

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

开发高性能的WebService应用

王兴明

微软认证金牌讲师 (MCT)

微软最有价值专家 (MVP)

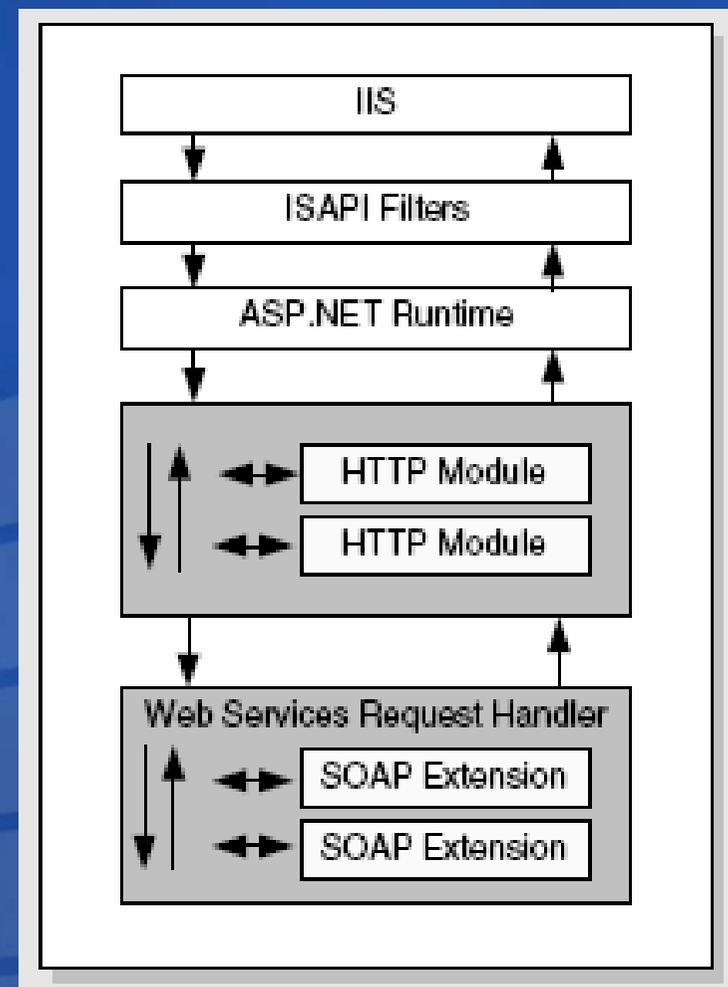
- **WebService性能概述**
- **WebService性能最佳实践**
 - 连接与线程优化
 - WebMethod的优化
 - 高效地调用WebMethod
 - 设置超时
 - 缓存
 - 状态管理

性能是一项功能

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 设计时就考虑性能
- 不要在事后再加入性能!
- 在项目开发的整个过程中反复测试
- 两种量化Web性能的方法:
 - 1) 机器吞吐率 (requests/sec)
 - 2) 响应时间 (time to first/last bytes)



面向对象

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- 应用于同一平台和运行环境
- 共享类型而不是schemas
- 采用便宜, 透明通信
- 对象的标识与生命周期由系统维护
- 客户机和服务器的同步的部署
- 容易概念化, 因而提供一个自然的模型
- 一般不需要状态管理
- 应用于一个可预测的序列、期限 和结果
- 目标是远程透明地使用方法和类型

面向服务

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- 应用于异构平台和运行环境
- 共享schemas而不是类型
- 采用低成本, 明确通信
- 服务是自治的: 安全和失效是隔离
- 允许的连接, 分离部署客户机和服务器
- 基于软件组件和分布的对象, 依赖服务的协议
- 拥有并维护状态
- 基于消息, 异步的, 和长通信
- 目标是提供服务隔离和调用的标准。

WebService性能设计的考虑

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- 设计"大块头"的接口减少往返。
- 基于消息的编程而不是远程过程调用(RPC)。
- 使用XML字符串作为参数。
- 尽量使用原始数据类型参数。
- 避免在调用之间维护服务器状态。
- 考虑为复杂的WebMethod提供输入校验。
- 考虑对WebMethod的结果进行缓存。
- 选择适用的大数据包传送方式。
- 避免调用本地的WebService。

消息 VS RPC

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

■ RPC

```
Serv.SendItemsToBePurchased(Array[] items);  
Serv.ShippingAddress(string Address);  
Serv.CheckOut();
```

■ 消息

```
//Client
```

```
string msg = "<Items>...</Items>";  
MyMethod(msg);
```

```
//Server
```

```
[WebMethod]
```

```
void MyMethod(string msg){ ... }
```

