

C/S 结构数据库应用系统的性能调整

王玉奇 (大庆石化总厂信息中心 163714)

摘要:本文首先分析了影响 C/S 应用性能的主要因素,在此基础上提出了进行 C/S 应用系统性能调整的一般原则,最后具体介绍一个 C/S 结构 ORACLE 应用系统性能调整的思路和方法。

一、问题的提出

C/S(Client/Server 的简称,意为客户(机)/服务器)是近年来非常流行的一种协同式分布处理体系结构。它克服了传统主机处理结构和局域网/共享文件处理结构的缺点,具有高度的灵活性、开放性和可伸缩性,能够有效地保护用户以往的系统投资。C/S 结构要求将整个系统处理任务根据具体情况合理分布在 Client 端和 Server 端,使 Client 和 Server 能够相互配合,密切协作,以便最大限度地发挥 Client 和 Server 各自的工作潜力,更好地满足用户对应用系统的整体要求。C/S 的这种分布式处理思想顺应了 90 年代以来出现的系统规模适度化(rightsizing)的潮流,因此得到了迅速发展并被广泛采用。

目前,国内许多单位已经开发或正在开发基于这一体系结构的应用系统。但 C/S 是一种典型的协同计算模式,它的高度开放性和可伸缩性使系统有可能融合多家厂商的技术和产品,从而会大大增加整个系统的复杂性和系统性能调整的难度。由此造成按照 C/S 结构模

式开发的应用系统最初往往整体性能较差,不经过系统地性能调整工作,很难满足用户的需要。随着应用系统规模的不断扩大,人们越来越深刻地认识到,这种 C/S 结构模式在给信息技术注入新的生命力的同时,也给我们提出了新的技术课题,即如何协调、适配 C/S 应用系统中各方面的关系,才能使系统整体性能达到最优? 本文将从分析影响 C/S 应用性能的主要因素入手,在此基础上提出开展 C/S 应用系统性能调整的一般原则,然后具体介绍应用这些原则调整一个 C/S 结构 ORACLE 应用系统性能的思路和方法。

二、影响 C/S 应用系统性能的主要因素

要想正确分析影响 C/S 应用系统性能的主要因素,必须首先弄清 C/S 体系结构的组成和 C/S 应用系统中数据存取的工作过程。关于客户机/服务器这个名词,虽然在近年来的专业媒体中经常出现,但这一术语至今尚无统一的定义,一般都将其理解为是一个物理上分布的

逻辑整体,它由客户机、服务器和连接支持等三部分组成,其中客户机是体系结构的核心部分,是一个面向最终用户的接口设备或应用程序,它是一项服务的消费者,可向其他设备或应用程序提出服务请求,然后再把得到的结果信息显示给用户;服务器是一项服务的提供者,它包含并管理数据库和通信设备,为客户请求过程提供服务;连接支持是用来连接客户机和服务器的部分,如网络连接、网络协议、应用接口等。

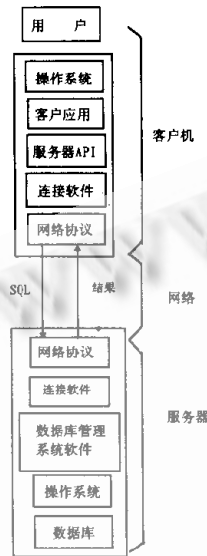


图1 C/S应用中典型的数据存取过程

图1所示是C/S应用中一种典型的数据存取过程,该图示应用把应用处理逻辑、数据表示及用户交互界面等部分放在Client端,把数据库、数据存取管理、完整性及并发性控制等部分放在Server端。Server提供的服务主要包括数据查询、更新、事务管理、索引、高速缓存、查询优化及多用户存取控制等。

用户对数据库的存取要求具体将表示为客户应用中的SQL语句,SQL语句经过服务器API、连接软件、网络协议传到网上,然后传到服务器一端,再经过服务器端的网络协议、连接软件传给数据库服务器软件,由数据库服务器软件具体执行SQL语句,实现对数据库的访问,当然具体的磁盘输入/输出是再通过服务器端的操作系统来完成的,从数据库中取得的结果数据,将按相反方向送回给用户。从图1中可以明显看出,在网络上传输的是SQL语句及其执行结果。SQL语句从Client传向Serv-

er,其执行结果从Server传向Client。

根据以上分析,可以把影响C/S应用性能的主要因素粗略归纳为下列五个主要方面:客户机、网络、服务器、数据库和应用系统,具体来讲就是:

1. 客户端计算机

- 硬件系统处理能力:CPU、主频、内存等。
- 操作系统设置:内存分配(诸如Windows环境下expanded memory的分配,各种drivers的内存分配)等。

2. Server端计算机

- 硬件系统处理能力:CPU、主频、内存等。
- 操作系统的选择及系统参数的配置等

3. 网络连接

- 硬件设备:网卡的处理速度、bridge/Modem的性能指标、电缆/电话线的传输速率?主干网(backbone)的性能?网络通信控制器的配置?等。
- 网络软件:采用何种通信协议?采用何种网管软件?等

