
ORACLE®

ORACLE

ORACLE THIRTY YEARS 30

利用Oracle 流技术实现数据复制

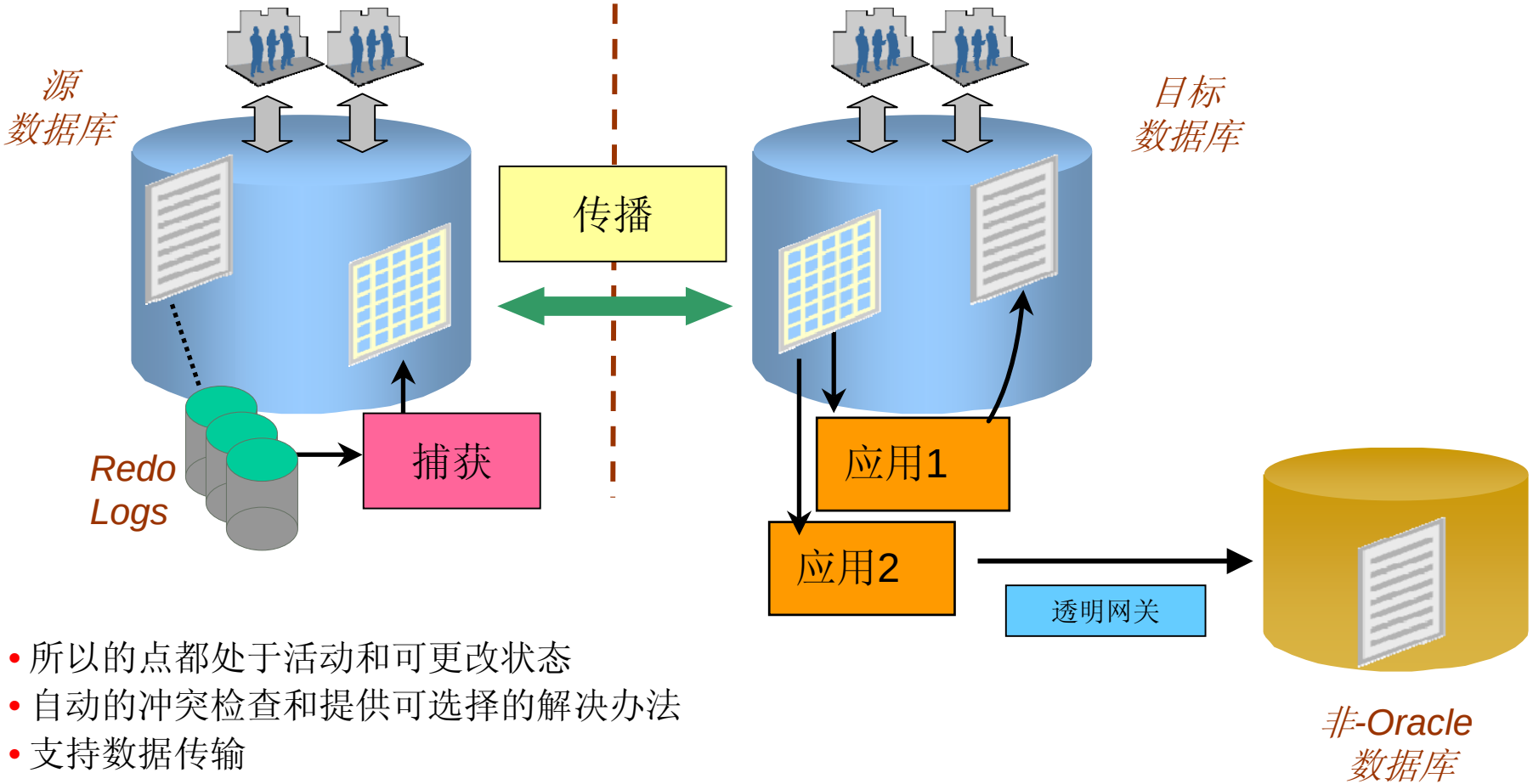
Ashish Ray
Director of Product Management
Database High Availability, Server Technologies

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.
The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

什么是 Oracle Streams?

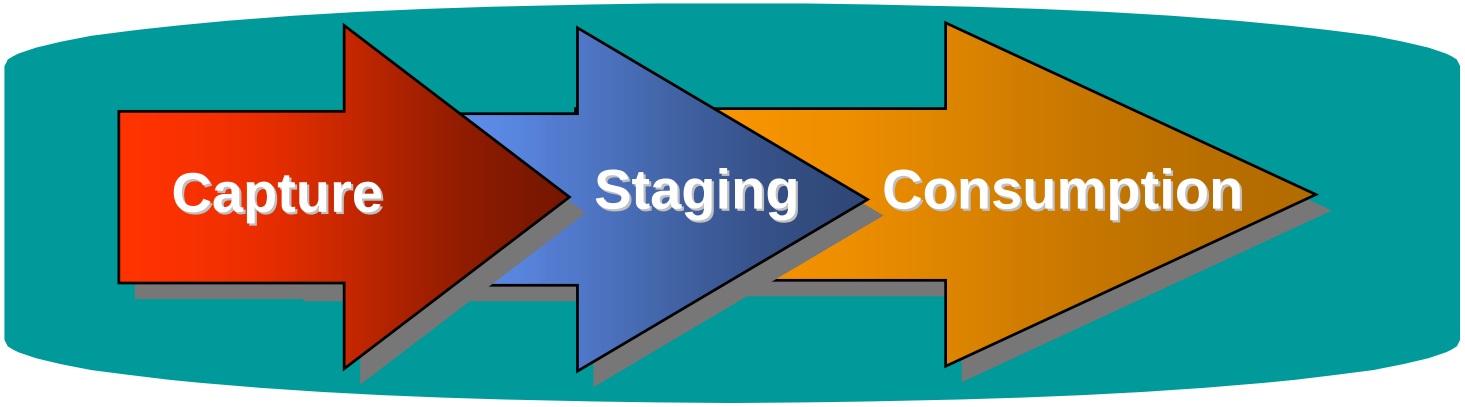
- 一个信息共享的简单解决方案
- 它提供如下功能：
  - 复制
  - 消息队列
  - 数据仓库的装载和简单的 ETL
  - 数据库迁移
  - 应用升级
  - 事件管理和通告

