



数据仓储和数据仓库

数据仓储和数据仓库

20年前人们首次创造了“数据仓储”这一术语。从那以后，Fortune 1000 家公司的主流活动都会涉及到这个词，因为在后来的十多年中这些公司的数据仓储的项目有很多。但是20年后的市场上，数据仓储仍然被人们误解。本节主要介绍了数据仓储的概念、给数据仓储初学者的建议、数据仓库的管理技巧、有关数据仓库设计问题以及如何进行数据仓库故障或灾难恢复。针对这些问题，数据仓储相关方面的专家进行了详细的解答。

数据仓储的概念

数据仓储开始并不产生许多个数据仓库，但经常发生这样的事情。栏目嘉宾 Rick Sherman 表示，主要问题就是数据仓储经常被人们误解。

20年前人们首次创造了“数据仓储”这一术语。从那以后，Fortune 1000 家公司的主流活动都会涉及到这个词，因为在后来的十多年中这些公司的数据仓储的项目有很多。但是20年后的市场上，数据仓储仍然被人们误解……。什么是数据仓储？它和数据仓库有什么区别？

❖ 人们真正理解数据仓储吗？

给数据仓储初学者的建议

针对 Oracle DBA 在开始学习过程需要从两个方面入手：数据仓储和 Oracle 数据仓库。前者需要了解什么是数据仓库、怎样建立数据仓库。专家建议你看看 Ralph Kimball 和 Bill Inmon 这两位作者的书，因为他们代表了两种截然不同的方法，但是这两种方法会在以后帮你找到最佳解决方案。另外，Oracle 数据库 10g 在数据仓储方面排在前 10 位的新特征包括 SQL Model 子句、SQL Access Adviser、多维 OLAP 引擎的完善、Tune MView Advisor 和 Query Rewrite 的完善等。

- ❖ 有关 Oracle DBA 的 DW 原理学习资料
- ❖ Oracle 数据库 10g 前 10 位的新数据仓储性能（一）
- ❖ Oracle 数据库 10g 前 10 位的新数据仓储性能（二）

数据仓库架构和数据仓库设计

在决定购买什么样的数据仓库硬件时有没有什么规则能遵循？你有没有在 DSS 工作负载中的 RAC 节点缩放相关方面的经验？对于任何 Oracle DBA 来说，一个最有挑战性的领域就是对复杂的 job stream 进行管理。幸运的是，Oracle 已经有 dbms_scheduler 包和 Oracle Warehouse Builder MetaBase Plus (OMB) 脚本语言可用来对它进行管理。

- ❖ 如何购买数据仓库硬件
- ❖ 在 DSS 工作负载中的 RAC 节点缩放
- ❖ 基于操作型数据模型的数据仓库
- ❖ 使用导出和导出方法复制到数据仓库

数据仓库管理

当我们设计数据仓库时，通常就是在操作型数据模型上模拟数据库。请问在创建基于操作型数据模型的数据仓库时有没有什么指导方针可以遵循吗？在给数据仓库硬件的定义方面并没有什么规则遵循，但是我们可以来看看硬件的组成元件，如磁盘、存储器、CPU、RAID 等。

- ❖ Oracle 数据仓储 DBA 的 MetaBase 脚本

数据仓库故障或灾难恢复

数据仓库已经成为紧急任务系统，它能帮助行政人员进行日常决策。在这个月的专栏里，BI 专家 Wayne Eckerson 就如何有效地保护你的数据仓库提出了具体建议。当你在进行故障恢复时，第一步就是识别你的业务程序。ETL 是处理你业务的唯一程序，例如你的产品可能就是将数据库里的信息。在识别所有的业务程序并存档之后，接下来的部分就是量化业务，

- ❖ 在数据仓库故障恢复方面给予帮助
- ❖ 数据仓库灾难恢复 101
- ❖ 数据仓储可能在灾难恢复热站点上进行吗？

人们真正理解数据仓储吗？

数据仓储开始并不产生许多个数据仓库，但经常发生这样的事情。栏目嘉宾 Rick Sherman 表示，主要问题就是数据仓储经常被人们误解。

20 年前人们首次创造了“数据仓储”这一术语。从那以后，Fortune 1000 家公司的主流活动都会涉及到这个词，因为在后来的十多年中这些公司的数据仓储的项目有很多。但是 20 年后的市场上，数据仓储仍然被人们误解。

Rick Sherman, Athena IT 解决方案

数据仓储的目标就是创造“真相版本之一”。但是在这些努力的过程中，许多公司常常就是用复制数据仓库代替。以下是一些典型的例子：

公司在数据仓储伞上安装一个多余的数据库——它们之中有一些是矛盾的或者互不相容的。2004 年的数据仓储研究调查表明，“平均每个公司有 2 个数据仓库、6 个独立的数据集市、4.5 个操作型数据存储以及 28.5 个 Spreadmarts。”

公司在 ERP 报告和商业智能 (BI) 方面仍然要从组织上以及技术上努力。尽管 ERP 厂商有数据仓储并且已经建立了数据仓储模块，他们也仍需要在上述方面做出努力。遗憾的是，这些模块通常是在公司已建立的公共的数据仓储模块上面单独安装的。

公司性能管理机构 (CPM) 给出了一个企业安装全面的、和谐一致的数据仓储的十分完美的理由。然而，CPM 程序经常让 BI 厂商先安装了一个解决方案，如同上述的 ERP 厂商将公司共有的数据仓库分开。这种方法产生了短期、快速的商业利润。在段时间内这种方法还可以，但是从长远来看，它仅仅是扩大了数据仓库和数据之间的矛盾。

