随着现实世界中国数据库处理事务的日益增加，信息系统的价值也更加清晰。事务构成了商业计算系统的基础。事实上，据我们所知，事务处理（TP）已成为商业计算的推动力。20世纪70年代和80年代，诸如财务和工作等核心商业处理的批量自动化推动了大型计算机的发展。沿着这个方向，TP开始从批处理转向与系统直接交互的用户，在线事务处理（OLTP）诞生了。在20世纪80年代，计算基础设施从带哑终端的大型集中式主机转向分散式客户/服务器计算，它们具有以下特点：带有图形用户接口（CUI），运行在PC机上，并通过网络访问其他主机上的数据库。

客户端/服务器革命提供了更好的用户接口，降低了硬件和软件的成本，但是它带来了系统开发、管理以及部署方面的复杂性。经过几十年的使用，系统管理员已不堪忍受管理数千个客户端以及几十个服务器这样的繁重的工作，因此在20世纪90年代后半期又转换回到集中式，包括网络。自始至终，Oracle数据库都在使用其内部固有的架构，并不断地增强其功能，从而为不断增长的事务负荷服务。

本章将探讨Oralce数据库用于处理大量事务负荷的所有特性。

## 1、OLTP基础知识

### 1.1什么是事务

事务是工作的一个逻辑单位，它必须作为一个整体来运行，运行结果要么成功，要么失败。每个事务包含一个或多个数据库操作语句，如INSERT、UPDATE或者DELETE，并且以COMMIT来对改变做永久保存，或者以ROLLBACK撤销改变来结束事务。

一个事务必须具有：

原子性：整个事务必须作为一个完整单位来执行，要么成功，要么失败。

一致性：一个完整事务执行的结果将保持所影响数据的一致性或使其处于正确状态。

隔离性：每个事务都独立运行，不受其他事务状态的影响。

持久性：已提交的事务产生的变化时永久性的。

如果事务串行执行，即一个接着一个执行，那么他们的ACID属性相对来说比较容易保证。每个事物开始于前一个事务的一致性状态，相应地为下一个事务留出下一个一致性状态。并发使用则需要引入复杂的锁技术以及其他的协调机制来确保并发事务的ACID实现，同时还要保证吞吐量和性能。

## 2、OLTP意味着什么

在线事务处理有两种不同的方式定义：

* 一种定义具有某些特性的计算类型；
* 另一种定义是与传统的批处理相对的计算类型。

### 2.1通用特性

大多数OLTP系统都具有下面几种通用特性：

事务吞吐量高并且用户数量庞大：

OLTP系统是许多公司的关键运营系统，因而这些系统通常支持其内部的很多歌部门的各种大容量的系统。

明确的性能需求：

OLTP系统是核心业务运营的中心，因而它必须能够为用户提供一致的响应时间。OLTP系统通常会附带服务级别协议（Service Level Agreements），其规定了期望的响应时间。

高可用性：

这些系统一般被视为是任务关键的，宕机会导致成本剧烈上升。

可扩展性：

在事务量上升时，OLTP系统的性能并不会明显地下降，这就使得OLTP系统能够处理商业活动过程中事务的波动。

### 2.2在线与批处理

在线事务处理意味着在事务处理的过程中系统将会与用户进行直接的会话交互。用户使用表格填写与查询数据，而表格直接与后台数据库交互。用户提交事务的时候，系统将会对数据进行编辑和验证。

批处理过程不会与用户产生交互。事务批量地从源文件输入到运营系统。期间产生的错误通常记录在异常文件或者日志中，供用户或操作员之后查看。实际上，所有的OLTP系统都具有一个批处理组件，即存在一个作业负责在非高峰时期处理报表、工资单以及邮寄账单等工作。

许多大型公司都拥有面向批处理的大型机系统，这些系统通常与公司的基础设施紧密结合在一起，无法被替换或删除。最常见的做法就是在这些遗留系统的前面放置OLTP系统交互来输入事务。OLTP系统产生批处理文件，并把它们嫁给下游的遗留系统。

一旦批处理完成，批处理系统会产生相关摘要，并交给OLTP系统；OLTP系统根据摘要进行更新。摘要系统提供了一个复杂的用户界面，用于在线验证和编辑，不过它保存了所有流进批处理系统的数据。虽然这个过程看起来似乎代价比较高，但是比起替换这个老系统来说更可取。现存的一些问题也使得改造难度增大，例如，某些旧系统的文档不完整，而且哪些清除内部工作方式的职员已退休或转到其他公司。