Streams: 体系结构



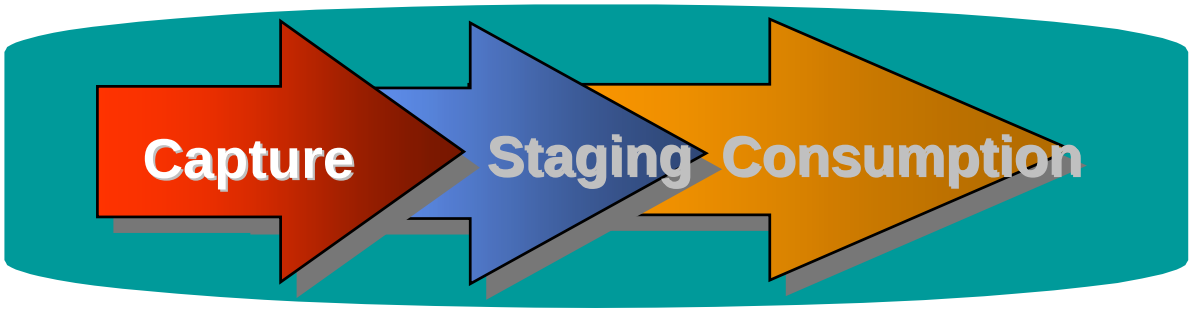
- 所以的点都处于活动和可更改状态
- 自动的冲突检查和提供可选择的解决办法
- 支持数据传输
- 灵活的配置 – n-way, hub & spoke, …
- 数据库平台/版本/和schema的结构可以不同
- 为用户的应用提供一个高可用的方案，这个方案中的update的冲突可以避免或可以管理

Streams: 功能组件



异步的信息共享架构

Capture-捕获

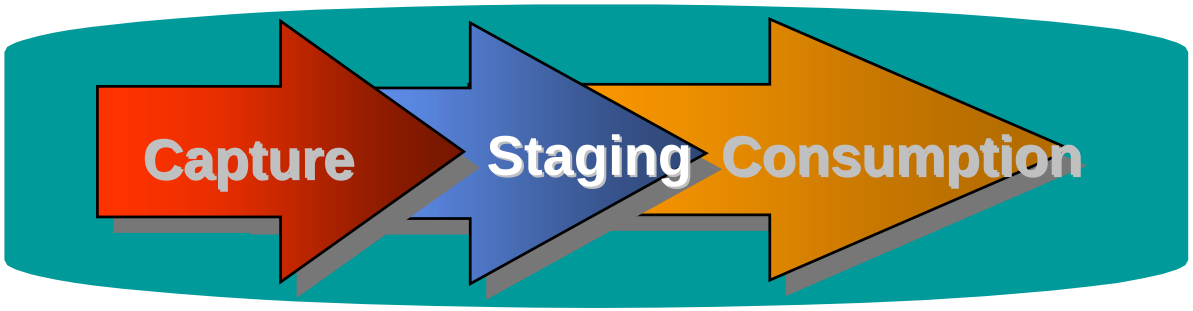


- 低开销，低延迟的变更捕获
  - 数据库的变更被写入在线的redo log
  - 当变更被写入数据库log后，Oracle Streams 提取这些变化
    - Redo or 归档 log 文件
    - 能在本地或 or Downstream 中捕获
  - 变更被格式化成逻辑变更记录(LCR)，一个变更的表示方式
  - 自动重启（实例重启或者RAC实例发生错误保护）
  - 自动的流控制

逻辑变更记录 (LCR)

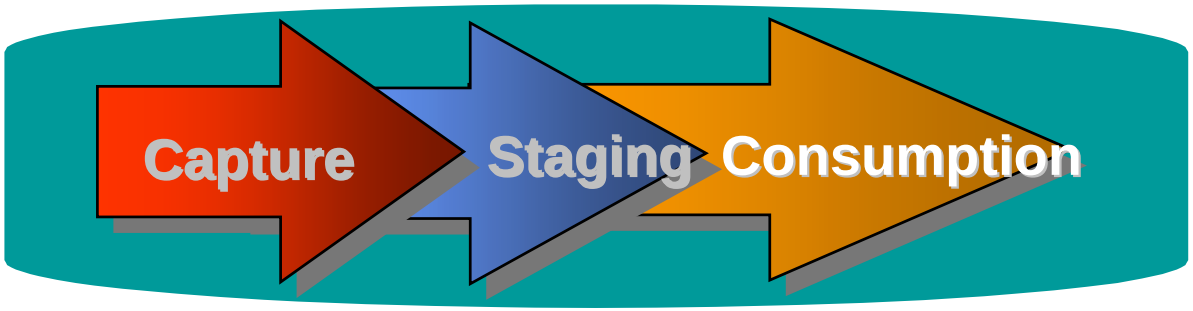
- 数据库变更 = LCR
  - DML
    - 行变更 = LCR
    - 对象的名字，属主，DML的类型，SCN,…
    - 老值, 新值
    - 可选的属性: 用户名，会话，线索,…
  - DDL
    - 对象的名字，属主，DML的类型，SCN,…
    - DDL 文本
    - 可选的属性: 用户名，会话，线索,…
  - LOB 和 LONG 字段
    - 每个 LOB 或者 LONG 字段column生成多个 LCRs
    - 大数据分段