公司数据仓储产生了更多的数据仓库并不是说数据仓储无从选择，它可能意味着人们完全不懂或者不能获得如何在公司范围内正确实行数据仓储的产品。

为什么在数据仓储的目标和公司实际安装的数据库的现实之间还存在着这么大的差距？大多数人都不清楚。

人们常把“数据仓库”和“数据仓储”两者混淆。数据仓储不仅仅包含数据仓库，它还包含一个完整的结构和步骤。数据仓储就是将数据转换成为信息，通过这种方式来检验它的操作和性能。这一任务将通过将数据源升级以及数据转换来完成并获取、分析信息。数据存储的时间将会持久（在磁盘上保存）或者短暂（用磁盘或存储器）。此外，工作流程通常包括多个数据存储到支持升级和数据转换到信息，如操作型数据存储、数据仓库、数据集市、在线分析处理体、文档如 flat file（用都好隔开的函数值）XML 数据和电子表。

只要你从操作系统中获取数据并执行报告和分析的任务，那你就正在实施数据仓储的过程。过去人们把它叫做决策支持，现在的术语就是商业智能。数据仓储的对象就是 BI 的曲线图和枢轴之下的数据。事实上，BI 仅仅是数据仓储架构上的表述层。

数据仓储通常只和数据仓库而不是整个架构和步骤联系在一起。数据质量、连贯性和完整性——更不用说能审计数据——只有在考虑到整个数据升级（或数据仓储）架构时才可能实现。

数据仓储程序比数据仓库要多得多。稍微看一下 DW，单独的 ERP、CPM 以及 BI 重建了 DW 的架构以及他们自己的数据仓库。从观念上说来，架构全景对相互之间的工作、重新使用的工具、密码、步骤、数据和标准起到了平衡作用。一个公司能够以更高的 ROI、更低的全面操作和维护费用，很经济地安装这些系统、但是他们仍然要为“真相的唯一版本”而努力。

（作者：Rick Sherman 译者：April 来源：TT 中国）

有关 Oracle DBA 的 DW 原理学习资料

问：我是一名 Oracle DBA。我想开始学习 DW 原理和其他的 DW 的其他资料。我还想紧跟这个行业变化形势。你能给我们相关书籍和资料的建议吗？

答：将 DBA 的工具包的数据仓储加起来是一个很不错的主意。使用数据仓库的观念和方法很少用于其他环境，但实际应该这样做。

开始学习过程你需要从两个方面入手：数据仓储和 Oracle 数据仓库。前者需要了解什么是数据仓库、怎样建立数据仓库。我建议你看看 Ralph Kimball 和 Bill Inmon 这两位作者的书，因为我觉得他们撰写的书籍最能帮助人们学习和最终设计以及安装数据仓库。他们的却代表了两种截然不同的方法但是要知道这两种方法会在以后帮你找到最佳解决方案。至于 Oracle 数据仓库相关知识，我建议你看看《Oracle 数据仓储指南》。我清楚它是文档，但是它还讨论了一些能作为起点的特征。

我还要建议你加入数据仓库的组群。他们有很多资料能够扩大你的知识库，如文章和会议等等，所以我建议你加入数据仓储协会或者其他组群。

希望这些有助于你开始学习。

(作者: Ian Abramson 译者: April 来源: TT 中国)

Oracle 数据库 10g：数据仓储前 10 位的新特征（一）

1、SQL Model 子句

也许现在最受关注的就是 Oracle 数据库 10g 新数据仓储特征中的 SQL Model 子句。该子句让用户将类似数据表模型插入 select 语句中，以前这在某种程度上属于的 OLAP 服务器关注的范围，如 Oracle Express 以及 Oracle 9i OLAP。SQL Model 子句给 Oracle 分析性的查询一个全新的标准并且说明了 SQL 一般工作模式的一些缺陷。

SQL Model 子句是为了说明以下的情况：在过去，客户先从关系型数据库中取出数据、然后将它们导入数据表如 Microsoft Excel 中。通常，这些模式涉及一系列宏，这些宏里的数据集合包括一系列的商业尺度、时间变更周期以及一套复杂的、很难用平常的 SQL 表示的商务尺度。我设法说服一些客户 SQL 的局限就是利用一系列独立的 Excel 表格，而这些给客户提供了他们所需要的分析能力、复制数据的可测量性和可靠性、然而不久缺乏对全局的控制能力也显露出来。

SQL Model 子句让一般的 SQL 语句能够在 select 语句的结果中创建多维数组，在这一数组上执行一些行间和列间的计算，然后对包含模式结果的数据表格进行升级。使用 MODEL 子句的 SQL 示例如下：

```
select SUBSTR(country, 1, 20) country, SUBSTR(prod, 1, 15) prod, year, sales
FROM sales_view
where country IN ('Italy', 'Japan')
MODEL RETURN updateD ROWS
PARTITION BY (country)
DIMENSION BY (prod, year)
MEASURES (sale sales)
```

```
RULES  (
    sales[' Bounce' , 2002] = sales[' Bounce' , 2001] +
        sales[' Bounce' , 2000],
    sales[' Y Box' , 2002] = sales[' Y Box' , 2001],
    sales[' 2_Products' , 2002] = sales[' Bounce' , 2002] + sales[' Y Box' , 2
002])
ORDER BY country, prod, year;
```

2. SQL Access Adviser

Oracle 数据库 10g Server Manageability 的部分特征就是，SQLAccess Adviser 给指定的工作量介绍了最佳索引和物化视图的结合。可行方案就是（通过 DBMS_ADVISOR 包）或者通过新推出的基于 Web 的 Oracle Enterprise Manager 中的 Advisor Central 元件，SQLAccess Adviser 最初就和 Oracle9i 一起建立在索引和概要 advisor 并且为调优和总结存储数据提供了一站式的服务。

更多有关 SQLAccess Adviser 的信息可以在 OracleWorld 这一白皮书上找到。

3、多维 OLAP 引擎的完善

利用 Oracle9i，起初单独的 Express 多维 engine 现在就合和 Oracle 数据库连成一体了。通过 Oracle10g，传统的关系型 Oracle engine 就开始明显起来。

