

提高 ASP.NET 网站速度的措施

张建华

(中国石化齐鲁分公司胜利炼油厂, 山东淄博, 255434)

摘要 ASP.NET 具有功能强大及使用容易上手的特点, 但对于初期使用者, 却存在执行速度慢、运行效率低的问题。本文将从程序开发、数据库优化、客户端处理等方面介绍提高 ASP.NET 网站速度的措施。

关键词 ASP.NET 网站 细节 执行速度

中图分类号: TP393.1 文献标识码: C 文章编号: 1009-9859(2010)03-0263-04

ASP.NET 是 Microsoft.NET 的一部分, 它提供了一个统一的 Web 开发模型。不同于 ASP 即时解释程序, 而是将程序在服务器端首次运行时进行编译, 这样的执行效果, 理论上要比逐条语言解释的 ASP 强很多。但是就初期使用 ASP.NET 改版的几个网站来说, 执行效率却不尽人意。通过总结分析 ASP.NET 架构网站的问题, 在提高 ASP.NET 运行效率方面采取了一些措施, 取得较好效果。

1 编程方面

1.1 避免重构函数中的重复事件

使用 ASP.NET 可以方便地开发所需要的新控件, 那是继承所提供的。比如为了保持开发的风格一致, 开发了 DefaultDataGrid 继承自 DataGrid 将颜色、边框、高度等都作了限定。这样开发人员使用 DefaultDataGrid 时, 所呈现给用户的界面整齐、一致, 开发人员也不必过多的关注网站风格。

使用继承需要注意的是事件的重复定义。例如, 控件 DefaultDataGrid 中定义了 PageIndexChanged 事件, 在具体页面使用时又再次定义了 PageIndexChanged 事件, 结果每次执行 DefaultDataGrid 翻页时, PageIndexChanged 事件执行了 2 遍, 造成了速度的降低。

1.2 使用 ViewState 要注意的几个问题

ViewState 带来许多方便, 但用时间较多。默认情况下页面的 ViewState 的大小为 20 K, 因此, 放入 ViewState 中的值尽量要少。

(1) 使用 ViewState 只存储简单的数据类型。遇到有大量数据类型的, 例如 ArrayList DataSet DataGrid 等, 要使用 Cache 来存储。否则效率降低。

(2) 一些简单的数据类型也可用 Session 替代 ViewState 来存储。使用 Session 时, 需要综合考虑 Session 的有效时间。

(3) 页面中的 ViewState 个数不宜过多, 否则页面加载时需要大量时间来处理。

1.3 尽量减少 AutoPostBack 值的设置

类似 DropDownList TreeView 等控件都有 AutoPostBack 属性。当 AutoPostBack 为 true 时, 每次都会引起页面的刷新, 刷新的同时就会将所有代码都执行一次。因此减少此类控件的 AutoPostBack 的设置也会提高运行效率。

但是只有将 AutoPostBack 设置为 true 才能触发一些事件, 例如 SelectedIndexChanged 等。这时就要看是否有其他方法来代替, 尤其在刷新的数据量很大的时候。

在使用控件 Microsoft.Web.UI.WebControls.TreeView 显示时会遇到这样的问题: TreeView 是显示在页面左侧的一棵导航树, 导航信息是由用户点击具体栏目后查询数据库, 获取相关数据后进行动态填充。右侧是具体内容展示, 含有 Long 型字段, 数据量较大。

收稿日期: 2010-04-27 修回日期: 2010-07-12

作者简介: 张建华 (1976-), 女, 工程师。2000 年毕业于烟台大学计算机及应用专业, 现任齐鲁分公司胜利炼油厂自动化信息所软件开发室主任。电话: 0533-7587412。

设计中没有使用 `SelectedIndexChanged` 事件, `TreeView` 的 `AutoPostBack` 设置为 `False`。用户使用一段时间后提出, 点击导航树后, 被点击的导航节点需设置不同的背景。采用页面地址核对的方式, 找到对应结点, 将背景色进行显示。

1.4 尽量使用 `Page.isPostBack`

通常页面加载时, 要处理控件初始化、数据绑定等, 希望只在页面第一次加载时完成这些工作即可。使用 `Page.isPostBack` 即能达到这个目的。

比如 `DropDownList` `DataGrid` 的数据填充, 这个过程也许会处理大量的数据, 但是回传的时候则不必再进行这部分的处理了。编程时应将这部分代码单独写在一起, 再由 `Page.isPostBack` 来判断, 如果为假就执行, 如果为真就可以省略了。

但是并不是所有的数据绑定都如此, 例如, 实现 `DataGrid` 数据导出至 Excel 表中, 就需要在每次页面加载时将 `DataGrid` 的数据绑定一遍, 否则点击导出按钮时, `DataGrid` 还没有数据, 导出的也只是个空的 Excel 表。

1.5 及时释放资源

一个对象不用时就要关闭、销毁, 尤其是数据库连接、`ExcelWord` 等占用大量资源的类型。

很多时候用户需要对数据进行导入、导出操作, 打开一个 `ExcelApplicationClass` 会占用非常巨大的资源, 因此及时的释放尤为重要。在开发的几个系统中就曾经出现过由于 Excel 操作失败, 没能及时释放, 导致服务器 CPU 占用极高的现象。

