

## 第41章 复制管理

本章要点：

什么是复制

多重主机复制

传播的类型

复制的类型

冲突解决

快照

一些有用的工具

Oracle 8和Oracle 8i的新特性

### 41.1 什么是复制

复制是在分布式数据库环境中维护表副本的过程。插入、更新、删除存储在本地并应用于分布式环境。复制对象包括在一台以上机器上存在的表、索引、数据库触发器、包与视图。Oracle支持复制对象的完整拷贝与复制对象的部分拷贝。Oracle使用复制组来轻松复制过程的管理。Oracle也实现了许多复制方法。当一个数据库环境诸如Oracle支持复制结点的可更新能力时，可能在复制对象的数据之间存在冲突。这种多结点更新能力称为高级复制环境。Oracle提供了一些解决复制冲突的方法。

提示 Oracle 8支持分区表与索引。

Oracle复制使用主结点/快照结点方案。主结点含有所有要被复制到其他结点的对象（表、索引、视图等等）。主组含有一组要被复制到多个结点的对象，快照结点直接与一个并且只与一个主结点相关，它可以含有主结点的所有对象，但通常只含有对象的一个子集。

工作进程是复制过程的重要组成部分。每个主结点需要一个工作进程。快照结点需要一至三个工作进程。这些工作进程在用户定义的工作、净化与按时间表链接时是必须的。需要周期性地初始化这些工作。job\_queue\_intervals机制控制这些工作按多长时间间隔开始，按秒记录。job\_queue\_intervals控制工作的时间。

注意 确保在带有五个其他主结点的一个主结点的环境下，开始复制设置之前先设置下面的INIT.ORA参数，每分钟复制一次。

```
global_names= True
job_queue_processes = 7
job_queue_intervals = 60
```

### 41.2 多重主机复制

复制也可以由多个主机执行，下面通过主群组与快照组的概念进行讨论。

## 41.2.1 主群组

主群组是复制对象的一个集合。主结点存储所有的对象与所有的数据。通常有一个主定义结点，或一个含有正被复制对象的定义的主结点，作为其他主结点的控制点。这个主定义

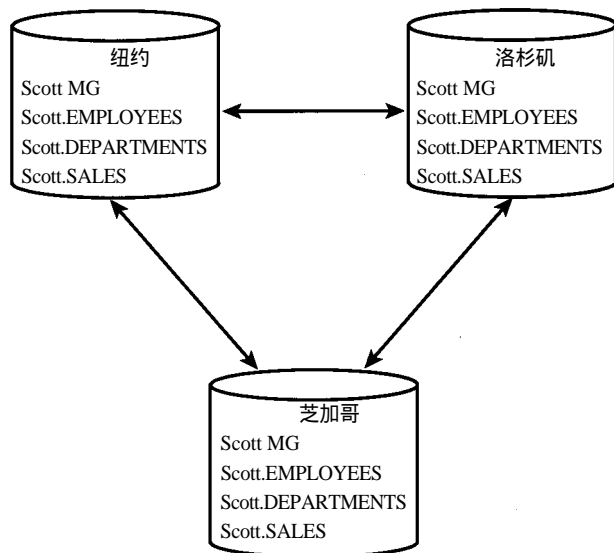


图41-1 多重主结点复制示例

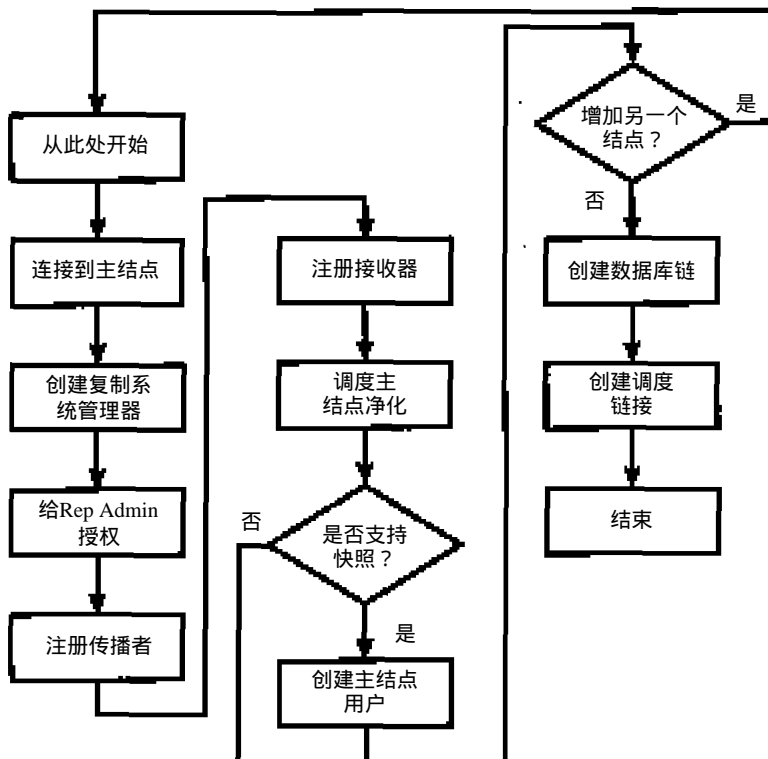


图41-2 设置一个主结点的步骤

结点是DDL改变被应用与复制开始和结束的地方。对数据的改变将从一个主结点推动到另一个主结点，这也称为n种方法复制。

当复制组放在多个机器上时，称他们为多重主群组，多重主群组必须在每个位置含有相同数量的复制对象。这些多重主群组执行真正的对等复制，每个组含有相同的对象与相同的数据。图41-1显示了一个多重主机复制模式。注意复制组与对象在每个结点是相同的。还要注意主组只能与相邻的结点共享数据。这给予了Oracle一种双主机更新路径。如果所有的主结点可以在同一时刻更新所有其他的主结点，潜在的冲突将无法预测，更不用说管理了。