金融服务业是事务处理信息技术的先导。在银行和保险公司内部，这种为下游遗留应用系统供给数据的做法非常普遍。例如，用户经常会通过前端在线系统提交保险索赔。一旦所有数据被提交，如果索赔申请被批准，它就被提交给遗留系统进一步的处理和支付。

可传输表空间和流等Oracle特性能够辅助分布式OLTP系统完成比传统批处理更加及时的作业。

## 3、OLTP与商业智能

OLTP与报表的混合工作负荷通常会引发许多性能挑战，它也是一个争论激烈的话题。由于OLTP系统需要支持大量的历史数据以及可帮助商业分析员进行多年趋势分析的即兴查询，因而它往往无法实际提供所需要的事务吞吐量；正是由于认识到这一点，诞生了数据仓库。

解决这个问题仅靠充足的机器马力是不行的。确切地说，它依赖于数据建模、存储以及访问的方法，而通常被采用的方法大小相同。在OLTP系统中，设计的核心工作就是分析商业流程，并将其自动化，从而为常见的事务及其用户提供一致的性能。这些工作负荷往往涉及大量小型的定义良好的事务，其中写事务占了较大比例。

商业智能的操作对象通常是汇聚了多个数据源数据的大型数据存储，通常包含较长的历史信息。数据仓库的模式设计与完全规范化的设计有着很大的不同。后者最适合于设计OLTP数据存储。而且任何数据仓库都支持即兴查询，虽然由于即兴查询自身的复杂性和访问数据量的原因，它会给仅有少量请求的系统带来巨大的负荷。

报表和查询功能是OLTP系统的一部分；相对于数据仓库环境而言，OLTP系统通常对报表和查询的范围及频率有较多的控制。例如，一个银行OLTP系统具有查询客户状态和账户余额的功能，但是它并不支持多年的事务模式。

OLTP系统提供的表格通常用于目标明确的查询，而且这些查询能够被高效地执行，而不会消耗过多资源。不过，有些严格的规则现在未必正确——例如，OLTP系统不能包含过多的查询功能。在大多数OLTP系统执行的I/O操作中，读操作通常占70%~80%，写操作占20%~30%。大多数事务会设计数据查询，例如，查询产品编码、客户姓名、账户余额、库存级别等。用户为特定商业功能提交经过调优的查询是OLTP的关键部分之一。而针对大量数据的即兴查询并不包含在内。

商业智能数据仓库系统与OLTP系统可以访问大部分的相同数据，但是这两种类型的系统对CPU、内存以及数据布局等有着不同的需求。这些需求通常会使得这两种类型的系统无法最好地支持混合型的工作负荷。从Oracle 10g开始，实时应用程序集群具备动态业务供给能力，它使得Oracle能够为每个工作负荷分配一个服务节点。这也使得将混合型工作负荷部署在单个数据库（具有多个数据库实例）上更加可行。

## 4、Oracle OLTP的继承性

Oracle数据库广泛应用于中型机算机环境的OLTP系统中。Oracle 6引入非升级的行级别锁以及读一致性（它们是Oracle核心OLTP特性中最重要的两个），但是从Oracle 7开始，Oracle才真正大量应用于OLTP系统中。Oracle 7引入了许多关键的特性，具体如下：

* 多线程服务器（MTS），现在叫做共享服务器。
* 共享SQL。
* 存储过程和触发器。
* XA支持。
* 分布式事务和两阶段提交。
* 数据复制。
* Oracle并行服务器（OPS）

Oracle 8增强了现有的功能，并引入了附加的OLTP相关特性：

* 连接池。
* 连接复用。
* 高级队列（AQ）。
* 根据索引组织的表。
* Oracle并行服务器的分布式锁管理器（DLM）的内部优化。
* 复制表的触发器和赋值事务的并行传播的内部化。

Oracle 8i在OLTP方面则提供了下列附加的增强功能和技术：

* 在数据库内核中提供对Java的内部支持。
* 支持分布式组件技术：CORBA V2.0和Enterprise JavaBeans（EJB） v1.0。
* 基于高级队列的发布/订阅消息。
* 在线索引重建和重组。
* 数据库资源管理器（DRM）
* 使用备用数据库来查询。
* 用于在远程站点应用事务的复制包的内部化。

