

Re: Running GUI application in separate application domain

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.clr/2005-10/msg00042.html>

- *From:* "Willy Denoyette [MVP]" <willy.denoyette@xxxxxxxxxxx>
 - *Date:* Thu, 6 Oct 2005 18:18:14 +0200
-

See inline.

Willy.

"nickdu" <nickdu@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message news:F68BB525-6283-4D54-97F5-863E7516DB0F@xxxxxxxxxxxxxxxxxxx
>I don't think remoting is the issue here and I believe the remoting
>counters
> back up that claim. Wait, let me take that back. I did make a change
> that
> seemed to provide some performance relief. The thread I spin up to
> execute
> the assembly in the new appdomain I now set the apartment state to STA. I
> was not setting it in the past so I guess it was defaulting to MTA. I had
> marked my Main() with [STAThread] so I don't know what kinds of problems
> that
> was causing previously.
>

Windows.Forms must run in an STA, however this is only an issue when you host COM based controls (ActiveX) or a .NET control that wraps a COM AX control in your forms. Note that when you need OLE drag and drop support your thread must run in a STA too.

Whenever you host a control that is COM based (some of the VS2003 supplied controls are!), that control must live in an STA thread apartment, when your thread is not running in an STA, the control will end on the process wide default STA thread (created by COM itself).

Now, If you really noticed a performance increase when setting the apartment state to STA, you must have a COM component on your forms, the slowdown is due to the marshaling between the callers thread and the host supplied STA thread which is shared by all AD in the process (there is only one host STA) and as this thread lives in the "system" AD! you will incur cross AD marshaling overhead and the inevitable security walks when crossing the AD borders.

Question is "what controls you are hosting in your applications". If they are COM based you are safe by setting the apartments stae to STA.

Now, if you don't take care, you might run into several issues (as I said AD

Re: Running GUI application in separate application domain

and UI programs don't play well together), for instance if you host a Single threaded COM component in you multidomain application, this component will live on the host STA in a single COM created thread. All calls to this component will be synchronized, so only a single application at a time can call into such component. Note, I don't say you are having this issue, only try to warn you for possible pitfall in such scenario's.

> In addition, I'm guessing that Code Access Security (CAS) checks are the
> main issue here. The performance counters indicate that there are a ton
> of
> runtime checks being made on the version that makes use of appdomains.
> What
> I would like to do is somehow turn off CAS for the appdomains I create. I
> don't know how to do that yet. I'm guessing that these CAS checks are
> also
> putting the added stress on the GC.
> --

No, the CAS check don't add GC pressure, and honestly I don't believe there are security walks unless you or one of the components are calling into unmanaged code without suppressing the security walks. (check the SuppressUnmanagedCodeSecurityAttribute i MSDN.

Did you check the runtime checks after you set the Main threads apartment to STA?

Note that you can disable CAS checks for the whole system by executing "caspol -s off" from the command line, but do this only for testing purposed, you can re-enable CAS by calling caspol -s on.

> Thanks,
> Nick
>
>
> "john conwell" wrote:
>
>> This is an interesting one. its obvious from your perf results that you
>> are
>> generating a LOT more short lived objects when running on multiple
>> appdomains. My guess is there is still some sort of cross domain
>> communication going on.
>>
>> If you have time it would be interesting to see the following counters as
>> well:
>> # Gen 1 and 2 collections
>> gen 1, 2 an large obj heap sizes
>>
>> Also, it would be interesting to run the tests with 1 appdomain, 5
>> domains,
>> 10, 15, and 20...too see what kind of curve you get on the increase of
>> gen
>> 0,1,and 2 collections as the app domains get greater in numberr.

