WCF service reaching high memory usage on first call

Asked 12 years, 1 month ago Modified 10 years, 8 months ago Viewed 7k times

6

We have a WCF service as a BL service. The service is in Mixed Transport mode, have more than 10 different endpoint, binded by BasicHttpBinding, with different contracts and the same address for all of them. The service runs on its on application pool on IIS-7.

The problem is, the service works fine, but after the first call, even the get the WSDL, the memory usage of the w3wp.exe goes straight to 300 mega, the service memory usage keeps to increase constantly, taking over all the physical memory of the server (98 - 100 %). We didn't get out of memory exception, but this situation slows down other applications and the service so we need to manually refresh the application pool once every couples of days. I already tried to use memory profiling tool and didn't find any leads to the cause of the problem.

Did anyone encounter this issues? and if yes, what did you do?

Additional information:

- The BL service is located above a DAL framework based on NHibernate, we've already ruled out the memory leak is originating from there.
- Config file

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
    <appSettings>
    </appSettings>
    <system.web>
        <compilation debug="true" targetFramework="4.0" />
        <httpRuntime maxRequestLength="20000"
requestLengthDiskThreshold="20000" />
    </system.web>
    <system.serviceModel>
        <behaviors>
            <serviceBehaviors>
                <behavior name="DefaultServiceBehavior">
                    <serviceMetadata httpGetEnabled="true" />
                    <serviceDebug includeExceptionDetailInFaults="true" />
                </behavior>
            </serviceBehaviors>

            <endpointBehaviors>
                <behavior name="AnonymousBehavior">
                </behavior>
            </endpointBehaviors>
      </behaviors>

      <bindings>
          <basicHttpBinding>
              <binding name="SecureBinding"
               closeTimeout="00:10:00"
               openTimeout="00:10:00" receiveTimeout="00:10:00"
```

```
                sendTimeout="00:10:00" allowCookies="true"
                hostNameComparisonMode="StrongWildcard" maxBufferSize="65536000"
                maxBufferPoolSize="524288000" maxReceivedMessageSize="65536000"
                transferMode="Buffered">
                   <readerQuotas maxDepth="20000000"
                    maxStringContentLength="8192000"
                    maxArrayLength="16384000"
                    maxBytesPerRead="4096000"
                    maxNameTableCharCount="16384000" />
                      <security mode="None">
                          <transport clientCredentialType="None"/>
                      </security>
             </binding>
         </basicHttpBinding>
    </bindings>

<services>
<service name="BL.Services.MyService"
behaviorConfiguration="DefaultServiceBehavior">

<endpoint address=""
binding="basicHttpBinding"
bindingConfiguration="SecureBinding"
bindingNamespace="Security/Anonymous"
behaviorConfiguration="WithSecurityContextInspector"
contract="BL.Services.Contracts.IAnonymousClaimsService" />

<endpoint address=""
binding="basicHttpBinding"
bindingConfiguration="SecureBinding"
bindingNamespace="Domain/App"
behaviorConfiguration="WithSecurityContextInspector"
contract="BL.Services.Contracts.IAppService" />

<endpoint address=""
binding="basicHttpBinding"
bindingConfiguration="SecureBinding"
bindingNamespace="Domain/App"
behaviorConfiguration="WithSecurityContextInspector"
contract="BL.Services.Contracts.IAttachmentService" />

<endpoint address=""
binding="basicHttpBinding"
bindingConfiguration="SecureBinding"
bindingNamespace="Domain/Site"
behaviorConfiguration="WithSecurityContextInspector"
contract="BL.Services.Contracts.ISecurityService" />

<endpoint address=""
binding="basicHttpBinding"
bindingConfiguration="SecureBinding"
bindingNamespace="Domain/Transaction"
behaviorConfiguration="WithSecurityContextInspector"
contract="BL.Services.Contracts.ITransactionService" />

<endpoint address=""
binding="basicHttpBinding"
bindingConfiguration="SecureBinding"
bindingNamespace="Domain/ActiveDirectory"
behaviorConfiguration="WithSecurityContextInspector"
contract="BL.Services.Contracts.IActiveDirectoryService" />