首先就是对较大的分析型 workspaces 进行划分、目前在 Oracle OLAP 世界引进了一些高级分区选项，它们备受 Oracle 数据库用户的欢迎。现在分析型 workspaces 储存在 Oracle schema 的 AW\$ 表中，这一分析型 workspaces 能够用 AW\$ 表再进行细分并允许你将单独的 analytic workspace 列入 10GB 字段，每个表的行中都包含着一个部分。这些表还可以像其他 Oracle 表一样进行细分，你也可以将行放入每个表空间。再者，每个表空间当然储存在不同的物理磁盘单元的数据文档里。虽然这样做很有益，细分分段大小也就是细分数据的唯一方法，你也不能在 analytic workspace 对每个分区进行详细说明。

Oracle 10g OLAP 现在还详细说明在 analytic workspace 里的哪一个对象划分进哪一个区域，如果对象相当大还能通过分段大小再进行细分。

分析空间的变化现在能够被区分开来，通过一些尺度或 CONCAT 尺度参考。10g 的多维 engine 将每个变量部分当作单独的物理对象储存，这样就可以直接在 AW\$ 表里区分行（允许你越过不同的表空间和物理磁盘驱动器）；而变量则对于应用软件、简化数据模型以及允许 Oracle 执行所有高难度的任务来说仅仅只是一个对象。

另外一个杰出的新特征并在含 Express 的参量真正有所提高的是对多用户读写访问个人分析工作空间的支持。过去，Express 的缺点之一就是只有一个用户可以在读写模式里访问 Express 数据库，这样就导致了 Express 开发人员开发了一整套可以选择的解决方案允许写入 Express 数据库。在 Oracle 10g OLAP 中，分析性工作空间可以附加在 MULTI 模式里。然后应用程序 ACQUIRE 单个变量附在用于读写的分析工作区间中。一旦获得一个对象（被 Oracle 多维 engine 锁定），就可以进行更新了，并且应用程序随时都可以进行修改。之后 update 命令的执行结果跟变量相反，该命令后跟有 COMMIT。RELEASE 命令的执行结果也和变量相反，这样做就是为了使其它的应用程序能够被写入。

观察多维引擎如何处理 multi-write access 很有趣。过去当用户读写口连接数据库时，带 Express 的数据库空间就能增大。其他的数据库在 read mode 中访问它，这是因为 Express 要为每个用户复制数据库以便让他们对待数据的意见一致。如果单个变量从 10g 的 analytic workspace 复制到临时的 workspace 中，这样的更新我一点都不觉得奇怪。随着更新过程往后推移（旧版本的 Express Excel Add-In），最后发生变化时——主要问题就是如何数据库大小，因为旧版的 Express 的处理方法比理想状态下的要少

Oracle10g OLAP 改善了集合体，并且还允许现在一些规则成为 AGGREGATE 命令的数据源，这样就排除了计算和存储数据的必要，尤其是动态集合又是另外一个领域，在这里 Oracle9i 和现在的 10g OLAP 都得到了显著的改善，超过了 Express。如果现存的 Express 系统还存在着这个问题，那么它还是值得我们更加仔细观察的。

更多有关 Oracle10g OLAP 型特征的详细信息在《OLAP 应用开发指南》中能够找到，《OLAP DML 指南》以及 Bud Endress 写的白皮书 OLAP Option to the Oracle10g Database 也能找到相关信息.。

4、Tune MView Advisor 和 Query Rewrite 的完善

Query Rewrite(Oracle 从具体层级到总表的直接的查询能力)是 Oracle8i 和 9i 数据仓储的最佳特征之一，但是这些特点不稳定，你通常会发现这些查询不能被重写。有时是因为你打破了 Query Rewrite 的某一项限制，有时是因为你的实体化视图没有包含正确的 column 和集合。Oracle10g 对 Query Rewrite 作了一系列的改善并且实体化视图调优过程应该让这一改善过程的收获更加丰富。有了 Oracle 数据库 10g，当你的 select 语句含有分析功能时，query rewrite 现在就可以实现了，快速联结和设置操作系统，如 union、MINUS 和 INTERSECT。此外，你现在能用到一个 hint: /*+ REWRITE_OR_ERROR */，它在 query rewrite 不发生的情况下将会终止 SQL 语句。

```
SQL> select /*+ REWRITE_OR_ERROR */
  2    s.prod_id,
  3    sum(s.quantity_sold)
  4  FROM      sales s
  5 GROUP BY  s.prod_id;
FROM      sales s
```

第四行的错误:ORA-30393: 该语句中的查询块不能重写

Oracle9i 自带两个包: DBMS_MVIEW.EXPLAIN_MVIEW 和 DBMS_MVIEW.EXPLAIN_REWRITE，这两个包可用来诊断为什么实体化视图不能用于 query rewrite。然而，尽管这些包已经告诉了你为什么不会产生 rewrite，它们仍然需要你解决如何更改你的 create MATERIALIZED VIEW 语句确保正确产生 rewrite。Oracle 数据库 10g

还带有一个新的 advisor 包: DBMS_ADVISOR.TUNE_MVIEW, 这个包输入了一个 create MATERIALIZED VIEW DML 语句后, 结果就得出了一个支持 query rewrite 和快速更新的特征。

更多有关 Query Rewrite 和 Tune MView Adviser 功能改善的信息和能在 TUNE_MVIEW 在线文件上找到, 即《Query Rewrite 在线文件》和《Oracle 数据库 10g Oracle-by-Example 指南》。