议程

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- WebService概述
- WebService性能最佳实践
 - 连接与线程优化
 - WebMethod的优化
 - 高效地调用WebMethod
 - 设置超时
 - 缓存
 - 状态管理

连接

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 配置maxconnection 属性。
- 对WebService按优先级分配连接资源
- 使用单一身份进行调用。
- 为集成认证使用
UnsafeAuthenticatedConnectionSharing 。
- 为基本验证使用PreAuthenticate 。

配置maxconnection 属性

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

```
<connectionManagement>  
  <add address="*" maxconnection="12"/>  
</connectionManagement>
```

```
<connectionManagement>  
  <add address="WebServiceA" maxconnection="8">  
  <add address="WebServiceB" maxconnection="4">  
</connectionManagement>
```

使用单一身份进行调用

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

```
serv = new WebService1();  
serv.PreAuthenticate=true;  
ICredentials conCredentials =new  
    NetworkCredential("UId", "Pwd", "NPSTest" );  
serv.Credentials = conCredentials;  
serv.AllowAutoRedirect=false;  
serv.ConnectionGroupName = "SameForAllUsers";
```

线程

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 优化线程池减少竞争。
- 考虑minIoThreads 和minWorkerThreads 为断断续续的集中的负载。

Configuration setting	Default (.NET 1.1)	Recommended value
maxconnection	2	12 * #CPUs
maxIoThreads	20	100
maxWorkerThreads	20	100
minFreeThreads	8	88 * #CPUs
minLocalRequestFreeThreads	4	76 * #CPUs

议程

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- WebService概述
- WebService性能最佳实践
 - 连接与线程优化
 - **WebMethod的优化**
 - 高效地调用WebMethod
 - 设置超时
 - 缓存
 - 状态管理

WebMethods

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 使用原始类型参数。
- 考虑使用缓冲。
- 将WebMethod的返回缓存。
- 只在需要会话状态时才使用它。

大数据包的处理

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- 使用字节数组作为参数或返回值
- 返回一个URL
- 使用串行化流

```
<configuration>
  <system.web>
    <httpRuntime maxRequestLength="8096"
      useFullyQualifiedRedirectUrl="true"
      executionTimeout="45"/>
  </system.web>
</configuration>
```

串行化

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 使用XmlIgnore减少串行化。
- 减少往返。
- 考虑使用XML 压缩。

```
public class MyClass
{
    // The str1 value will be serialized.
    public string str1;
    /* This field will be ignored when
    serialized--unless it's overridden. */
    [XmlIgnoreAttribute]
    public string str2;
}
```

异步 Webmethod

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- 用异步 Webmethods 对 I/O 操作。
- 当 Webmethods 依赖于工作线程时, 不要使用异步。

```
//webservice  
[WebMethod]  
IAsyncResult BeginMyProc(...)
```

```
[WebMethod]  
EndMyProc(...)
```

```
//the WSDL will show the method as  
MyProc(...)
```

单工通信

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 如果不需要返回, 考虑使用OneWay属性。

```
[SoapDocumentMethod(OneWay=true)]  
[WebMethod(Description="Return immediately")]  
public void SomeMethod()  
{...}
```

COM 交互

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 建议: 尽量避免交互
- 最好将COM代码移植到.NET
 - 可能会很昂贵, 尤其对于数组操作
 - “受管”与“不受管”转换
- 注意单元线程组件
 - 缺省情况 ASP.NET 使用MTA线程
 - 损害STA 组件性能

注意所有VB6组件!

议程

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- WebService概述
- WebService性能最佳实践
 - 连接与线程优化
 - WebMethod的优化
 - 高效地调用WebMethod
 - 设置超时
 - 缓存
 - 状态管理

异步调用

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 当有并发任务时例用异步调用WebService。
- 使用异步调用多个不相关的WebService。
- 异步叫调用WebService可以增加UI 响应。

您的潜力. 我们的动力

Microsoft[®]
微软(中国)有限公司

Demo

异步调用WebService

议程

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- WebService概述
- WebService性能最佳实践
 - 连接与线程优化
 - WebMethod的优化
 - 高效地调用WebMethod
 - 设置超时
 - 缓存
 - 状态管理

超时

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 适当地设置代理的超时。
- 设置您的ASP.NET 超时大于您的WebService的超时。
- 如果页面超时，则放弃页面与WebService的连接。

```
MyWebServ obj = new MyWebServ();  
obj.Timeout = 15000; // in milliseconds
```

