驱动器 3.0MB/秒(持续,无压缩)

控制器 9.0MB/秒(持续,3:1 压缩)

. 记录容量　1.2GB545 英尺、EDRC 压缩记录方式
　　2.4GB1100 英尺、EDRC 压缩记录方式
. 磁轨数　36 轨 DD－－NRZI(18 轨蛇形)
. 压缩算法　兼容 IDRC
　　压缩(ANSI X3.225－1994)
. 典型压缩比　3:1
. 主机接口　ANSI 标准 SCSI－2(ANSI X3.131－1994)
. 机器接口选择　SCSI－2,F&W(16 bit)单端
　　SCSI－2,F&W(16 bit)差分
　　SCSI－2,F&N(16 bit)单端
　　SCSI－2,F&W(16 bit)差分
. 缓存　2MB
. 重写缓存　64K(最大)的数据可以重写,并且不需主机干预
. 带速　正常:2.0M/秒(78.6IPS)
　　检索:4.0M/秒(157.2IPS)
. 噪音　<50db

驱动器性能

. 加载/卸载时间:13 秒(典型)
. EOT 倒带时间　:55 秒(550 英尺)
　　100 秒(1100 英尺)
. 读写时间:　65ms
. 加电时间:　40 秒(正常)
. 选择配置:　桌上型、机架型
　　自动加载器可选
　　内置自动加载器可选

可靠性

. 保修期:1 年
. 平均维修时间(MTTR)　<30 分钟
. 平均故障时间(MTBF)　50,000 小时
. 磁头寿命　6,000 小时(50%负荷周期)
. 机械寿命　加载/卸载 200,000 次
. ACL　100,000 次
. FACL　200,000 次
. 误码率　读　<10 [alan1][alan2]－11
　　写<10－9

OLAP:将数据转化成信息的技术

Pilot Software 公司－－金岭

作为对联机事务处理(OLTP,On－Line Transaction Processing)进行补充的一种信息技术,联机分析处理(OLAP,On－Line Analytical Processing)开始在现代管理过程中被人们大量使用。OLAP 技术的概念最早由 E.F.Codd 在 1993 年提出。OLTP 系统处理的对象是大量的事务,每个事务中有相对小容量的细节数据;而 OLAP 系统则注重于对相对大容量的,主要是聚合的数据进行分析。OLAP 是行政信息系统(Executive Information System)进一步演变的方向,它使用户脱离了表单,脱离了传统关系数据库的制约。

多维数据服务器(MDD)是 OLAP 技术的基础,是表单模型的一个自然数据库服务器扩展。MDD 用于事务分析工作最为理想,由于它是一个服务器,MDD 可以被许多分析家共同使用。与关系数据库所提供的表单系统不同,一个 MDD 服务器并不限于两维查询处理,它可以管理许多维的数据。

MDD 服务器一般用作关系型数据库管理系统(RDBMS)的补充。关系型数据库管理系统服务器在通用商业操作系统中用作基本的存储管理器,如数据的输入和用户的支持。同时,关系型数据库管理系统对于各种报表和决策支持系统也工作得很好。但是,对于联机分析处理进程的那类业务分析应用需求程序而言,最好的选择仍是 MDD 多维数据服务器。

什么是 OLAP

大多数提供商和用户设计了关系型数据库管理系统,并用它来支持联机事务处理进程(OLAP)应用程序,这些环境包括有限的记录集,而这些记录集通过简单的主/非主关键词彼此相关。尽管关系型数据库管理系统也在推出查询优化器,希望可以灵活的处理较复杂的查询事件,关系型数据库管理系统至今仍最适合处理简单列表结果的查询,但对于一系列需要实时处理分析数据的交互查询,关系型数据库管理系统处理相当困难。

OLAP 应用程序事实上是在反映 OLTP 应用程序的另一极端。OLAP 查询通常检查历史数据,并且找到和决定发展趋势以及发现有偏差的特定区域。数据总是与一些维数(如销售区域,产品类型和时间)和不同级别(如部门,领域,地区和国家)的统计(或合并)有关。服务器必须有一套快速查询不同级别的统计机制,同时也应具备处理这些统计数据以分析那些隐藏的详细数据。

OLAP 是专门设计用于支持复杂的分析操作的,这种操作正是 OLAP 程序的特点。MDD 多维空间的不同截面定义了坐标系,其中存储了所需的"测量数据"(即统计数据)。一个对某一特定的信息"片"感兴趣的管理者能够通过联机方式自由定义其感兴趣的业务范围和度量。OLAP 应用则可通过多维坐标查询定位并返回一套与其指定规范相符的数据集。

大多数 OLAP 应用程序提供在给定的时间框架里进行信息检索的功能。基于这个原因,MDD 服务器常常提供在

一定时间范围和时间数据类型的某种直接支持。对于时间范围的直接支持免除了分析家和编程人员的大量工作，他们不再需要定义和编制必须的逻辑程式以支持在不同时间框架(如小时，日，周，月，年)的统计。