5、数据泵：导入和导出的代替者

数据泵代替了旧版 IMP 和 EXP 应用程序, 它用于创建 Oracle 表、schemas 以及数据库的逻辑备份。数据泵属于服务器应用程序 (与 IMP 以及 EXP 相反, 二者都属于客户应用程序), 在 Beta 测试数据导出时的速度是旧版 EXP 的两倍, 数据导入时是旧版 IMP 的 10 倍。数据泵是可以随时获取, 通过 DBMS_DATAPUMP 或通过 IMP 和 EXP (又叫 IMPDB 和 EXPDB) 还可以通过 Oracle Enterprise Manager 10g. 的一部分 wizard delivered 来获取。

数据泵 (和新版的 IMPDB 以及 EXPDB 应用程序) 在过去的 IMPORT 和 EXPORT 上进行了很大的改善, 包括可恢复/可重新开始的任务、automatic two-level parallelism、使用 DBLINK/listener 服务名称代替 pipes 的网络模式 (所以能选择独立的表、视图、包、索引等等进行导入或者导出而不仅仅是表或者含有 IMPORT 和 EXPORT 的 schema), API 允许第三方 ETL 包带有数据泵功能。

(作者: Mark Rittman 译者: April 来源: TT 中国)

Oracle 数据库 10g：数据仓储前 10 位的新特征（二）

6、对存储管理的改善

自动存储管理（ASM）是 Oracle10g “最酷的新特征”之一，它能减少 Oracle 数据库管理员的工作量。自动存储管理完全使创建逻辑 volume、文件系统以及文件名过程自动化。DBA 只需要规定原始磁盘的位置，接下来自动存储管理就会完成其他事情。磁盘的 I/O 通过块来均匀分布数据，另外 ASM 将处理 disk 镜像并创建镜像群或故障群。

ASM 能处理由快速扩大数据仓库引起的问题，管理员不再处理吸盘单元、结点以及逻辑族群里的 sheer 数字，它也是 Oracle10g 的网格架构的主要特征之一，目的就是为了“虚拟”计算机能力以及目前数据库的特征如处理和存储对自身的有效管理。

更多有关自动存储管理的信息能够在 Lannes Morris-Murphy 撰写的 OTN 这篇文章中找到。

7、快速全表扫描

全表扫描在数据仓储环境中很常见，认识到了这一点， Oracle10g 中的表格扫描性能得到了改善。Oracle 数据库 10g 中的代码优化减少了对 CPU 的占有空间，并且还实行了快速全表扫描（queries 受 CPU 的限制而不受 I/O 的限制）。和受 CPU 限制的 query 比起来速度提高了 30-40%。

8、自动调优和维护

自动维护和调优通常是区别 Microsoft SQL Server 和 Oracle10g 的主要产品之一，能够满足并和其他产品竞争的特征包含在服务器技术里。调查表明 DBA50%以上的时间都用于调优和检测数据库服务器，很重要的任务通常很复杂并且很难做到相当准确。有 Oracle

数据库 10g 的帮助, Oracle 引进了一系列的构成成分, 这些元素在一起使数据库服务器能够检测自己、使配置发生智能变化并在需要进行手工操作的情况下提醒 DBA。

在这个范围内首要条件就是 Automatic Workload Repository, 它用了一个加强版的 Statspack, 每 30 分钟收集一次实例 statistics, 保存周期为七天, 七天后自动更新一次。这种增强版的 Statspack 现在收集了大量的 statistics 和 optimizations, 简化了获取高价 SQL 语句的方法, 确保在上一个快照被收集之后只有 SQL 行为在才能在很大程度上影响其性能。保存在 Automatic Workload Repository 里的用法信息就是 Oracle 数据库 10g 所有自我管理功能的基础。

其次就是 Automatic Maintenance Tasks 这一特征, 作用于 Automatic Workload Repository 集聚的 statistics 的基础之上并执行例如索引重建、更新 statistic 等任务。在执行这些任务时不需要管理员进行手动操作。计划特征如 Unified Scheduler 在预定义维护窗口中执行这些命令, 将默认值设置为第二天的 10: 00pm 和 6: 00am, 虽然这些时间能设定以便减少对其他任务的影响 (如批量下载)。

第三个自我管理框架范围的条件就是服务器告警 (Server Generated Alerts), 这是一种数据库服务器通过电子邮件将通知发送给管理员的方法——包括关于如何处理这种情况。如果数据库自己不能处理这些例如在磁盘单元里有充足的空间扩展数据文件的情况, 通常就会发出警告。

最后, 也是自我管理所有因素框架范围最重要的因素, 就是自动数据库诊断监视器 (Automatic Database Diagnostic Monitor)。它主要分析在 Automatic Workload Repository 中获取的数据并且在使用了一种人工智能运算法则, 和 Oracle Expert 中找到的一样, 都用于分析以下的区域: 对锁的争议、CPU 瓶颈、I/O 使用和争议以及监测点的问题等等。和管理员目前通过分析 statspack 报告进行的操作方法几乎一样。

9、异步变化数据捕捉

Oracle 变化数据捕捉 (Change Data Capture, CDC) 和 Oracle9i 是同时引进的，它为表提供了记录变化的能力并将他们保存在变化表中供 ETL 使用。Oracle9i 变化捕捉通过在元数据表上创建触发器同时迁移数据但却创建了一个处理程序，还要求访问元数据表结构。由于触发器对 underlying tables 的影响，许多数据仓库计划在通常在没有变化数据捕捉的情况下进行或者使用其他方法进行数据捕捉。

