

Oracle高可用概述

杨廷琨

ACUG
member

高可用之概念

- 狭义：出现硬件、网络、系统或人为故障时，数据库不受影响，或数据库可以在很短时间内恢复并提供服务。
- 广义：不局限于数据库的故障处理，涵盖数据库的各个方面，小到数据库日常维护对系统可用性的影响，大到数据库的迁移和升级。甚至扩展到应用程序、硬件、存储和网络的层次。

高可用之技术——RAC

- 多个实例同时提供服务，一个实例崩溃不会影响数据库的可用性。
- 设计合理的RAC环境可以避免单点故障对系统的影响。
- RAC的ROLLING UPDATE支持不停机升级补丁

高可用之技术——DATA GUARD

- 和RAC环境搭配是高可用系统的最佳配置
- 解决了人力不可抗拒的灾难发生后系统的可用性问题
- 快速的解决人为引入的错误
- 逻辑STANDBY可以用来进行快速升级

高可用之技术——STREAM

- 可以提供主环境崩溃后的**FAILOVER**服务
- 利用**STREAM**配合进行迁移，可以极大的缩短迁移停机时间
- 比**STANDBY**数据库提供更多的灵活性

高可用之技术——传输表空间

- 传输表空间提供跨平台迁移的高可用方案
- 传输表空间提供跨版本迁移的高可用方案
- 传输数据库是数据库版本升级和跨平台迁移的高可用方案
- 传输表空间提供了快速、增量的数据加载方案

高可用之技术——FLASHBACK

- FLASHBACK可以快速解决人为错误
- 当发生错误的DML语句后，可以通过FLASHBACK QUERY来获取操作发生之前的数据
- FLASHBACK TABLE可以恢复误删除的表
- FLASHBACK TRANSACTION可以恢复指定的事务
- FLASHBACK DATABASE可以将整个数据库恢复到指定的时刻

高可用之技术——ONLINE REDEFINITION

- 通过在线重定义功能，可以为表重新设置物理存储参数。比如普通表转换为分区表。
- 在线重定义功能可以改变表的逻辑结构
- 有效的降低了表、索引的维护操作对系统可用性的影响

高可用之设计

- 高可用环境是设计出来的
 - RAC、DATA GUARD等高可用环境需要在系统设计时就详细考虑
 - 合理利用DATA GUARD、流等功能产生的数据冗余，提高系统的可用性
 - 用户、表空间等逻辑结构的设计会直接影响系统日后迁移、升级过程。

高可用之设计

- 高可用方案是设计出来的
 - 利用现有环境
 - 利用高可用技术
 - 发挥想象力

高可用之维护——合理利用**ONLINE REDEFINITION**

- 利用**ONLINE REDEFINITION**可以降低表、索引的维护操作对系统可用性的影响
- 适用情况
 - 表的物理、逻辑结构的改变
 - 索引的创建

高可用之维护——充分利用**STANDBY**数据库

- 利用**STANDBY**数据库实现读写分离，避免大的查询操作对在线系统的影响。**STANDBY**数据库可以只读打开，类似报表查询等只读操作可以部署到**STANDBY**数据库
- **11G**中**STANDBY**数据库可以在只读打开的同时应用日志，查询可以随时看到最新的修改
- 利用**STANDBY**进行备份，可以降低备份操作对主环境的影响