Staging-暂存



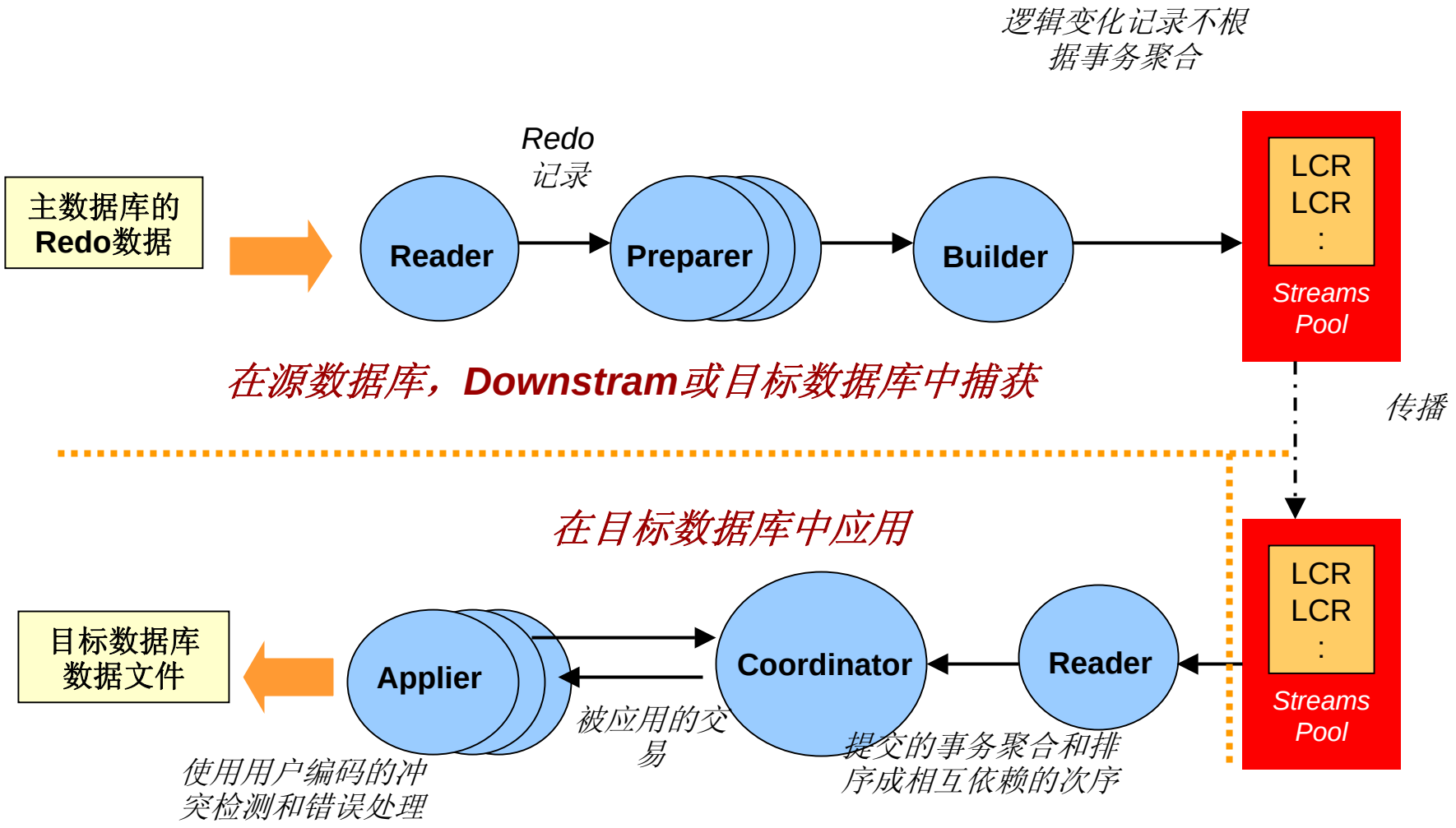
- **Streams 发布捕获事件到内存中的暂存区域**
  - 暂存区是用类型的队列实现
  - Streams 捕获使用缓存的 ANYDATA 类型的队列
    - SGA中的Streams缓存池
  - 订阅者：另外一个暂存区或Streams进程
  - 消息一直保留在暂存区直到被所以的订阅者都使用
- **在暂存区之间的传播通过传播任务定期进行**
  - LCRs 从本地的数据库通过队列被传播到目标数据库的队列里

Apply-应用

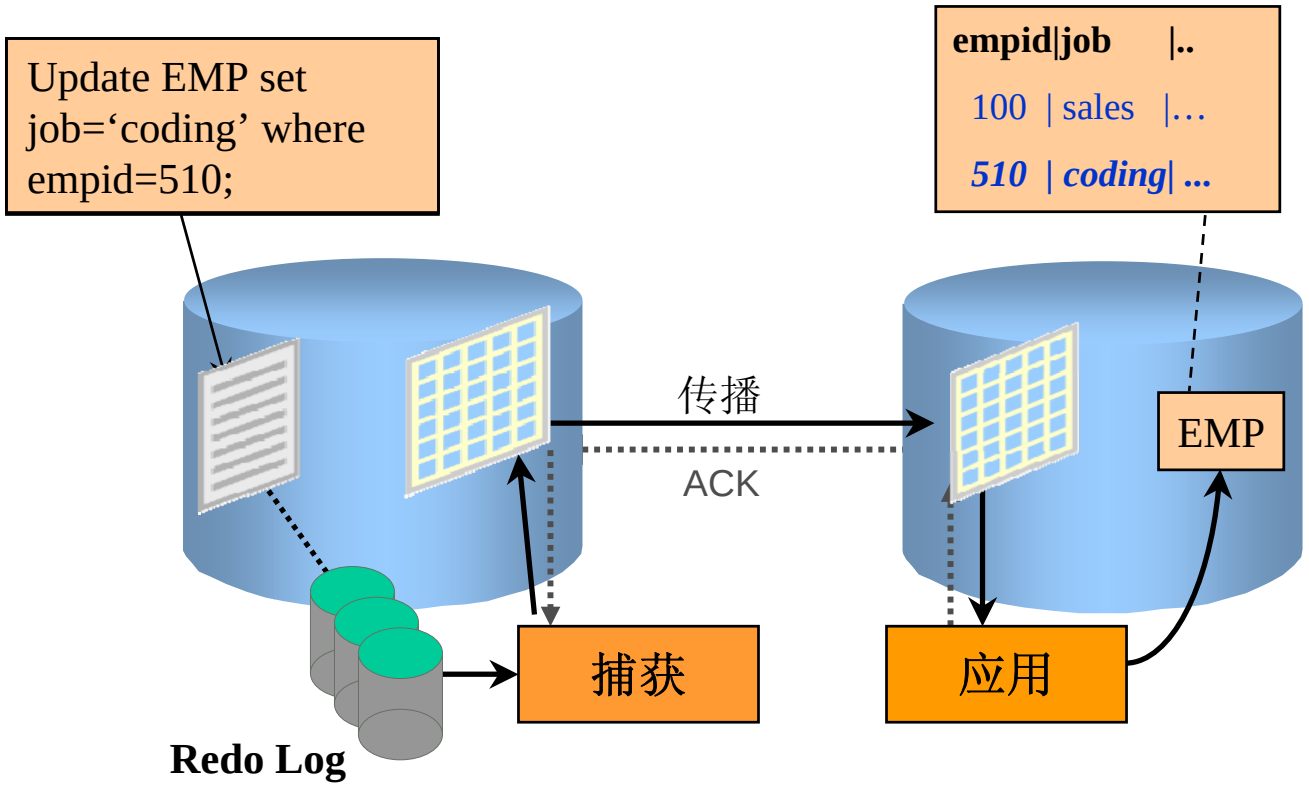


- 缺省的应用引擎将间接地应用在LCR里的DML或者DDL
  - 应用到本地的Oracle表
  - 通过数据库链接应用到非Oracle表
- 自动的带有选项的冲突检测
  - 没有解决的冲突放在错误队列中
  - 事务可以从错误队列中重新应用或者删除
- 最大地并发使用并行应用