Oracle10g 引进了异步变化数据捕捉，通过触发器使用数据记录文件捕捉变化并将它们应用于集合表。异步变化数据捕捉因此不需要变换表结构，也不影响数据库性能。

更多关于异步变化数据捕捉的详细信息在 OTN 上能够找到。

10、Oracle 数据挖掘的完善

除了包含 Oracle Express 多维 OLAP 引擎外，Oracle9i 还包含了数据库数据挖掘功能，这一数据挖掘功能通过 Oracle 数据库 10g 增强。Oracle10g 增加了对两种新类型程序的支持，Support Vector Machine（用于从上至下而不是从下至上的计算）和 Non-Negative Matrix Factorisation 以及对 Frequent Itemsets 的支持，用于市场购物篮分析 (market basket analysis) 以及潜在分析 (propensity analysis)。

(作者: Mark Rittman 译者: April 来源: TT 中国)

(插入分页符，空一行)

如何购买数据仓库的硬件

问：在决定购买怎样的数据仓库硬件时有没有什么规则能遵循？我们想写出多个集合点、控制器以及 RAID ARRAY 的必要条件。可是我们已经为寻找建立数据仓库硬件的指南花了很多时间。

答：很遗憾告诉你，在给数据仓库硬件的定义方面并没有什么规则遵循。我们大致来看一下下面的确定硬件的组成元件：

1、磁盘

必要数据保存（表、索引、临时表空间等）

数据束落区（反向文件和文档）

必要软件（数据库、ETL 和 BI 软件）

BI 软件空间必要条件

2、存储器

必要数据库元件（数据库工作存储器和动态存储器）

ETL 工具的必要条件

必要的 BI 工具

3、CPU

取决于数据库大小和安装方法

ETL 能够使用多个 CPU

4、RAID

和 RAID 方法比较性能要求

一般由于受性能的影响我们会避开 RAID

5、Growthz——确保硬件可以扩大

假设等比级数能够超出预想范围

我知道这些都是基本因素，但是在最后，你会知道对硬件性能的要求以及建立在推断之上的你目前的硬件结构，你会知道怎样利用硬件。购买硬件的关键就是它们能够扩展。在前面你可能会发现一些似乎正确的规则，但它并不是长期都适用。所以并没有规则可循，但是你应该能够根据你的环境决定购买什么样的硬件，这样不仅满足你当初的愿望，而且还能为你以后做准备。

(作者: Ian Abramson 译者: April 来源: TT 中国)

在 DSS 工作负载中的 RAC 节点缩放

问：你有没有在 DSS 工作负载中的 RAC 节点缩放相关方面的经验？同样，是不是在缩放过程中主要瓶颈就是架空的 cachefusion 锁管理协议，而不是实际的互连？

答：如果不是那样的话，我们假设你说的数据仓储支持的是群体决策支持系统（DSS），而我支持 RAC，有很多原因能够说明 DSS 不是最好的数据仓库架构。

数据仓库通常依赖 Oracle 并行查询选项（PQO）对大型表、报表或物化视图进行快速扫描。PQO 的功能就是和有用的 CPU 紧密结合，因为 RAC 经常被用来作为水平刻度而不是垂直（更多的机器与更多的 CPU）。和对称式多处理服务器比较起来在 RAC 环境中依靠并行查询选项通常能降低性能，因为在 RAC 环境中每个节点必须将数据迁回至 PQ 协调程序，这一程序能产生在一台服务器上易于解决的比较大的瓶颈并且经常会在水平单一的服务器上依靠一类型的 RAM 添加较大而且典型的系统操作，这样一来 RAC 数据仓库的问题就越来越大了。

当然，服务器有一些限度，除了数据库的属性之外，系统还要求有一些集群支持一些大型用户团体，加上这些问题的能够用非常高级的 proactive tuning 减轻。但也有例外。在这里我们讨论一些常见规则。我不会用 RAC 说明数据仓库的性能问题。

首先我会将 RAC 看作是有效解决方案，其次就是将它看作是大型 OLTP 数据库。考虑到数据仓库性能解决方案，我会介绍一种很复杂的情况，在这种情况下系统允许操作一些有益的解决方法，但是不能通过一台单独的服务器来解决。

我希望对你有所帮助！

（作者：Bill Cullen 译者：April 来源：TT 中国）

基于操作型数据模型的数据仓库

问：当我们设计数据仓库时，通常就是在操作型数据模型上模拟数据库。请问在创建基于操作型数据模型的数据仓库时有没有什么指导方针可以遵循吗？

答：数据仓库环境应该有很多层级支持它执行多个 Roles。操作型支持系统将数据导入数据库，所以审计层应该在非常接近操作支持系统进行建模，在这里可以使用 Application Triage 技术。一定要在这一层面上改变数据捕捉并记录这一过程，因此你需要适当地在这些表中增加元数据。

这些基层/历史层应在反映业务范围建模。在设计中体现主题定位、完整性、3NF，还使用一些 Data Vault 的建模因素将历史栏记录和秘密或安全栏分开。这里可以用到 Table Triage 技术。接触到表格就获取表格。

用户能访问的层级应该在反映业务客户如何使用信息的基础上建模、一些空间模型支持功能和性能，这也正是需要分析工具（记分卡片、dashboard、loading cube 等）的地方。还有那些 flat “profile”类型数据支持下载数据挖掘工具，如 SAS 企业工具。用户需要的方法加入必要的数据元素支持存档的信息传送要求。

所以最简单的回答就是“yes”，操作型模型对于数据仓库里的一些数据架构层面很适用。

(作者: Mike Lampa 译者: April 来源: TT 中国)

使用导出和导入方法复制到数据仓库

问：我现在为一所大学工作，我们已经为存档数据建立了数据仓库和两个数据集市。目前我们在使用导入和导出程序将联机系统复制到我们的测试数据仓库。下面是有系统信息：数据库是 Oracle9i、操作系统是 Sun Solaris 上运行的 Unix。2007 年将会换成 ODS。

