

如何提高 Visual Basic 应用程序的性能

张庄 (福建省经济信息中心 350003)

摘要: Visual Basic 是目前主要程序设计语言, 大部分的应用开发都采用了 VB 编写, 但编写程序时不采用合理的设计, 不但会使计算机资源浪费, 而且给应用系统的改进造成困难, 本文就介绍用 VB 程序编写时应该注意的事项和提高的方法。

关键词: Visual Basic 程序设计 提高

只有几年的时间, Visual Basic 已经成为快速开发 Windows 应用程序最重要的工具之一, 正如微软公司总裁比尔·盖茨所说: “全球绝大多数 Windows 应用程序都是用 Visual Basic 开发的”。这是因为编写 VB 应用程序者无需学习过更复杂或更正式的开发就可以设计出运行在 Windows 平台上的各类应用程序。但是 VB 是一种解释式的语言, 不能指望它达到优化编译器语言(如 C++/C)的性能, 同时 VB 程序编写者很大一部分为非正式计算机语言设计人员, 因此, 提高 VB 应用程序的运行速度是非常重要的。怎样设计使 VB 应用程序达到最佳性能呢? 本文提出的一些方法在编写 VB 程序时能起一定的作用。

一、使用 VB 代码剖析器

VB 代码剖析器是调试 VB 程序非常好的工具之一, 它可以跟踪 VB 程序执行代码的每一行, 程序运行完后, 代码剖析器建立了反映应用程序各部分运行时间报告。这些报告会显示出子程序被调用的次数, 实际运行时间以及执行应用程序各部分所花费的时间百分数, 从中很清楚地看出应用程序的哪些部分能够或者应该进一步改进优化; 能够查找出没有被调用过的模块或执行次数为零的模块; 还能够查找出死码。

VB 代码剖析器必须从 VB 安装光盘中的 TOOLS \ VBCP 子目录中将它拷到硬盘中的某个子目录下。代码剖析器在添加到 Add - Ins 菜单之前先要运行一次 VBCP 目录中的 VBCP32.EXE(即双击 VBCP32 图标)程序, 然后运行启动 VB, 这时 VB 代码剖析器就会作为一个可用的添加工具出现在添加工具管理器中(Add - Ins Manager)中, 单击选中它, VB 代码剖析器就作为一个选项出现在 Add - Ins 菜单下, 这时程序员就可以用 VB 代码剖析器对程序中的所有模块给予分析, 分析的结果主要有如下 4 个类型:

(1) 行运行时间(Line Timing): 显示执行每一行代码所花时间, 如果要检查某些行是否花了很多时间, 那就设置行这个类型。

(2) 函数运行时间(Function Timing): 显示 VB 程序中所有函数执行所花的时间, 如果要检查哪些函数运行时间最长, 就设置这个类型。

(3) 行运行次数(Line Hit Counts): 显示程序中每一行代码被执行次数, 如果检查是否有些代码行从没有执行过, 就设置这个类型。

(4) 函数运行次数(Function Hit Counts): 显示函数执行次数, 如果检查 VB 程序中某些函数是否从没有执行过, 请设置这个类型。

当某个应用程序要运行代码剖析器时, 必须先将它加到应用程序中。单击 Add Profile Code 按钮向项目中添加一个称为 VBCP CODE.BAS 模块, 然后运行应用程序, 代码剖析器就会跟踪每一个例程完成后, VB 返回到代码剖析器中, 从 File 菜单中选择 View Result 菜单项就可以很清楚看到应用程序代码运行的结果。

二、优化 VB 程序代码

VB 程序代码编写的好坏对 VB 程序性能的影响很大, 例如 txtTextBox = “default text” 这个语句虽然和 txt-TextBox.text = “default text” 语句功能一样, 但是 VB 运行第一条语句要快 25%, 这是因为解释程序不必分析那个圆点。所以说优化 VB 程序代码可以提高执行速度, 减少内存占用, 减少对系统资源的要求, 减少占用磁盘空间等性能, 一般来讲对 VB 程序代码优化有以下五个方面。

1. 一般性的代码优化

(1) 选取最佳数据类型。VB 有五种数值数据类型, 每一种类型都有一个特点的用途和空间需求, 例如整型(INT)占 2 个字节; 长整型(Long int)占 4 个字节, 双精度

(double)和货币型占到8个字节,当然选用INT类型的数据运行最快,选用货币型的数据运行就最慢,相对地INT类型数据范围最小,双精度和货币型数据范围最大。因此,选用哪种数据类型一定要根据实际的需求来确定,不要以为最大最好,这样就会造成运行速度变慢。