Streams 进程的体系结构

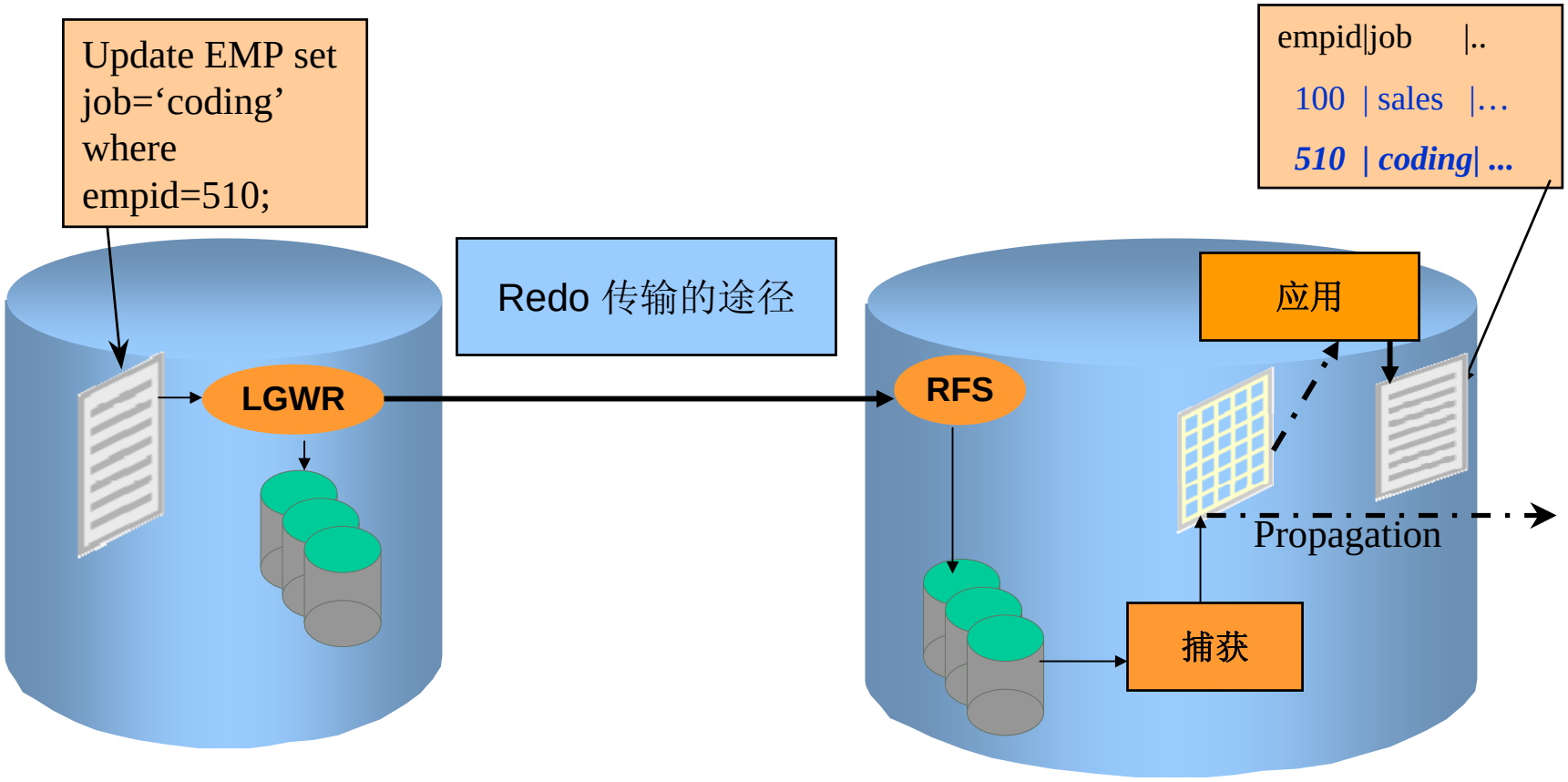


Streams 捕获和应用: 复制配置



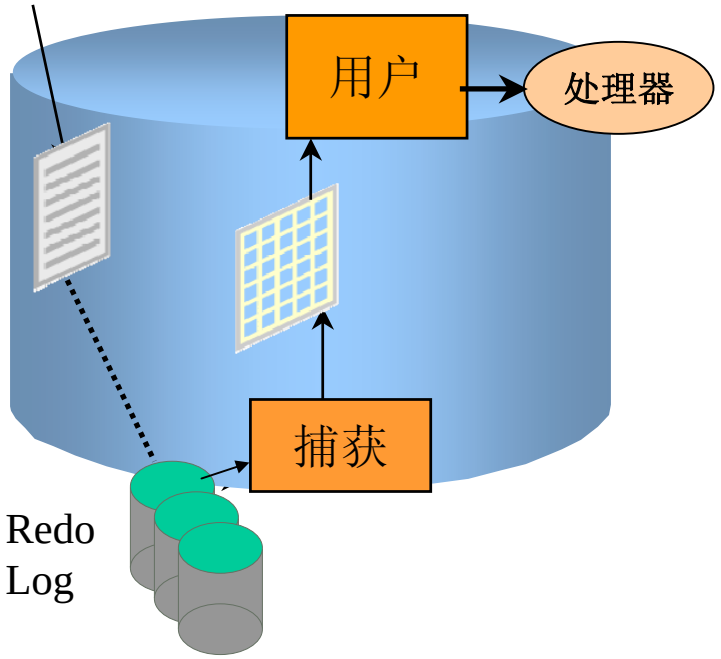
本地捕获

Downstream 的捕获



使用本地客户化的Streams 捕获

Update EMP set job= ‘coding’ where empid= 510;



本地捕获

PL/SQL Package: **DBMS_APPLY_ADM**

LCR:

DML 处理器
- 每个表每个操作 (Insert/Update/Delete/Lob_Update)
- **SET_DML_HANDLER()**

DDL 处理器
- 定义整个应用进程
- **ALTER_APPLY()**

应用指示
- **SET_ENQUEUE_DESTINATION()**
- **SET_EXECUTE()**

事务:

Pre_Commit
- 定义整个应用进程
- **ALTER_APPLY()**

基于规则的配置

- 消费者预定发布的事件
- 规则使用SQL WHERE 子句表达

```
dbms_rule_adm.create_rule(
   rule_name=>'scott.rule1',
condition=>':dml.get_object_owner() = "SCOTT" AND
            :dml.get_object_name()="EMP"');
```

- 规则集控制捕获、暂存的应用的行为
  - 包含
  - 排除
- 动态规则维护
- 裁剪的复制API

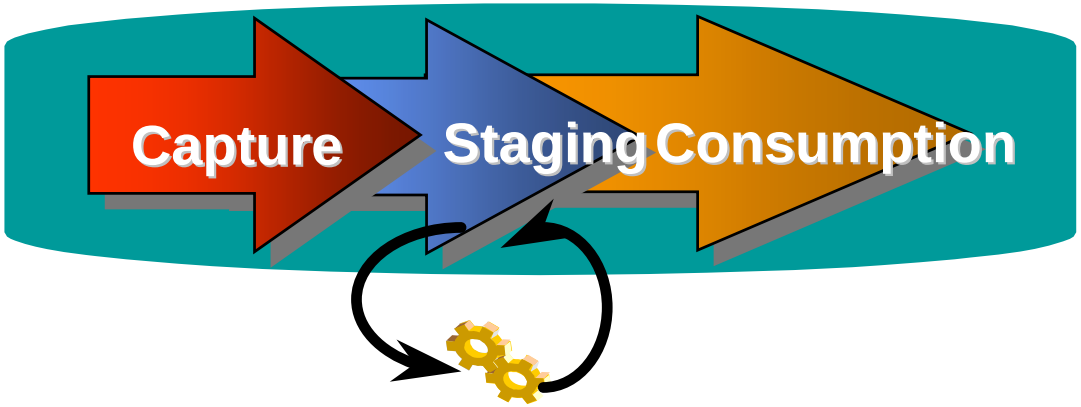
DBMS_STREAMS_ADM.MAINTAIN_*
    {Tables | Schemas | Global | TTS}
DBMS_STREAMS_ADM.ADD_*_RULES
    {Table | Schema | Global |Subset}

10.2: 简单的配置

Schema 复制

```
BEGIN dbms_streams_adm.MAINTAIN_SCHEMAS(
    SCHEMA_NAMES       => 'HR,SCOTT',
    SOURCE_DIRECTORY_OBJECT => null,
    DESTINATION_DIRECTORY_OBJECT  => null,
    SOURCE_DATABASE  => NULL,
    DESTINATION_DATABASE    => 'TARGET_global_name',
    PERFORM_ACTIONS  => TRUE,
    BI_DIRECTIONAL => FALSE,
    INSTANTIATION  => DBMS_STREAMS_ADM.INSTANTIATION_SCHEMA_NETWORK
   ,SCRIPT_DIRECTORY_OBJECT => 'SCRIPT_DIR',
   SCRIPT_NAME   =>  'generated_schemas_script.sql',
);
END;
/
```

基于规则的转换



- 转换可以在如下几个步骤实施:
  - 在事件进入暂存区时
  - 当事件离开暂存区时
  - 当事件在两个暂存区传播时 areas
- 声明转换
  - schema,表,字段改名
  - 增加或删除字段
- 客户转换
  - 用户提供的 PL/SQL 函数

声明转换

```
BEGIN
   DBMS_STREAMS_ADM.RENAME_SCHEMA(
   rule_name => 'STRMADMIN.HR51',
   from_schema_name => ‘HR',
   to_schema_name => 'HR_REPL',
   step_number => 0,
   operation => 'ADD');
END;
/
```

```
SELECT rule_owner||'.'||rule_name
  rule,transform_type,from_schema_name,to_schema_name from
  DBA_STREAMS_TRANSFORMATIONS;
```

```
RULE              TYPE                          FROM         TO
--------------    ------------------------      ----------   ------------
STRMADMIN.HR51    DECLARATIVE TRANSFORMATION    HR           HR_REPL
```

冲突的解决

- 如果被DML允许在多个地方修改一张表的同一个字段冲突可能发生
- 冲突总是能被检测到
- 如果有相应的配置，应用(Apply)试图解决冲突
  - 可以使用多个现成的冲突解决程序或者用户自己写的
  - 冲突字段被主键或替代键所标识
  - 为解决目标数据库发生冲突的字段的问题，源数据库可以提供附加的日志

处理错误

- 错误队列存储不能成功应用到目标数据库的事务
  - 当接受队列建立后自动建立
  - 只包含本地的错误信息
  - 包含每个错误事务的 LCRs
- **DBMS_APPLY_ADM.EXECUTE_ERROR()** 中有可选的错误处理程序
- 应用错误管理
  - **MESSAGE_NUMBER** column in **DBA_APPLY_ERROR**
  - 参考在使用手册中的脚本 "*Oracle Streams Concepts and Administration*", Chap 22 – "*Monitoring Streams Apply Processes*", and Section – "*Displaying Detailed Information About Apply Errors*"

错误事务重试

- Execute_error can specify a *user_procedure*

```
BEGIN DBMS_APPLY_ADM.EXECUTE_ERROR(
   local_transaction_id => '5.6.924',
   execute_as_user => false,
   user_procedure => 'strmadmin.modify_emp_salary');
END;
/
```

- User Procedure signature

```
PROCEDURE strmadmin.modify_emp_salary(
   in_any IN ANYDATA,
   error_record IN DBA_APPLY_ERROR%ROWTYPE,
   error_message_number IN NUMBER,
   messaging_default_processing IN OUT BOOLEAN,
   out_any OUT ANYDATA)
```

例子：user_procedure

- Execute_error(‘1.1.1’,user_procedure=>’MY_ERR_PROC’);