我们想决定是否每晚都该进行增量更新和插入而不是复制产品系统。我知道增量表更新速度要快得多，但是有没有对该程序持反对意见的呢？有没有可能让文件脱离 sync？

你使用导出/导入方法更新你的数据仓库吗？

你每天都执行增量更新吗？

更新你的数据仓库/数据集市需要多长时间？

你创建任何概要数据/物化示图了吗？

进行备份需要多长的时间？

如何进行注册？

增长率如何？它怎样影响你的系统？

你是在什么平台上运行的？

谢谢。

答：哇，这是一个很长的问题，涉及到架构决策类型的多个方面。通常在创立和维护数据仓库时，我们会考虑到这些问题还有是否将导入/导出作为复制方法的基本问题。

我们从把导入和导出当作复制方法说起吧。首先我可以告诉你那样做可行，和你的目标是相符的。然而，在不知道你的数据库有多大的情况下，这种方法不可取。考虑到一个仓库有一到两个千兆字节，那么导入/导出方法就不可取。我们使用的方法取决于必备的条件。例如，一个企业的数据仓库，我们就用一个系统层级上的方法。我们使用 Oracle 便携式的表空间或者使用 EMC 快速拷贝磁盘。这两种方法都行之有效，因为它们能减少对数据库的影响，但是更适合更大型的数据仓库。另外一种方法就是复制，但我一般不会将它用在 test 数据库上。因为你表示你关注记录更新，我对此表示理解，现在你需要建立一个 ETL 架构用于更新你的 test 数据库。通常我会认为我们没有必要复制。

至于我运行数据仓库的类型，目前有一种将多个业务流程融合在一起的企业仓库。仓库是目前有多个 TB，并在过去已看到显着的增长，然而它还属于有 EMC 的阵列的惠普 12-way Itanium。我们通过 Informatica 执行了所有的下载程序，所以我们用直接的数据库访问和批量下载的方法，这取决于我们需要处理的事情。我们所有的处理都是三角形式的处理。从原始资料中我们了解到了数据变化，然后将它们传送给数据仓库，现在我们的下载窗口根据日常工作量大约要花 5-8 个小时下载。至于备份，我没有时间花在它上面因为我已经把它留给操作系统了。至于更新开发和 test 数据库，首先我们进行导入/导出，这一过程要花费三天的时间。然后将它移入简单的 Informatica 解决方案，这一过程也需要 12 小时。

我希望这能对你有所帮助。如果你还有其他问题，请你告诉我，很高兴为您解答。

你有没有在 DSS 工作负载中的 RAC 节点缩放相关方面的经验？同样，是不是在缩放过程中主要瓶颈就是架空的 cachefusion 锁管理协议，而不是实际的互连？

(作者: Ian Abramson 译者: April 来源: TT 中国)

Oracle 数据仓储 DBA 的 MetaBase 脚本

对于任何 Oracle DBA 来说，一个最有挑战性的领域就是对复杂的 job stream 进行管理。幸运的是，Oracle 已经有 dbms_scheduler 包和 Oracle Warehouse Builder MetaBase Plus (OMB) 脚本语言可用来对它进行管理。

就 Oracle 数据仓库管理而言，Oracle Warehouse Builder 产品提供一种能够整合抽取、转换和装载等命令的 OMB 脚本，这种整合要依靠能进行错误检测、确认以及数据配置的内部工具才能进行。对 OWB 环境来说，它就是一种类似 TP 的检测器 (CICS)。

一方面 Oracle 公司在 Oracle 10g Enterprise Manager 图形用户界面 (GUI) 方面取得重大进展，另一方面一些高级 Oracle 专家仍然觉得有必要利用更加活跃的脚本环境。一个图形用户界面不能完成所有的任务，并且有些高级操作需要更加灵活、精确的工作控制逻辑：

预定执行——工作流在预先确定的时间、先决条件下完成。确保“遗漏”工作都是预先确定的。

条件执行——在已完成任务情况的基础上执行新的任务。

对错误警告和工作流的确认——Oracle DBA 需要确定所有可能出错的范围并创建用户退出窗口终止执行直到解决所有重要事情。

Oracle 数据仓库专家在管理 ETL 时有两种方法：

亲自动手 (DIY) ——如果 Oracle DBA 已经懂得怎样创建 batch stream、检测错误和进行有条件执行，那么写 ETL 就是一种方法。Oracle10g dbms_scheduler 包能够用来安排 job-stream，或用习惯做法：OS shell 脚本。DIY 方法更加灵活，这样你就可以选择最佳的处理方法和技术，而不是依靠 OWB 提供的工具。Tim Hall 在这方面撰写了一本很好的

书，书名为 Oracle Job Scheduling，它是在 Oracle 环境中执行复杂工作任务的最全面的参考书。

使用 OracleMetaBase 脚本——OMB 是 Tcl 语言的扩展（全称为 tickle），它具有下列强有力的特性：程序设计语言，包括复杂的布尔运算符、变量支持和 loop 循环（FOR, WHILE, REPEAT-UNTIL）。Oracle 数据仓库 Tcl 扩展是为 OMB 定做的，所以进行复杂的数 据配置以及规划 job stream。跟 Java 一样，OMB 有独立的平台而且在 Mainframe 或者 Macintosh 上运行 MetaBase 时不需要做任何的改变。

当然，MetaBase 需要你额外购买 Oracle Warehouse Builder 来处理 ETL，但是 OMB 的脚本语言是一种具有吸引力、并能代替“亲自动手”的方法，这种方法取决于你 ETL 处理的经验水平。

内部 OMB 语句

让我们更仔细地观察 OMB 环境。OMB 跟 SQL*Plus 很相似，它可以调用内部命令目录来处理 OMB 数据映射。这些数据映射能嵌入（一个层级结构），并且 OMB 给出命令允许你从 Tcl 程序中调用这些映射。以下是两种 OMB 命令：