为设置一个主结点进行复制，遵循图41-2中列出的步骤。遵循图41-3中的步骤以设置主群组。

**注意** 只在Oracle 8i企业版中支持多主机复制。

**提示** 在尝试创建一个主结点之前，确定你已作为SYSTEM注册。

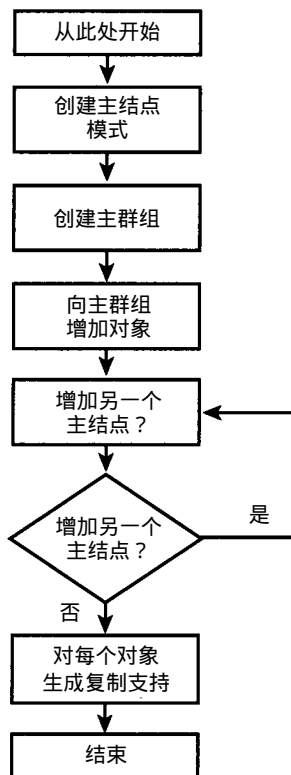


图41-3 设置主群组的步骤

#### 41.2.2 快照组

快照是一个主机对象的子集。快照可以在主群组中含有对象的完全拷贝或只是一个子集。

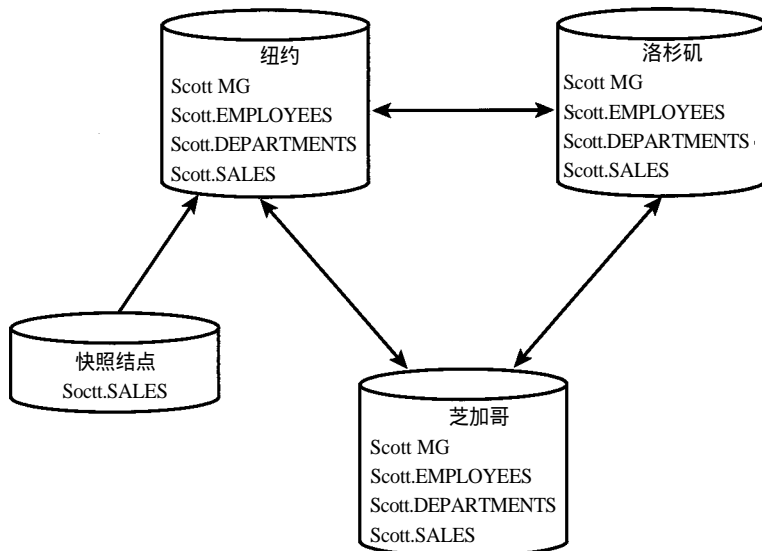


图41-4 多重主结点/快照复制

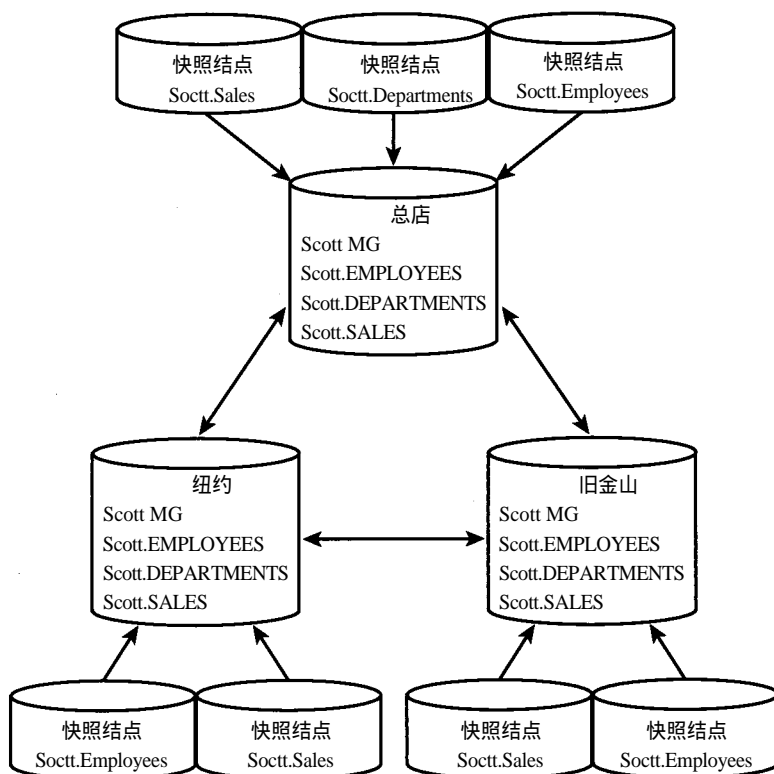


图41-5 复杂主群组/快照组示例

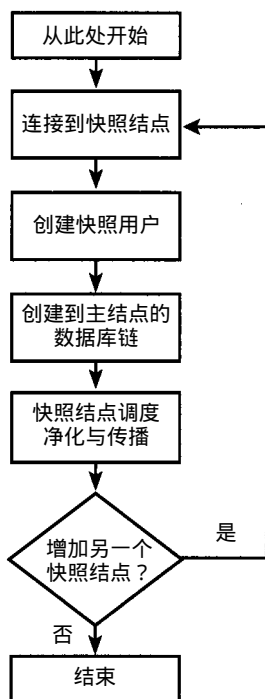


图41-6 设置一个快照结点的步骤

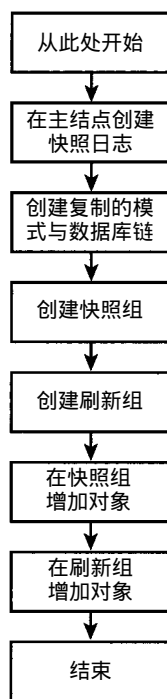


图41-7 设置一个快照组的步骤

快照组与主群组相似，它们都是维护复制的一种便利的方法。快照组只与一个单独的主群组相关，可以被更新。图 41-4显示了一个多重主/快照复制组配置，注意快照组只能与一个单独的主群组共享数据。