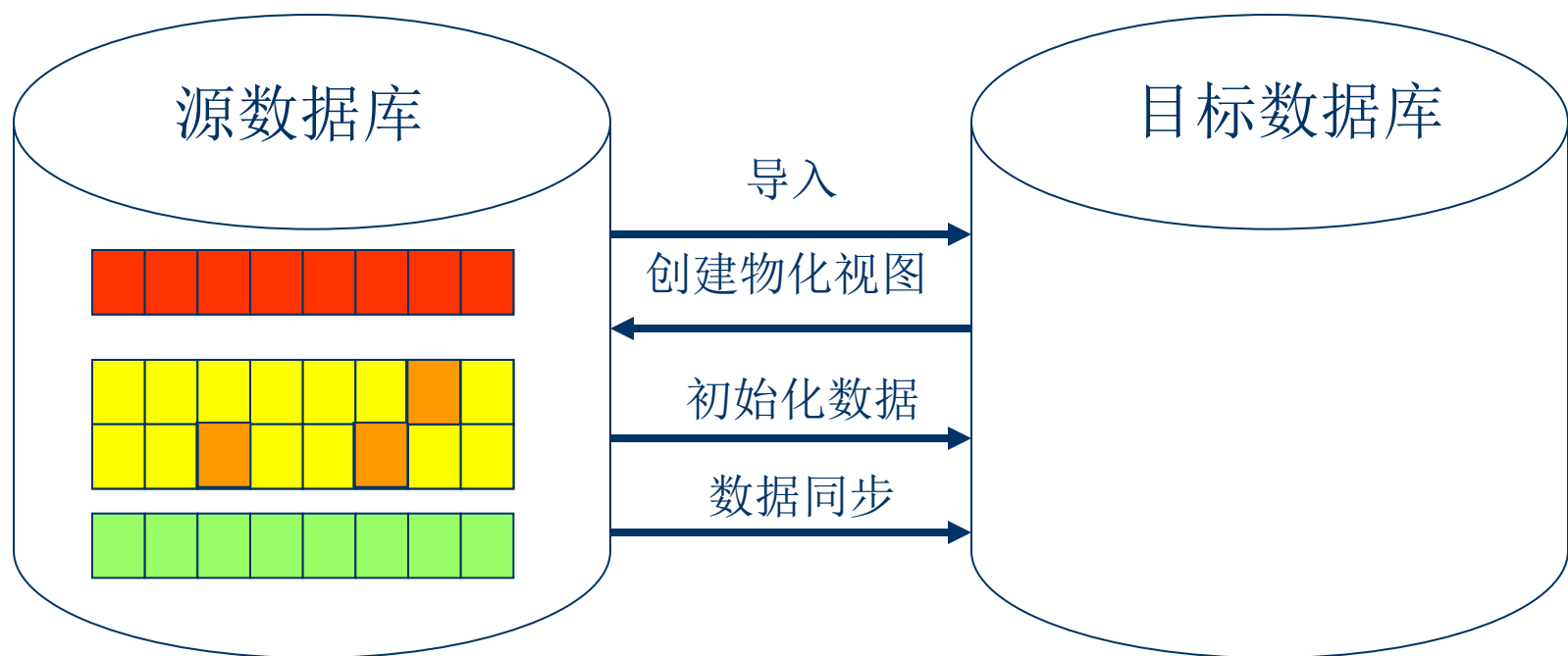
高可用之维护——善于利用新特性

- 闪回（FLASHBACK）
- 不可见索引（INVISIBLE INDEX）
- 添加非空字段
- 细粒度依赖性判断
- 版本（EDITION）

高可用之升级、迁移方案——物化视图迁移方式

- 利用ON PREBUILT TABLE物化视图来实现快速迁移或升级
- 迁移或升级所需停机时间很短
- 可以实现跨版本、跨操作系统、跨平台的迁移
- 逻辑迁移方式，对源和目标库几乎没有限制，用户可以不同、存储参数可以不同、表结构也可以不同
- 要求表必须包含主键