Oracle 9i则在这个方向继续发展，引入了实时应用程序集群，将Oracle并行服务器的优点扩展到OLTP应用程序。从Oracle 10g开始，实时应用程序集群支持部署在新的计算模型——网络计算模型。不过，Oracle许多支持OLTP的特性在很多年前就已经被加入到数据库产品的核心。

## 5、OLTP体系结构

虽然所有的OLTP系统都面向共同的目标，但是在部署OLTP的过程中，仍然有几种不同的体系结构可供选择，具体包括传统的两层模型、三层模型以及围绕Web和网格使用的集中模型。

### 5.1传统的两层客户端/服务器模型

在这种结构中，PC相当于通过网络访问一个独立数据库服务器的客户端。客户可以运行GUI和应用逻辑，因此产生了胖客户端。数据库服务器处理SQL语句并且向客户返回请求的结果。虽然数据库服务器通过可视化工具比价容易建立，但是整个客户端/服务器系统却难以部署和维护，因为它们通常需要较高的网络带宽，而且在每个用户的PC机上都要安装特定的客户端软件，而且还需要定期升级。



存储过程：

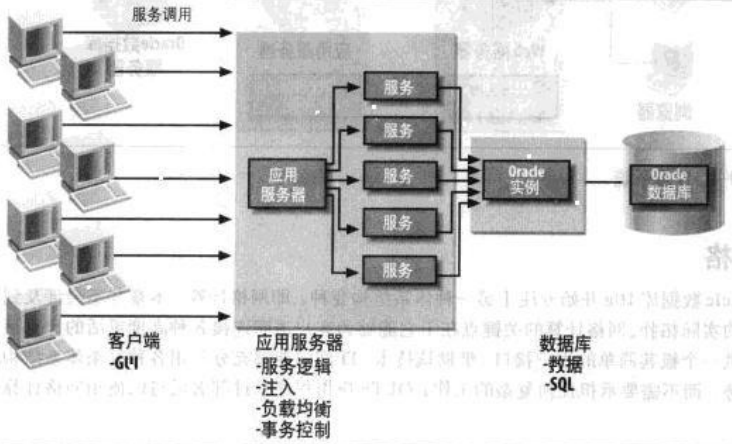
Oracle 7引入使用PL/SQL语言编写的存储过程，PL/SQL是Oracle专门用来编写应用逻辑模块的语言。这些过程存储在数据库中，通过客户端发出的远程过程调用（RPC）来执行，这区别于一般的SQL语句的执行。客户只需要发出一个携带所需参数的过程调用来，就可以代替发出多个SQL调用，从而完成一个任务。数据库利用它接受到的参数来执行所有需要的SQL语句和逻辑模块。

存储过程还能屏蔽数据库结构和程序逻辑的改变对客户端落户模块的影响。只要客户端传递的参数和返回的信息没有改变，客户端软件就不需要任何改变。存储过程将一部分用应用逻辑从客户端移到了数据库服务器，这样做能够显著地减少网络流量。这个功能增加了两层结构的可扩展性。



### 5.2三层系统

用于庞大用户数量和巨大事务吞吐量的OLTP系统通常部署为三层体系结构。客户端通过访问中间层的事务处理监控器来访问后台的数据库服务器。事务处理监控器的概念可以追溯到最初的大型机OLTP系统。当然，在大型机环境中所有逻辑都在一台主机上运行。而在开发式系统环境中，应用服务器一般运行在一台或多台独立的主机上，从而在客户和数据库服务器之间增加一个中间层。



几种不同类型的应用服务器：

* 旧的私有服务器，例如，运行在UNIX和Windows上的BEA系统公司的Tuxedo，以及运行在大型机上的IBM的CICS。
* 基于Java 2企业版本（J2EE）的行业标准应用服务器。
* 微软.NET应用服务器环境，它是Windows服务器版本的一部分。

应用服务器为客户端调用的服务提供了运行环境。客户端并不会直接和数据库服务器交互。事务处理（TP）监控器在远程机器上提供服务的例子在许多方面都类似于前面一节简述的存储过程体系结构；这也是为什么有时候把基于存储过程的系统称作“TP-Lite”的原因。