(2)有用时才选择可变数据类型。VB的可变数据类型相当灵活,但运行负担很重,因为可变数据类型不管存储什么数都占16个字节,而它存储字符信息时要占用22个字节外加字符串长度,如果没有对变量强制声明,那么没有声明的所有变量VB会使用可变类型,这点请大家请注意。

(3)内联(inline)代码快于函数调用。函数调用增加了程序堆栈的大小,并会引起另一处代码的跳转,内联处理没有这种情况,如果某个函数只用到一、二次,那么内联方法是最好的,但如果函数调用次数很多,就不要用内联方法了。

(4)合理设置窗体(FORM)上的控件类型和数目。窗体上的控件越多,加载与这些控件关联的内容花费时间越长。在VB应用程序的第一个窗体上考虑这种情况尤其重要,如果必须在第一个窗体上设置许多控件,那么在加载该窗体对考虑用飞溅式的屏幕显示方法,这样在应用程序就绪的过程中向用户提供一些可看的東西。

(5)去掉无用代码。无用代码虽然不占用执行时间,但增加了可执行文件的大小,如果可执行文件很大,那么它就会经常被交换出内存。在程序中出现无用代码是太容易了,例如,删除一个控件而不删除与它相关的代码,那么代码就会放到应用程序通用段的函数中。

(6)用Select语句取代嵌套的if...then...else语句。如果只检查一个条件,那么if...then语句效率很高。如果检查许多条件,VB处理一条select语句要比处理一系列嵌套的if语句更快,而且select语句更让人看得懂。

(7)用for...next循环语句取代do...while循环语句。do语句是在循环的结尾赋值,这意味为实现每个条件整个循环要执行,然后分析while语句。for语句是在开始处赋值,它至少节省整个代码的一次运行。如果在整个循环中用短整型做索引,就会获得最大性能。

(8)直接调用windows API或DLL。由于VB最终要将VB代码转变为对API或DLL的调用,那么自己实现这种调用,可减少运行负担,但要小心的是:直接调用的最大问题是降低了捕获错误的能力,若出错,应用程序无法知道。

2. 选用最佳的纪录集对象

VB的Jet引擎提供三类可用来填充纪录的对象,每一种对象都有各自的优点,这取决于数据源。

(1)Table对象的特点

- 对单个表进行读/写访问的最快方法;
- 在执行搜索时具有索引方法的全部优点;
- 能立即执行更新操作;
- 不能用于ODBC数据库、挂接表和跨越多个表;

(2)Snapshot对象特点

- 它能最快地访问多个表;
- 能用SQL语句的SORT和FILTER;
- 能够访问选择判据中的所有数据;
- 只读访问,不能编辑,添加或更新所依赖的数据源中的纪录,而且如果将纪录添加到数据源中,必须重建Snapshot对象以保持当前状态。

(3)Dynaset对象的特点

- 它是最灵活的、能够对多个表读/写访问;
- 能使用SQL语句的SORT、FILTER、JOIN;
- 即使修改了底层的数据源也能保持当前状态;
- 操作过于灵活,是最慢的对象;

对所有对象来讲,“少”就是“好”。选择判据越少,对象操作就越快。对②和③而言,最好使用它能选择指定字段的能力,对①而言,如果只想引用表中的几个字段,那么用大表(即含有很多字段)作为数据源可能没有好处,而用Dynaset对象更快。

3. 加速数据控件

对编写VB应用程序者来讲,数据控件是访问数据的最容易的方法,但它给应用程序增加了许多负担。实际上数据控件本身运行并不慢,是它减慢了FORM-LOAD事件的速度,每个数据控件拥有几十个属性,确实能够降低速度。但在编写FORM-LOAD事件时采用如下方法能改变这种情况。

(1)将ReadOnly属性设置为True去掉隐藏在背后的更新例程;

(2)将Visible属性设置为False不影响控件的性能,但是屏幕上不显示该控件,这样能改进视觉效果。

数据控件使用很方便,但因为性能方面的原因,尽可能减少使用它,建议使用SQL语句来代替数据控件,一条SQL语句,一些命令按钮和一个文本框就能完成数据控件完成的事,并且速度更快,这在删除成组的数据时特别显著。

4. 优化ODBC

开放式数据库连接(ODBC)能访问非Jet支持的数据度,所以能够用于clients/server和模块应用程序的开发。这对程序设计者来讲是很好的福音,但没有一个包含ODBC的应用程序的性能和包含Jet的应用程序的性能一样好。这是因为受到ODBC驱动程序本身引起的,

通过下面方法能提高 ODBC 的查询性能:

(1) 只对做了索引的字段查询。因为索引字段的访问总是要比非索引的快, 如果要排序, 那就对索引的字段排序。

(2) 在建立查询时, 不要用通配符 * 返回不要的字段, 用清晰的列表限制要检索的列。

(3) 不要用 BETWEEN AND、IN 和等号 (=) 操作符限制查询, 除非是访问已做了索引的列。

(4) 避免使用 SORT 和 GROUP BY 表达式, 它们给驱动程序添加了额外的负担。

(5) 向本地数据库挂接远程表, 而不直接打开 ODBC 源中的表。挂接远程表就建立了对它的引用, 这种引用保存在本地数据库中, 保存的引用纪录了远程表的结构, 尽管它在本地, 也能够通过引用访问到远程表。

(6) 对客户/服务器应用程序, 使用 Dynaset 的下列数据高速缓冲特点: Cache Start、CacheSize 和 Fill Cache。高速缓冲允许在本地内存中存储 Dynaset 中的纪录, 同通过 ODBC 驱动程序与获得数据相比, 它可以大大加快访问速度。高速缓冲区越大, 就会越少使用驱动程序。

(7) 不用 find 方法, 而是用一个指定范围的 WHERE 子句建立一个限定行数的记录集, 因为这是 ODBC 服务器的单独操作, 所以比建立记录集然后查找其中指定行效率更高。

(8) 调整 ODBC 超时延迟值。网络阻塞延迟或大量使用 ODBC 服务器能降低对 ODBC 调用的响应时间。可指定在产生超时错误之前等待多长时间。要使应用程序看上去运行更快, 降低超时值, 以便应用程序能够很快获得控制权, 并且能通过不同的更有效的调用或请求用户响应报文处理超时问题。

(9) 其他一些要注意方法

- 只在需要的时候刷新集合;
- 查询有关纪录集而不是重新打开它;
- 用书签取代 Find 方法返回纪录;
- 用默认集合;
- 用 SQL 语句取代代码循环;
- 用集合取代 List 方法;
- 在向 Dynaset 类型的纪录集对象添加纪录对使用 dbAPPENDONLY。

5. 加快数据访问对象(DAO)的运行

要提高 DAO 运行的性能就要减少对磁盘的访问和减少 Jet 对语句的分析次数。采用下面的方法有助于提高 DAO 对象的运行性能。

(1) 用 Seek 方法代替 find 方法。虽然 find 方法是针对表的, 但是 seek 方法利用表的索引, 这使它成为访问

表记录集对象的最快方法。(Seek 不能用于 ODBC 数据库和挂接表)。

(2) 用索引代替排序。访问数据的最快方法是通过索引。如果可能的话, 用满足需要的索引直接打开一个表。若一个表记录集对象要求限制很多, 那么就建立一个查询并使用 ORDER 子句。实际上这是建立一个临时索引, 能大大缩短访问时间。

(3) 用 Seek 或一个查询语句代替过滤器使用。过滤器的使用是一个两步骤的过程, 先获取所有的记录并将它们放到一个记录集中, 然后在此基础上通过建立另一个记录集削减记录集的大小, 这大概是访问数据子集最慢的方法。如果能通过索引访问数据, seek 方法能够给予最快的访问, 如果没有合适的索引, 那么用 SQL WHERE 子句建立一个查询获取数据子集, 其结果与两步骤的记录集过程是一样的, 但是这种方法只有一步。

(4) 如果只有单个用户修改数据就使用 dbDenywrite 标志, 当只有一个用户修改数据时用 dbDenywrite 标志锁定整个表式库, 减少对库加锁次数。这样可提高数据集更新性能, 并且其他用户仍可以用只读方式访问数据库。

(5) 用事务方法管理数据库更新。如果用户对具有很多记录的数据只修改一个字段内容, 那么每修改一条记录就进行磁盘写操作这样加重了系统的运行负担。大家可以采用事务处理方法解决这个问题, 事务缓冲区只有在受到委托之后才会写盘。Begon Trans 方法可启动委托过程, Commit Trans 方法可访问磁盘。如果在用 Commit Trans 方法之前存在错误, 那么用 Rollback Trans 方法能恢复所有的修改, 成组地更新事务能减少对磁盘的访问, 不要每更新一条记录就访问一次磁盘, 只有在完成事务后才访问磁盘。因此, 对上面所讲的例子, 用户可用事务方法先成批修改, 然后启动事务, 进行一次性的写盘, 这样就提高了应用程序的性能。

(6) 除了上面的方法外, 还有如下方法:

- 用 SQL 语句代替循环代码;
- 用预存的查询语句代替动态 SQL 语句;
- 用! 操作符引用记录集中的字段;
- 查询一个记录集而不是重新打开它。

三、结束语

本文讲述了怎样提高 VB 应用程序性能的一些方法, 通过这些方法可以帮助大家提高编程质量。要确信通过使用这些方法能提高 VB 程序的性能, 最好的办法是使用本文中所讲的 VB 源代码剖析器这项测试工具来测试验证。

(来稿时间: 1998 年 8 月)