现在看一个在图 41-5中含有的主群组与快照组的更复杂的模型。注意总店组也是主定义结点。总店组含有三个分离的快照组，每个部门有一个可更新的快照，负责控制内部的数据。总店组含有所有的对象，旧金山与纽约主群组结点是营业所，它们也含有所有的复制对象，但它们的快照只含有与特定店有关的销售数据与雇员数据。

快照结点也可以很容易地设置，遵循图 41-6中列出的步骤以设置快照结点，在这些步骤完成后，遵循图41-7中的步骤以设置快照组。

### 41.3 传播的类型

对主群组的改变从一个主结点被推动到另一个主结点。改变从主结点被拉到它的快照结点。对快照结点的改变被推回到它们的相应主结点。

#### 41.3.1 异步传播

异步传递，也称为延迟事务，是当改变被延迟或在它们发生的结点排队然后隔一定时间推动到其他结点。这种推动也可以手工启动，这种方法称为保存并转发方法。图 41-8显示了在高级复制模式下在两个主结点之间的 Oracle 触发器与队列的使用。这种相同的配置也可以被用于以异步方式在快照结点与主结点之间传播改变。

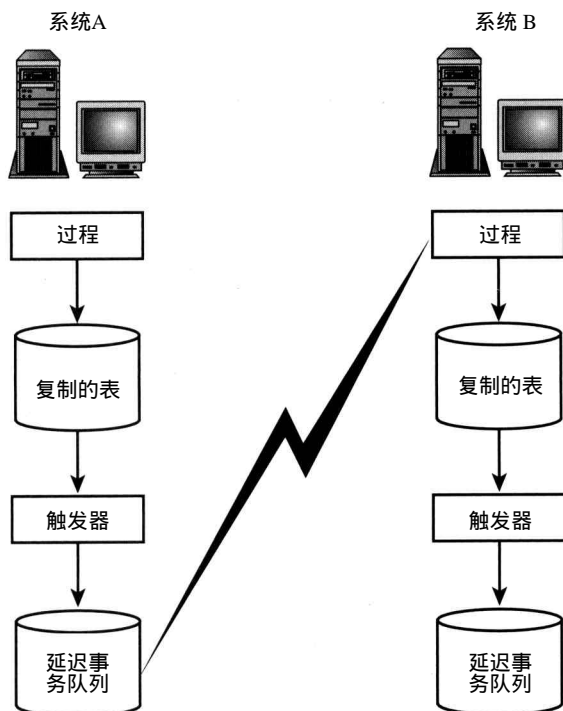


图41-8 在两个主结点之间的Oracle触发器与队列

在高级复制模式中触发器的使用很关键。Oracle使用Oracle的工作队列机制与延迟事务管理这种传播模式。内部的Oracle触发器建立RPC，用于在主结点或快照结点上复制数据的改变。这些RPC是真正为以后的传播进行的存储。工作队列中的工作是一种典型的高级复制模式，包括推动事务的工作、净化整个事务的工作等等。

异步传播也具有高度可用性，即使在快照结点不可获取时，用户仍然可以继续工作。这种传播的方法也执行得更好，因为用户在继续他们的工作之前，不需要等待改变应用到其他结点。然而可能发生数据冲突，正在被复制的数据不立即应用到其他结点，异步传播环境更难于管理。

### 41.3.2 同步传播

同步传播或实时传播，是在一个单独的事务中，将改变应用到本地复制与所有其他的主复制结点与快照复制结点。这种传播的方法更容易配置，但非常依赖网络与系统资源的随时可用性。图41-9显示了同步复制中包括的少数步骤。

