

基于分层的 Web 系统的性能优化研究与探讨

罗福强¹, 熊永福²

(1. 电子科技大学成都学院, 四川 成都 611731;

2. 东软学院, 四川 成都 611844)

摘要:良好的性能对于一个 Web 系统来说是至关重要的,但 Web 系统的异常复杂性使得其性能优化问题无法通过某个单一技术就能有效解决。文中的主要目的是全面探讨 Web 系统性能的具体优化方法与策略。为此,首先根据 Web 系统的组成提出了分层优化思想,将 Web 系统性能优化问题划分为硬件与网络层、数据与文件层、代码与架构层、页面层、服务器配置层等 5 个层次,然后深入研究了各层影响 Web 系统性能的各种因素,同时详细分析了相应的具体优化方法与策略,从而为软件生产企业和用户找到了切实可行的解决方案。

关键词:Web 系统; Web 服务器; Web 应用程序; 性能优化; 分层优化; 网站

中图分类号: TP31

文献标识码: A

文章编号: 1673-629X(2012)11-0085-04

Web System Performance Optimization Research and Discussion Based on Layering

LUO Fu-qiang¹, XIONG Yong-fu²

(1. Chengdu College of University of Electronic Science and Technology, Chengdu 611731, China;

2. Institute of Neusoft Information and Technology, Chengdu 611844, China)

Abstract: Good performance for a Web system is very important, but Web system is extremely complex, so its performance optimization problem can't be solved effectively by a single technology. The main purpose is a comprehensive study of the Web system performance optimization methods and strategies. Firstly, according to the Web system components, propose a hierarchical optimization idea, and divide the Web system performance optimization problem into 5 layers, including hardware and the network layer, data and file level, code and architecture layer, page layer, server configuration layer, and then delve into the effects of Web system performance on a variety of factors, and analyze in detail the corresponding specific optimization methods and strategies. Lastly find a complete solution for software producers and users.

Key words: Web system; Web server; Web application program; performance optimization; hierarchical optimization; Web site

0 引言

对于一个 Web 系统来说,无论其网站的页面多么精美,功能多么完善,操作多么便捷,如果整个系统的性能不足,对客户端的响应缓慢,甚至经常出现访问超时,或者莫名其妙地打不开网页,那么它的开发就是失败的。因此,良好的性能对一个 Web 系统来说是至关重要的,而如何破解其性能优化问题是软件生产企业和用户都非常关心的。

1 分层优化 Web 系统的性能

一个 Web 系统由 Web 服务器系统、Web 客户端系

统和 Internet 网络组成^[1]。其中,Web 服务器系统除硬件系统之外,还包括操作系统、Web 服务器(如 IIS)、数据库管理系统、Web 应用程序等软件系统。Web 客户端系统比较简单,主要关注 Web 浏览器(如 IE)及浏览器插件等。Internet 网络是指从服务器到客户端之间的所有相关网络系统,包括服务器端的网络、电信级的骨干网络和 Web 客户端的接入网络。因此,一个 Web 系统是异常复杂的,其性能优化问题是无法通过某个单一技术就能有效解决的。

由于大多数的 Web 系统都是“胖服务器-瘦客户端”系统^[2],因此可重点讨论 Web 服务器系统的优化。结合 Web 系统的组成,笔者认为 Web 系统的性能优化应采用分层的思想,进行逐层优化,其基本结构如图 1 所示。

其中,硬件与网络层是整个分层体系架构的物质基础,其主要任务是破解客户端与服务器端通信的瓶

收稿日期: 2012-02-15; 修回日期: 2012-05-18

基金项目: 2010 年第一批四川省科技支撑计划项目(2010GZ0148)

作者简介: 罗福强(1970-),男,四川眉山人,副教授,主要研究方向为计算机网络、软件工程。

颈效应,提高客户端与 Web 服务器之间的数据吞吐量。数据与文件层的任务是整合各种数据存储与访问技术,优化数据存储结构,改善各种数据的访问性能。代码及架构层的任务是优化 Web 应用系统的软件架构和数学模型,提高代码的执行效率。页面层的任务是优化页面内容、外观、布局、特效处理,努力降低客户端与服务器端之间的网络流量。服务器配置层的任务是优化服务器的配置,使之高效地、稳定地为客户端提供访问服务。

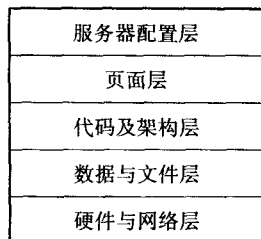


图 1 Web 系统性能的结构

下面将详细分析各层优化方法。

2 硬件与网络层的优化方法

硬件与网络层主要解决客户端与服务端通信的瓶颈问题,可从以下 4 个方面入手解决。

2.1 选择更高的网络带宽

网络带宽是系统性能的关键因素,它决定了系统的吞吐总量^[3]。因为网络带宽不足,在高峰期整个 Web 系统可能响应缓慢或超时,甚至根本无法访问,而此时 Web 服务器自身却可能非常空闲。因此,对于核心交换机和网卡,应该尽量选择千兆或万兆带宽。