元数据定义语言（MDL）——如 DDL 一样，通过它你能创建和下拉 OMB 对象。MDL 对象类型包括项目、模块、表、映射以及工作流程步骤。

OMBalter——用来为 Warehouse Builder 元件改变元数据。在下列的示例中，要注意类似 UNIX 反斜线符号命令延续字符：

```
ombalter mapping 'MAP_NAME' modify operator 'join'  
set properties join_condition)  
values (' fact.empno = emp.empno)
```

OMBcreate——用这一命令在 OWB 中创建一个元件，也许是一个项目、模块、表、映射或者工作流程步骤。

OMBdrop——用来下拉存放中的命名对象。

元数据维护语言 (MML) ——类似 Oracle DML，MML 命令允许我们改变（增加、更新、删除）命名对象。一些 MML 命令还包括：

OMBCC——这是“更改内容”命令，其用法和 UNIX 以及 DOS “cd”命令一样，都是用于更改 OWB 目录。

OMBCOMMIT——负责业务办理。

OMBCOPY——这是一个手动命令，用于复制和该命令同以类型的一个或者多个对象。

```
ombcopy 'AN_EXTSTING_MAP' to 'A_NW_MAP' replacement
```

OMBDEPLOY——它通过 OWB 运算时间表给具体的数据库部署一个具体的对象。

OMBLIST——OMBLIST 命令包含在目录命令内，该目录在具体的子目录结构中列出了所有 OWB 对象。

OMBRECONCILE——用于协调 OWB 元数据定义和数据库元数据定义。

ODTUG 文件明确提出“用 Oracle 的 MetaBase Plus 语言创建和部署映射和工作流”，使用 Oracle 的 MetaBase 对传统工作流应用软件来说确实是个很不错的选择，下面是 MetaBase 范例：下面的例子同样说明部署 OWB 流程指向 Oracle Workflow 引擎。首先，它连接存储库，改变了 STUDENT_YEAR_4 模块内容。然后开始用一个特殊的命名模式找回所有的工作流包，内部 loop 循环还能对该部署什么样的工作流进行控制。

```
proc owb_deploy_workflows_all {} {  
    OMBCONNECT OWB_DEV/XXX@SERVER:1521:DBPROD  
    OMBCC ' STUDENT_YEAR_4'  
    OMBCONNECT RUNTIME ' SASI_CURR_RT' USE PASSWORD ' XXX'  
    set ModList [ OMBLIST PROCESS_FLOW_MODULES ' CURR_W.*. ' ]  
    set i 1  
    foreach ModName $ModList {  
        puts "Accessing WF package: $ModName"  
        OMBCC '$ModName'  
        set mapList [ OMBLIST PROCESS_FLOW_PACKAGES ' LD_ACNR.*. ' ]  
        set J 1  
        foreach mapName $mapList {  
            puts " deploying: $mapName"  
            OMBcreate TRANSIENT DEPLOYMENT_ACTION_PLAN  
            ' DEPLOY_PLAN' ADD ACTION ' WORK_FLOW_DEPLOY'  
            SET PROPERTIES (OPERATION)  
            VALUES (' create' ) SET REFERENCE PROCESS_FLOW_PACKAGE  
            ' $mapName'  
            OMBDEPLOY DEPLOYMENT_ACTION_PLAN ' DEPLOY_PLAN'  
            OMBdrop DEPLOYMENT_ACTION_PLAN ' DEPLOY_PLAN'  
            OMBCOMMIT  
            incr J  
        }  
        OMBCC ' ..'  
    }  
}
```

小结

Oracle 数据仓库 DBA 能否像选择 ETL 脚本语言一样选择 OMB 脚本？由于 OWB 通常带有数据仓库许可证，因此它一般都是免费工具，任何数据仓库 DBA 都想希望能够在非 GUI 环境中处理 OWB。唯一的缺点就是需要学习 Tcl 语言。

(作者: Donald K. Burleson 译者: April 来源: TT 中国)

在数据仓库故障恢复方面给予帮助

问：我目前在为 Illinois State Police 工作，我们现在正安装一个大型的（州际的）罪犯审判系统。该系统的一部分就是一个数据仓库（用 Oracle10g, OWB 建）。我们需要有人在业务持续性与故障恢复方面给与帮助。但是我们有一个问题：你何时在故障里运用 ETL？你能向我们推荐其他的免费资源吗？

答：当你在进行故障恢复时，第一步就是识别你的业务程序。ETL 是处理你业务的唯一程序，例如你的产品可能就是将数据库里的信息。在识别所有的业务程序并存档之后，接下来的部分就是量化业务，例如像 Amazon.com 这样的公司如果客户不能下订单、公司不能完成订单计划就会损失惨重，这些程序（下订单和完成订单）对他们来说是很热门的话题，而数据挖掘客户订单模版却不是那么热门。在故障里，接收和完成订单比了解客户的习惯要重要得多。这些完成后，有人可能在这些过程的某些地方划上一条线，线上面的所有程序需要进行持续性和故障恢复，现下面的程序在你储存这些必要的过程以后再进行处理，而且有人必须在这里划一条公开的线。公司还需要支付额外的资金确保操作的持续性，当然公司只需要支付那些在故障恢复后必须的一些程序。

(作者: Brian Peasland 译者: April 来源: TT 中国)

数据仓库灾难恢复 101

数据仓库已经成为紧急任务系统，它能帮助行政人员进行日常决策。在这个月的专栏里，BI 专家 Wayne Eckerson 就如何有效地保护你的数据仓库提出了具体建议。

你最后一次回顾和测试数据仓库灾难恢复计划是在什么时候？你有数据仓库灾难恢复计划吗？虽然自然和地理政治概念上的灾难——包括龙卷风、雷暴天气以及油价上涨——正逐步增多，他们还不是你商业智能环境（BI）的最大的威胁。据 Information Age 调查，英国某主要的准们为主管编辑的杂志表示大多数的 IT 行政人员都相信对 IT 操作持续性的造成最大威胁是内部系统故障（65%）和病毒（45%）。同时，自然灾难占 32% 能量和运输损耗占 33%。

