Re: .NET 2.0 performance bug in ArrayList.Sort

Source:
http://www.tech−archive.net/Archive/DotNet/microsoft.public.dotnet.general/2006−04/msg01099.html

- *From*: "Frans Bouma [C# MVP]" <perseus.usenetNOSPAM@xxxxxxxxx>
- *Date*: Sat, 29 Apr 2006 02:09:02 −0700

Alex Chudnovsky wrote:

> I have come across with what appears to be a significant performance
> bug in .NET 2.0 ArrayList.Sort method when compared with Array.Sort
> on the same data. Same data on the same CPU gets sorted a lot faster
> with both methods using .NET 1.1, that's why I am pretty sure its a
> (rather serious) bug. Below you can find C# test case that should
> allow you to reproduce this error, to run it you will need to put 2
> data files into current directory where executable is or just change
> filename pathes accordingly, the data files can be obtained from here:
>
> fast_data.txt:
> http://majestic12.co.uk/files/other/dotnet2bug/fast_data.txt
> slow_data.txt:
> http://majestic12.co.uk/files/other/dotnet2bug/slow_data.txt
>
> The data are strings (URLs) of about similar size in bytes and number.
>
> The following are the console outputs from code on the same machine
> (AMD x2 3800, 2 GB RAM, XP Pro SP2):
>
> VS2003 .NET 1.1 (with SP1) run:
>
> −−
> Loaded 29974 lines from file slow_data.txt
> Time to sort strings in ArrayList is: 250 msec
> Time to sort strings in string[] array is: 234 msec
> Loaded 31688 lines from file fast_data.txt
> Time to sort strings in ArrayList is: 250 msec
> Time to sort strings in string[] array is: 250 msec
> −−
>
> Note that sorting times are almost exactly the same here, so all good
> in .NET 1.1 .
>
> −−
> VS2005 .NET 2.0 run:

```
Loaded 29974 lines from file slow_data.txt
Time to sort strings in ArrayList is: 1531 msec
Time to sort strings in string[] array is: 187 msec
Loaded 31688 lines from file fast_data.txt
Time to sort strings in ArrayList is: 703 msec
Time to sort strings in string[] array is: 171 msec
Press ENTER to exit
−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
```

Notice that on the same machine with the same data sorting times are MUCH slower in ArrayList.Sort, and particularly for the "slow_data.txt" file, Array.Sort times are actually better, so I am not complaining there, but clearly ArrayList sorts are seriously flawed − this appears to be data dependent and by that I don't mean size of the data or number of strings: I have got lots of such data files and about every 10th of them is 10−20 times slower than the other even though it has got about the same number of lines and bytesize.

Note: I am aware of boxing/unboxing overheads when dealing with ArrayLists, however in this case the slowdown is really bad comparing to .NET 1.1 and it appears to be data dependent − I am getting it on about 10% of my dataset from which I have selected 2 examples (slow and fast) to demonstrate that its a very serious performance issue.

Here is the code that should allow you to replicate the problem:

```
//////////////////////////////////////////////////////////////////////
//// using System;
using System.Collections;
using System.IO;

/*
This is a test case of what appears to be a major performance
issue in ArrayList.Sort method for strings
in .NET 2.0 − it appears to be data dependant as some similarly
sized data files have got a lot less
performance penalty when sorting them using Array.Sort method on
string[] array.

The kicker: both versions run fast in .NET 1.1

Author: Alex Chudnovsky <alexc@xxxxxxxxxxxxxxx>
Date: 28 Apr 2006
*/


namespace Majestic12
{
/// <summary>
```

```
/// Tests sorting performance of Array.Sort of string[] versus
ArrayList.Sort of the same strings
/// It appears that in .NET 2.0 in some cases ArrayList will take
a LOT more time to do the sorting
/// </summary>
class SlowArrayListSortTest
{
static void Main(string[] args)
{
// load strings from file: assumed to be in the same
place as the executable
TestFile("slow_data.txt"); // <−−− this data file has got
10 times slower
TestFile("fast_data.txt"); // <−−− more reasonable 2
times slower

Console.WriteLine("Press ENTER to exit");
Console.ReadLine();
}

static void TestFile(string sFile)
{
FileStream oFS=File.OpenRead(sFile);

ArrayList oLines=new ArrayList();

StreamReader oSR=new StreamReader(oFS);

while(oSR.BaseStream.Position<oSR.BaseStream.Length)
{
oLines.Add(oSR.ReadLine());
}

oFS.Close();

Console.WriteLine("Loaded {0} lines from file
{1}",oLines.Count,sFile);

// now copy same strings into string array for speed
comparisons string[] sLines=new string[oLines.Count];

for(int i=0;i<sLines.Length;i++)
sLines[i]=(string)oLines[i];

DateTime oTime=DateTime.Now;
oLines.Sort();
Console.WriteLine("Time to sort strings in ArrayList is:
{0}
msec",(DateTime.Now.Ticks−oTime.Ticks)/TimeSpan.TicksPerMillisecond);

oTime=DateTime.Now;
```

```
Array.Sort(sLines);
Console.WriteLine("Time to sort strings in string[] array
is: {0}
msec",(DateTime.Now.Ticks−oTime.Ticks)/TimeSpan.TicksPerMillisecond);
} }
}

//////////////////////////////////////////////////////////////////
////
```

Strange, as ArrayList.Sort uses Array.Sort() :D (use reflector, and see for yourself). So it's not the sort that's flawed, as it's precisely the same routine that gets executed.

The only difference is that ArrayList.Sort() uses a default comparer, Array.Sort() doesn't. I think this is where the problem occurs.

I.o.w.: your sorts aren't exactly the same: the array sort is sorting an array of strings. The arraylist sort is sorting an array of objects using a comparer.

FB

−−
−−
Lead developer of LLBLGen Pro, the productive O/R mapper for .NET
LLBLGen Pro website: http://www.llblgen.com
My .NET blog: http://weblogs.asp.net/fbouma
Microsoft MVP (C#)
−−
.