1.6 选择合适的控件

选取控件要根据具体的需求和设计来进行。比如在 `SqlDataReader` 和 `DataSet` 的选择上。

`SqlDataReader` 优点是读取数据非常快。如果对返回的数据不需做大量处理的情况下, 建议使用 `SqlDataReader`, 其性能要比 `DataSet` 好很多。缺点是直到数据读完才可 `close` 掉数据库的连接。

`DataSet` 是把数据读出, 缓存在内存中。缺点是对内存的占用较高。如果对返回的数据需做大量的处理用 `DataSet` 较好, 可以减少对数据库的连接操作。优点是只需连接一次就可 `close` 数据库的连接。

因此, 一般情况下读取大量数据时, 对返回数据不做大量处理用 `SqlDataReader`, 对返回数据大量处理用 `DataSet` 比较合适。对 `SqlDataReader` 和

`DataSet` 的选择取决于程序功能的实现。

控件使用上的统计数据: `DataReader` 比 `DataSet` 快 15%; `SqlDataReader` 比 `OleDbDataReader` 快 50%; 用 `DataReader` 的 ASP 风格的表格显示比 `DataGrid` 绑定 `DataReader` 快 60%; 用 `Ordinal` 的 `DataReader` 访问字段比用名字访问快 15%; `DataGrid` 中用 `AutoGenerateColumns = true` 比用显式绑定快 24%。

在页面跳转时, 多使用 `Button` 控件。然而 `Button` 控件有向服务器回传的过程, 回传回来后再进行 `Click` 事件的处理。这样会耽误时间, 不妨使用 `HyperLink` 控件进行页面的跳转。`HyperLink` 控件在页面初始化时已在 `NavigateUrl` 属性中记录了跳转的页面, 点击 `HyperLink` 控件不必进行回传过程, 即可实现页面的跳转。如果单击控件时只是执行页面间的跳转, 可使用 `HyperLink` 控件以达到更加节省访问时间的效果。

1.7 Web.config 设置

注意 `web.config` 中的 `< compilation faultLanguage = “#” debug = “false” />`。`Debug = “true”` 是在调试时需要设置的, 然而正式上线后的程序需设置为 `false`, 这是通常会忽略的。在 `web.config` 的提示语中有这么一句: “设置 `compilation debug = “true”` 以启用 ASPX 调试。否则, 将此值设置为 `false` 将提高此应用程序的运行性能。”

1.8 建立网站特有的主题架构

使用 `Theme` (主题) 和 `Master` (母板), 便于开发者从全局上控制网站的布局, 最大程度上优化了 CSS 代码, 并减少相当程度的代码编写量。如果开发人员与美工人员是分开的, 那么更要如此。

2 客户端方面

2.1 减少服务器端的负荷

既然 ASP.NET 的执行都是由客户端发出请求, 再到服务端执行后返回客户端展示的过程, 那么适当的减少服务器端的负荷会提高运行效率。

JavaScript 是一种由 Netscape 的 LiveScript 发展而来的原型化继承的动态型客户端脚本语言, 主要目的是为了解决服务器端语言遗留的速度问题, 为客户提供更流畅的浏览效果。使用 JavaScript 可以实现以下几点。

(1) 使用 JavaScript 和正则表达式可以限制

输入数据的格式。

(2) 处理一些简单的公式计算。

(3) 构建搜索框的自动补全功能。搜索框可以在用户进行输入的时候提供建议结果。

(4) JavaScript 可以解决布局问题。利用 JavaScript 开发人员可以得知页面上任何一个元素所在的位置和区域, 以及浏览器窗口的尺寸。利用这些信息可以避免元素的重叠和解决不同分辨率下控件的显示问题。

(5) 使用当前流行的 Ajax (异步 JavaScript 和 XML)。通过利用 Ajax, JavaScript 会通过其 XMLHttpRequest 对象使客户端只和服务器交换有用的数据, 而不必要的的数据则不再重新加载, 使得应用过程更自然, 操作更流畅, 给予用户更好的应用体验。

2.2 编写符合标准的 HTML 代码

虽然网页开发工具众多, 但是很少有开发软件能生成标准的 HTML 代码。只有 HTML 代码符合标准, 网站的访问速度才能提高。对软件开发来说, 首先要熟悉这个标准, 这样才能编写出合格的代码, 提高访问速度。

最好的开发模式是将所有的格式和布局都使用 CSS, 并将这些 CSS 代码写在一起, 不仅能提高网站的代码清晰度, 同时能提高下载速度。

2.3 将 JavaScript 代码和 HTML 代码分离

将 JavaScript 代码和 HTML 代码分离, 其效果和 CSS 与 HTML 分离一样。将所有的 JavaScript 代码写在一起, 并尽可能优化 JavaScript 代码, 能提高下载速度。JavaScript 代码不在多, 而在简。

