

第三部分 Oracle 8.x RDBMS

第9章 从Oracle 7.x到Oracle 8.x的移植

本章要点：

三种主要方法

移植工具

常见问题

移植之后的任务

这一章将讨论从Oracle 7.x数据库到Oracle 8.x数据库移植的三种主要方法。重点将集中在移植工具方法上，讨论伴随任何移植方法所必须的移植前的任务，并考虑常见问题和简要了解一些可能需要的移植后的任务。

9.1 三种主要方法

从Oracle 7数据库移植到Oracle 8数据库的三种主要方法如下：

- 移植工具——Oracle 8提供能够使用GUI或命令行模式的移植工具，用于帮助将 Oracle 7（7.1.6或更高版本）内建的数据库移植到 Oracle 8数据库中。
- 导入/导出——这是与使用所有的 Oracle 二进制导出和导入格式完成通常（同种版本到同种版本）整个数据库备份和恢复完全相同的技术。
- COPY或CTAS——这是一种用于拷贝整个数据库的方法，通过 SQL*Net（Net 8）的 COPY命令或者经过DDL Create Table As Select（CTAS）命令，从一个Oracle 7数据库到一个Oracle 8数据库拷贝整个数据库，每次仅能拷贝一个表。

表9-1概括列出每种方法的优点和缺点。

表9-1 移植方法的优点和缺点

方 法	优 点	缺 点
移植工具（内建移植）	由Oracle提供和支持。几乎是自动的。 速度相对快。空间相对有效地利用	源Oracle 7数据库必须是7.1.6或更高版本。仅能全部移植。移植是破坏性的（对源Oracle 7数据库）
导出/导入	任何Oracle 7源数据库。可以部分移植。 有利于消除碎片。移植无破坏性	速度相对慢，全库导出经常受操作系统和分区限制。除非直接写到磁带，否则在磁盘上要求有足够空间存放一个数据库完全导出文件
COPY或CTAS	任何Oracle 7源数据库。可以部分移植。 有利于消除碎片。移植无破坏性	速度相对慢，需要明显的手工操作。 源和目标数据库都必须是打开的

现在简要地看一看这些优点和缺点的比较。在移植之前或移植进行过程中，新的 Oracle 8基本表和V\$视图被增加，另外有些 Oracle 7基本表和V\$视图被修改。更新数据文件头的结构，

包括控制文件和重做日志文件的文件头结构。任何方法必须在 Oracle 7数据库转换到 Oracle 8数据库之前或其间完成这项工作。

现在讨论许多 Oracle 7数据库系统管理员应该已经熟悉的两种方法：导出/导入和 COPY/CTAS。用导出/导入方法，当 Oracle 8导入工具从 Oracle 7导出文件导入时，所需要的 Oracle 7到Oracle 8的转化自动发生。使用导出/导入方法的最大好处可能是使用工具管理的方便性，而最大缺点是对整个数据库导出文件需要大的空间。然而用导出/导入方法“逐个地”移植也是可能的，尽管每次你仅使用小的空间，但将需要花很长时间来完成移植整个数据库的工作。然而，在多数情况下的讨论几乎不关心移植速度的快慢。相反，在多数安装中最主要关心的是进行成功的移植。

COPY/CTUS方法对于Oracle 7的数据库管理员与同导出/导入一样熟悉。然而这种方法很少作为主要的备份和恢复手段使用，特别是对 VLDB。它经常被用户用作标准备份（如导出/导入或 RMAN）之外的备份方法。这种方法最大的优点是操作简单，而这种方法最大的缺点是移植要花很长的时间。发生后面一种情况是因为表必须被拷贝，不管表有多少，每次只能拷贝一个表。在所有的拷贝完成之前，两个数据库必须保持运行状态。另外，与导出/导入一样，移植也可分阶段进行，但这只是更进一步加长了移植时间。与导出/导入相反，在COPY/CTAS情况下，移植所花的时间是最重要的，因为它与会潜在地影响正常操作的时间。要注意的另一方面是，INSERT INTO.....SELECT能够替代CREATE TABLE AS SELECT，唯一不同是目标表已经被创建。