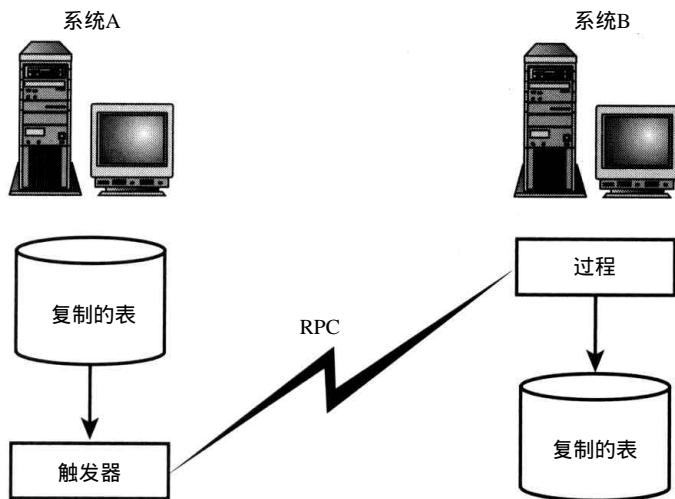


图41-9 同步传播过程

同步复制的优点包括没有数据冲突，使数据可以立即被所有的复制结点使用，并且更易于管理。但通常同步复制环境效率更低，因为用户必须等待改变应用到整个复制模式。

注意 如果在同步传播环境中，任何复制结点是不可获取的，改变不会应用到任何结点。

## 41.4 复制的类型

在Oracle 8与Oracle 8i复制环境中有四种主要的复制类型：行级（或直接接口）、串行、并行与过程化。

### 41.4.1 行级复制

行级复制，或直接接口复制，使应用可以使用标准的 DML 语句更新本地副本。内嵌自动

冲突检测，可以使用内嵌的冲突解决例程。

行级复制更易于实现，因为 Oracle 为真正的传播与冲突检测创建了必要的代码。这种方法也更有灵活性，因为不论使用什么工具，改变都被传播并进行数据的修改。

行级复制的常用对象包括如下：

<表名>\$RT——在每条 DML 语句触发这个触发器，并将改变放入延迟日志队列（只用于 Oracle7）。

<表名>\$TP——当正应用同步传播时，这个触发器与 \$RT 触发器共同使用。

<表名>\$RP——这个包用于在目的地环境中执行 DML 语句，这个包也含有冲突检测（Oracle8）。

<表名>\$RR——这个包与 \$RP 包一起工作，执行必要的冲突解决方案（Oracle8）。

图41-10含有实现行级复制的必要步骤。

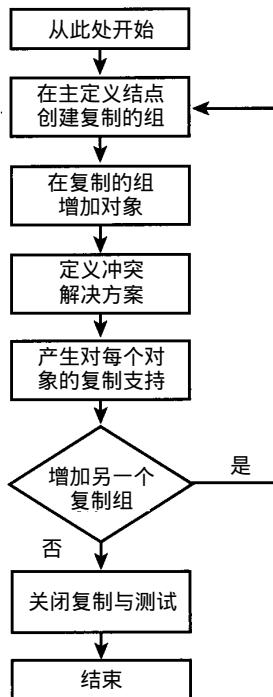


图41-10 行级复制的步骤

#### 41.4.2 串行传播

Oracle 按与初始结点上改变的提交完全一致的次序将改变传播到复制结点。

#### 41.4.3 并行传播

Oracle 使用与 Oracle 用于并行查询、载入、恢复等相同的并行机制同步更新复制结点。

#### 41.4.4 过程化复制

过程化复制也称为封装接口，开发者只书写数据操作的必要过程，Oracle 书写一个封装处理传播，在此没有提供自动冲突检测，没有可用的内嵌冲突例程。这种方法比行级复制更快，因为开发者可以控制被复制的数据。

提示 过程化复制用于批处理最理想，因为应用可以在一个单独的事务中改变大量的数据。数据并没有真正地复制，更新代码被传播到每个结点并在每个结点执行。

图41-11含有实现过程化复制的必要步骤。

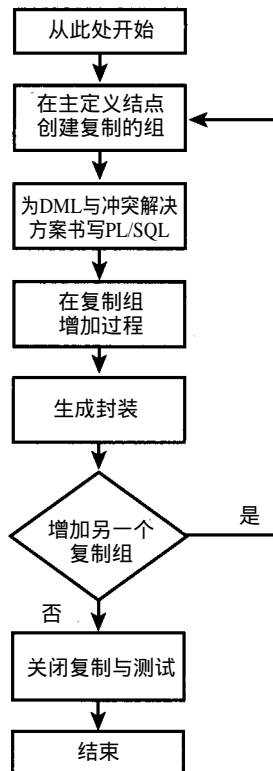


图41-11 过程化复制的步骤

## 41.5 冲突解决

每当在多环境中更新相同的行时，就有可能产生数据冲突。DBA或开发者必须考虑如果不同结点的事务在几乎相同的时间试图更新相同的记录时，应该做什么。

大多数冲突可以通过预先的应用设计被避免。诸如航空订票的应用不能具有冲突。其他类型的应用，如销售导向应用，可以通过数据的拥有者进行本地更新。有一些应用确实需要数据可以在多个结点更新，这种情况时，需要以这样一种方法设计应用：在冲突发生之前就处理数据冲突。

### 41.5.1 冲突的类型

有三种类型的冲突：

更新冲突，发生在当两个或多个结点在几乎相同的时间试图更新相同的记录时。

唯一性冲突，发生在当一个更新产生了一个违反主键或唯一性约束时。

删除冲突，发生在当一个事务删除了另一个事务已更新的一行时。

图41-12显示了多结点更新怎样导致数据完整性冲突。在这个例子中，纽约结点更新了一条部门代码，几乎与此同时，洛杉矶结点正在更新相同的部门名称。

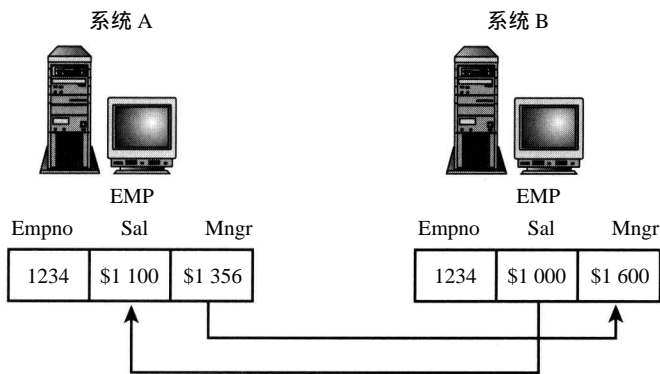


图41-12 多结点更新冲突

### 41.5.2 避免冲突

避免冲突的第一条建议是限制具有相同数据更新能力的结点的数量，这可以由主结点拥有者达到这个目的。在这种情形下，只有一个结点可以更新数据，所有其他的结点只有特定数据的读权限。更粒状的控制形式能够通过仅运行某个结点更新某些数据列来实现。动态拥有，或令牌环传递，与主拥有不同，具有更新从一个结点移动到另一个结点的数据的能力，这对那些流水作业应用特别有用。在这种情形下，应该设计应用按需要自动改变拥有者。

