMENT				
1	PX COORDINATOR				
2	PX SEND QC (RANDOM)	:TQ10002	Q1,02	P->S	QC (RAND)
3	HASH GROUP BY		Q1,02	PCWP	
4	PX RECEIVE		Q1,02	PCWP	
5	PX SEND HASH	:TQ10001	Q1,01	P->P	HASH
6	HASH GROUP BY		Q1,01	PCWP	
7	HASH JOIN		Q1,01	PCWP	
8	BUFFER SORT		Q1,01	PCWC	
9	PX RECEIVE		Q1,01	PCWP	
10	PX SEND BROADCAST	:TQ10000		S->P	BROADCAST
11	VIEW	VW_GBC_5			
12	HASH GROUP BY				
13	TABLE ACCESS FULL	SALES			
14	PX BLOCK ITERATOR		Q1,01	PCWC	
15	TABLE ACCESS FULL	CUSTOMERS	Q1,01	PCWP	

像前面这种情况，部分并行化执行计划可能会导致两方面效果都很差：消耗的时间并没有改善，因为串行操作形成了整个执行的瓶颈。然而，该 SQL 还捆绑了并行服务器进程，而且可能影响其他并发执行 SQL 的性能。

如果我们为表“SALES”设置一个默认的并行度，该串行瓶颈就消失了。对“SALES ”表的全扫描现在是按并行执行了，而且“串到并 S->P”瓶颈被全并行的“并到并 P->P”操作替代了：

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

Id	Operation	Name	TQ	IN-OUT
0	SELECT STATEMENT			
1	PX COORDINATOR			
2	PX SEND QC (RANDOM)	:TQ10003	Q1,03	P->S
3	HASH GROUP BY		Q1,03	PCWP
4	PX RECEIVE		Q1,03	PCWP
5	PX SEND HASH	:TQ10002	Q1,02	P->P
6	HASH GROUP BY		Q1,02	PCWP
7	HASH JOIN		Q1,02	PCWP
8	PX RECEIVE		Q1,02	PCWP
9	PX SEND BROADCAST	:TQ10001		P->P
10	VIEW	VW_GBC_5	Q1,01	PCWP
11	HASH GROUP BY		Q1,01	PCWP
12	PX RECEIVE		Q1,01	PCWP
13	PX SEND HASH	:TQ10000	Q1,00	P->P
14	HASH GROUP BY		Q1,00	PCWP
15	PX BLOCK ITERATOR		Q1,00	PCWC
16	TABLE ACCESS FULL	SALES	Q1,00	PCWP
17	PX BLOCK ITERATOR		Q1,02	PCWC
18	TABLE ACCESS FULL	CUSTOMERS	Q1,02	PCWP

在优化并行执行计划时，要确保所有相关步骤都在并行执行：“DBMS_XPLAN”中的串到并 S->P 标记或者“PLAN_TABLE”中的“PARALLEL_FROM_SERIAL”通常指示在并行计划的某些方面存在串行瓶颈。

● 确保请求的 DOP 是可实现的

超过调优限度增加 DOP 可能给系统增加额外负载，而不会提升性能。在最坏的情况下，超过调优限度增加 DOP 可以导致查询运行时间减少。因此，设置合适的 DOP 对于数据库整体的健康和并行查询的性能优化都是非常重要的。

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

确保你请求或预期的 DOP 是现实的;过高的 DOP 可以导致数据库服务器负载过度，而不会提升 SQL 的性能。

● 监视实际 DOP

你请求的 DOP 可能被优化，但是并不总是能成功。当多个并行查询竞争有限的并行执行资源时，DOP 可能会减少，或者 SQL 语句可能会以串行模式运行。

我们前面讨论了 Oracle 如何决定实际 DOP;最重要的参数“PARALLEL_MIN_PERCENT”，“PARALLEL_DEGREE_ POLICY”和“PARALLEL_ADAPTIVE_MULTI_USER”控制了 Oracle 改变 DOP 的方式，不管语句运行在降低并行，出错终止，还是在现存资源不足以请求的 DOP 运行该语句时推迟到后续处理。

DOP 的减少可以导致你的并行 SQL 表现出令人失望的性能。你应该监视查询执行来看是否 DOP 的减少确实出现了。我们前面看到了我们可以如何使用“V$PQ_TQSTAT”来度量实际 DOP，以及我们可以如何利用“V$SYSTAT”统计来度量整体并行降级。

如果你发现降级的并行导致了令人失望的性能，你可能想重新审查你的系统资源(内存，IO 带宽)，调度并行 SQL，或者重新检查服务器配置。可能的选择包括：

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

- 重新调度 SQL，以便它们不要并发运行。Oracle 11g R2 可以自动调度 SQL，只要你设置“PARALLEL_ DEGREE_POLICY”参数为“AUTO”。
- 调整并行配置参数，以支持更大的并发并行。你可以通过增加“PARALLEL_THREADS_PER_ CPU”或者“PARALLEL_MAX_SERVERS”的值来做到这一点。这里的风险是并行执行的总量将会比你系统能支持的数量要多，这会导致 SQL 性能退化。
- 增加你数据库服务器的性能。你可以增加 CPU 数量，RAC 集群中实例的数量，你磁盘阵列中磁盘的数量。
- 调整“PARALLEL_MIN_PERCENT ”参数，使 SQL 可以运行在较少的并行状态下，而不是报错。

令人失望的并行性能可能是 Oracle 由于并发负载或者并行执行资源的限制降低了 DOP 导致的结果。

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。

我们的编辑团队

您若有何意见与建议，欢迎与我们的编辑联系。

诚挚感谢以下人员热情参与 TechTarget 中国《Oracle 系列电子书》的内容编辑工作！

诚邀更多的数据库专业人士加入我们的内容建设团队！

Guy Harrison
TechTarget特邀专家，Quest公司Spotlight系列产品的架构师，并领导开发了Quest 公司的云数据库产品Toad，他在数据库设计、开发、管理和优化方面拥有超过20年的工作经验。

所有内容版权均归 TechTarget 中国所有，未经许可不得转载。