稍微讨论一下第四种存在的方法：卸载/装载方法。用这种方法卸出所有的 Oracle 7表数据到一个大的平面（ASCII）文件，然后将它们用 SQL*Loader装入到你新的 Oracle 8数据库中。这种方法的好处是像导出一样，在被重新装入之前，能够卸载脱机数据。因此，它是固有的异步模式，然而，它也是固有的最耗时的手工操作。除了最小的和最简单的数据库之外，我认为对所有其他的数据库而言，这种方法不是真正可行的方法。

9.2 移植工具

因为移植工具（mig和mig80）对Oracle 8是新工具，它特别对大的数据库在空间和时间的有效性方面可能提供全面的解决。本节将讲述它。

9.2.1 高层次的步骤

使用移植工具的高层次步骤概括如下：

- 1) 安装移植工具。
- 2) 对移植的准备（参看9.2.3节“移植前的任务”）。
- 3) 安装Oracle 8软件。
- 4) 运行移植工具。
- 5) 准备将转换的Oracle 7数据库。
- 6) 从Oracle 7数据库到Oracle 8的转换。
- 7) 完成最后的整理工作（参见9.4节“移植之后的任务”）。

9.2.2 低层次的任务

接下来你将花一些时间来了解低层次的任务，在使用移植工具方法时必须遵循这些步骤，

查看每一步骤以获得详细解释。

1) 运行Oracle安装程序 (orainst)。选择“Install Migration Utility O7->O8”。这个工具被拷贝到Oracle7的\$ORACLE_HOME/bin子目录下。它也将相关的信息文件——一个新版本的migrate.bsq文件安装到\$ORACLE_HOME/dbs子目录,并安装所需要的NLS文件。

2) 按照下一节“移植前的任务”中的提纲,概念化和策略化地准备移植。明智地执行移植所要做的唯一最重要的事情是在继续进行移植之前,确保获得一个成功的、完全的 Oracle 7 数据库备份。如果可能,用两种不同方式获得两个完全备份,例如一个完全冷备份,另一个完全导出。关闭实例。

3) 运行Oracle安装程序 (orainst),选择Install/Upgrade安装Oracle 8软件。这后一种选择阻止安装程序创建新的实例和数据库。

4) 从任务2往下实例应该始终关闭。运行任务1在Oracle 7的\$ORACLE_HOME/bin子目录下安装的移植工具。这个可执行文件的命名在UNIX下是mig,在NT下是mig80。在运行mig之前,要知道它的主要参数。

- check_only——如果是FALSE (缺省值),计算使用的空间,发生实际移植。如果是TRUE,仅计算使用的空间。它与用主要导出来决定是否有足够的空间用来移植数据库 (源和目标,在这种情况下)的noexec参数相同。
- no_space_check——当它被置为TRUE时,不执行空间检查,check_only被忽略。如果为FALSE (缺省值),check_only起作用。
- dbname——被移植的旧的源数据库名 (在init.ora中的db_name)。
- new_dbname——新的目标数据库名 (在新的init.ora中的db_name),可以和旧的名相同;缺省值是DEFAULT。
- pfile——旧的源数据库的init.ora文件名。
- spool——日志输出的相对或绝对路径的文件名,以双引号括起。

警告 因为Oracle 8数据字典要比Oracle 7大50%,所以手动增加数据文件到SYSTEM表空间,或至少设置这些数据文件已经支持的自动区间扩展 (假设在操作系统级还有空间),这是十分重要的。另一个选择是手动重新调整或自动扩展 (AUTOEXTEND) SYSTEM表空间已经存在的数据文件。