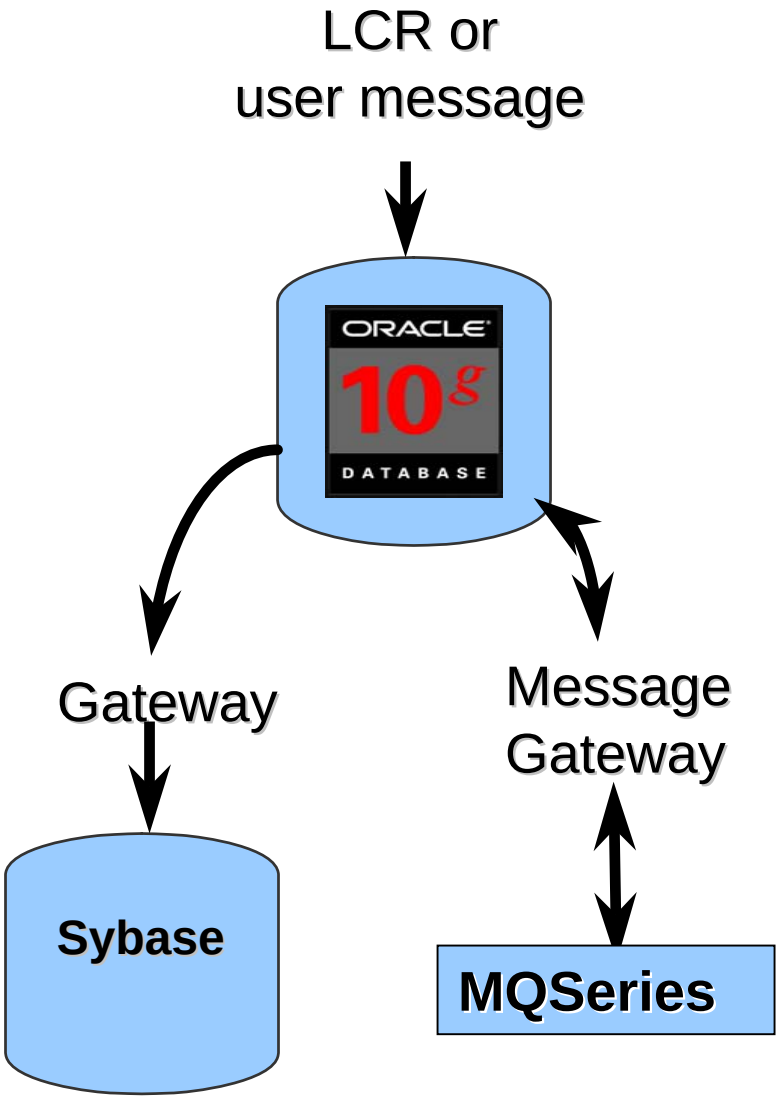
```
create or replace procedure strmadmin.my_err_proc (a  in anydata,
    ae in dba_apply_error%rowtype,
    en in number,
    messaging_default_processing in out boolean,
    out_any  out anydata) AS

  typenm   VARCHAR2(61);
  rowlcr   SYS.LCR$_ROW_RECORD;
  res      NUMBER;

BEGIN
 out_any := a;
 if (en = 107) then
   res := a.GETOBJECT(rowlcr);
   rowlcr.set_value('old', 'SALARY', Sys.AnyData.ConvertNumber(8000));
   messaging_default_processing := TRUE;
   out_any := Sys.AnyData.ConvertObject(rowlcr);
 end if;
END;
/
```

‘en’ is the lcr that failed:

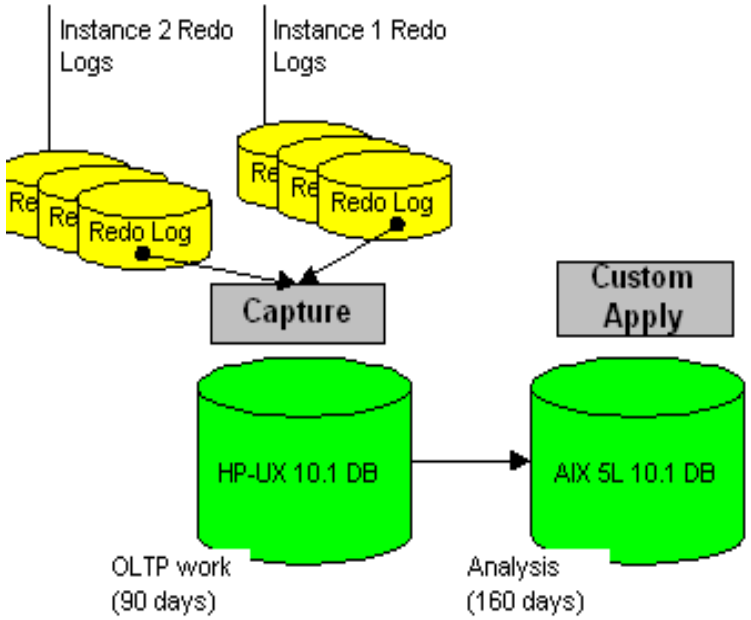
MESSAGE_NUMBER in DBA_APPLY_ERROR

支持异构数据库

- Oracle通过网关到非 Oracle 的应用
  - Apply process on Oracle node applies change
- 支持非Oracle数据库到Oracle数据库的LCRs的捕获
- 消息网关
  - MQSeries
  - Tibco

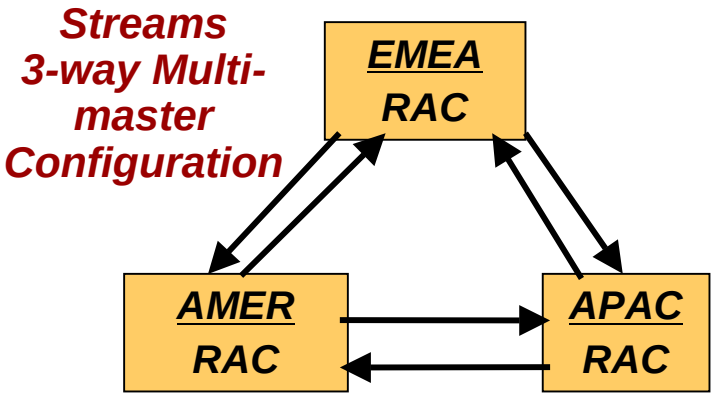


案例学习 #1: 10g RAC->10g



- 一个大型制造工厂
- 分离的报表系统(Offload reporting)
- 持续的负载 (24/7制造)
- 客户化应用: 跟踪所以的事务

案例学习 #2
10g RAC<-->10gRAC



- 大型投资银行:
  - Linux 10g RAC (2节点) 数据库:
    - 在不同的地理区域(WAN)
    - *一天之内负载在三大洲不断轮转 (Follow the sun* workload)
  - 客户化的 Streams，用于把 LOBs 转换成 XMLType

每个数据库

3 个队列: 每个进程一个

1 个捕获进程 (本地变化)

