基于ASP.NET的Web程序中MSMQ消息传递技术的应用

辛士光 哈尔滨理工大学计算中心 150080

摘 要

针对基于ASP.NET开发的Web应用系统，学生作业等大量信息传递时存在的问题，提出了应用MSMQ技术的解决方案，以及在ASP.NET应用程序中队列的建立、消息发送、消息读取的实现方法。

关键词

Web应用；网络教学系统；消息队列；异步消息传递；ASP.NET

我们基于ASP.NET开发了Web应用系统，在大量并发用户访问某一应用时，比如网络教学辅助系统中大量学生作业的集中上交，常常会遇到这些情况：某个ASP.NET程序花费了过长的时间而导致在客户端过期或服务器上阻塞了大量的死队列，系统出现错误信息“服务器太忙”，而导致访问失败。我们在使用ASP.NET开发Web应用程序时应用Microsoft Message Queue (MS MQ) 技术解决了此类问题。

一、什么是Microsoft Message Queue

MSMQ(MicroSoft Message Queue，微软消息队列) 是运行在Windows NT的服务，在多个不同的应用之间实现相互通信的一种异步传输模式。相互通信的应用可以分布于同一台机器上，也可以分布于相连的网络空间中的任一位置。它的实现原理是：消息的发送者把自己想要发送的信息放入一个容器中（我们称之为Message），然后把它保存至一个系统公用空间的消息队列(Message Queue)中；本地或者是异地的消息接收程序再从该队列中取出发给它的消息进行处理。这些队列能够确保MSMQ的传送，而不管当前网络连接的状况如何。在基于ASP.NET的应用，消息的发送者通常是Web Server(IIS)。

采用MSMQ带来的好处是：由于是异步通信，无论是发送方还是接收方都不用等待对方返回成功消息，只要消息成功发送出去，就可以认为处理完成，就可以执行余下的代码，因而大大地提高了事务处理的能力。当信息传送过程中，信息发送机制具有一定功能的故障恢复能力。MSMQ的消息传递机制使得消息通信的双方具有不同的物理平台成为可能。如：Sun Solaris、HP-UNIX、OS/2、VMS、AS/400、RS/6000等平台。

在微软的.NET平台上利用其提供的MSMQ功能，可以轻松创建或者删除消息队列、发送或者接收消息、甚至于对消息队列进行管理。在其产品中，提供了MSMQ类库“System.Messaging.dll”。它包含了两个类分别对消息对象和消息队列对象进行操作。

二、如何在ASP.NET编程时应用MSMQ技术

在能够使用MSMQ功能之前，你必须确定你的机器上安装了MSMQ消息队列组件，并确保服务正在运行中。ASP.NET编程时，应在头部使用以下代码将MSMQ类库引入ASP.NET文件。

```
<%@ Assembly Name=”System.Messaging”%>
<%@ Import NameSpace=”System.Messsaging”%>
```

（1）通过Create方法创建使用指定路径的新消息队列

```
public static void Createqueue(string queuePath)
{
MessageQueue.Create(@".private$myQueue");
}
```

（2）连接消息队列并发送消息到队列

```
public static void SendMessage()
{
// 连接到本地的队列
MessageQueue myQueue = new MessageQueue(".private$myQueue");
Message myMessage = new Message();
myMsg.Body = "消息内容";
myMsg.Formatter = new XmlMessageFormatter(new Type[] { typeof(string) });
// 发送消息到队列中
myQueue.Send(myMsg);
}
```

（3）连接队列并获取队列的全部消息

```
public static void GetAllMessage()
{
// 连接到本地队列
MessageQueue myQueue = new MessageQueue(".private$myQueue");
Message[] message = myQueue.GetAllMessages();
XmlMessageFormatter formatter = new XmlMessageFormatter(new Type[] { typeof(string) });
for (int i = 0; i < message.Length; i++)
{
message[i].Formatter = formatter;
Console.WriteLine(message[i].Body.ToString());
}
}
```

三、结论

本文所给出的在ASP.NET应用开发中采用MSMQ技术的异步Web服务模型，很好地解决了异步消息传递，为通信双方提供了松散的异步交互环境，该模型为像网络教学系统这样的Web应用中，大量并发用户的数据传输和处理带来了很好的可靠性和灵活性，能够很好地解决由于ASP.NET程序运行超时和服务器等待而导致的访问失败。

作者简介

辛士光，工学硕士，主研方向：网络和数据库、计算机基础教学。