高可用之升级、迁移方案——物化视图迁移方式

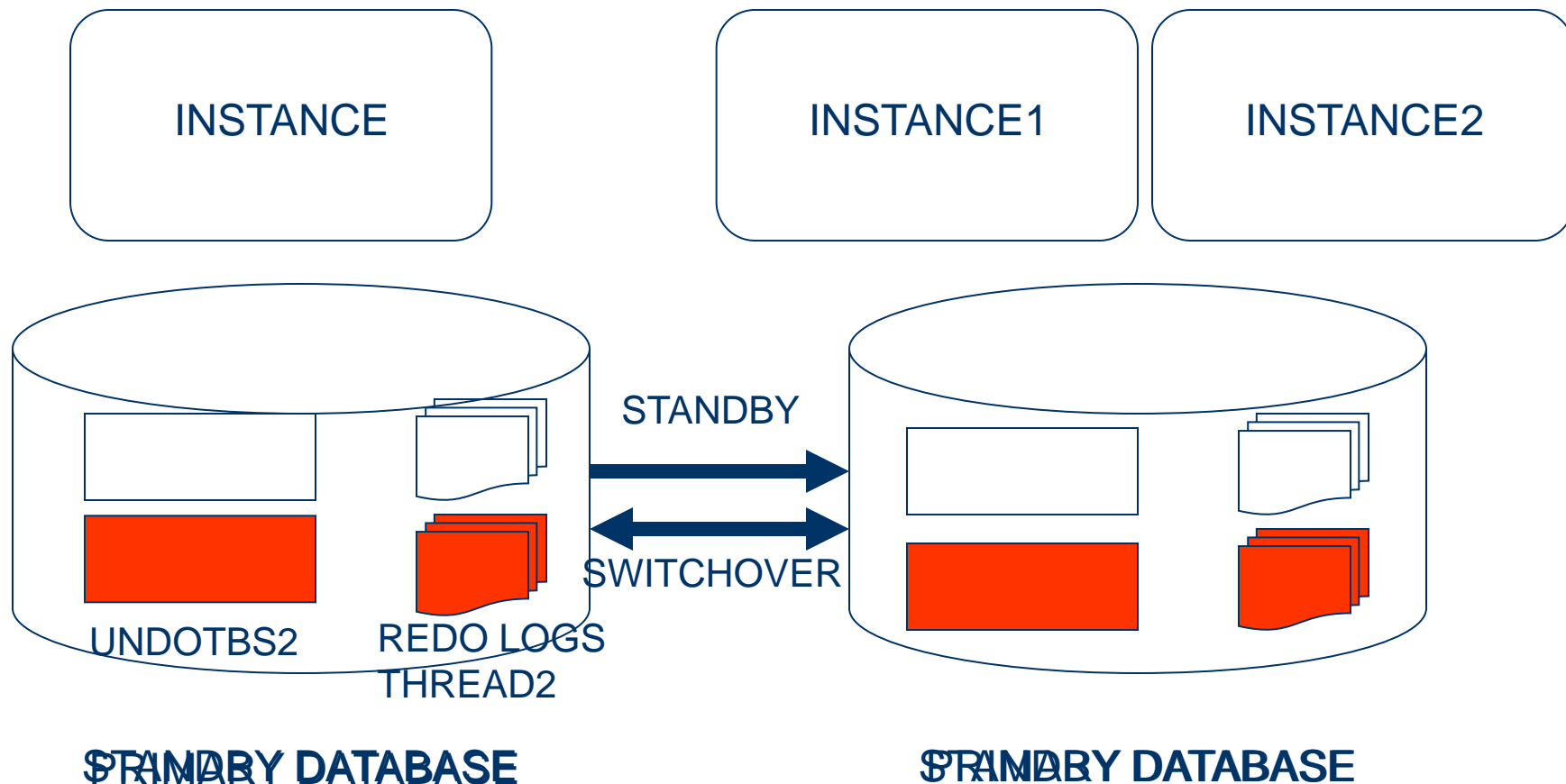


■ 除表、索引外的其他对象 ■ 表和数据 □ 表结构 ■ 物化视图日志

高可用之升级、迁移方案——单实例数据库迁移到RAC环境

- 利用STANDBY数据库的SWITCHOVER切换，实现单实例数据库到RAC环境的迁移
- 减少了迁移所需的时间
- 迁移失败不会对源系统造成损害
- 迁移结束后，源数据库环境自动配置了STANDBY环境
- 一旦系统在RAC环境下工作异常，可以迅速的切换回单实例模式，且不会造成数据丢失。