时代变了

十年前根本就没有必要为数据仓库制定恢复计划、报表和它们支持的应用程序。当时，绝大多数数据仓库每月都被装在半打或者开源系统上。大多数负载量都很小，即使是最大的数据仓库都比一些兆字节大小要小得多。所以大多数数据仓储团队都没有灾难恢复计划一点都不感到奇怪，更不用说备份策略了。一般来说如果数据仓库被损坏，你可以简单对它整体做一下更新。

今天，大多数数据仓库已经成为了紧急任务执行系统。一些数据仓库在实时的基础上捕捉业务并对它进行了更新，并且这些数据仓库支持许多业务运营应用程序。如果系统脱机几个小时的话。业务运营程序用户就对制定日常决策的数据仓储信息非常依赖所以这些程序对数据仓储主管的要求也特别苛刻。再者，作为决策程序，数据仓库可以在灾难发生时帮助企业或机构作出最理想的反应。数据仓库报表能够帮助行政人员解决如何区分活动顺序、进行资源配置以及分配人员处理紧急事务。

你的公司将操作所需的业务过程和应用程序区分开了么？如果数据仓库应予以有先考虑的话，那么数据抽取、转换和装载过程（ETL）呢？ETL 可以组装数据仓库以及绘制、分发报表的 BI 服务器。这条链很脆弱，数据仓库环境很复杂，包含多个系统和应用程序、并且和内部、外部系统相互依赖，所以数据仓库的环境不能在它所有元件都联机之前完全恢复。

你最后一次真正测试数据仓库恢复计划是什么时候？如果你在数据库故障中测试恢复，那你还只完成了一部分。你还需要恢复客户、服务器、网络、存储、应用程序和数据库，对恢复环境进行完全模拟。如果你是在一年之前进行的测试，那这就是测试你的恢复计划是否过期的好机会。由于数据仓库是个能更新的系统，所以它会经常回答业务人员提出的新问题。因此查询、报表、元数据、ETL workflow 等也许从你上次测试开始就已经改变了。再次，业务人员在这种紧急情况下问的问题和他们在平时问得问题可能会完全不一样。

快速恢复的关键不仅仅是灵活多余的系统的，还有人。在灾难发生的过程中一定很混乱。许多主要的人员也许不在或不能登录操作系统。因此你不仅需要系统的冗余量而且还需要进行分配。你的团队应该锻炼具备适应各种紧急情况的能力并且还要准备扮演多个角色。

准备

灾难恢复重视质量好、最新的、端到端的元数据，一些公司已经成功安装了这些数据。元数据至关重要的，它影响评估——当一些数据在开元系统里发生变化时，你要知道它将怎样在终端用户的报表里影响系统里的其它元件。在紧急情况下，数据仓储团队的能力可能会受到影响，达不到恢复时间的目标（如恢复业务功能）、危急数据点（需要及时恢复数据的点）以及恢复点对象（恢复数据的时间）而不需要登录动态综合元数据管理系统。

数据当然是数据仓储环境的心腹，所以企业必须确保数据不会出现能量故障、储运损耗或其它灾难。许多企业都作了低成本的磁带备份并且进行远距离储运。虽然从磁带中恢

复数据仓库要花很长的时间，这些数据都是历史性并且在紧急情况中没有很高的价值。为了保护更多更新的信息，公司还应该在数据通过 ETL 过程中对数据进行复制和快照然后把它们储存在灾难恢复系统里的磁盘上，这种方法能在一段时间（通常是几天或几周）后将数据存档或删除。大多数数据仓储团队都了解处理数据仓储信息的生命周期的必要性。

遗憾的是，这些团队通常不参加两次灾难恢复。最理想的状态就是，联机备份系统远距离保存，因此数据中心并不能同时破坏原来的系统以及备份的系统（这样做的费用明显要高得多并需要高速网络连接）。如果远程系统出现故障，他们同样没有对备份文件进行备份。许多企业没有预料到灾难会持续许多天。如果在 Katrina 的激发下一些业务仍然不能完全恢复，我们就需要延长对灾难预计的时间。最后，许多远程备份系统不能保护公司免遭病毒攻击，而且这些病毒还在系统内部进行繁殖。远程系统应该有一扇内置门，这扇门能将真正的病毒繁殖时间能的延长几个小时从而保护软件免遭攻击。

总结

在这种郁闷的环境中提出意见并不是很有趣，并且没有人想在转移那些可能永远不会发生的东西上面花钱。但是似乎对我来说，我们目睹了许多由环境退化和政治两极分化引起的危险期、灾难期和地理政治的紧张期。一种比较好的老式的灾难促使我们中断灾难恢复计划。这样做比等待现实灾难测试来我们的恢复计划效果要好得多。

(作者: Wayne Eckerson 译者: April 来源: TT 中国)

数据仓储能在灾难恢复热站点上进行吗？

问：我们已经被批准将灾难恢复热站（高可用性）聚集在一起，还计划从 Oracle 购买合适的 DR 许可证并且安装复制工具 DataGuard。我们的 CIO 在这些灾难恢复热站点上进行数据仓储或者放置报表。如果在备用的 DR 里复制使用 DataGuard 时也可以这么做吗？

答：Data Guard 的优点之一就是你能将备用示例放在 READ ONLY 模式里，这样你就能在不影响 Primary 示例的情况下使报表接触到数据库。所以数据分歧还将存在。如果出现了故障，那就只好等待 Standby 申请重做直到它成为新的数据库。将 standby 放置在 READ ONLY 模式里的时间能够延长出现失效转移的时间。

(作者: Brian Peasland 译者: April 来源: TT 中国)