今天 OLAP 技术已经被广泛地使用在金融、市场和销售，企业财务分析等部门，在那里管理者们经常需要参照过去的数据来制定新的业务计划，而质量和产品控制部门也常常需要查询历史数据来分析发展态势，并预测潜在的弱点地区。下面就是一个市场分析者调查某一特定产品销售情况的例子：

1：产品 A 在上一个月销售情况如何？与前五年的相同时期销售情况相比，情况又如何？在不同的部门、地区和领域，情况又如何？

2：在其它地区，这种产品的销售是否更好？是否存在地区性发展趋势？

3：与去年相比，产品 A 是否有更多的退货？这种退货是什么原因造成的？是否其他厂家也存在这种情况？

在第一个查询集合中，市场分析者想了解时间发展态势。市场分析者首先查看各种统计，然后进一步寻找一些数据以更好的掌握隐含的细节，一个问题总是可以导向另一个问题，管理者必须能够轻易快速地进行这一系列的分析性查询，而不需要花时间等待响应。这些分析的结果可直接的影响业务决策的进程。

至于为什么 OLAP 更适合管理信息系统的开发则要从管理信息的特征说起：

管理信息有 4 个基本特征：高量度、高度多样性、高分散度和低集成度。这些特征在 OLAP 环境下的实现是十分轻而易举的。

高量度

管理信息的数量和公司的规模并不直接相关，它往往是一个集中和分散的管理概念。其数量是和想要的细节等级相关的。

在一个中层管理人员和低层管理人员都有业务智能的环境中，管理信息是非常详细的。我们可以划出成百个成本中心或一个有 2500 名顾客的公司的财政数据模型来。举例来讲，制药工业必须和高度多样性的产品打交道，某种产品可能会以药片、药粉或冲剂的形式供应市场，而且有不同的药物含量和数量。这就造成了成百或上千种产品。单位销售量和营业额必须存储一个 2 年到 5 年的时间，以监测趋势和进行季节性预测。

多样性

管理信息的多样性是和公司所活跃的市场领域和对管理公司起关键作用的信息域的个数直接相关的。集成管理信息常常涉及到很多不同的信息域，如财政信息、销售信息、人事信息、后勤信息、客户满意度信息等等。关键的运行指数常常包含和这些域相关的一些比例，如年收入/雇员或废品/生产小时。为了维护和控制这些多样性的管理信息，必须将注意力集中在操作过程和自动检察(Validation)操作上。

分散度

分散度的度量对象是信息被逻辑地或是物理地存储在不同的地点或计算机上的系统，如分散信息系统。

对于每一个信息源，当要进行月终操作的时候，应准备一个时间表，以便和不同信息域的结果进行符合比较。这叫做拷贝管理。除此以外，该信息还应能以不同方式被存取，从通过广域网(WAN)存取或用磁盘传送文件。

特别是当要处理高分散度的信息时，应制定严格的步骤，以及时和完整地接收准确的信息。想象一下，如果公司的一个部门把某客户的帐记在总公司身上，而另一个部门却把帐目送交子公司。在这个例子中就很难确定针对该客户的总收入是多少。

这种情况是很常见的，即：每个源信息系统都有它们自己的一套对如"产品"或"客户"的标准，或者不同的总体分类帐目都有它们自己的会计图表。集成度越低，维护的解释过程就会越复杂。

下面，我们将讨论为什么在管理信息开发中多维 MDD 服务器要优于关系型数据库管理系统服务器的三个重要原因：

MDD 易于使用

当关系型数据库第一次引入时，它们较以前的层次数据库管理系统和网络数据库管理系统更易使用和理解，这个优点是显而易见的。但是管理分析员们仍然发现关系型数据库令人试图把上卷(roll - up)查询和纵向钻入(drill - down)查询功能嵌入关系型数据库设计中是笨拙的和耗时的，而且关系型数据库管理系统总是不能满足 OLAP 应用程序的性能指标。

MDD 服务器有强大的内嵌计算能力，获得比率、方差和其它有关多维度量的复杂的数学关系。关系数据库没有这类强大的内嵌数学功能，这意味着大多数的工作必须用前端工具进行。用户可以试用存储程序来完成其中的一些函数，但是存储程序的建立和使用存在着困难，并且不能很好地与决策支持工具接口。同时存储程序必须由一个 DBA 预先建立，而不是由业务分析员在一个随机(Ad - hoc)的基础上定义的。

数据跟踪是多维 MDD 服务器的另一个重要的特性。商业管理人员需要从不同的角度观察数据。在前面的例子中，市场分析员首先查看销售的产品，然后是工厂地址，最后是出售产品的销售商。OLAP 的主要目标是迅速地钻透数据并变换角度来进行查询，这种能力是以管理人员所拥有的知识为基础的。过长的响应时间是不可接收的，这样可能会丢失某些最新且十分有用的信息。