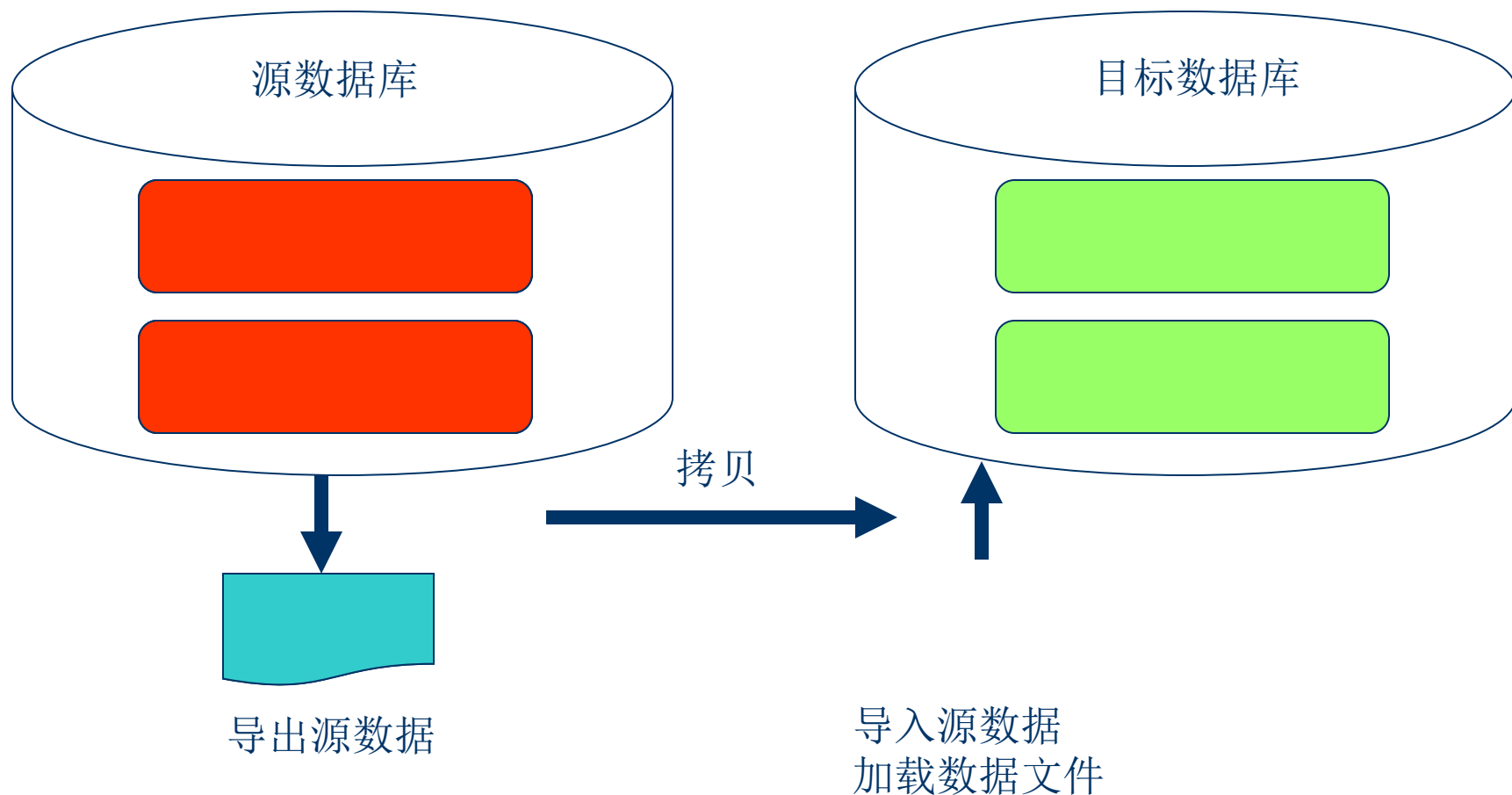
高可用之升级、迁移方案——单实例数据库迁移到RAC环境



高可用之升级、迁移方案——迁移表空间

- 直接拷贝物理数据文件的迁移方式，迁移速度很快
- 支持原数据库和目标数据库版本不同
- 迁移结束后，保留原始环境，一旦迁移失败，不会对原有系统产生影响
- 对操作系统和平台有一定要求

高可用之升级、迁移方案——迁移表空间



高可用之升级、迁移方案——ROLLING UP升级

- 逻辑STANDBY数据库进行ROLLING UP升级
- RAC数据库进行ROLLING UP升级
- STREAM环境进行ROLLING UP升级

高可用之警告

- 使用不当
- 过分相信