<endpoint address=""
binding="basicHttpBinding"
```

```
bindingConfiguration="SecureBinding"
bindingNamespace="Domain/Report"
behaviorConfiguration="WithSecurityContextInspector"
contract="BL.Services.Contracts.IReportService" />

<host>
<baseAddresses>
<add baseAddress="//MyService.svc" />
</baseAddresses>
</host>
</service>
</services>

<serviceHostingEnvironment multipleSiteBindingsEnabled="true" />
</system.serviceModel>
<system.webServer>
<modules runAllManagedModulesForAllRequests="true" />
<defaultDocument>
<files>
    <add value="MyService.svc" />
</files>
</defaultDocument>
</system.webServer>
</configuration>
```

c# .net wcf web-services iis

Share Improve this question

Follow

edited Nov 19, 2012 at 15:24

asked Nov 19, 2012 at 9:47

Mr Mush

1,538 ● 3 ● 25 ● 39

I had memory issues with WCF that refused to go away. The documentation around the myriad of configuration options for WCF is pretty poor IMO. Never got to the bottom of it and ended up writing my own remoting framework which put me back in control of memory consumption. Hope you find a better solution. – spender Nov 19, 2012 at 9:59

whats your config looking like? particularly around the objectItemsInGraph option etc? Are you running concurrently? How many active sessions etc are you allowing? All of these have an effect. – Chris Nov 19, 2012 at 10:21

3 Answers

Sorted by: Highest score (default)

6

The 300MB is not unusual as AnkMannen notes. Heavily used service can easily plateau out around 700MB or more. Your second observation of the service consuming most available server memory but not triggering an out of memory exception is likely due to the **non-default** config values:

```
binding:
maxBufferSize="65536000"
maxBufferPoolSize="524288000"
maxReceivedMessageSize="65536000"
```

```
transferMode="Buffered"

readerQuotas:
maxDepth="20000000"
maxStringContentLength="8192000"
maxArrayLength="16384000"
maxBytesPerRead="4096000"
maxNameTableCharCount="16384000"
```

You are actually configuring WCF to consume excessive memory with the values you have chosen. Unless you have encountered a specific condition that required changing the default value for any of those attributes, don't change them. The only value I routinely change is `maxReceivedMessageSize` from a default of 64K to around 1 to 2 MB, if unavoidable. If you're routinely slinging around messages that are bigger than 3 MB, you should reconsider your data contract design. A lot of the performance issues WCF is accused of are actually misconfigurations not performance problems in WCF itself.

Share Improve this answer

Follow

edited Nov 19, 2012 at 14:46

answered Nov 19, 2012 at 14:29

Sixto Saez

12.7k ● 5 ● 47 ● 53

Thanks for your comment, I'm aware of this issue, we already checked it and this is not the issue. – Mr Mush Nov 19, 2012 at 15:53

5

After a long search we found the problem. Our service used a lot of logic units in a unit of work pattern. Each logic unit inherited from a BaseLogic class. In the the BaseLogic unit there is an Enterprise Library UnityContainer property which created a factory. Each call created many instances of this factory, changing this property to a static property fixed the problem.

Share Improve this answer

Follow

edited Apr 8, 2014 at 10:42

answered Dec 9, 2012 at 9:10

Mr Mush

1,538 ● 3 ● 25 ● 39

4

The first initial jump to 300MB is consistent with what I've seen in our applications. Haven't really found a way to decrease that number but it stays at that figure over time.

For the increasing part of memory it sounds like a standard memory leak or at least a GC issue. Are you using entity framework and did you profile with a tool like Red Gates Memory Profiler, not the built in VS profiler?

It's hard to give any more specific answer based on the information in the question.

In the mean time, try to use the IIS auto refresh of the application pool. Set it to a threshold of your choice and let it automatically handle the refresh.

Share Improve this answer Follow

answered Nov 19, 2012 at 10:04

Johan Karlsson

918 ● 1 ● 10 ● 25

I know the question is a little vague, I will add any information needed. – Mr Mush Nov 19, 2012 at 10:08