Re: Running GUI application in separate application domain

Re: Running GUI application in separate application domain

>>
>> Also, since all cross appdomain comms are through remoting (i
>> think...could
>> be wrong here) take a look at some of the remoting perf counters such as:
>>
>> Context-Bound objs Alloc / sec
>> Remote calls / sec
>>
>>
>> "nickdu" wrote:
>>
>>> I just ran some more tests and captured perfmon logs for the two
>>> different
>>> scenarios. When I run the GUI components in the default app domain I
>>> get the
>>> following results (I only lists the ones that are significantly
>>> different and
>>> could potentially indicate the problem).
>>>
>>> .NET CLR Security\Total Runtime Checks: Min=122,443 Max=358,890
>>> Delta=~236,000
>>> .NET CLR Memory\# Gen 0 Collections: Min=149 Max=9927 Delta=~9800
>>> .NET CLR Memory\% Time in GC: Average=6
>>> .NET CLR Memory\Allocated Bytes/sec: Average=12MB
>>> .NET CLR Memory\Finalization Survivors: Average=344
>>>
>>> The same counters for the scenario where the components are run in an
>>> appdomain different from the default domain is:
>>>
>>> .NET CLR Security\Total Runtime Checks: Min=124,677 Max=1,475,320
>>> Delta=~1.3M
>>> .NET CLR Memory\# Gen 0 Collections: Min=762 Max=69,766 Delta=~69,000
>>> .NET CLR Memory\% Time in GC: Average=17
>>> .NET CLR Memory\Allocated Bytes/sec: Average=32MB
>>> .NET CLR Memory\Finalization Survivors: Average=38
>>>
>>> For for some reason running these components in a different application
>>> domain drastically increased the # of runtime security checks, # of gen
>>> 0
>>> collections, % Time in GC, and the allocated bytes/sec. Note that the
>>> server
>>> which generates the real time data feed is the same in both cases.
>>>
>>> Any ideas?
>>> --
>>> Thanks,
>>> Nick
>>>
>>>
>>> "nickdu" wrote:
>>>

Re: Running GUI application in separate application domain

>>>> All of the components which communicate with each other should be in
>>>> the same
>>>> application domain. Let me try to explain this a little more.
>>>>
>>>> I have this current application, which you can think of as your
>>>> typical
>>>> application. It has a mainform and some controls (though all of my
>>>> controls
>>>> are modeless dialogs) and they all run in the default application
>>>> domain.
>>>> Now you decide you want to run this whole thing in a different
>>>> application
>>>> domain, other than the default. So in Main() of the default
>>>> appdomain you
>>>> create a new appdomain and execute an assembly on a different thread.
>>>> The
>>>> main of this new assembly does exactly what the original program did.
>>>> Don't
>>>> know if this explains it any better. I hope so.
>>>> --
>>>> Thanks,
>>>> Nick
>>>>
>>>>
>>>> "Jon Shemitz" wrote:
>>>>
>>>>> nickdu wrote:
>>>>>
>>>>>> I don't believe remoting/marshaling is an issue.
>>>>>>
>>>>>>> grid components ...
>>>>>>> displaying real-time data updates over a TIB bus. When these
>>>>>>> same
>>>>>>> components are all run in the default application domain the
>>>>>>> application can
>>>>>>> keep up with the real-time updates. When I create two of the
>>>>>>> grids in one
>>>>>>> appdomain and two in another the performance decreases
>>>>>>> drastically.
>>>>>>
>>>>>>> where is the component that accesses the "TIB bus"? if it's in the
>>>>>>> default appdomain, you are remoting the data. if it's in each
>>>>>>> appdomain, you may have synchronization (lock) overhead.
>>>>>>
>>>>>> --
>>>>>>
>>>>>>> i'm midnightbeach.com and a mtn bike
>>>>>>> accident means i'm left handed until mid october,
>>>>>>> so please pardon lower case and terseness.
>>>>>>>

- *Follow-Ups:*
 - ◆ **Re: Running GUI application in separate application domain**
◇ From: nickdu

- *References:*
 - ◆ **Re: Running GUI application in separate application domain**
◇ From: Henning Krause [MVP – Exchange]
 - ◆ **Re: Running GUI application in separate application domain**
◇ From: nickdu
 - ◆ **Re: Running GUI application in separate application domain**
◇ From: Jon Shemitz
 - ◆ **Re: Running GUI application in separate application domain**
◇ From: nickdu

- Prev by Date: **Re: Running GUI application in separate application domain**
- Next by Date: **Problem with Diagnostics.Process.GetProcesses(String machineName)**
- Previous by thread: **Re: Running GUI application in separate application domain**
- Next by thread: **Re: Running GUI application in separate application domain**
- Index(es):
 - ◆ **Date**
 - ◆ **Thread**