每当一行将被更新时，需要做以下工作：

定位行的当前拥有者。

锁定行，当拥有者进行更新时阻止其他更新。

改变行的拥有者（更新行域）。

执行更新。

图41-13显示了这在应用中是如何实现的。

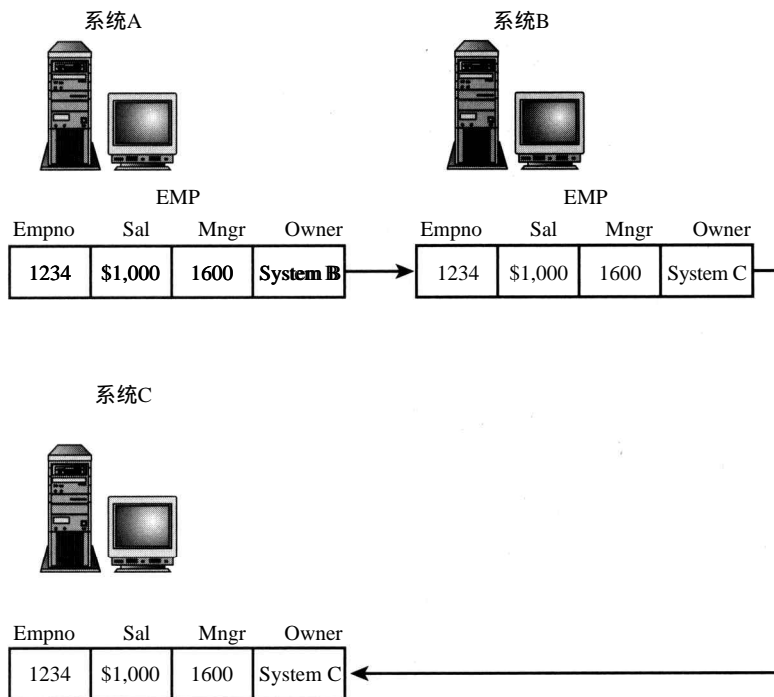


图41-13 动态拥有示例

在每个复制结点创建一系列唯一序列号可以帮助避免整个复制结点的唯一性冲突。另一种观点是将本地序列号作为组合键值的一部分。

可以通过不使用 DELETE DML 语句避免删除冲突。需要删除记录能力的應用可以标志要删除的行，并使用过程化复制将它们定期地清除。

在高级复制环境中的更新冲突很难避免。试着理解所有冲突情形的可能，并配置复制环境以自动处理冲突。最省事的方法，提供一个警告处理就可以查找并修正冲突。

### 41.5.3 鉴别冲突

Oracle需要能够鉴别与匹配不同复制结点上的相关联的行。Oracle使用主键实现这个功能，如果需要更新主键，需要指明一个替代键。

提示 不允许应用修改复制对象的主键值。

### 41.5.4 解决冲突

如果不能避免数据冲突，使用 Oracle 的自动冲突解决方案在冲突发生时修正它们。如果没有解决数据冲突，不一致性会产生更多的不一致性。如果 Oracle 的自动冲突解决策略没有实现，数据冲突被简单地记录到日志，不得不由手工进行处理。这种手工方法，在没有及时执行时，将导致数据不一致性的增长。

可以使用列组以检测并解决更新类型的冲突。列组是一个或多个列的集合。这些列在主

结点被指明，方法是首先创建列组，然后向这些列组分配列。图41-14说明了创建列组的必要步骤。列组使你可以配置不同数据冲突的不同解决方案，要注意解决方法不能相互冲突。

如果不能避免唯一性冲突，可以为主键或唯一键约束指派冲突解决方法以在这些类型的冲突发生时解决它们。为解决这些类型的冲突，需要书写你自己的冲突解决方法，因为 Oracle 没有提供这种冲突解决方法。

列组提供了五种 Oracle 预建的更新类型数据冲突的冲突解决方法：覆盖与放弃值、最小值与最大值、最早与最晚时间戳值、求和与平均值、优先组与结点优先。

覆盖与放弃预建了可以忽略初始结点或目标结点值的解决方法。这种方法意图成为具有多个快照结点的一个单独的主结点。可以设置覆盖与放弃来强迫主结点的改变应用到快照结点。

最小与最大值预建了比较初始结点与目标结点值，并使用两个值中小小的值或者是大的值创建数据收敛的解决方法。

最早与最晚时间戳值预建的解决方法与前面提到的最小与最大值解决方法相同，只是它应用在日期数据域。

求和与平均值预建的解决方法用于只有一个数字域的列组。这种方法的求和部分将目标结点上的原始列值与当前列值的旧值与新值的不同求和，平均值部分平均原始列结点的新值与目标结点上的当前值，这种平均值方法只能用于单主机结点配置中。

优先组与结点优先可以为优先组指派列值的优先级，或者为结点优先选择分配优先级。Oracle 使用较低的优先级解决冲突。图 41-15 说明了五个销售状态列值及相关优先级。当解决一个冲突时，Oracle 使用具有较高优先级的列值更新具有较低优先级列值的结点。结点优先以相同的方法工作，只是给予不同的结点一个优先值。

	优先组	优先级	值	
...	结点优先级	1	...	系统A
..	结点优先级	2	...	系统B
..	Sales	1	..	合格
...	Sales	2	..	测试
..	Sales	3	..	合同

图41-15 优先级组示例

还有三个 Oracle 预建的唯一性冲突解决方法。下面的方法可以分配给主键或唯一键约束：追加结点名为双倍值、追加序列号为双倍值，或废除双倍值。

注意 Oracle 没有任何用于删除类型的冲突或主键列更新的预建的解决方法。

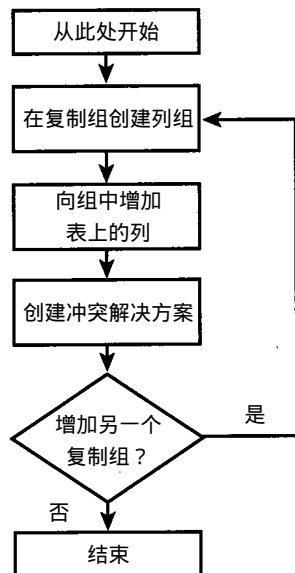


图41-14 创建列组的步骤

除了前面提到的预建的解决方法之外，Oracle还提供了列组的多个冲突解决方法，当一个冲突解决失败后修正这个冲突时，这特别有用。图 41-16说明了多重冲突复制方法是怎样实现的。