2 个应用进程

- 变更来自于其他的源数据库

Streams 复制的最佳实践

Ref. Chap. 14 – "Best Practices for Streams Replication Databases", Oracle Streams Replication Administrator's Guide 10g Release 2 (10.2)

一般的配置提示

- 对每个捕获，应用和每个源数据库进程使用不同的队列
- init.ora:
  - **_job_queue_interval=1** (在源数据库设置，表示任务队列扫描频率的间隔(秒))
  - **streams_pool_size=200M** (在源和目标数据库设置, 参见 **V$STREAMS_POOL_ADVICE**)
- 对于广域网: (SQL*NET 参数)
  - 增加 SDU (sqlnet.ora, tnsnames.ora, listener.ora)
  - 增加 send_buf_size, recv_buf_size
  - 对于 downstream 的捕获, 参见 MAA 最佳实践白皮书: *Oracle Database 10g Release 2 Best Practices: Data Guard Redo Transport & Network Configuration*

10gR2 Streams 进程参数

捕获 (DBMS_CAPTURE_ADM package)

- 如果需要，设置捕获检查点的持续时间

  Alter_capture(‘captureName’, checkpoint_retention_time=>7)

- 减少捕获检查点频率的参数

  Set_parameter(‘captureName’,’_checkpoint_frequency’,’1000’)

传播 (DBMS_PROPAGATION_ADM package)

- 使用 queue_to_queue 参数 (设置成 TRUE)
  - 源和目标数据库必须是10.2 或者是以上版本

应用 (DBMS_APPLY_ADM package)

- Set_parameter(‘applyName’,’parallelism’,’4’)
- Set_parameter(‘applyName’,’_hash_table_size’,’10000000’)
- Set_parameter(‘applyName’,’disable_on_error’,’N’)

应用性能的提示

- **ALTER TABLE SYS.STREAMS$_APPLY_PROGRESS INITRANS 16 PCTFREE 10;**
  - **INITRANS** 的值应该和应用的并行度相匹配

- 批量处理
  - 频繁提交　(事务大小 < 1000 LCRs)
  - 考虑使用存储过程复制 – 复制 PL/SQL 存储过程调用, 而不是复制存储过程产生的DML – 有示例程序

规则设置提示

- Spelling counts!
- source_database_name = 源数据库的GLOBAL_NAME
  - 必须正确设置
- 消除重复和相互覆盖的规则集
- 使用否规则集(negative ruleset) (10*g* and above)
  - 规则用于不允许的的对象和不支持的数据类型
- 避免复杂的规则
  - LIKE
  - Functions
  - NOT (9.2)

操作设置提示

- 是否复制 DDL?
  - 避免复制由系统生成的完整性约束和索引名
  - 修改手工的热备份脚本，设置应用标签
    - 例如: **dbms_streams.set_tag('99')**

- 是否从磁盘上删除归档日志?
  - 不要删除捕获进程重启后所需要的日志文件
    - 最小的 **REQUIRED_CHECKPOINT_SCN值**
    - 配置源数据库归档日志文件存储的位置，不要放在闪回回复区，对于downstream数据库也一样

源数据库: 心跳

- 使用周期性的任务更新心跳表来实现
  - 2个字段: Dbname, timestamp
  - 配置 Streams 捕获/应用 “heartbeat” 表
  - 配置冲突解决覆盖或最晚的时间标签
- 在数据库中生成活动
- 为DBA快速报告状态