议程

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- WebService概述
- WebService性能最佳实践
 - 连接与线程优化
 - WebMethod的优化
 - 高效地调用WebMethod
 - 设置超时
 - 缓存
 - 状态管理

缓存

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 对时效性不强的数据进行缓存。
- 考虑提供与缓存所相关的信息给客户。
- 考虑使用边界缓存，如ISA上的缓存。

您的潜力. 我们的动力

Microsoft[®]
微软(中国)有限公司

Demo

缓存WebServicer的输出

议程

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- WebService概述
- WebService性能最佳实践
 - 连接与线程优化
 - WebMethod的优化
 - 高效地调用WebMethod
 - 设置超时
 - 缓存
 - 状态管理

状态管理

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 只在需要会话状态时才使用它。
- 减少应用状态与服务器的亲合力。
 - InProc
 - StateServer
 - SQLServer

您的潜力. 我们的动力

Microsoft[®]
微软(中国)有限公司

Demo

WebService使用Session状态

回顾

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- WebService概述
- WebService性能最佳实践
 - 连接与线程优化
 - WebMethod的优化
 - 高效地调用WebMethod
 - 设置超时
 - 缓存
 - 状态管理

再看一次

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- 设计"大块头"的接口减少往返。
- 基于消息的编程而不是远程过程调用(RPC)。
- 使用XML字符串作为参数。
- 尽量使用原始数据类型参数。
- 避免在调用之间维护服务器状态。
- 考虑为复杂的WebMethod提供输入校验。
- 考虑对WebMethod的结果使用缓存。
- 选择适用的大数据包传送方式。
- 避免调用本地的WebService。

获取更多MSDN资源

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- **MSDN中文网站**

<http://www.microsoft.com/china/msdn>

- **MSDN中文网络广播**

<http://www.msdnwebcast.com.cn>

- **MSDN Flash**

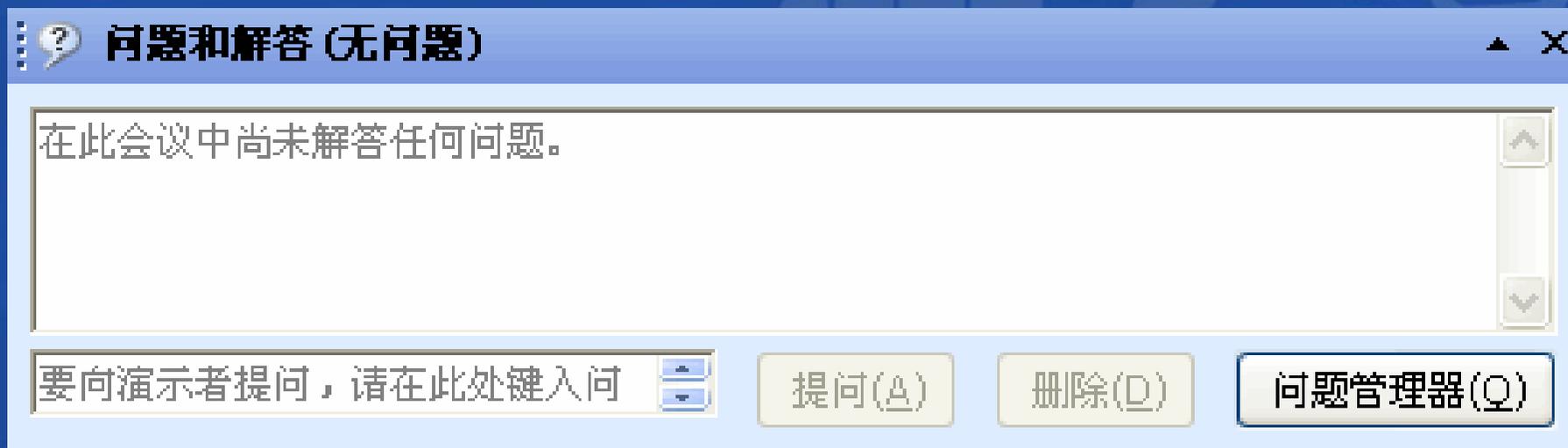
<http://www.microsoft.com/china/newsletter/case/msdn.aspx>

- **MSDN开发中心**

<http://www.microsoft.com/china/msdn/DeveloperCenter/default.msp>

Question & Answer

如需提出问题，请单击“提问”按钮并在随后显示的浮动面板中输入问题内容。一旦完成问题输入后，请单击“提问”按钮。



您的潜力，我们的动力

Microsoft[®]
微软(中国)有限公司

Microsoft[®]

msdn


MSDN Webcasts