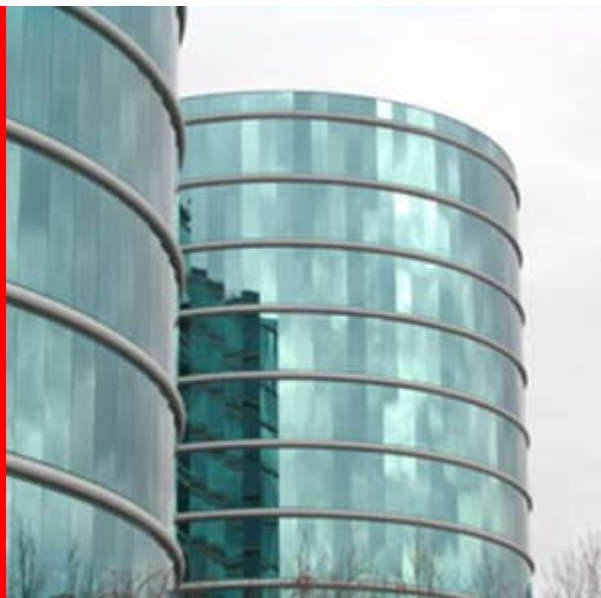


ORACLE®



ORACLE®



数据库移植案例分析

王克非
开发经理,
Oracle深圳合作伙伴解决方案中心

日程

- 案例背景
 - ISV
 - 应用简介
 - 移植需求
- 移植中的挑战
- 规划
- 工具
- 移植流程
 - 大纲移植
 - 数据移植
 - 应用移植
- 问题和解决方案
- 总结

案例背景

ISV – 新意科技

- 新意科技是一家专注于金融证券业应用软件开发的专业软件公司
- 新意科技是中国证券一级、二级清算系统软件提供商的领导者
- 新意科技产品在一级清算领域的客户占有率达到100%，二级清算领域达到70%以上

案例背景

应用 – 新意证券综合管理平台 (E-SIM)

- 新意证券综合管理平台(E-SIM)，新意的旗舰产品
- 主要处理证券公司法人与登记结算公司的一级资金清算和券商总部和下属各分支机构的二级资金清算业务，包括股份清算业务处理、资金清算业务处理管理、开户与存管业务和银证资金往来管理系统等
- 客户包括了国内大部分的证券公司

案例背景





为什么需要移植？

 Oracle 特性

挑战

- 现有的数据库和硬件已经不能满足日益增长的需要处理的数据量
- 需要绝对精确的计算结果
- 交易互相关联，整个清算业务必须作为一个大事务来处理，对底层数据库构成巨大挑战
- 处理必须及时完成，以保证第二天交易的正常进行

ORACLE的解决之道

- SQL*Loader 实现快速数据装载 
- 增强的SQL和PL/SQL 引擎提供高性能的数据处理
- MVRC机制保证了大量并发情况下最少的死锁并且避免行级锁的升级 
- RAC 提供更好的性能，可扩展性和高可用性 
- 已经被证明能够高效的处理海量数据 

价值

- 更快的系统
- 对于未来业务增长具有可扩展性
- 更好的可管理性

移植的挑战

- 新意科技的开发队伍缺乏Oracle专家
- 客户需要短时间内完成移植
- 需要移植的数据库规模较大
 - 多个源数据库
 - 数百个数据表
 - 数十个存储过程（绝大部分代码复杂）
 - 使用了T-SQL的特有功能 (语法/函数/过程)

移植计划

- 分析待移植的源数据库
 - 使用了自增长的ID类型
 - 大量使用了临时表
 - 过程内调用shell命令
 - 没有用户自定义类型
 - **Most of procedures are complex (over 200 lines)**
- 移植策略
 - 第一步：移植数据大纲结构
 - 第二步：迁移数据
 - 第三步：转换存储过程和触发器
 - 第四步：测试移植后的应用
 - 第五步：生成打包安装的脚本
- 方法
 - 使用自动工具移植大纲和数据
 - 使用自动工具移植存储过程
 - 人工校验并修改未完成移植的过程

移植计划 (续)

- 利用Oracle 深圳合作伙伴解决方案中心的帮助
 - 免费帮助合作伙伴应用Oracle产品和技术
 - 数据库移植，挑优，中间件移植，嵌入式
 - 提供移植最佳策略建议
 - Oracle移植专家协助合作伙伴开发人员进行移植