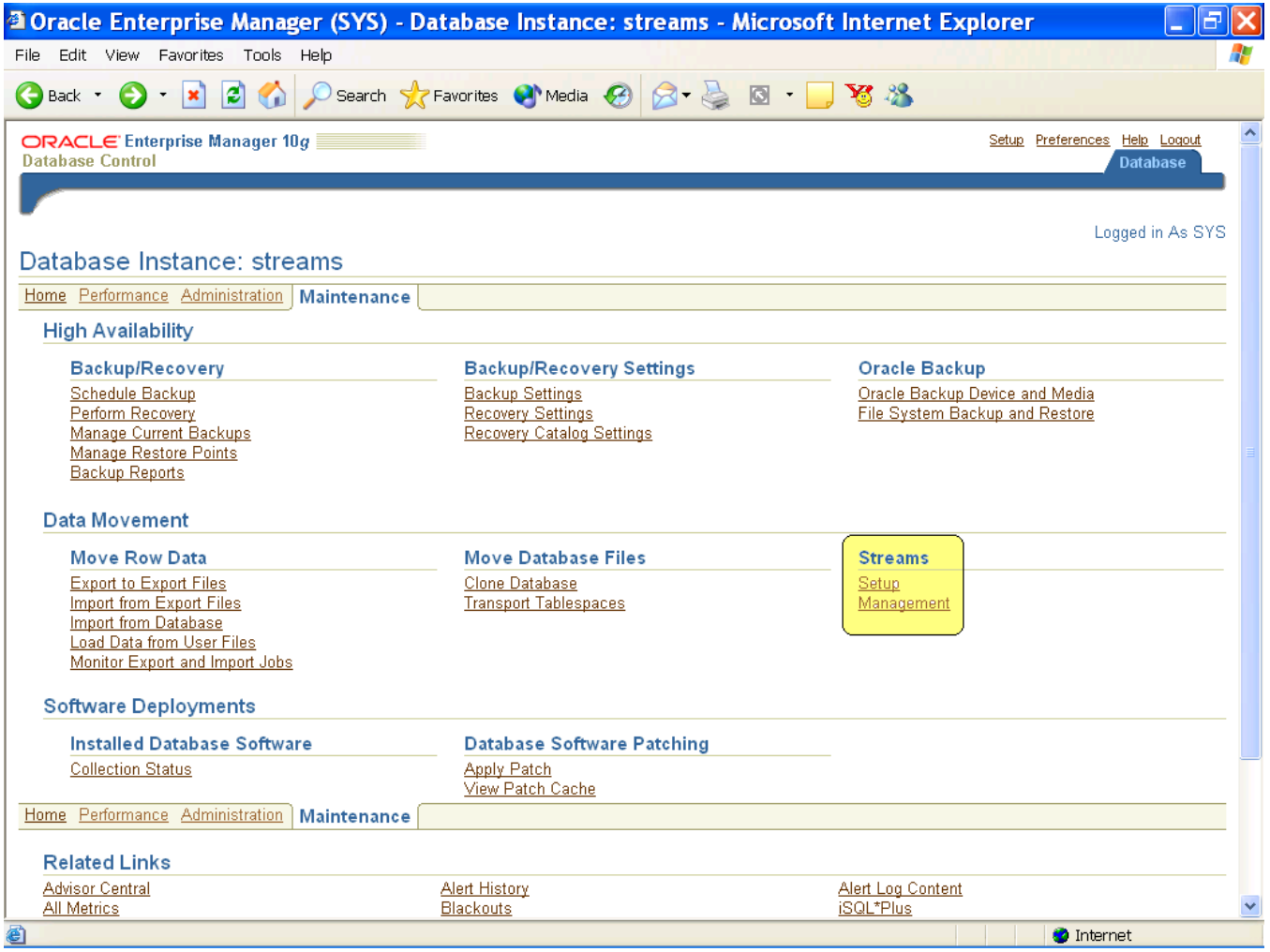
周期维护

- **DBMS_CAPTURE_ADM.BUILD()**
- **DBMS_CAPTURE_ADM.PREPARE_GLOBAL_INSTANTIATION()**

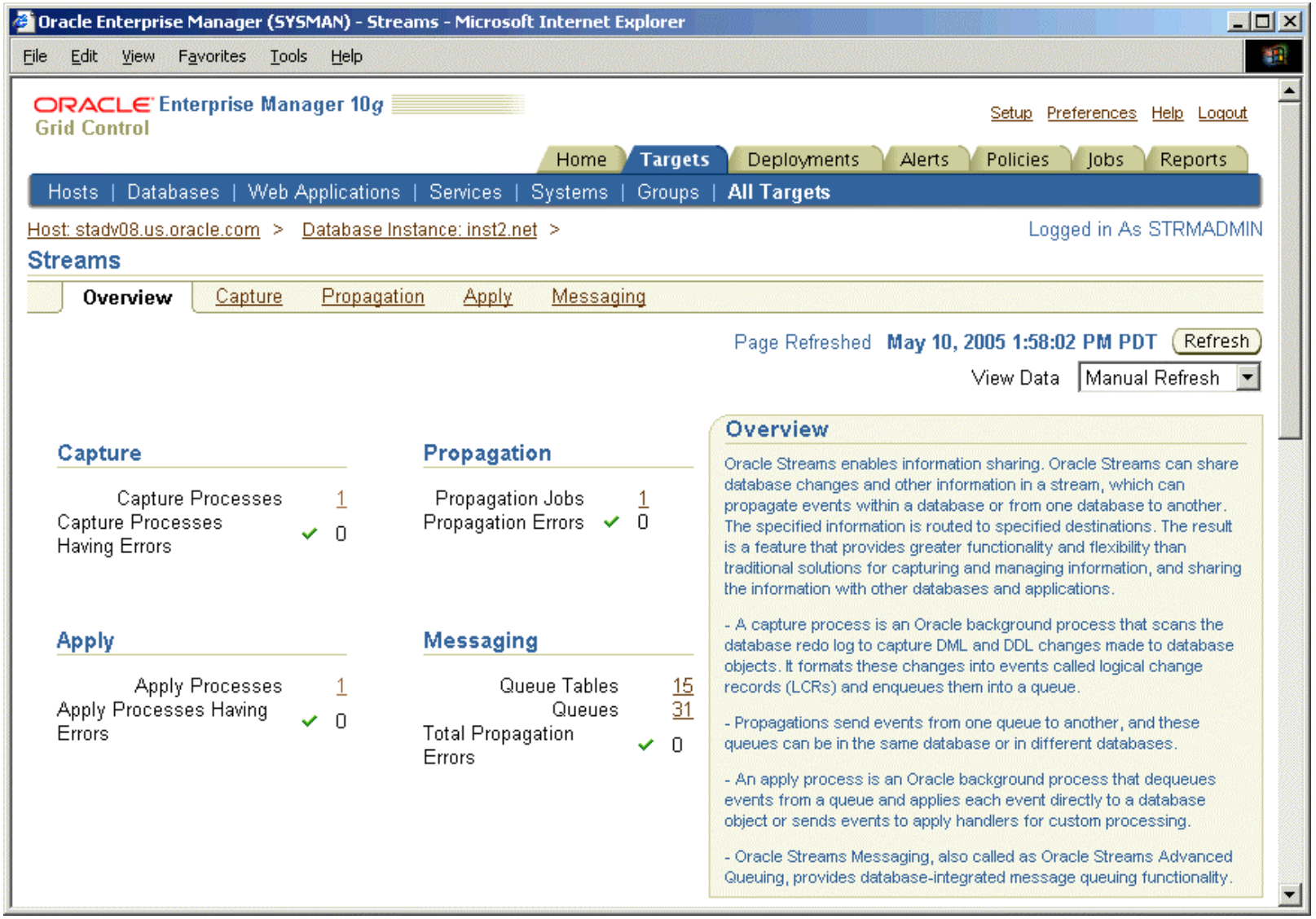
监视

- 视图
- DBControl, Grid Control
- Healthcheck – MetaLink Note:273674.1 – *Configuration Report and Health Check Script*
- STRMMON – MetaLink Note:290605.1 – *Oracle Streams STRMMON Monitoring Utility*
- Alert Log – 留意Capture进程生成的消息，当捕获进程看到一个大的或者长事务会在日志中吉林信息

Enterprise Manager



Enterprise Manager



Streams 总结

- 特性:
  - 基于日志变更捕获
  - 客户化的应用引擎
  - Schema 演化
  - 转换
  - 支持异种数据库
- 多面手:
  - 复制数据
  - 合并信息
  - 在数据库迁移和升级阶段提供高可用的特性

Streams 的增强

ORACLE 11g DATABASE

- 源和目标数据的比较和汇集
- Streams 性能顾问视图
- Streams在Hub和Spoke中的分离/合并复制机制
  - 为所有的复制提供高可用性
  - 重启的复制可以自动快速地追溯
- 跨数据库的LCR跟踪
  - 在一个视图中可以跟踪一个消息的开始和结束过程
- 性能优化
- Streams 同步捕获
  - 在Oracle数据库11g中都可用
  - 高效率的内部机制去即时地捕获数据库变更

资源

- Collateral, best practices, sample code: http://otn.oracle.com/products/dataint/
- MetaLink Note 418755.1 – 10.2 Streams Recommendations


Q&A
QUESTIONS
ANSWERS

更多的信息

search.oracle.com

或者

oracle.com


ORACLE®

ORACLE IS THE INFORMATION COMPANY