

# 分布式“一卡通”系统数据复制技术应用研究

## Research and application of replicating datum technique in distributed system of the all - purpose with a card

何月顺 (南京航空航天大学 210016)  
(江西抚州 东华理工学院 344000)  
丁秋林 (南京航空航天大学 210016)

**摘要:**在分布式系统中要维持整个系统数据的完整性和保持各节点数据的一致与同步,采用数据复制技术是一种行之有效的措施。本文简要介绍了 ORACLE 的高级复制功能与机制,并通过一个实际项目,讨论了 ORACLE 数据库在分布式系统中如何采用高级复制使各地数据达到统一的配置方法,最后给出了配置脚本供大家参考。

**关键词:**分布式系统 复制 配置脚本 一卡通

### 1 引言

随着网络、分布式数据库及多媒体等技术的发展与成熟,数字化校园、企业信息化的建设已为大家认同,并产生了许多相关的业务与应用系统,其中“校园一卡通”、“企业一卡通”等系统就得到了不断普及和深入,在这些案例中一般采用分布式系统结构,即分布在各地的分部(分校)分别设有独立的数据库服务器,各地数据库服务器采用数据库同步复制的方式更新本地数据库内容,与总部保持一致。从而使得各地终端用户及时、快捷、可靠地获得最新数据及相关信息。

Oracle 数据复制是实现分布式数据环境的一种技术,通过在不同的物理站点拷贝数据来建立分布式数据环境,数据复制实现了所有的站点都有相同数据对象的可用拷贝。本文通过具体的实例,展示了一个异步复制的过程及复制的脚本配置。

### 2 基本概念

#### 2.1 同步和异步的概念

同步分发数据库技术是一种实时远程存取和实时更新数据的技术。这种技术可以保证应用的完整性降低了应用的复杂性,但是如果系统存在网络存取速度很慢这样的问题,相应响应时间就会很慢。

异步分发数据库技术是一种延迟远程存取和延迟传播对数据更新的技术。这种技术具有很高的可用性

和很短的响应时间。相比同步分发数据库技术就显得复杂一些,为了确保应用的完整性需要仔细考虑和设计。

对于实际的商业问题,必须权衡这两种技术的利弊,最终选择最佳的解决方案。

#### 2.2 复制的概念

复制,顾名思义就是将数据库中的数据拷贝到不同物理地点的数据库中以支持分布式应用,它是整个分布式计算解决方案的一个重要组成部分。上面介绍了同步和异步的概念,这里针对复制也存在同步复制和异步复制的问题。

同步复制,复制数据在任何时间在任何复制节点均保持一致。如果复制环境中的任何一个节点的复制数据发生了更新操作,这种变化会立刻反映到其他所有的复制节点。这种技术适用于那些对于实时性要求较高的商业应用中。

异步复制,所有复制节点的数据在一定时间内是不同步的。如果复制环境中的其中一个节点的复制数据发生了更新操作,这种改变将在不同的事务中被传播和应用到其他所有复制节点。这些不同的事务间可以间隔几秒,几分钟,几小时,也可以是几天之后。复制节点之间的数据临时是不同步的,但传播最终将保证所有复制节点间的数据一致,根据实际应用,在配置复制时可以设置同步的时间间隔。

### 3 “一卡通”数据复制技术

#### 3.1 系统结构设计

某高校的“校园一卡通”综合应用系统包括三个信息中心,分布在三个不同的校区,校区之间在空间上跨度较大,三地之间通过专线连接。系统采用 Oracle 作为数据库平台并采用分布式数据库存储架构,后台数据库实时同步复制技术使三个异地服务器数据库内容一致,将远程访问转化为本地访问,从而提高终端用户访问速度。系统简单拓扑结构如图 1。