应用服务器还提供了一些有价值的服务：

* 注入：
* 连接池：
* 负载均衡：
* 容错：
* 事务路由：
* 异构事务：

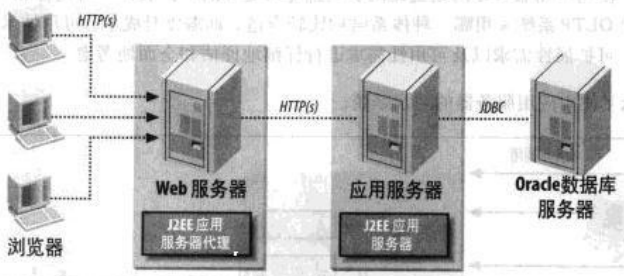
使用应用服务器的系统比两层系统提供更高的可扩展性、可用性和可行性。要确定一个OLTP系统采用哪一种体系结构比较合适，则需要对成本、可用技术、工作负荷概述、可扩展需求以及可用性需求进行仔细评估和全面地考虑。

### 5.3应用服务器和Web服务器

在基于Web的系统中，应用服务器和/或Web服务器通常位于中间层。这些服务器所提供与前面我们描述的应用服务器相似，不过它们是以Web为中心，负责HTTP、HTML、CGI以及Java程序。

在过去十年，J2EE和.NET应用服务器得到了长足的发展，在今天的N层结构系统中，它们承担起了遗留TP监控器的职责。不同的公司有不同的标准和偏好——由于.NET是私有产品，很多公司使用J2EE，不过，也有些公司喜欢.NET将众多的微软产品集成在一起。有关J2EE和.NET的各自特点以及通过应用服务器技术详细讨论超出了本书的范围。可以说，应用服务器在今天的系统环境中扮演者重要的角色，因而数据库管理人员必须懂得N层系统架构。

如图，描绘了N层系统的架构，其中包含客户端，Web服务器、应用服务器以及DBMS服务器。



### 5.4网格

Oracle 10g开始专注于另一种体系结构变种，即网络计算。网络计算的关键点在于它能够为客户透明连接各种高度灵活的计算平台提供一个及其简单的用户接口，借助该技术，IT部门能够充分利用各种复杂体系结构的优势，而不需要承担任何复杂的工作；OLTP应用程序经过部署后可以使用网络计算资源。

## 6、Oracle OLTP特性

Oracle有许多有助于提供OLTP的性能、可靠性、可伸缩性和可用性的特性。这一节将说明这些特性的基本属性。

### 6.1通用并发性能

在OLTP系统中，Oracle已经实现了对一致性与性能很好的支持。OLTP相关的一些关键特性如下所示：

非升级行级所技术：

多版本读操作一致性：

共享SQL：

存储概要：

### 6.2可扩展性

多线程服务器和数据库资源管理器有助于Oracle支持拥有大量用户的环境以及混合的环境。

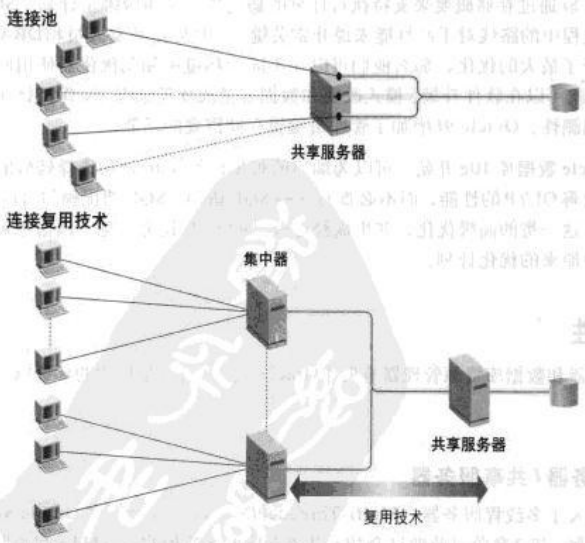
1）多线程服务器/共享服务器

Oracle 7引入了多线程服务器（Multi-Thread Server，MTS，在Oracle 9i中被更名为共享服务器）以便支持更多的用户。当MTS减少服务器进程数量时，每个客户仍然使用他自己的物理网络连接。通常，用于网络连接的资源是很有限的。Oracle 8在操作系统级别提供了两种增加实际网络套接层容量的解决方案：