2.2 配置足够的内存

物理内存是另外一个重要因素,这是因为内存的消耗会直接随着客户端请求数量的增加而增加。此外,缓存(Cache)和输出缓存(OutputCache)技术严重依赖内存容量。只有足够的内存才能将磁盘的访问频率降到最低,才能明显提升服务器的总体性能。

2.3 使用 RAID 技术

一个 Web 系统包含了大量的网页、程序、数据、图片、音频、视频文件,文件缓冲区技术因为物理内存有限并不能解决快速访问磁盘的问题。为此,要选择性能指标更高的存储介质(如 SCSI 硬盘),并引入 RAID 技术,用多个物理硬盘组成磁盘阵列,进行同时读写数据,以提升磁盘系统的数据吞吐量和容错性^[4]。

2.4 中央处理器的选择

如今,一个 Web 系统包含了大量的服务器端程序,程序之间相互频繁调用,为了更快地执行程序,需要更多地关注 CPU 的性能,应尽量选择多核处理器或多 CPU 系统的专用服务器。当然,出于经济性的考虑,也可采用服务器群集、负载均衡技术。

3 数据与文件层的优化方法

数据与文件层主要解决如何更高效地存取数据或文件问题。在设计 Web 应用系统时,是否采用数据库技术或 XML 技术,使用何种 DBMS,是否使用分布式文件系统等,都直接关系一个 Web 应用系统的性能。

3.1 数据库及其访问

针对数据库而言,可从 4 个方面改善相关设计。

(1) 使用连接池。

访问数据库需要创建连接、打开连接和关闭连接^[5],这些操作需要反复与数据库交互以确保数据库的安全,因而比较耗费服务器资源。为此,可使用连接池技术来改善连接管理,需要时从连接池取出连接对象,关闭时收回,等待下一次的连接请求。当连接池满时,若还要求创建新的连接,则拒绝。可见,连接池本身也构成了一种系统资源,必须合理使用,减少每个连接打开的时间以避免连接池满的发生。

(2) 使用存储过程。

存储过程是存储在 DBMS 中的一组 SQL 语句,它独立于 Web 服务器,在数据库服务器中运行^[6]。使用存储过程的最大好处是,它使得 Web 应用程序在与数据库服务器通信时只需传递存储过程的名字或参数值,而不必传递冗长和复杂的 SQL 语句代码,因此可以有效减少 Web 应用程序与数据库服务器之间的网络通信量。

(3) 优化 SQL 语句。

由于 Web 应用系统大量利用 ADO.NET、ODBC 或 JDBC 等数据访问技术来访问数据库,数据库连接所消耗的系统资源也相当大,SQL 语句运行的时间越长,它所占系统资源的时间也越长^[7]。因此,尽量使用优化过的 SQL 语句以减少执行时间。例如,使用索引、避免子查询、避免返回所有记录或字段等。

(4) 完善数据库的管理。

一个 Web 系统长期运行之后,其数据库必将堆积大量的临时记录或无效数据,如果不及时地删除这些垃圾记录,数据库将变得越来越庞大,其性能就会大幅下降。因此,必须增强 Web 系统的数据管控性能。一方面,使用 SQL 规则和触发器进行数据信息的自动管理和维护,另一方面,完善数据库的管理功能(包括备份、扎账、压缩等),降低人工管控难度。

3.2 文件及其管理优化

针对文件而言,可从以下 3 个方面进行改进。

(1) 良好的目录结构。

为了方便管理,Web 系统的文件必须分类存放,将网页、程序、数据、图片、音视频等文件存放到不同的目录之中。此外,还要注意区别临时文件和永久文件。例如,由访问者上传的未经审核的图片文件可先放到

临时目录,经审核后再移入永久目录。

(2) 垃圾文件管理。

对于已经失效的或过期的文件,Web 系统应具备自动删除功能,否则天长日久系统将堆积大量垃圾文件。此外,要注意数据库记录与文件的匹配问题,要确保在删除数据记录时其相关文件一起删除。