·网络结构布局设计:是采用Ethernet/IEEE 802.2还是采用Token Ring/IEEE 802.5?是采用电缆/bridge方式还是用电话线/modem方式,或者是两者的混合方式?是否采用router?等等。

4. 数据库服务器软件的系统参数设置

- 内存分配方案:百分之多少用于Data,百分之多少用于Stored Procedure?
- 用户数目指定:最大并发用户数过大会影响每个用户所占内存量;过小则会造成用户队列的延长。
- 数据块尺寸的设定:它影响数据空间的使用效率
- tablespace及各种rollback segments的设置与分配等等。

5. 数据库设计

- 若Server端计算机超过一台,相互间如何划分、连接数据库?
- 数据库的规范化程度(Normalize)?
- 用哪些Field来定义index和clustered index等。

6. 应用系统的分布设计

- 功能分布:应用处理逻辑是整个驻留于client端,是整个驻留于Server端,还是将应用处理逻辑分割开来,合理分布到Client和Server上?
- 数据分布:数据是集中存放还是分布存储?如果是分布存储数据,那么应怎样分割整个数据集?如果是分布存储数据,那么应把哪些部分存放于

Client, 哪些部分存放于 Server 上? 整个数据集的完整性如何保证等等。

以上大致列举了可能影响 C/S 应用性能的主要几方面因素。必须指出, 它们之间不是彼此孤立的, 而是密切相关甚至是相互制约的。常常会出现这样的情况: 为改善某个性能问题而采取的措施却引起了另一方面性能的下降。从这种意义上讲, 性能调整往往是一种得与失的权衡, 是各有关方面利与弊的折衷。那么作出这样权衡与折衷的依据或者说开展 C/S 应用性能调整的一般原则是什么呢?

三、C/S 应用系统性能调整的一般原则

系统性能差的原因有时很简单: 计算机的处理能力不够, 或缺少应用系统所需的资源。为了便于深入讨论问题, 本文假定所有硬件系统的处理能力和硬件资源都能满足整个应用系统的需求, 即上述情况不在本文讨论之列。

概括地讲, 开展 C/S 应用系统性能调整的一般原则是:

1. 合理划分应用功能, 把应用功能分布在最适合完成其任务的计算机上运行, 以便使 Client 和 Server 上的负载得到均衡分配, 从而最大限度地发挥 Client 和 Server 端计算机的处理能力, 并减少网络上的传输量。如当 Server 端计算机具有强大的处理能力时, Server 端就可以承担更多的处理任务, 把原来由 Client 端完成的应用处理任务部分地移到 Server 端, 由 Server 计算机来实现, 例如通过调用存储过程来实现。所谓存储过程就是存放在 Server 上的预先写好并经过编译的代码段(过程)。这段代码段能完成一定的功能, 可以在 Client 的应用程序中直接调用该过程, 这样既发挥了 Server 计算机的强大处理能力, 又能减轻 Client 端计算机的工作负载, 避免了出现忙闲不均的现象。由于在该过程中可以频繁地使用 SQL 语句, 从而不必在网上频繁地传输 SQL 语句及其执行结果, 只需在网上传输过程调用命令及该过程的执行结果, 这样就可防止形成网络瓶颈, 明显提高系统的整体性能。

2. 合理地分布数据, 把数据分布在距其用户最近的地方, 比如某台 Client 计算机上, 而不是把数据都放到 Server 机上, 这样就可以避免该 Client 频繁地向 Server 发出数据存取请求, 从而也就减少了网络传输量和 Server

端的磁盘 I/O 次数。

3. 合理配置和调整服务器操作系统和数据库管理软件的可调系统参数, 全面提高服务器操作系统的工作效率和数据库的存取效率。

4. 紧密结合数据库管理软件的技术特色, 充分优化数据库应用系统。主要应从数据库的规范化、索引的建立与使用、连接(join)的表的个数、连接操作中驱动表与被驱动表的顺序等方面着手开展优化工作。

四、一个 ORACLE 应用系统的性能调整

由于用户需求的多样性和 C/S 体系结构的灵活性, 不可能提出, 事实上也不存在一种适用于所有 C/S 应用的性能调整方法, 而只能根据实际情况做具体分析并采取与之相适应的调整方法。笔者曾参与开发过一个 C/S 结构的大型 ORACLE 应用系统。该系统的服务器全部采用 VAX 计算机, 并配置 VMS 操作系统, 客户机全部是 486 以上的微机; 网络部分采用 IEEE 802.3 以太网并运行 TCP/IP 通信协议。尽管有很强的硬件配置, 但系统实施后其整体性能表现很差。后来我们按照前述的性能调整的一般原则并结合这个系统的实际情况着重从服务器操作系统、数据库管理软件及应用系统本身等影响系统全局的几个主要方面对系统进行了全面的性能调整, 取得了十分明显的效果, 系统性能得到很大改善。下面就简要介绍一下这个系统的性能调整思路和方法。