Oracle Net连接池

Oracle Net连接管理器

就可扩展性而言，可以把连接池看成中量级解决方案，而把连接管理器的复用技术看成重量级解决方案。



在Oracle 10g中，连接管理器变得更加灵活，添加了不需要重启连接就可以动态修改配置参数的功能，并且改进了其中的CMAN流量过滤访问规则。

绑定变量和共享SQL：

Oracle共享SQL是构建高性能应用程序的一个关键特性。在OLTP应用中，相似的SQL语句可能被重复利用，但是每个提交的SQL语句一般在WHERE子句都有不同的选择标准，以确定操作的目标行。Oracle可以共享SQL语句，但要求语句必须完全一致。

那些除了WHERE子句中特定值不同其他部分相同的SQL语句都可以共享。为了重复利用这个特性，可以在SQL语句中绑定变量。SQL语句中代替绑定变量的值可以不同，但语句本身必须相同。

为员工加工资的例子。应用程序提交以下的SQL语句：

UPDATE emp SET salary = salary\*(1 + 0.1)

WHERE empno = 123;

UPDATE emp SET salary = salary\*(1 + 0.15)

WHERE empno = 456;

这些语句显然不同，它们分别更新由员工号标识的不同员工信息，而且每个员工工资的涨幅也是不同的。为了充分利用共享SQL的优点，可以通过在应用中绑定变量来表示工资的增幅和员工号。这样，修改后的SQL语句如下：

UPDATE emp SET salary = salary\*(1 + :v\_incr)

WHERE empno = :v\_empno;

UPDATE emp SET salary = salary\*(1 + : v\_incr)

WHERE empno = :v\_empno;

经过这样的修改，这些语句就完全相同了，可以共享。这段代码的不同之处是为两个变量:v\_incr和:v\_empno提交了不同的值。Oracle在SQL语句中使用这些实际值来代替变量。这种替换发生在绑定阶段，即紧跟在分析阶段和优化阶段之后。最后的结果和使用两条不同的语句是一样的。

Oracle 10g以及最新的版本提供了相关的调优工具，用于方便地检测出潜在的这种类型的应用程序优化。

2）数据库资源管理器

Oracle 8i引入了数据库资源管理器（DRM），从而使得混合工作负荷管理得到简化和自动化；混合工作负荷是指不同用户根据不同目的来访问同一数据库。可以定义不同的消费者组来包含不同的用户组。DRM基于资源计划为消费者组分配CPU及并行资源。资源计划为用户使用的特定计算机资源定义了一组总量限制，这就保证DBA可以根据执行需要为某类用户分配足够的主机资源。

Oracle 9i对DRM进行了显著改进。DRM现在允许DBA为消费者组制定可用的活动会话数量。该消费者组的其他任何连接请求都将被排队。通过限制活动会话数量，可以避免下列情况的发生；一个新的请求进入，可能使得改组的资源需求超过了规定的限额，从而影响该组其他用户。

Oracle 9i中的DRM还具有估计每个请求操作占用的CPU时间的功能。如果某个操作看起来将会超过所在资源组规定的最大CPU时间，那么该操作将不会被执行，这样可以从一开始防止执行不适当的大型操作。

DRM还能在某个消费者组活动太久的情况下在佛那个转向另一个消费者组。这个特性适用于自动将面向OLTP的小型操纵消费者组切换到另一个更适合于批操作的消费者组。

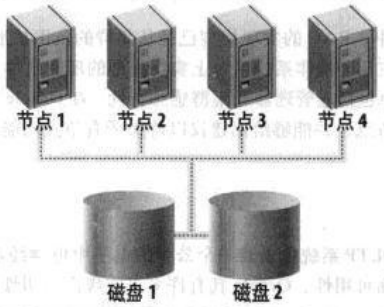
从Oracle 10g开始，可以根据服务名、应用程序、主机或操作系统用户名来定义消费者组。

### 6.3实施应用程序集群

可以证明，在Oracle 9i中最大的进展就是实时应用集群特性。该技术是在以前的Oracle并行服务器（OPS）的基础上发展起来的。

OPS是某些数据仓库应用程序中用来改进性能和可扩展性的一种特性。由于存在着ping现象，OPS只限用于数据仓库实现中。

在OPS额RAC的环境中，多台机器访问共享磁盘上相同的数据库文件（这个共享磁盘通过物理方式连接或者通过软件以看似物理连接的方式连接）。



这个体系结构允许在一个主机集群中增加更多的主机，相应地增加系统的处理能力。但是也存在这样一个问题，那就是因为集群中的每台主机都有自己的缓存。如果一台主机试图修改另一台主机正在修改的页中的一行，这个页必须被写入共享磁盘的数据文件中——这个过程又叫做ping。这些关联的事件会导致磁盘I/O的增加，从而降低了这种解决方案的整体性能。