(3) 分布式文件存储。

对于大型 Web 应用系统,可将静态资源文件部署到多个独立的 Web 服务器之中,通过 URL 超连接它们,这样就能成倍地提高整个系统的性能。

4 代码及架构层的优化方法

代码及架构层主要解决如何更高效地执行政程序的问题。可从以下几个方面入手。

4.1 代码的架构

现在,Web 应用系统通常采用 MVC 模式或 N 层架构进行设计^[8]。其优点是提高了 Web 系统的可维护性和代码可重用性,其缺点是增加了 Web 服务器最终要执行的代码总量。因此,对于突发访问量巨大、实时性和事务性要求较强的 Web 系统(例如,全国火车票预订系统、网上银行转账系统),建议不要使用 MVC 模式或 N 层架构进行开发。

4.2 XML Web Service

如今,XML Web Service 技术已经非常成熟。在开发大型 Web 应用系统时,如果能根据系统各模块的耦合性进行合理的分包,将整个系统切割成若干个独立的 XML Web 服务,那么就能实现程序的分布式运行,有效地提高系统性能,同时还能增强网站的安全性。

4.3 云计算

自从 2006 年 8 月 9 日,Google 首席执行官埃里克·施密特(Eric Schmidt)在搜索引擎大会(SES San Jose 2006)首次提出“云计算”(Cloud Computing)的概念^[9]以来,“云计算”技术逐渐走向成熟,在 Web 应用系统中,借助“云计算”所提供的功能强大的服务管理平台和技术(例如,IBM 的蓝云、亚马逊的 Amazon EC2、谷歌的 Google App Engine 和微软的 Windows Azure 等),就能以较低的经济成本获得较好的 Web 系统的整体性能。

4.4 代码本身的优化

代码本身的优化可从以下 5 个方面入手。

(1) 字符串操作。

字符串操作是应用程序的常用功能,如果能够提高字符串操作的性能,就可以提高 Web 应用程序的性能。例如,在 C# 中进行字符串连接时,可直接使用“+”将数字添加到字符串中;当大量拼接字符串时最好使用 StringBuilder 来完成,这样耗时会更短。

(2) 装箱与拆箱。

装箱和拆箱都是需要大量运算的过程。尤其是对值类型的装箱操作,所耗时间比赋值操作长 20 倍,拆箱操作时间是赋值操作的 4 倍^[10]。因此,要尽可能避免装箱和拆箱操作。例如,在 C# 中,可使用值类型的 ToString 方法进行避免;在操作集合元素时建议使用泛型集合以替代普通集合。

(3) 异常处理。

程序通常按正常的控制流程进行编写,但有经验的程序员总会仔细地分析各种可能的异常,例如内存不足、磁盘出错、网络连接中断、数据库无法使用、非法输入等,一旦发生异常,程序将无法正常运行。因此,使用 try-catch-finally 语句必须成为一种习惯。

(4) 释放资源。

为了及时地释放资源,避免系统因死锁而崩溃,对于数据库、XML 文件、网络连接等系统资源,必须保证打开与关闭操作的成对调用。同样,对于跨用户共享的 Application 变量也必须保证 Lock 和 UnLock 的成对调用。释放资源时可使用 using 语句,让系统自动释放。

(5) Cache 对象。

ASP.NET 技术内置了 Cache 对象,该对象将数据缓存在服务器的内存之中,实现跨用户、跨页面的数据共享,通过多种缓存策略实现缓存项的自动管理。例如,借助该对象基于依赖项的缓存策略,当代码需要反复引用磁盘文件或数据库中相同的数据时,可在首次引用时把数据缓存到 Cache 对象之中,同时建立与该文件或数据表的依赖关系,当再次引用时则直接从 Cache 中读取,这样就可以大大提高数据的访问速度,所依赖的文件或数据表一旦更新,则只需重新缓存即可。

5 页面层的优化方法

页面层主要解决如何更有效地设计、更新、浏览网页的问题,可从以下 4 个方面入手。

5.1 会话状态的恰当选择

因为 HTTP 协议是一种无状态的通信协议,所以需要维护客户端请求的会话状态。具体方法有多种:一是直接使用内存,它具有最佳的性能;二是使用状态服务器,可在多 WEB 服务器之间维护会话状态;三是使用数据库技术,以避免丢失重要信息,不过会造成系统性能下降。因此,必须根据实际需求进行恰当选择。

5.2 服务器控件的优化选择

ASP.NET 提供了大量的服务器控件,其强大功能 HTML 标记所不能比拟的,但不足之处在于它将消耗更多的服务器端的资源。因此,过多的使用服务器

控件会影响 Web 系统的性能。使用服务器控件时, 需要注意以下几点:

(1) 减少不必要的服务器控件的使用。

应该优先使用 HTML 标记来显示静态信息, 当无法满足功能要求时可考虑使用 Javascript 脚本, 迫不得已才考虑选择服务器控件。

(2) 禁用不必要的状态视图。

服务器控件的状态视图属性能够自动地在页面往返过程中维护服务器控件的状态, 它减少开发者的工作量, 但是需要占用大量的服务器内存资源。因此, 在不需要状态视图功能时, 应将其 EnableViewState 属性设置为 false。

