

中图分类号: TP393 文献标识码: A

对象池技术在 WebGIS 中的应用

赵珏¹, 张胜

(湖南商学院, 湖南, 长沙, 410205)

摘要: 文章分析了现有 WebGIS 的缺陷, 阐述了对象池技术的思路和特点, 设计了对象池最大阈值计算方法, 提高了现有系统的并发能力, 并且在 .NET 环境下进行了实现和测试。事实证明, 合理设置对象池有利于提升系统性能。

关键词: WebGIS; 对象池; 阈值

The application of Object pool technology in WebGIS design

ZHAO Jue¹, ZHANG Sheng

(Hunan University Of Commerce, Hunan, Changsha, 410205, China)

Abstract: This article analyzes the shortcomings in the existing WebGIS and introduces the concept of the object pool technology and its features. Based on Object Pool, the maximum pool threshold calculation method is proposed to improve the concurrent capability in the existing systems. An experiment system is developed and implemented to test the calculation method in .Net environment. The test findings support that the proper object pool settings will increase the system performance.

Keywords: WebGIS; object pool; threshold

1 引言

WebGIS 是在 Internet 或 Intranet 网络环境下的一种存储、处理、分析、显示与应用地理信息的计算机信息系统, 它的基本思想就是在互联网上提供地理信息, 让用户通过浏览器浏览获得地理信息系统中的数据和功能服务。目前 WebGIS 发展速度很快, 应用也非常广泛, 利用它可获取各种地理信息, 这给人们的生活带来了极大的便利[1]。但是, 现有的 WebGIS 系统目前仍存在一些技术缺陷, 制约了其应用的扩大和业务需求复杂度的增加。比如: 并发访问能力不适应日益扩大的系统规模^[1]。

由于 WebGIS 应用涉及数据量大, 处理复杂, 在服务器端需要实现大量的业务逻辑, 加重了服务器端的负担, 因此, 并发访问能力和响应速度一直是 WebGIS 软件的主要改进方向。

2 对象池阈值的设计

目前, 大部分的 WebGIS 仍采用的是 CGI/Sever API 方法。在实现过程中, 每一次请求就会创建一个进程服务。本文拟应用对象池技术予以改进。对象池的基本思路是^[2, 3]: 一次性创建多个可重复使用的对象, 从而在一定程度上减少频繁创建对象所造成的开销, 这对于大众化的 GIS 应用服务的实现是非常有帮助的。

对于每个 ASP.NET 应用程序, 都可以在服务器端创建一个 COM+ 对象池。当 ASP.NET 请求开始时, WebSessionActivator 从对象池获得 Session 实例, 如果不再需要对象则将该对象放回对象池, 等待下一个请求。这就意味着, 在使用前就创建多个 Session 对象, 使用过程中 Session 对象不会再被创建和释放, 这些 Session 对象一直可用于 Web 请求。

¹收稿日期: 2009-04

作者简介: 赵珏, 女, 1980 年 4 月生, 湖南怀化人, 软件工程硕士, 讲师, 研究方向为网络软件与应用。

CGI/Sever API 方式中请求服务越多, 进程实例越多, 使用的资源就越多。对象池可以帮助降低创建并发进程实例的数量, 优化服务器的资源。

当面对管理成千上万个客户端并发请求的大规模系统时, 确认能够建立对象池最大阈值是非常重要的。一般来说, 大的对象池有利于提高系统的性能, 对占用资源也越多, 当系统资源耗尽时, 系统反而会停止响应。因此, 需要监控连接池和应用程序的性能, 以便确定系统的最佳池大小。同时, 最佳大小还取决于用来运行该系统的硬件。

经过反复的研究、演算, 得出对象池最大阈值计算公式 (1) :

$$\frac{\text{Mem} - \text{Op} - \text{Oth} - \text{Res}}{K} \quad \text{式 (1)}$$

假设系统的内存空间为 Mem, 操作系统用去 Op, 其他程序和服务占用的空间为 Oth, 需要预留的空间为 Res (一般来说, 不要完全使用所有的内存, 否则会出现不可预料的问题, 如: 溢出), 每个对象池对象耗费的空间为 K, 那么对象池最大值阈值应该小于公式 (1), 参数都可以通过测量得出, 具有可操作性。

对象池化的目的, 是要通过减少对象生成的次数, 减少花在对象初始化上面的开销, 从而提高整体的性能。然而, 池化处理本身也要付出代价。因此, 并非任何情况下都适合采用对象池化。

当给出的一组条件相同时, 使用与不使用对象池技术的实际性能是有差异的^[4]。实测结果表明: 对于类似 Point 这样的轻量级对象, 进行池化处理后, 性能反而下降, 因此不宜池化; 对于类似 Hash table 这样的中量级对象, 进行池化处理后, 性能基本不变, 一般不必池化 (池化会使代码变复杂, 增大维护的难度); 对于类似 JPanel 这样的重量级对象, 进行池化处理后, 性能有所上升, 可以考虑池化。根据使用方法的不同, 实际的情况可能与这一测量结果略有出入。在配置较高的机器和技术较强的虚拟机上, 不宜池化的对象的范围可能会更大。

3 系统实现

可以通过程序配置对象池大小的最小值和最大值。最小值指定了首次访问对象池时应该创建的 ISession 实例的数量。最大值指定了在对象池中要创建的 ISession 实例的最大值。

