Durgaprasad Gorti's WebLog

Only one usage of each socket address (protocol/network address/port) is normally permitted ★★★★★

Lets say that you are invoking a web service from another web service. Both are on the same box. You might be making authenticated
or unauthenticated calls and perhaps you are setting the KeepAlive = false.
Intermittently, (under load) you might get "Only one usage of each socket address (protocol/network address/port) is normally permitted (typically under load)."
You might be wondering why are you getting a *SOCKET* Exception...

Here is the scoop
1. When you make authenticated calls, the client is closing connections. And when you are making authenticated calls
repeatedly to the same server, you are making and closing connections repeatedly
2. The same might happen when you are making regular http [un authenticated] calls but setting keep-alive = false.

When a connection is closed, on the side that is closing the connection the 5 tuple
{ Protocol, Local IP, Local Port, Remote IP, Remote Port} goes into a TIME_WAIT state for 240 seconds by default.
In this case, the protocol is fixed - TCP
the local IP, remote IP and remote PORT are also typically fixed. So the variable is the local port.
What happens is that when you don't bind a port in the range 1024-5000 is used.
So roughly you have 4000 ports. If you use all of them in 4 minutes - meaning roughly you
make 16 web service calls per second for 4 minutes you will exhaust all the ports. That is the cause of this exception.

OK now how can this be fixed?
1. One of the ways is to increase the dynamic port range. The max by default is 5000. You can set this up to 65534.
HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\MaxUserPort is the key to use.

2. The second thing you can do is once the connection does get into an TIME_WAIT state you can reduce the time it is
in that state, Default is 4 monutes, but you can set this to 30 seconds
HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\TCPTimedWaitDelay is the key to use.
Set this to 30 seconds

3. If you don't have access to registry you can do this through code.
In System.Net 2.0 we addeed what is called a BindIPEndPOintDelegate
What this does is to give you a chance to choose the local end point for the
connection that is being made.
See http://blogs.msdn.com/malarch/archive/2005/09/13/466664.aspx
for more info.

Using this, the following sample code attempts to choose between 5001 and 65534
and wrap around when we reach 65534.

```
Req.ServicePoint.BindIPEndPointDelegate
 = new BindIPEndPoint(BindIPEndPointCallback);
public static IPEndPoint BindIPEndPointCallback(ServicePoint servicePoint,
     IPEndPoint remoteEndPoint,
     int retryCount)
{
  int port = Interlocked.Increment(ref m_LastBindPortUsed); //increment
  Interlocked.CompareExchange(ref m_LastBindPortUsed, 5001, 65534);
  if(remoteEndPoint.AddressFamily == AddressFamily.InterNetwork) {
 return new IPEndPoint(IPAddress.Any,port);
  }
  else {
 return new IPEndPoint(IPAddress.IPv6Any,port);
  }
}
```

Durgaprasad Gorti
Durgaprasad.Gorti@microsoft.com.dontspam
Test Lead
System.Net

Posted: Sunday, September 18, 2005 12:03 AM by dgorti

Comments

Kirk Allen Evans' Blog said:

I love posts like this that explain how to tune System.Net.&amp;nbsp; System.Net.ServicePoint has long been...

# September 18, 2005 7:56 AM

joncole said:

Note that this BindIPEndPoint functionality is new to v2.0 of the .Net Framework.

# October 10, 2005 12:55 PM

dmitryr's blog said:

I've posted an update to Cassini v2 sources that fixes a leak of Connection objects pointed out to me...

# March 9, 2006 11:21 PM

Anonymous comments are disabled