(3) 合理使用数据源绑定控件。

ASP.NET 的数据源绑定控件允许以零编程模式设计访问数据库的网页。但各控件内置的功能并不完全相同, 一般来说功能越强相应的性能开销就越大^[11], 因此不能滥用数据源绑定控件。例如, GridView 控件具有数据显示、修改、删除、排序、分页等功能, 如果只需显示简单的数据列表, 可使用功能更少的 DataList 或功能最少的 Repeater 控件。

(4) 定义用户控件。

用户控件(.ascx 文件)是一种由用户创建的复合控件, 用来保存各页面相同的操作界面及其功能逻辑, 以实现界面复用和代码复用^[12]。在设计 Web 页面时, 首先尽可能地把相同用户界面从页面中独立出来并创建为用户控件, 然后再将用户控件添加到各页面之中。这样, 不仅可以加快页面的开发速度、降低页面维护和更新的工作量, 还会因用户控件具有只需编译一次的特性而提高 Web 服务器编译各 Web 页面的速度。

5.3 输出缓存

ASP.NET 的输出缓存允许缓存整个 Web 页面或用户控件, 其好处是避免重建页面, 使页面请求响应时间最短, 但要注意, 它可能会大量消耗内存资源, 如果使用不当, 反而会造成性能下降。

5.4 页面布局与效果设计

不要使用表格来定义整个页面, 尤其要避免表格的多重嵌套, 要尽量使用<DIV>来控制页面布局。还要坚持页面内容设计与效果设计分离的原则, 尽量把 CSS 代码和 JavaScript 代码从页面中分离出来单独保存, 这样不仅可以实现 CSS 和 JavaScript 脚本的共享, 还能减少相关代码在 Internet 网络中的字节流量。

6 服务器配置层的优化方法

Web 服务器的配置决定了整个 Web 系统性能的

好坏, 需要注意以下几点:

(1) 在部署之前, 一定要记住禁用调试模式。

(2) 必要时调整应用程序每个辅助进程的线程数。进程中线程太多, 由于额外的上下文交换, 将导致 CPU 资源被消耗在维护线程而不是处理请求之上。

(3) 不要将日志文件与 Web 站点文件存储在同一个磁盘中。这将造成操作日志文件的线程和检索站点文件的线程之间相互干扰。

(4) 在 IIS 中启用 HTTP 压缩可以减少页面加载时间。

(5) 在 IIS 中限制可用的连接请求数量以避免服务器因过载而崩溃。

7 结束语

Web 系统是异常复杂的, 其性能优化问题是无法通过某个单一技术就能有效解决的。文中根据分层优化的思想对其性能优化问题进行了层次结构划分, 同时详细阐述了各层次的优化方法, 为软件生产企业和用户找到了切实可行的解决方案。

参考文献:

- [1] 罗福强. Web 应用程序设计实用教程[M]. 北京: 清华大学出版社, 2010.
- [2] 罗福强, 白忠建, 杨 剑. Visual C#. NET 程序设计教程[M]. 北京: 人民邮电出版社, 2009.
- [3] 岳付强. ASP.NET 企业网站的性能优化研究[J]. 西昌学院学报, 2010(3): 63-66.
- [4] 兰景英, 王永恒. Web 系统性能测试研究[J]. 计算机技术与发展, 2008, 18(11): 90-93.
- [5] 阳小兰, 钱 程, 赵海廷. Web 日志分析系统研究[J]. 计算机技术与发展, 2011, 21(9): 211-215.
- [6] 赵 洁, 沈苏彬. Web 服务访问控制的设计和实现[J]. 计算机技术与发展, 2010, 20(10): 159-162.
- [7] 田 冲, 李兴国. Web 软件系统事务处理模型设计[J]. 计算机技术与发展, 2009, 19(10): 62-65.
- [8] 付燕宁, 金 英, 刘 磊, 等. 基于语义的 Web 服务体系结构[J]. 计算机技术与发展, 2008, 18(3): 28-31.
- [9] 周建鲲. 基于 ASP.NET 应用系统性能研究与探讨[J]. 硅谷, 2009(2): 77-78.
- [10] 李 隽. ASP.NET 应用程序的性能优化[J]. 信息化建设, 2010(7): 36-37.
- [11] 丁 汀, 孙卫民, 刘克冰. WEB 应用的性能问题研究[J]. 中国科技创新导刊, 2008(13): 30-31.
- [12] 明 仲, 尹剑飞, 杨 巍, 等. 一种 Web 系统性能测试框架及其混合建模过程[J]. 计算机研究与发展, 2010(7): 1192-1198.