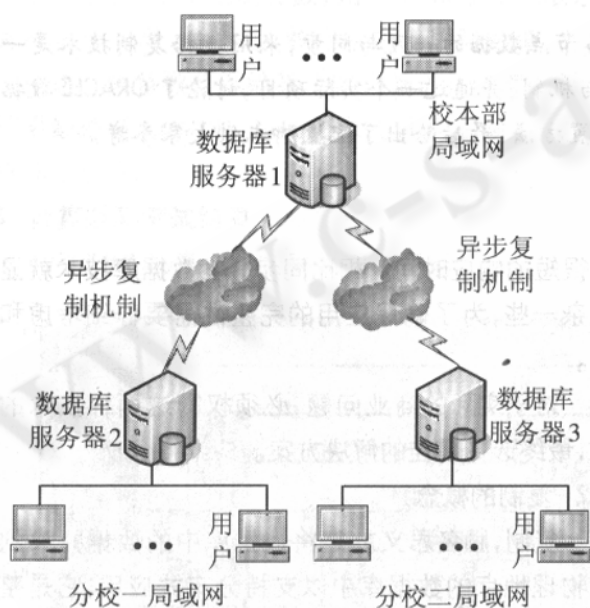


图 1 “一卡通”网络拓扑结构

在图 1 中校本部处于相对中心的位置,所以校本部和分校一之间,本部和分校二之间分别建立了异步复制机制,为了降低网络负担和流通环节,没有建立分校一和分校二之间的复制,各分校直接通过与校本部异步来保持三地数据的一致。一般而言,如果各分校或分支机构处于平等位置,相互之间的数据流量相当,也可以两两之间建立复制机制。

异步复制通过将一个主系统数据复制到另一系统,可以起到对主系统的崩溃提供保护的作用,这也就是所说的 Fail-Over 系统,如果主系统出现问题,业务依旧可以通过复制数据库进行,因此在本系统中采用异步复制机制。

#### 3.2 数据复制应用模型

为确保实际应用数据的一致性,必须在异步复制

应用模型中考虑冲突避免或者冲突检测和消除的方法。首先必须在逻辑上了解系统中采取的冲突避免方法和在某个节点有那些数据以及这些数据中那些是可更新的,而那些又是不可更新的。有两种常用的冲突避免方法:主站点所有权模型和动态所有权模型。

Oracle 复制管理器提供了几个内建的冲突解决方案,诸如“最近时间戳”和“站点优先”等来解决潜在的冲突。对于不同的表可以选择不同的方法,甚至可以对一个表中的不同列选择不同的冲突消除方案,用户对于一些特殊的需求也可以创建自己的冲突解决方案。限于篇幅,本文不再介绍实践冲突解决方案与配置。

#### 3.3 复制方案设计

在校本部的数据库(数据库名:xyykt0)维护所有的表,在其余数据库建立相对于校本部的所有表的快照。为了维护快照的方便,在分校 1 创建一个单独的快照表空间和一个模式(schema),也可以称为用户(USER),系统中的其他用户通过一个私有同义词来存取这些快照。这里私有同义词相对公共同义词要好,这是因为在校本部存在一个同样的系统,它的表是通过公共同义词来存取的。对于快照的刷新,采用 Oracle 系统包 DBMS\_REFRESH 进行,并将该刷新过程的运行定时在每天早上 2:00,这样可以减少网络流量。对于快照的刷新形式,由于对表的维护不是很多而且表的数据量相对较少,所以在此选择了完全刷新,这样就避免了管理快照日志的麻烦。下面以一个节点分校一(数据库名:xyykt1)为例来说明具体的技术实现细节。

(1) 在数据库 xyykt1 上增加 xyykt0 的 services name,可以直接在 tnsnames.ora 文件中增加 xyykt0 的 services name,包括 IP 地址, SID 以及端口号等。services name 命名为 rep.world。

(2) 创建一个用于连接 xyykt0 的数据库连接(dblink)

```
CREATE PUBLIC DATABASE LINK rep.hr_group@
xyykt0 CONNECT TO xyykt0 IDENTIFIED BY admin USING
xyykt0; //创建数据库连接
```

```
select * from cat@rep.hr_group@ xyykt0; //测试数据库连接
```