可以为解决一个特定的冲突分配多个方法。例如，如果时间信息不能解决一个冲突，如果配置了结点优先级，Oracle仍然能够解决冲突。

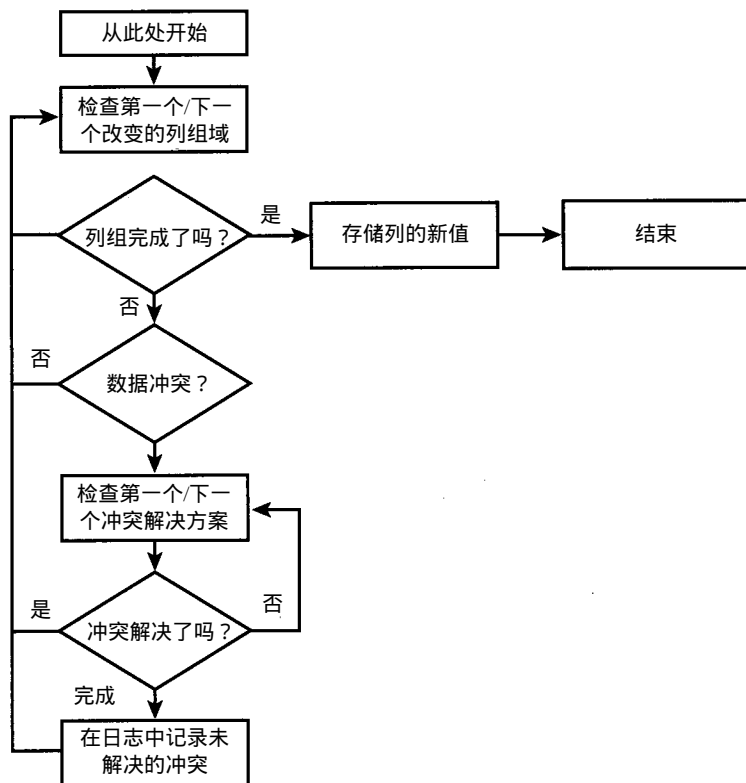


图41-16 多重冲突解决示例

## 41.6 快照

快照是主结点表的一个副本。多个主结点表可以从其他的主结点进行更新，其中快照从一个单独的主结点使用刷新进行更新（参见图 41-17）。将刷新考虑为一个批处理工作。快照可以使用 WHERE 子句含有主结点数据的子集，当不需要在每个远程结点维护整个数据集时，这特别有用。快照的改变被推进一个单独的主结点，然后这些改变被传播到其他主结点，如前面讨论过的那样。主结点的改变被拉进快照结点，这些改变需要一个专有网络连接，使销售人员连接到网络时，从销售人员的便携式电脑刷新数据成为可能。

**提示** 快照可以用于减少网络流量。当使用异步传播方法时，可以通过传播的定时控制网络流量。

快照体系结构是非常直接的，取决于快照是否是可更新的（参见图 41-18），或者快照是否是只读的（参见图 41-19）。如果快照是可更新的，如图 41-18所示，有一个快照日志含有受

更新影响的行的主键与 /或ROWID。刷新可以是快速刷新，其中只应用快照的日志内容，或者是一个完全刷新，其中整个快照表被重新拷贝或重新刷新。快照日志也可以含有快照上主结点子集的筛选信息。