3 数据库方面

3.1 数据库优化

数据库方面采取过程、函数、触发器等方法来替代一些程序中的功能。

(1) 使用过程或 Job 替代定期服务。在胜利炼油厂自动化信息所的“所内动态系统”中就用到存储过程来处理公共事务的每月更新, 建立 Job 每月执行一次此过程。

(2) 建立函数来处理标志字符的存储值与代表含义的转换。

(3) 利用触发器处理父、子表的操作。

(4) 使用事务处理多次对数据库的操作。这样做的好处是: 一组对数据库的操作可以一次性

提交, 如果产生错误, 则全部 Rollback, 提高了正确性和速度。

3.2 优化 SQL 语句

多次打开数据库连接, 进行数据处理需很多时间。比如类似父子表的情况, 尽量使用复合查询的 SQL 语句, 将多次查询缩减到一次查询, 速度会大幅提高。

3.3 建立索引

索引是数据库查询速度快慢的重要影响因素, 在建立表的同时, 就要考虑索引的建立。还有一个重要的地方就是查询 SQL 语句的 order by, 按照 order by 的顺序建立索引会有不同的效果。所以建立索引时要考虑 order by 的顺序, 按照这个顺序建立第一、第二……索引。

3.4 数据库服务器与 WEB 服务器分开

建议 2 台服务器均采用负载均衡的机制, WEB 服务器与数据库服务器分开后, 由原来 1 台机器变为由 2 台机器处理, 减少了 WEB 服务器的负荷。一些数据库需处理的大型过程、Job 等从 WEB 服务器的占用资源中分离出来, 提高了 WEB 服务器的处理速度。同时 WEB 和数据库的运行互不干扰, 提高了数据的安全性。

4 其他方面

4.1 优化网站上的图片

网站上的图片很占资源。很多时候只需要将图片的格式改一下, 例如将 .jpg 改成 .gif 那么图片的大小将减少很多。对于图片清晰度要求不高的情况下, 适当减小图片的大小也是可以的。用户希望在打开网站的时候, 所有东西一瞬间就能下载下来。

4.2 使用 Cookie 存储用户信息

Cookie 很多时候是作为一种通用的数据让不同的语言进行数据交换。如果使用的好, 将减少很多不必要的代码, 并且能很好地结合许多技术。

在开发的系统架构中, 就使用了 Cookie 存储用户登录的许多信息。这样只要第一次登录成功, 下次就不必再次登录。除非 Cookie 中的信息被删除了。

5 结束语

ASP.NET 提供了为建立和部署企业级 Web 应用程序所必需的服务。要使它高效运行, 需要

开发人员深入了解其内部架构及实现机制, 在应用的同时发挥其优势, 避免其劣势。从总体上来说, 就是要使用适合的控件、减少不必要的数据往返于客户端和服务端端的次数和数量、减少 WEB 应用平台和数据库之间的开关次数、发挥数据库

自身的速度优势。ASP.NET 现在已经推出了 4.0 版本, 微软也在不断地改进其运行速度。相信在 ASP.NET 的制造者和使用者的共同努力下, ASP.NET 网站的运行速度将会不断的提高。

COUNTERMEASURES TO IMPROVE THE SPEED OF ASP.NET SITE

Zhang Jianhua

(Shengli Refinery of Qilu Branch Co., SINOPEC, Zibo Shandong 255434)

Abstract ASP.NET possesses powerful functions and is easy to use, but it exists some problems of slow execution speed and low running efficiency for the beginning users. This paper elucidated measures to raise running speed of ASP.NET site from the following aspects, including program development, database optimization, client treatment and etc.

Key words ASP.NET, site, detail, execution speed

欢迎订阅 2011 年《石化技术》

《石化技术》是由中国石化集团资产经营管理有限公司北京燕山石化分公司主办的石油化工综合性技术期刊, 创刊于 1980 年, 1994 年 1 月经国家科委批准国内外公开发刊。

《石化技术》报道范围: 石油炼制、石油化工、合成橡胶、合成树脂、精细化工等领域的工业生产技术总结、技术改造与技术开发成果、科学研究报告、经营管理与发展规划研究、技术进展与发展规划研究、技术进展综述等, 以及与上述领域有关的仪表、设备、机械、自动控制、计算机应用、工业装置检修与维护技术。读者对象: 从事石油化工工业生产和技术开发研究的工程技术人员、科研工作者, 以及高等院校相关专业的师生。

《石化技术》为季刊, 采用国际流行的标准大 16 开本, 内容丰富, 每季末 21 日出版, 国内邮发代号: 82-812, 每期定价 8 元, 全年 32 元, 全国各地邮局均可订阅。(如错过订期, 可随时向编辑部订购)。本刊国内统一连续出版物号: CN 11-3477/TE, 国际标准连续出版物号: ISSN 1006-0235, 由中国国际图书贸易总公司国外发行, 国外代号 1352 Q。

编辑部地址: 北京市房山区燕山岗南路 1 号燕化公司发展研究中心

电 话: (010) 69345106

邮 政 编 码: 102500

传 真: (010) 69342439

E-mail: pcit_yssh@sinopec.com

欢迎惠赐稿件

欢迎刊登广告