(3) 创建一个名为 snapshot\_xyykt0 的表空间来存放 xyykt0 快照,并创建一个和该表空间有关的名

Snapshot 的用户。CREATE TABLESPACE "snapshot\_xyykt0" LOGGING DATAFILE 'D:\ORACLE\ORADATA\ORA81\SNAPSHOT\_xyykt0.ora' SIZE 30M DEFAULT STORAGE ( INITIAL 30K NEXT 15K MINEXTENTS 100 MAXEXTENTS UNLIMITED PCTINCREASE 0 ); // 创建表空间

```
CREATE USER snap IDENTIFIED BY snap DEFAULT TABLESPACE "snapshot_xyykt0";
```

(4) 运行下面的脚本文件 snapsql.sql 来生成创建 xyykt0 数据库上 userinfo 用户表的快照脚本:

```
CREATE SNAPSHOT userinfo PCTFREE 10 PCTUSED 40 TABLESPACE "snapcost_xyykt0" STORAGE ( INITIAL 40960 NEXT 57344 PCTINCREASE 0 ) USING INDEX REFRESH WITH PRIMERY KEYFOR UPDATEAS select * from userinfo@ rep. hr_group@ xyykt; // 创建快照
```

```
EXECUTE DBMS_SNAPSHOT. REFRESH ( 'snap. userinfo'
```

运行脚本文件 create\_snapshot.sql 后,就在模式 snap 中创建了所需要的快照。下一步就是考虑该如何刷新快照。对于快照的刷新,可以通过一些桌面 DBA 工具来刷新快照也可以通过系统包 dbms\_snapshot.refresh 来刷新一个快照。

创建或选择实现数据库复制的用户和对象,给用户赋权,数据库对象必须有主关键字,如果数据库对象没有主关键字,可以运行以下 SQL 命令添加:

```
alter table userinfo add ( constraint userinfo_sfzh_pk primary key ( sfzh ) );
```

(5) 创建一个定时刷新过程来定时刷新快照

```
CREATE OR REPLACE PROCEDURE sp_snapshot_refresh ISBEGINDBMS_REFRESH. MAKE ( NAME = >'userinfo', LIST = >'snap. userinfo', NEXT_DATE = > TRUNC ( SYSDATE + 1 ) + 2/24, INTERVAL = >' ( SYSDATE + 1 ) ', IMPLICIT_DESTROY = > FALSE, LAX = > TRUE );
```

```
END sp_snapshot_refresh; // 创建了一个定时任务来每天早晨 2:00 定时刷新快照
```

```
EXECUTE sp_snapshot_refresh; // 执行这个过程
```

```
SELECT JOB, WHAT FROM DBA_JOBS; // 查询刚刚
```

加入的这个任务

```
CREATE SYNONYM system. useinfo FOR snap. useinfo; // 创建快照的私有同义词 GRANT SELECT ON dm_gy_rydm TO system; // 向 system 用户授予 select 权限。
```

(6) 同样的步骤在其它位置建立校本部 ( xyykt0 ) 的表快照和定时刷新任务。

无论任何时候只要出现网络连接问题,刷新就会失败。这些错误信息可以在 alert.log 文件中找到。

## 4 应用效果与结论

本文设计的是 ORACLE 中的一种高级数据库复制技术,除此外,还可通过配置 Oracle 复制管理器来实现,当然通过高级复制更能定制所需的功能,更能优化复制性能。

本系统也经历了升级与多次灾难,通过数据复制有效地保证了系统的正常运行和数据的完整性与一致性。

## 参考文献

- 1 Oracle9i Distributed Database Systems [ M ]. Oracle corporation, 2004.
- 2 何月顺、丁秋林, Oracle 数据库的灾难恢复, 计算机系统应用 [ J ], 2002 ( 9 ) .
- 3 Steve Bobrowski. Oracle 9i for Windows NT Starter Kit [ M ], The McGraw - Hill Companies. 2004.
- 4 何月顺、丁秋林, 调整优化 oracle 9i 数据库的性能 [ J ], 2004 ( 6 ) .
- 5 杨成樞、荣芳, 基于客户/服务器结构的空数据分布式处理研究 [ J ], 计算机工程与应用, 2004; 37 ( 8 ) .
- 6 <http://lutian.blogchina.com/blog/refer.194093.html> [ J ] .
- 7 刘辰, Oracle 数据库系统—管理与应用 [ M ], 2003.