移植工具

- **Oracle Migration Workbench (OMWB)**
 - 免费工具 – 可从 <http://otn.oracle.com> 下载
 - 支持多种第三方数据库，包括SQLServer 7/2000.
 - 提供完整功能 – 可以移植表，函数，存储过程，数据
- **SQLDeveloper Migration Workbench**
 - 新产品 (同样免费)， SQLDeveloper的插件
 - 目前仅支持 SQLServer, Access 和 MySQL
 - 提供了更好的针对 T-SQL 的解析转换能力，并且有更好的代码质量和可用性
- **OMWB** 被用来移植大纲和数据
- **SQLDeveloper** 用来转换存储过程

SQLDeveloper Migration Workbench

The screenshot displays the SQL Developer Migration Workbench interface. It features two side-by-side editors comparing SQL code. The left editor, titled 'Scratch Editor', shows a script with several lines of SQL code, including a SELECT statement and a SET statement. The right editor, titled 'Oracle Diff', shows a similar script with a CREATE OR REPLACE PROCEDURE statement. The code in both editors is highlighted in green and blue, indicating differences or specific sections of interest. The interface includes a toolbar with navigation and execution icons, and a status bar at the bottom.

```
(View-Only)
, @ComparePrice money OUTPUT
, @ListPrice money OUT
AS
SELECT p.[Name] AS Product, p.ListPrice AS 'List Price
FROM Production.Product AS p
JOIN Production.ProductSubcategory AS s
ON p.ProductSubcategoryID = s.ProductSubcategoryID
WHERE s.[Name] LIKE @Product AND p.ListPrice < @MaxPri
-- Populate the output variable @ListPrice.
SET @ListPrice = (SELECT MAX(p.ListPrice)
FROM Production.Product AS p
JOIN Production.ProductSubcategory AS s
ON p.ProductSubcategoryID = s.ProductSubcategory
WHERE s.[Name] LIKE @Product AND p.ListPrice < @Me
-- Populate the output variable @compareprice.
SET @ComparePrice = @MaxPrice;
```

```
(View-Only)
CREATE OR REPLACE PROCEDURE uspGetList
(
v_Product IN VARCHAR2,
v_MaxPrice IN NUMBER,
v_ComparePrice OUT NUMBER,
v_ListPrice OUT NUMBER,
cv_1 IN OUT SYS_REFCURSOR
)
AS
BEGIN
OPEN cv_1 FOR
SELECT p.NAME Product,
p.ListPrice "List Price"
FROM Product p
JOIN ProductSubcategory s
ON p.ProductSubcategoryID = s.ProductSubcategoryID
WHERE s.NAME LIKE v_Product
AND p.ListPrice < v_MaxPrice;
-- Populate the output variable @ListPrice.
SELECT MAX(p.ListPrice)
INTO v_ListPrice
FROM Product p
JOIN ProductSubcategory s
ON p.ProductSubcategoryID = s.ProductSubcategoryID
WHERE s.NAME LIKE v_Product
AND p.ListPrice < v_MaxPrice;
-- Populate the output variable @compareprice
```

工作模式

- 新意工程师向**Oracle**工程师介绍应用架构
- 共同分析源数据库并制定移植策略和计划
 - **Oracle** 具有丰富的相关项目经验
 - **Oracle** 会主导次过程
- 大纲/数据移植
 - 使用自动工具
 - 新意工程师检验移植结果的正确性和完整性
 - **Oracle**工程师协助解决相关语法和规范等问题
- 存储过程移植
 - 工具可以自动转换
 - **Oracle** 工程师协助解决转换中的问题
 - 人工转换复杂以及非标准的语句
- 测试并性能调优

大纲移植

- OMWB 捕获源数据结构
- 转换成Oracle模型
 - Database => Schema
 - ID => Sequence + Trigger
 - 自动转换表、视图、索引等的定义（除过程、触发器和函数以外的所有对象定义）
- 生成DDL脚本
- 在测试环境上应用DDL脚本
- 校验结果
 - OMWB 提供移植报告
 - 人工校验

数据移植

- OMWB生成数据导出脚本
 - 使用BCP将数据导出为文本文件
 - 一个表对应一个文件
- OMWB生成数据导入文件
 - Oracle SQL*Loader
 - 自动生成 SQL*Loader控制文件
- 需要人工校验ctl文件中的内容，包括日期时间格式
- 检查SQL*Loader 报告的未能成功导入的记录
 - 非法的约束条件
 - NOT NULL,SQL Server 允许 “ (空字符串). 但是Oracle当作NULL来处理。需要人工干预 (用空格代替或者废除非空约束)
 - 重新加载未能导入的记录

