

Re: Remoting large dataset – deserialization is slow

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.remoting/2006-11/msg00058.html>

- *From:* "Yuancai \(\Charlie\) Ye" <nospam@xxxxxxxx>
 - *Date:* Sat, 11 Nov 2006 12:26:13 -0500
-

Hi, All:

See our free remote database service at
<http://www.udaparts.com/document/articles/dialupdb.htm>

I estimate our remote database service can send your 68369 rows to a new machine in less than 5 seconds.

"tanitack" <tanitack@xxxxxxxx> wrote in message
<news:1162794993.167204.159640@xx>

Hi,

We have a client/server application windows based application. The client displays a number of data points to the user. We transfer all the necessary data over remoting. The data is transferred as a dataset containing all the required tables. We use the compression code given in Chapter 13, Extending .Net Remoting, by Ingo Rammer. The compression works fine and transfers the data over the remoting channel pretty fast even for around 68369 rows. But even though remoting is fast, there are other things which take time at the client side and server side.

For example, on the server side in the CompressioHelper class we have the GetCompressedStreamCopy which takes some 4 to 5 seconds.

```
public static Stream GetCompressedStreamCopy(Stream inStream) {  
  
    Stream outputStream = new System.IO.MemoryStream();  
  
    DeflaterOutputStream compressStream = new DeflaterOutputStream(  
  
        outputStream,new Deflater(Deflater.BEST_COMPRESSION));
```

Re: Remoting large dataset – deserialization is slow

```
byte[] buf = new Byte[1000];  
int cnt = inStream.Read(buf,0,1000);  
while (cnt>0) {  
    compressStream.Write(buf,0,cnt);  
    cnt = inStream.Read(buf,0,1000);  
}  
compressStream.Finish();  
compressStream.Flush();  
outStream.Seek(0,SeekOrigin.Begin);  
return outStream;  
}
```

Here the loop takes time. If we remove the loop and do it in one go like,

```
public static Stream getCompressedStreamCopy(Stream inStream) {  
    Stream outStream = new System.IO.MemoryStream();  
    DeflaterOutputStream compressStream = new  
    DeflaterOutputStream(outStream,  
    new Deflater(Deflater.DEFAULT_COMPRESSION));  
  
    int bufferSize = Convert.ToInt32(inStream.Length);  
  
    byte[] buf = new Byte[bufferSize];
```

Re: Remoting large dataset – deserialization is slow

```
int cnt = inStream.Read(buf,0,bufferSize);  
  
//while (cnt>0) {  
  
compressStream.Write(buf,0,cnt);  
  
//cnt = inStream.Read(buf,0,bufferSize);  
  
//}  
  
compressStream.Finish();  
  
compressStream.Flush();  
  
return outStream;  
  
}
```

Even this case there is a 4 second time to compress the data. The 4 second is just for the one statement "compressStream.Write(buf,0,cnt)". Is there any way to improve this? We want to bring up all the data on the client side in less than 10 secs. Though remoting over the wire doesnt take much time, data loading + plus this 4 secs makes it almost 20 secs on the server side alone. The buffer size in the above code for which it took 4 secs was 35139484.

Client Side:

The client side decompression happens pretty fast. But after it gets decompressed and moves through the call stack to the point where the original method call was made, it takes almost a minute. The bulk of the time is between two system calls:

The time taken to hit RemotingProxy.CallProcessMessage() after it hits Remoting.Channels.BinaryClientFormatterSink.SyncProcessMessage() is 40 seconds. So we are loosing all the time saving we have made by compressing the data. Basically you find it to be taking huge time to deserialize, the surprise thing is serialization is faster. Any clues as why all these are happening?

regards,

Catinat

Re: Remoting large dataset – deserialization is slow