在操作系统命令行提示符下运行 mig可执行文件,并在其中配以参数列表,如 <参数1>=<值1> <参数2>=<值2>等等。在NT操作系统在MS-DOS提示符下运行 mig80;在UNIX操作系统中,在shell提示符下运行 mig命令。假如你的 Oracle bin目录是在你系统启动的映象路径下,一个可能的例子如下:

```
C:\>MIG80 DBNAME=MYDB NEW_DBNAME=MYNEWD B SPOOL=MYLOG
```

移植工具将运行ASCII码脚本文件migrate.bsq,它接下来创建一个移植用户,在这个移植用户中建立视图以支持移植过程。当这样做之后,移植工具将在特定位置 (或缺省)的路径下留下一个日志文件 (从SPOOL选项),在Oracle 7 ORACLE_HOME/dbs子目录下留下一个二进制转换数据文件 (conv<SID>.dbf)。

5) 准备要转换的Oracle 7数据库,这是在运行移植工具之后独立的关键性的阶段。准备Oracle 7数据库,移走 (移动或重命名)控制文件,因为新的 Oracle8控制文件也在原来的位置上产生。确保所有其他文件、数据文件和重做日志文件原封不动。

同时，准备操作系统环境，重置 ORACLE_HOME 操作系统变量（在 NT 的注册表或 UNIX 的壳环境变量中）为新的 Oracle 8 目录位置。然后，重置 Oracle 二进制可执行文件路径（在适当的 NT 注册表分枝树中的映象路径，如果必要，或 UNIX 的 PATH 变量，尽管在 UNIX 系统中应该关心 ORACLE_HOME）。如果 ORACLE_SID 发生改变，重置这个变量。如果需要，设置 ORA_NLS33 参数。

编辑你的新的 Oracle 8 init.ora 文件以包含参数设置 compatible = 8.0.0（或更高）。同时移走过时的 Oracle 7 参数（请参考你的服务器手册，以找到最新版本）。确保你对 control_files = () 和其他文件位置（例如 bdump 和 cdump）的参数设置是正确的（也就是，不能包括任何 Oracle 7 的路径引用）。

从 Oracle 7 的 \$ORACLE_HOME/dbs 子目录到 Oracle 8 的 \$ORACLE_HOME/dbs 子目录复制转换数据文件 (conv<SID>.dbf)。

6) 转换你的数据库。这是不可返回的操作，换句话说，在你运行转换后，你的数据库要么转换成功，要么失败，没有完全冷备份，是不能返回到 Oracle 7 状态的。为做到这点，运行命令 svrmgrl(UNIX) 或 svrmgr30(NT):

```
svrmgrl> connect internal
svrmgrl> startup nomount
svrmgrl> alter database convert;
svrmgrl> alter database open resetlogs;
```

7) 完成最终的整理工作，涉及到几件事，包括删除移植工具运行时创建的移植用户（用 cascade 选项）并运行 catalog.sql 和 catproc.sql 脚本。然而，有一个新的脚本，cat8000sql，当它被运行时，它为你完成所有整理工作。

移植工具的图形化界面与命令行版本的 mig 相对应，它能从 Oracle 安装程序（orainst）被访问和运行，并正好在移植工具被安装之后被执行。选择如下选项 “Install” “Migrate from Oracle7 to Oracle8” “Run Migration Utility” “Migration、Utility、Oracle7 to Oracle8.x.x.x”。然后你将被提示做一些选择，这些对应于命令行 mig 选项，如 check_only。如果你做出选择仅检查空间要求，那么与命令行版本一样，你必须重新运行工具以执行真正的移植（不再选择那个选项）。因此，用 GUI 取代命令行版本，你应完成任务 1 和 2，但没有完成任务 3，因为 GUI 将安装 Oracle 软件和移植数据。

9.2.3 移植前的任务

在你开始运行移植工具之前，必须计划和做一些重要的事情，以下所列步骤概括了移植前的检查列表。

- 查看新的 Oracle 8 特性。
- 选择你的移植方法。
- 估计硬件和软件需要的资源。
- 获得需要的资源。
- 开发可接受的准则（用它来测试）。
- 制定移植规划。
- 备份 Oracle 7 源数据库。