存储过程移植

- 从SQLServer导出DDL脚本
- 用SQLDeveloper Migration Workbench翻译脚本并得到最初的PL/SQL版本
- 编译并发现语法错误
 - 人工修改
- 检查代码查找语义问题
 - 人工修改
- SQLDeveloper Migration Workbench 得到较佳质量代码
 - 通过行间注释可以快速发现问题
 - 80% 以上的代码可以正确无误的得到翻译

测试和性能调优

- 新意重用其功能测试计划
- Oracle帮助测试性能并调优
 - 优化系统设置
 - 优化SQL

问题和解决方案

临时表

- SQLServer的临时表可以在会话间保留数据，而Oracle的全局临时表只能在一个会话或者事务内保存数据
- Workbench 会自动生成临时表定义
- 必须人工判断如何移植SQLServer存储过程中的临时表
- Local temporary table – 转换为Oracle全局临时表
- Global Temporary Table (##), 这些临时表是用来在会话间共享数据 (## global)
 - 考虑在Oracle中使用普通表
 - 重写查询和应用，避免使用临时表作为查询结果的中间层（某些local temporary table同样适用）

问题和解决方案

执行Shell命令

- EXEC XP_CMDSHHELL 'copy a.txt b.txt',NO_OUTPUT
- Oracle 没有直接对应的功能
- 可以通过外部存储过程来实现
 - 需要额外的编程和配置工作，需要编译
 - 对于需要打包安装的应用相对麻烦
- 替代方法: 使用 DBMS_SCHEDULER包

```
EXECUTE IMMEDIATE '
    dbms_scheduler.create_job(
        job_name => yourName -- better be random
        job_type => ''EXECUTABLE'',
        job_action => ''copy a.txt b.txt'',
        enabled => TRUE); '
DBMS_SCHEDULER.run_job(yourName);
```

问题和解决方案

Identify Column

- Oracle中没有对应的自动增长的数据类型
- 通过触发器 + 序列来模拟
- 序列用来得到不断增长的数字
 - 通过 CURRENTVAL 和 NEXTVAL来获取
 - `SELECT YOUR_SEQUENCE.NEXTVAL FROM DUAL`
- 每次插入操作时，触发器获得 NEXTVAL
- OMWB 会自动完成转换
- 从性能角度出发，可以考虑不使用触发器+序列，而是直接在 Insert语句中引用序列值
- 移植过程中触发器应该禁用，序列也应该小心规划来避免主键重复
- @@Identity

问题和解决方案

异常处理

- 异常处理
 - T-SQL 通过 @@error 来检查错误
 - 异常后可以继续执行后面的语句
 - Oracle的异常处理机制类似于JAVA

```
BEGIN
```

```
...
```

```
EXCEPTION
```

```
WHEN ZERO_DIVIDE THEN -- handles 'division by  
zero' error          ... ;
```

```
WHEN OTHERS THEN -- handles all other errors
```

```
...
```

```
END ;
```

- 必须结合应用逻辑来决定移植方案

问题和解决方案

其他

- 自定义函数
 - Workbench 会自动转换
 - Workbench 会创建一个OMWB_EMULATION包来模拟 T-SQL 内置函数
- Convert / Cast => TO_NUMBER, TO_DATE, TO_CHAR...
- IS_DATE
 - 需要自己实现

总结

为什么通过PSC来进行移植

- 1 成熟的方法
- 2 成熟的工具
- 3 Oracle 产品专家
- 4 移植专家
- 5 免费服务
- 6 降低移植成本
- 7 更短的项目时间
- 8 更快推向市场
- 9 更好的移植质量
- 10 本地支持，没有语言障碍



引言 技术主任, 新意科技


“移植获得了完全的成功，我们也从中学到了许多经验。没有深圳合作伙伴解决方案中心的帮助是不可能如期完成的”

资源

- Oracle 移植技术中心
 - <http://www.oracle.com/technology/tech/migration/index.html>
- 合作伙伴解决方案中心 – 深圳
 - <http://www.oracle.com/cdc/sspocc/index.html>
 - sspocc_ww@oracle.com

了解更多

search.oracle.com



or

oracle.com

ORACLE®