解决这个问题的传统方法就是简单地避开它，仅在数据库大量的写操作不会导致ping发生的情况下，或者对写操作进行隔离不让它们访问其他节点正在使用的数据的情况下，才使用OPS。这个限制需要你仔细考虑部署OPS的应用类型，有时候甚至会要求你根据OPS限制来修改应用程序的设计。

使用实时应用集群，由ping导致的问题已经得到解决。实施应用集群支持的技术叫做高速缓存熔合（Cache Fusion）。高速缓存熔合特性将所有数据存放在实时应用机器中的每台主机的每个缓存内，这些数据对于集群中其他任意主机都是可用的。如果一个主机需要由另一个主机缓存的数据块，这个数据块通过超高速的互连通道直接转入请求的主机。

有了高速缓存熔合意味着不需要考虑ping的问题。实时应用集群可以使几乎所有的应用在可扩展性方面都有显著的改进，而不需要用户做任何改变。对于部署子啊RAC环境的OLTP应用程序而言（在这种环境中，会频繁修改某些叶子块中的索引），反向键索引可以将插入操作分散在索引的所有叶子健中，从而避免任何可能的性能问题。

实时应用程序集群还继承了OPS所有的可用性优点，由于实时应用集群中所有主机共享同一个磁盘，因而单个主机的故障并不意味着这个数据库的故障。用户连接到有故障的主机时，将被转到集群中的另一台主机，数据库服务器本身继续工作。

在Oracle 10g中，RAC的实现模型已经从集群的范围扩展到了网络计算。现在Oracle能够提供用于在几种操作系统平台上实现集群的所有组件，并将它们纳入Oracle软件栈的一部分，其中包括卷管理器和集群感知特性。

### 6.4高可用性

从运营的角度来看，OLTP系统代表着一个公司的电子中枢神经系统，因此支持这些系统的数据库必须具有高可用性。Oracle具有许多可实现高可用性的特性：

备用数据库：

透明故障迁移（TAF）：

Oracle流/高级队列（AQ）：

Oracle流复制：

实时应用程序集群：

## 7、Oracle流和高级队列

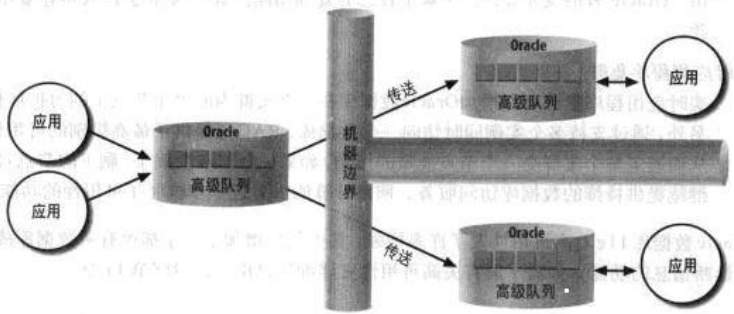
消息技术已经存在许多年了，而且在OLTP应用系统中也相当常用。典型的消息技术为消息从一台主机经过网络传输到另一台主机提供了一个可靠的传输层。高级队列（AQ）作为一个集成的数据库服务被引入到Oracle 8中。Oracle 9i在创建Oracle流的过程中将AQ与机遇日志的复制结合起来。

Oracle AQ除了具有简单消息产品的优点之外，还提升了数据库中队列的价值。消息队列中的信息代表关键的业务事件，应该被存储在一个可靠的、能扩展的、安全的并且可恢复的位置。将队列存放在数据库中，可以将数据库的核心优势扩展到队列本身。

队列中流过的数据代表着业务操作的流入流出。分析消息的类型和容量有助于确定不同的业务功能是如何运作和交互的，更进一步则可以提供业务操作相关很有价值的信息。Oracle AQ支持消息仓库的概念，由于消息是存储在数据库中，队列的内容及详细信息可以被查询和分析。Oracle还支持将消息从队列中取出，不过还可以在队列中留下历史数据以便日后分析。