可以通过在服务器控制面板中“组件服务”管理工具的控制台树中, 右键单击要置入对象池中的 COM+ 应用程序, 然后单击“属性”, 在“共用与回收”选项卡上, 根据要运行的应用程序实例的数目在“应用程序共用”下输入“池大小”的值 (如图 1)。

池大小的计算见式 (1)。

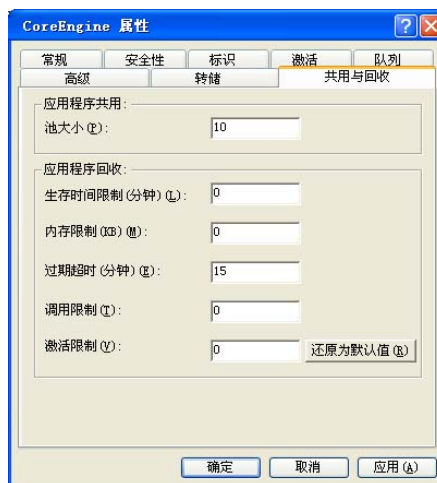


图 1 对象池大小的设置

也可以通过程序控制，下面是 VB 代码：

```
Function SetMyApplicationPooling( strApplicationName As String, lngPoolValue As Long ) As Boolean
```

```

' strApplicationName 表示应用程序名, lngPoolValue 表示设置对象池大小
SetMyApplicationPooling = False ' 初始化
Dim objCatalog As COMAdmin.COMAdminCatalog
Dim objAppCollection As COMAdmin.COMAdminCatalogCollection
Dim objApplication As COMAdmin.COMAdminCatalogObject
Set objCatalog = CreateObject("COMAdmin.COMAdminCatalog")
Set objAppCollection = objCatalog.GetCollection("Applications")
objAppCollection.Populate
For Each objApplication in objAppCollection
If objApplication.Name = strApplicationName Then
objApplication.Value("ConcurrentApps") = lngPoolValue
MsgBox strApplicationName & " pooling value set to " & lngPoolValue
Exit For
End If
Next
objAppCollection.SaveChanges
Set objApplication = Nothing
Set objAppCollection = Nothing
Set objCatalog = Nothing

```

End Function

4 系统测试

系统的测试环境如下：服务器为 2.4 GHz/512 MB；操作系统为 Windows 2003；网络环境为 100M 全双工；测试数据为 3.39M 地图数据，生成图像大小为 746×514 像素 GIF。

(1) 利用 ACT 工具录制压力测试脚本。录制过程从访问该系统开始，进行各种简单的操作如地图的缩放、拖动等。

(2) 开始模拟测试过程。浏览器同时连接数定为 25，对象池数从 5 开始，每次增加对象池数 5 个，然后记录客户端机器、服务器端机器的性能和相关的网络响应时间，每次测试时间为 3 分钟。

(3) 对测试结果进行统计分析，从末字节响应时间看整个系统的负载能力。

测试结果见下图 2。

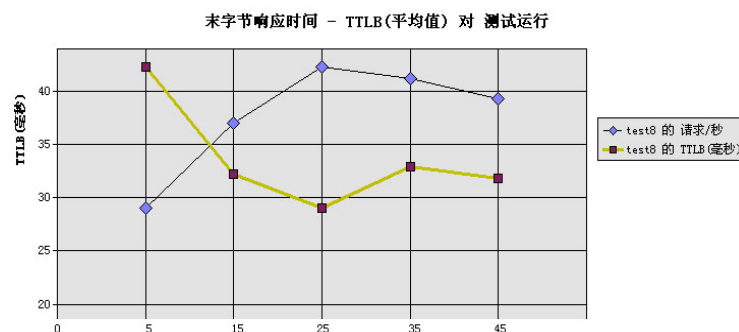


图 2 对象池测试报告图

从图 2 可以看出：当对象池增加到 25（注意：测试时设定浏览器同时连接数为 25），系统达到最佳性能，对象池数超过 25 后，性能趋于稳定，这时增加池的大小作用不大。同时，

设定过多对象池耗费了系统的内存资源，使得性能反而有下降的趋势。

5 结论

本文研究了对象池技术对服务器资源的优化，设计了对象池最大阈值计算公式。经过改变系统的对象池大小、增加并发访问的连接数测试系统的综合表现能力，证明改进后系统具有较好的稳定性和并发性，为完善 WebGIS 系统做出了一定的努力。

[1] 韩双旺, 王心源, 李德录. 利用缓存优化基于 ASP.NET 的 WebGIS 性能. 微计算机信息, 2006, 22 (8):152-155

[2] 朱江, 宋关福, 钟耳顺. 基于 Web Services 和 .NET 技术的新一代 WebGIS 研究. 地理信息世界, 2004, 2 (3): 17-20

[3] 刘南, 刘仁义. 基于 COM+的分布式 WebGIS 架构及实现方法. 中国图象图形学报, 2004, 9 (1): 99-104

[4] 何江, 刘仁义, 刘南. 基于 COM+的 WebGIS 系统设计与实现. 浙江大学学报, 2004, 31 (6): 712-715

[5] Dr. Cliff Click. Performance Myths Exposed. JavaOne, 2003,

作者介绍: 赵珏 (1980—), 女, 湖南怀化人, 软件工程硕士, 讲师, 研究方向为网络软件与应用。

Biography: Zhao Jue(1980 -), Female, Huaihua Hunan, Master of Software Engineering, Lecturer, Mainly for research on network software and applications.