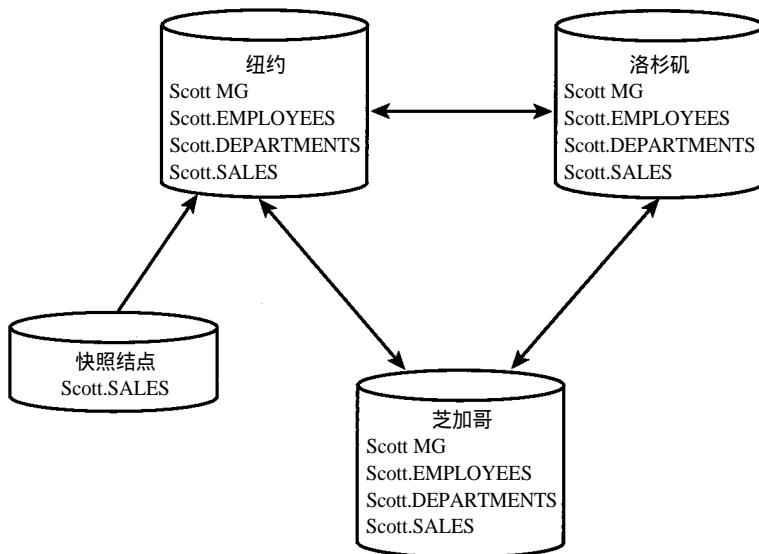


图41-17 一个单独的主结点的快照更新

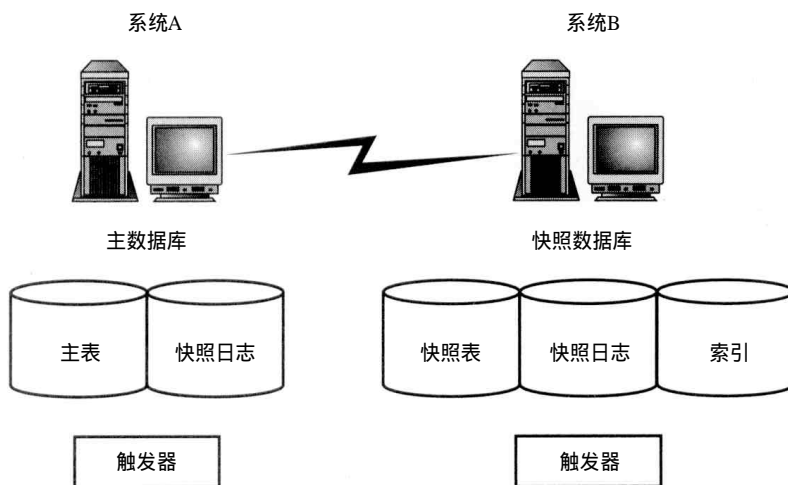


图41-18 可更新快照示例

**注意** 如果快照不依赖于ROWID并且主表被截断或重新构造，不能使用快照日志的快照刷新功能。

有三种可用的快照类型：主键快照、ROWID快照与复合快照。

主键快照是Oracle 8环境中的常用快照类型，改变只基于主键被记录并传播。

ROWID快照的维护与Oracle 7具有兼容性。

复合快照含有SELECT语句，其中有DISTINCT或其他集合函数、CONNECT BY 子句、

一个子集操作（UNION、INTERSECT或MINUS）。复合快照不能使用快照刷新技术，并且如果大量数据被传播，应该避免使用复合快照。

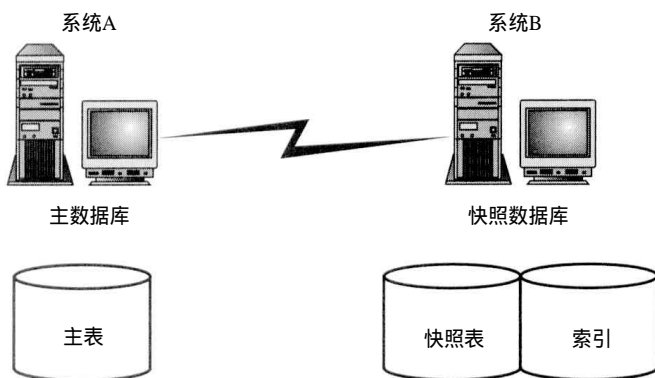


图41-19 只读快照示例

图41-20显示了简单快照与复合快照的不同。简单快照查询的性能是可接受的，这个快照方案可以使用快速刷新。复合快照早已将数据进行了连接，所以查询性能将更好，但代价是必须进行完全刷新。如果刷新是较频繁的，使用复合快照。如果需要刷新的速度与效率，使用简单快照。快照索引总是基于主结点的主键创建的，索引名为 I\_SNAP\$\_snapshot\_name。只为可更新快照创建快照日志。

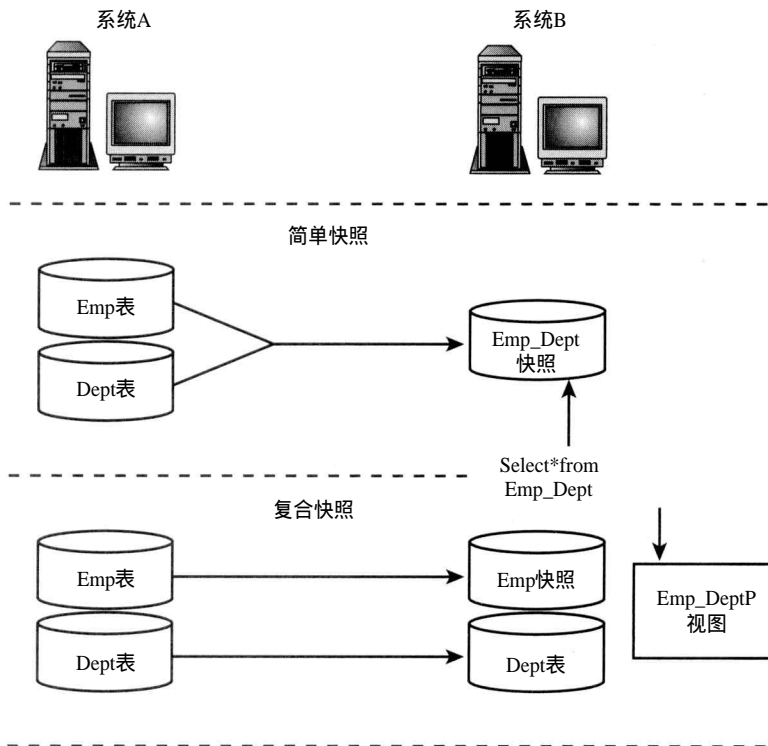


图41-20 简单快照与复合快照示例

只读快照使用许多相同的维护机制，但它们不需要属于一个快照组。一个只读快照是只读的，因为在CREATE SNAPSHOT语法中省略了FOR UPDATE子句。

使用WHERE子句创建具有主结点数据子集的快照减少了刷新时的网络通信量（意为更佳的刷新性能），减少快照数据存储需求，并且可以用于只为用户提供他们需要的数据。后一种方法可以用作一种安全手段，因为只有需要在一个结点存取的敏感数据才真正存储在那个结点上。

Oracle支持主结点列含有以下数据类型的快照：NUMBER、DATE、VARCHAR2、CHAR、NVARCHAR2、NCHAR、RAW、ROWID、LOB、BLOB及CLOB。

注意 在Oracle 8i与Oracle 7版本7.3的混合环境中不支持LOB数据类型。

### 展开模板

展开模板使在分布式环境下快照的安装对 DBA而言既简单又安全。展开模板可以像简单快照一样简单，如果需要，也可以很复杂。展开模板给予 DBA对整个安装的控制，在不同结点重复运行安装的能力，并根据授权用户列表帮助进行安全管理。

图41-21说明了建立一个展开模板的必要步骤。

在建立一个展开模板之后，需要创建一个脚本，它执行展开模板安装的必要步骤。这称为实例化。实例化可以是联机的，也可以是脱机的。联机与脱机实例化的区别在于脱机脚本同时含有创建快照对象的DDL与填充快照的DML。联机实例化只含有DDL，像常用刷新机制一样填充快照。