为了帮助你了解新的 Oracle 8 特性，阅读整个第三部分 “Oracle 8 RDBMS”。在这一部分

中,你能决定移植到 Oracle 8对你现在的安装是否是适当的。另一方面,当你决定移植时,还要考虑除新功能和增强之外的因素。其中最重要的一点是在逐渐进步的基础上 Oracle“不支持”Oracle旧版本的策略。例如,计划在2000年底不支持所有的 Oracle 7,特别是到1998年12月31日不再支持 Oracle 7.3.3和较早期的版本,而 Oracle 7.3.4是 Oracle 7最终的版本,计划到2000年12月31日不再被支持(但是最后的不再支持期限已经被改变了两次)。

当决定选择哪种移植方法时,考虑空间和时间需求是十分重要的。另外,考虑你的数据库的关键性和它所要求的正常运行时间(如果有的话)。如果你的数据库必须是高可用的,并且非常巨大,使用移植工具可能是最好的途径,特别是在实际处理移植产品之前,多次用于测试方式时。另一方面,如果你的数据库相对小或不是很关键的,导出/导入与 COPY/CTAS 都是很好的选择。在这种种情况下,使用移植工具的唯一原因,可能是为了熟悉工具自身,如果在移植路线中还需要移植其他的更大更关键的数据库,这也是一个原因。

从你选择的移植方法的当前任务中,回顾你需求的空间和时间,估计你的硬件和软件资源的需要。除了增加 Oracle 8系统表空间和二进制文件所需的存储空间(这是与移植方法无关的常量)之外,你将发现移植工具(内建)方法将需要最少量的额外量或进程、空间。这是真的,因为不需要中间的过渡媒介。同时,你将发现移植工具是最快速的方法,因为移植的时间与需要移植的数据库对象成正比,而不是与总的数据库大小成正比。最后,用移植工具方法,所有的事情中每件事物只需要一个:一台机器、一个服务器、一个实例、一个数据库等等。这是显然的,因为旧的数据库被“内建”移植成为新的数据库。

用导出/导入方法最明显的资源要求是对完全数据库导出文件需要相当大的总存储空间。但是你能将其直接导出到磁带中,至少在 UNIX系统中可以这样做,并且这将联机存储需要转换为脱机存储需要。用 COPY/CTAS方法,明智的存储空间需求是在新的数据库中的数据库对象至少需要与在旧的数据库中一样多的存储空间(实际上,由于新的 ROWID和其他因素,存储空间要更多一点)。对导出/导入和 COPY/CTAS这两种方法,它们几乎都需要每种两个事物;两个实例、两个数据库,甚至可能两台机器,例如如果是源数据库已经耗费几乎所有的机器资源。对 VLDB而言,相对于移植工具,导出/导入和 COPY/CTAS方法都是代价十分大的方法。

当你确定了你的资源需求,假定你选定了移植方法,获取这些资源。安装和配置第二台机器,镜像主机的环境、资源分配和尽可能接近的利用。如果有两台机器,一台机器明显地比另一台机器具有更强的能力和更多的资源,应使目标 Oracle 8实例形成在这台机器上,除非有不能不可行的或不能接受的原因。

9.2.4 接受的标准

接受的标准可以包括一系统测试,这些测试在表 9-2中描述。

假设选择了移植方法、大小、关键性和数据库的正常运行时间以及其他因素,决定实现移植的时间表。如果你的数据库较小或者是非关键的,那么可能在周末或甚至在工作日进行简单移植。但是,如果你的数据库是另一种极端,在你的企业中具有高度“可见性”,那么如果可能,考虑进行脱机移植。例如,如果你的数据库有一天 24小时一周7天不间断的正常运行时间要求,你怎样才能升级?如果你采取完全冷备份,并在另一台机器上重建(那就是,做一个克隆数据库),然后通过某一方式移植克隆体。但是如果你不能丢失从备份时到升级时过程中事务的活动,那么如果可能,考虑创建一个备用数据库或复制数据库,并升级第二个数据库,因