只要发出了特定的入队或出队语句，应用程序就可以将消息作为事务的一部分或者独立的事件入队或出队。包括在事务范围内的队列操作与事务一起被提交或回滚。如果发生了失败，队列操作和其他的数据库操作一起恢复。

Oracle提供一个消息传递的路由引擎用于将消息从一个队列传播到另一个队列。



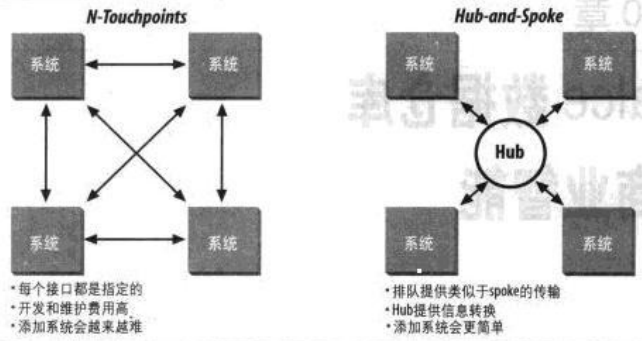
Oracle 10g以及之后的版本允许将批量的消息加入队列和退出队列，同时也减少了与队列交互的代码量，因而使用这些Oracle版本来编码实施流更加容易。

### 7.1系统接口的流

实现OLTP系统时总会涉及到与企业内部其他系统或其他公司的系统之间的接口。设计、创建和管理这些接口的任务非常艰巨，这些任务在大型ERP实现中可以很容易地达到总成本的40%~60%。此外，再将其他系统增加进来或者改变现有的系统都会导致重新设计接口，从而增加负担。

Oracle流可以帮助公司解决集成问题，其方法是通过结合使用消息、路由以及传输这三种技术提供一种“hub-and-spoke”的体系结构。在传统的方法中，需要在两个系统间并发专用的接口。当需要往这两者中加入第三个系统时，必须在每两个系统间开发更加专用的接口。要继承的系统越多，接口越需要定制，相应的开发和维护负担也就越多。

利用这些组件，每个系统通过辐条连接到一个中心，这样也就避免了系统对系统的直接接口。这个辐条负责发送和接收消息，中心提供路由和转换服务。这样就减少了系统之间接口的数量。不需要在两个系统之间建立专用接口。把系统增加到现有系统中也不需要开发许多新的接口。将新系统连接到中心。然后再由中心来负责路由和转换。如图，将定制方法和“hub-and-spoke”方法做了对比：



### 7.2Oracle与发布/预定技术

Oracle 8i引入发布/预定功能增强了高级队列技术。对于应用程序希望收到的消息，可以通过指定其属性预定一个消息队列。当另一个应用程序通过将消息加入队列进行发布时，Oracle会分析这个消息的内容，从而确定哪个预定应用程序对此感兴趣，并通知感兴趣的应用程序。例如，一个船运应用程序可以预定一个用于订货的队列，并制定自己对状态为“准备好船运”的订货消息感兴趣。当表示订货的消息流经这个队列时，该应用程序仅会收到满足要求的消息。这种发布/预定的方式与消息传播技术结合在一起，为系统之间的信息流动提供了一个非常强大的消息中枢。

## 8、对象技术与分布式组件

理论上来讲，信息量越大，其中可提取的智能就越多，将各个独立系统的信息集成起来可能是一个庞大的工作，特别是当许多系统混合在一起的时候，其中的集成复杂度将会成几何数增长。

消息技术可以用于不同系统间的交互，同时也常用于在线交互。例如，如果HR系统需要维护公司员工的信息（例如，员工的部门和职责），最理想的情况是在订购时订购系统可以在线访问HR系统的数据，为此，订购系统需要确定订购者的消费限额，以及将账单记在哪个部门上。实际上，由于需要系统之间协商如何通信，这些线接口通常难以建立。每个系统都有其私有的API，用于和其他系统通信。这些专用的API限制了每个系统功能的重复使用。

对象技术提供了一种很有竞争力的解决方案：系统通过调用对象方法而不是调用特定的API来实现通信。例如，如果要检查一个用户所在的部门，那么可以对HR系统管理的员工对象执行一个标准对象调用。

Oracle 8i以及之后的版本支持许多对象技术，包括使用Java作为面向对象的编程语言。第14章将详细描述Oracle对面向对象的支持。目前，面向服务体系结构的概念已成为应用程序开发和代码重用的焦点之一，这方面的内容将在第15章讲述。