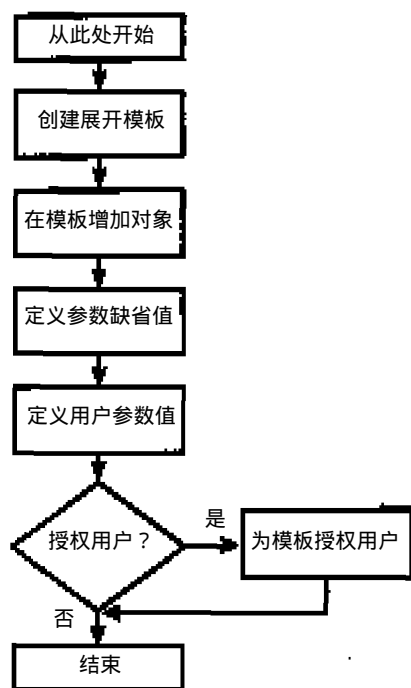


图41-21 建立展开模板的步骤

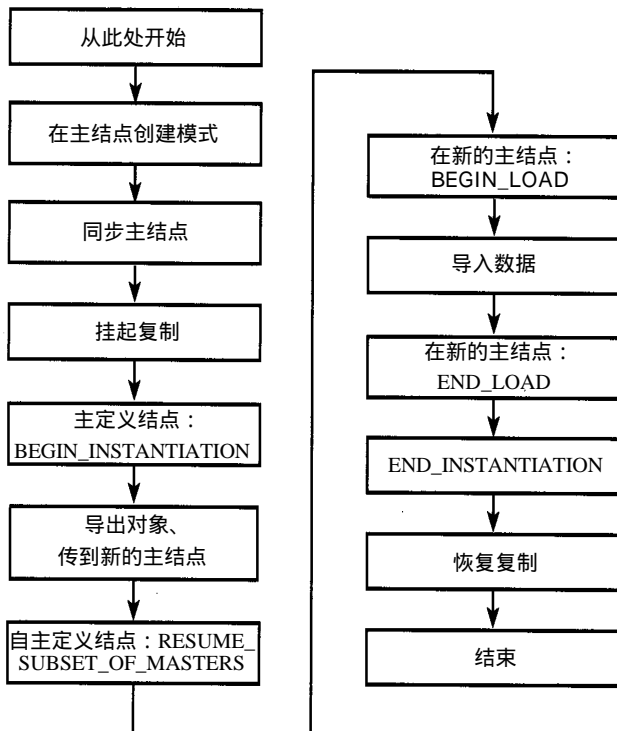


图41-22 主结点脱机实例化示例

图41-22展示了主结点上脱机实例化的必要步骤。图 41-23展示了快照结点上脱机实例化的必要步骤。

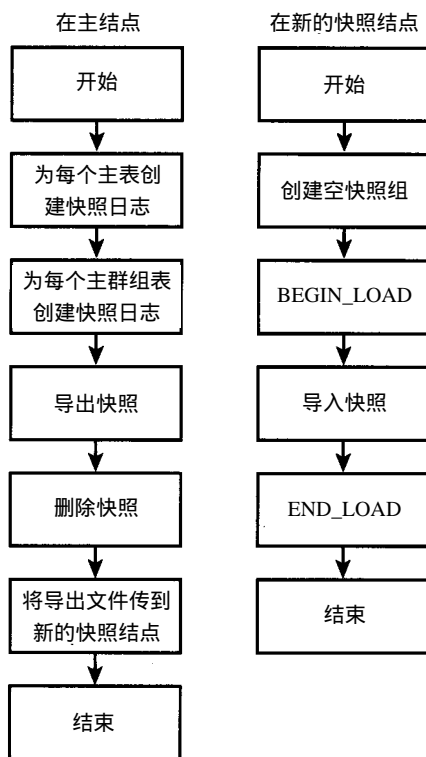


图41-23 快照结点脱机实例化示例

## 41.7 一些有用的工具

下面是用于创建与维护高级复制选项的包清单：

DBMS_REPCAT_AUTH	这个包用于从系统管理员帐户授予并回收 SYS 权限。
DBMS_REPCAT_ADMIN	这个包创建复制系统管理员帐户、授予并回收角色等等。
DBMS_REPCAT	这个包是高级复制模型的主力，执行许多类型的操作。这个包向复制组增加主结点，创建主组与快照结点组，启用并禁止传播，生成包、触发器与过程等等。
DBMS_REPUTIL	这个工具在会话级使能与禁止复制。
DBMS_OFFLINE_OG	这个工具使用户可以实例化结点，从主结点向其他复制结点拷贝数据等等。
DBMS_RECTIFIER_DIFF	这个工具比较两个复制表，如果发现不同，可以使它们同步。
DBMS_DEFER_SYS	这个工具使 DBA 可以计划时间表、执行与删除排队的事务。
DBMS_DEFER	这个工具用于建立延迟调用。
DBMS_DEFER_QUERY	这个工具提供获取传递给延迟调用的参数，并也可以

用于诊断报告。

DBMS\_SNAPSHOT

这个工具使DBA可以维护快照与快照日志。

DBMS\_OFFLINE\_SNAPSHOT

这个包用于实例化快照，特别是大的快照。

DBMS\_REFRESH

这个工具在快照结点初始化快照组的一次刷新。当远程没连接的PC连接到网络并且想要访问他们的数据时，它是非常有用的。

## 41.8 Oracle 8与 Oracle 8i的新功能

Oracle 8引入了许多性能改进与新的复制功能，这将在本章的剩余部分讨论。

### 41.8.1 Oracle 8复制的新功能

Oracle 8.0版本增强了性能并增加了一些功能，下面逐条列举。Oracle也在Oracle 8复制模型中增加了对LOB复制的支持。

功 能	描 述
延迟事务的并行传播	这个功能通过并行化数据的同步与异步传播极大地改进了性能
内在化复制触发器	这个改进减小了开销，需要更少的系统管理，并改进了响应时间性能
减少的数据传播	这个功能减少了通过网络传播的数据总量
基于子查询的快照子集	子查询的常用类型现在可以被快速刷新
LOB支持	Oracle8现在支持BLOB、CLOB与NCLOB的复制
主键快照	Oracle8继续支持ROWID快照，但这个新功能将成为缺省的快照模式。这个功能也允许重新组织主表并维护快速刷新功能及主机快照日志的一致性
分区表与索引	Oracle8支持分区表与索引的复制
增强安全模型	Oracle8改进了同步与异步环境下的一致性，事务极少有可能由于缺少权限而失败

### 41.8.2 Oracle 8i复制的新功能

下面包括了一些Oracle 8i的新复制功能。

特 性	描 述
内部应用包	为继续Oracle8内在化触发器与所需要的函数，Oracle8i内在化用于将改变应用到远程结点的PL/SQL生成的包
刷新组	这项技术导致快速快照刷新，一个单独的刷新组现在可以含有 400个快照
参数化快照展开模板	这项功能将推动非常巨大的应用的展开，例如自动销售、国际化组织等等
列级快照子集	Oracle通常可以支持快照结点上的记录数量的限制，但 Oracle8i现在也支持快照结点上的记录数量的限制

## 41.9 小结

复制无疑是Oracle的强项之一，Oracle继续支持并增强它的复制模型。本章介绍了 Oracle 复制模型，以及所有的术语与不同组件的体系结构。对每个复制选项与潜在的竞争问题的理解将促进对Oracle的高级复制功能的有效管理。