此不会丢失任何事务的活动。如果升级失败，你仅需重新克隆并更新你的第二个数据库。

表9-2 接受的标准

测 试	描 述
移植	移植源数据库的一个子集或者克隆整个数据库。子集能够很容易地用导出 / 导入方法测试，但是克隆数据库将需要测试移植工具，不需要某种类型的备份 / 重建
最少	在移植之后，测试目标数据库应用的最少功能
功能	在移植以后，测试目标数据库应用的全部功能。换句话说，遍历所有可能的功能性路径组合
集成	通过测试所有与目标数据库上核心应用发生交互作用的外部组件（接口），测试整个系统
性能	在移植之前，列出应用的所有性能基准，或者至少是关键的功能性组件的基准。移植发生之后，运行相同的功能性组件（基准），并测量同样的性能方面，将移植之前与之后进行比较，移植之后（Oracle 8）的执行至少应该与移植之前（Oracle 7）相同，可能会更好一些
装载	也称为容量测试或承受力测试。这种类型的测试可以与性能测试联合进行，通过改变用户数和/或数据容量执行测试。然后，比较移植之前和之后的性能，查看移植之后是否执行得与移植之前一样好或更好

在移植到 Oracle 8 之前，备份你的源 Oracle 7 数据库，其意义应该不言而喻的。然而由于许多 DBA 没有足够重视，而没有这样去做。但是如果你选择移植工具方法，结果失败了并且没有好的备份来恢复重建的话，你将处于最糟的境地。

9.3 常见问题

一个常见问题是资源限制：SYSTEM 表空间中空间不够。Oracle 8 需要的空间大约是 Oracle 7 空间总量的 1.5 倍，而且在 Oracle 8 的 \$ORACLE_HOME/bin 下将占有的空间大约是 Oracle 7 下的空间的 3 倍。

部分使用 ROWID 是另一个常见问题。参见第 14 章“Oracle 8i 附加主题”，这一章中说明，在 Oracle 7 应用中 ROWID 的部分使用将不能导入到 Oracle 8，除非使用新的 Oracle8 DBMS_ROWID 包重新写它们以适应新的 Oracle8 ROWID 结构。

移植使所有编码对象无效，所有表数据分布统计（对基于成本的优化器）丢失，有些索引可能被标为不可用。

GUI 移植工具（orainst）仅只使用 US7ASCII 字符集；与之相反，命令行版本可以指定任何字符集。

在 Oracle 7 只读表空间中的表的 ROWID 不能转化为新的 Oracle8 ROWID（因为它们只是只读的）。工作区需要临时地改变表空间到读 - 写（常规）状态，从这些表空间中每一个只读表 SELECT*（强迫 ROWID 转化为使用全表扫描），然后将它们改变回只读状态。不这么做，只读表空间中的表在每次读取时不得不经临时的 ROWID 转换。

9.4 移植之后的任务

除了明显的移植后的活动（备份新移植的 Oracle 8 数据库和测试你在移植前制定的接收标准）外，还有几个其他重要项目。这些包括（性能）调整你的 Oracle 8 数据库和安装非捆绑（可选）产品，如 Object Option。你所依赖的许多系统管理脚本必须进行修改，以帮助你监控和管理存储和增长，以及性能，有效地给出新的 Oracle 8 数据结构、对象和视图。如果没有其他原因，仅只是为利用 oracle 8 的某些新特性，你的 Oracle 7 应用将可能必须被修改。如前面所提到的，那些依靠部分 ROWID 操作的应用必须被重写。最后，如果 SQL*Net 连接字符串不是全部指向 Net 8，依赖它的应用将必须相应地修改。