1. VMS 系统参数调整

- (1) 调整工作集: 当一个 ORACLE 用户进程产生时, 内存中就为之产生一个工作集, 存放该进程的代码和数据, 工作集装满后, 发生访问的页面不在工作集中时就会产生换页。可见用户工作集越大, 发生换页的可能性就越小。但用户工作集越大, 内存能容纳的 ORACLE 用户进程就越少, 产生进程交换的可能性就越大。该系统 ORACLE 用户工作集一般都调整为系统原缺省值的 2-3 倍, 对经常需要多表查询的用户, 则将其工作集调整为原缺省值的 5-6 倍。

- (2) 调整自由页表和修改页表: VMS 操作系统中的自由页表和修改页表起高速缓存作用, 其大小合适与否直接影响 ORACLE 用户。若其值过小, 会发生所缺页面不在内存缓冲区, 从而会增加磁盘 I/O 数; 若其值过大, 则会减少用户进程数, 导致交换增多, 同样会增加磁盘

I/O数,影响系统效率。要用系统监控命令观察读缺页率和页交换率,以确定自由页表设定是否合适。若前者高,则说明自由页表偏小,应予增大;若后者高,则应减小自由页表,增大平衡集。

(3)调整系统调页文件和交换文件大小:系统调页文件和交换文件实际上就是VMS中的虚拟存储空间。前者用于转储需换出内存的修改页面,后者用于转储需换出平衡集的用户工作集内容。可用显示内存命令观察两者的使用情况再作相应调整。对于交换文件,一般运行时只需保持1/4的剩余空间,但我们在调整该系统时将交换文件的剩余空间调到1/3左右时才保证了所有ORACLE用户的正常运行。

2. 调整 ORACLE 数据库管理系统参数

(1)调整数据字典快存:数据字典快存与缓冲区快存相比所需内存空间较少,但数据字典缺页的影响比缓冲区快存大得多。因此,在ORACLE内存分配上应首先考虑满足数据字典快存的需要。数据字典快存由多项内容组成,可用ORACLE监控命令观察其运行情况。若数据字典快存中某项缓冲区的访问命中率小于85%,则应考虑将其加大。若其使用值远小于设定值,则可将其减少。

(2)调整缓冲区快存:缓冲区快存是ORACLE系统全局区(SGA)中用于存放数据块的区域,缓冲区中每个存储块包含一个ORACLE数据块。缓冲区快存中存储块的数目由ORACLE系统的某些初始化参数决定。用户可根据ORACLE监控命令提供的统计值,判断其大小合适与否,可用逻辑读与物理读之差与逻辑读的百分比来衡量。若这个值小于60%,则应增加相应的ORACLE初始化参数,增大缓冲区快存。调整时可以50为增减步长量,然后观察相应命中率的变化情况,从而确定最佳调整值。

3. 调整应用系统

(1)数据模式的规范化:若数据模式不符合第三范式,则容易引起数据不一致性,并造成数据冗余;而过分规范化则往往会增加连接(join)次数,使性能严重下降。一般情况下将数据库规范到第三范式为宜。

(2)索引的使用:在索引方式的选择方面,一般而言,主键(Primary Key)应采用clustered index,而外键(Foreign Key)则采用nonclustered index;在主键的索引方面,

不应有超过25%的列成为主键,而只有很少的普通列。在索引的使用效率方面,仅在选择数据量少于全表的20%,并且表的大小超过ORACLE的5个数据块时,使用索引才会有效,否则,用于索引的I/O加上用于数据的I/O就会大于做一次全表扫描的I/O数。

(3)结合ORACLE的技术特点,优化SQL语句

ORACLE核心使用自底向上,从右至左的处理规则。例如,FROM子句中的最后一个表才是ORACLE用作驱动表的表。WHERE子句的最后一个条件中所含的列,它所属的表才是最先被引用的表。同样,WHERE子句中的条件语句也是按从右至左的顺序处理的。因此,在优化SQL语句时应将最具约束性的条件(例如,限制性最强的,返回行数量少的)放在WHERE子句的最后面。概括起来就是:

在FROM子句中,要将表名按被驱动表——驱动表的顺序排列;

在WHERE子句中,要将条件语句按最少约束——最多约束的顺序排列。

五、结束语

C/S应用开发中最具难度的莫过于系统性能调整,它的挑战性在于:①系统性能调整工作不仅仅指出现性能问题后调整,它不包括在设计阶段为防止以后出现性能问题而采取的主动性的事前调整工作。②在C/S环境下的应用系统,不要说广域网,就是局域网,在系统真正投入运行之前,也很难进行真正有实际意义的性能模拟测试工作。③你可以按一定的规范去分别单独预测,调整Client端、Server端及网络系统各自的性能,然而当三者组合在一起实际投入运行时,真正的性能“瓶颈”究竟在何处?迄今为止尚未见到有成熟的规律和理论可循,很大程度上要凭专业人员的实践经验。由于没有一种性能调整方法可以适用于所有的C/S应用系统,因此掌握C/S应用性能调整的一般原则,找到解决问题的切入点就是非常重要的。本文是笔者近年来从事有关工作的经验与体会的总结,意在抛砖引玉,希望有更多的同行参与这方面的讨论。

(来稿时间:1997年7月)