MDD 与 OLAP 协调

客户/服务器环境里的一个重要目标是将进程逻辑传输给服务器.这种方式使网络流量最小化,并且防止工作站在工作期间死锁。相关的提供商已经在它们的 RDBMS 中补充了存储程序和触发器,以减少 OLTP 应用程序的流量和数据的移动。同样的原因,复杂的分析功能被设计在 MDD 服务器中。MDD 服务器可以在数据库服务器内核中进行选择、统计、计算并直接跟踪信息。这个功能意味着 MDD 服务器不需传送大量的数据,也不需为工作站装载进程,只需要将分析进程的结果传送到工作站。在工作站上,用户可以通过其它程序工具例如电子表格来显示结果,并完成任何附加的分析。

因为关系数据库内核没有计算和内嵌的上卷、导出的跟踪的功能,因此这项工作要被送往一个完成上述功能的一个前端的工具。因此,大量的数据被传送到这个前端工具,而且这个前端工具渗透网络并降低了应用程序的性能。

数据库设计者们把相关的数据分解成若干个规范化的关系型表格,可以减弱冗余的数据,但这样就导致数据难以维护和数据更新的不一致性。冗余的数据也造成了磁盘需求的急剧增加。完成 OLTP 更新具体数据的操作系统需要有一个规格化的数据库。但是,这些适合于更新程序使用的数据库不见得就适合于面向分析的应用程序。相反,当数据合并之后,分析应用程序则更易于实现和使用。问题是,管理分析家们往往发现处理那些分裂成大量二维表格的信息并不自然,不管他们使用了多少前端工具试图解决这个问题进展缓慢,至少目前沿不存在一种方法可以妥善地解决如何把二维数据组成多维形式数据的问题。

如果使用了多维 MDD 服务器,商业部门可以更好地跟踪信息(因为所有的信息都存放在同一个多维空间里),还可以更好地整理使用数据和执行在多维结构里复杂的数据演算。用户不在需要求助于复杂的联结,子查询和合一操作,也不需要与那些古怪的关系型操作打交道。这些问题均不复存在,因为所有的数据都存放在多维块中,而不是经过截断的表结构。

MDD 内嵌分析计算功能

MDD 服务器具有内嵌的分析和计算功能,这些特征在某方面反映和延伸表单和其它桌面分析工具提供的功能,把这些特性直接嵌入数据库的内核的目的时为了减少用户和专业开发人员的工作量,使用户无须再重复这些工作。同时,内嵌表单特性使得建立表单应用程序更为简单。

MDD 服务器不仅有着优秀的维护性能特性,而且因它支持上卷(roll - up)查询和纵向钻入(drill - down)查询而与众不同。MDD 服务器允许用户定义各种统计等级和查询任意截面范围中的全部查询级别的数据。它也可以直接支持时间轴的数据计算,而后者对数据进行历史分析是至关重要的。用户不需要再专门设计特定的数据模型或数据表格,也不需要设计编程逻辑式来对不同时间点的信息进行分析。为了保证这种及时性,MDD 服务器把这种统计作为数据库中的对象存储起来。这样,这些统计对象就不再需要动态计算得到了。

将工作负载传送到前端工具意味着可能会出现工作站因大量的数据和所涉及到的工作而过载。这种情况通常发生在所要处理的任务是解决分析量非常大的问题,并且要涉及到数十万行的数据时。在一些情况下,可以用附加容量来扩充工作站,但是当网络中有数百个工作站时这种方法可能不太可行。

OLAP 应用表现了 MDD 服务器的优良性能。MDD 服务器通过压缩数据和使用特殊的算法来处理稀松数据(即不存在数据的维交叉点),从而解决了这个棘手的问题。

RDBMS 提供商解决大数据库查询的方法是通过并行查询进程。这是一促笨拙的方法,它要求昂贵的硬件和软件,因此对于部门或工作组全面使用这种方法是不大合适的。但也可以只处理一小部分的问题。并行查询允许 RDBMS 迅速地提取信息,但是仍未解决信息送往的地点问题。传送到一个工作站或一个中间的应用程序服务器上的极大量的数据会使整个系统 - - 服务器、工作站和网络 - - 过载,如果系统还能运行的话,这将会导致极差的性能。一个典型的运行一个带有并行查询 RDBMS 的 SMP 计算机解决方案,需要几十万(如果没有几百万)美金,而一个运行 Windows NT 或 Unix 服务器的 MDD 的费用仅是这个费用的一小部分。

为什么使用 MDD?

一些人认为 RDBMS 和前端工具结合起来可以最终地处理所有的 OLAP 问题。这种方法可以采用,但我对此有所保留,已经设计并实现了关系模型和 RDBMS,以此来服务于某一组应用程序问题。OLAP 应用程序是在 RDBMS 的范围之外的;使用工具会比较好些,例如特别为 OLAP 设计的 MDD 服务器。试图强制使关系数据库的解决方案适用于所有的情况,这也许可能实现,但这也往往是不方便且昂贵的,并会导致可靠性和可用性上的问题。如今可以使用 MDD 服务器,它对 OLAP 应用程序是非常有用的。就我而言,当需要支持 OLAP 应用程序时,使用 MDD 服务器是十分有意义的。