Maarten Balliauw {blog}

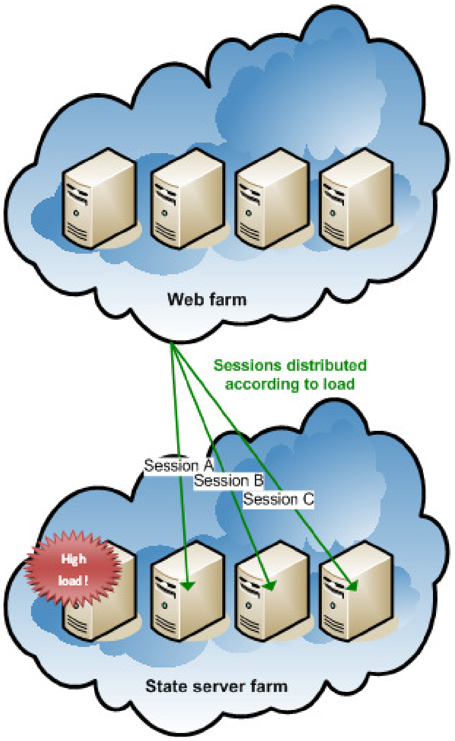
ASP.NET, ASP.NET MVC, Azure, PHP, OpenXML, VSTS, ...

Home | Talks & Presentations | Archive | Contact | Subscribe| Sign in

<< ASP.NET Session State Partitioning | Books I recently read... >>

ASP.NET Session State Partitioning using State Server Load Balancing

Posted by maartenba on Thursday, January 24, 2008 4:05 PM



It seems like amount of posts on ASP.NET's Session State keeps growing. Here's the list:

- ASP.NET Session State Partitioning
- ASP.NET load balancing and ASP.NET state server (aspnet_state)

Yesterday's post on Session State Partitioning used a round-robin method for partitioning session state over different state server machines. The solution I presented actually works, but can still lead to performance bottlenecks.

Let's say you have a web farm running multiple applications, all using the same pool of state server machines. When having multiple sessions in each application, the situation where one state server handles much more sessions than another state server could occur. For that reason, ASP.NET supports real load balancing of all session state servers.

Download example

Want an instant example? Download it here.
Want to know what's behind all this? Please, continue reading.

What we want to achieve...

Here's a scenario: We have different applications running on a web farm. These applications all share the same pool of session state servers. Whenever a session is started, we want to store it on the least-busy state server.

1. Performance counters

To fetch information on the current amount of sessions a state server is storing, we'll use the performance counters ASP.NET state server provides. Here's a code snippet:

```
if (PerformanceCounterCategory.CounterExists("State Server Sessions Active", "ASP.NET", "STATESERVER1")) {
    PerformanceCounter pc = new PerformanceCounter("ASP.NET", "State Server Sessions Active", "",
"STATESERVER1");
    float currentLoad = pc.NextValue();
}
```

2. Creating a custom session id

Somehow, ASP.NET will have to know on which server a specific session is stored. To do this, let's say we make the first character of the session id the state server id from the following *IList*:

```
IList<StateServer> stateServers = new List<StateServer>();

// Id 0, example session id would be 0ywbtzez3eqxut45ukyzq3qp
stateServers.Add(new StateServer("tcpip=10.0.0.1:42424", "sessionserver1"));

// Id 1, example session id would be 1ywbtzez3eqxut45ukyzq3qp
stateServers.Add(new StateServer("tcpip=10.0.0.2:42424", "sessionserver2"));
```

Next thing we'll have to do is storing these list id's in the session id. For that, we will implement a custom *System.Web.SessionState.SessionIDManager* class. This class simply creates a regular session id, locates the least-busy state server instance and assign the session to that machine:

```
using System;
using System.Diagnostics;


public class SessionIdManager : System.Web.SessionState.SessionIDManager
{
```

About the author

Maarten Balliauw is currently employed as .NET Technical Consultant at RealDolmen. His interests are mainly web applications developed in ASP.NET (C#) or PHP and the Windows Azure cloud platform.

More about me
E-mail me

Search

Enter search term | Search

□ Include comments in search

Pages

About me
Disclaimer
Talks & Presentations

Latest Twitter

Blogged: Windows Azure CDN updates - http://bit.ly/fXCnwA #azure #cdn 1 day ago
Home. And in 2 minutes, we depart on vacation for the second time. Yay! #fb 1 day ago
Also interesting: coffee drinking in the car on Belgian roads is like taking a shower... #fb #yam 1 day ago
Wife forgot something. Returning home after 1h drive... #fb #yam 1 day ago
A whole day of autobahn in sight. Gone for ski for a week. Cheers! #fb #yam 1 day ago
@sergejusb I also saw online that I passed last month. Today I got the paper confirmation from #prometric :-) 2 days ago
Blogged: Windows Azure CDN updates - http://bit.ly/fXCnwA #azure #cdn 2 days ago
The day before my vacation, and the #antwerp #beltway wants to take away some precious time. Darn you! #fb #yam 2 days ago

Follow me on Twitter...

My projects

PHPExcel
PHPLinq
PHPMEF
Windows Azure SDK for PHP
ASP.NET MVC SiteMap provider
Wandelaar.be
And many more... ask!

Related blogs

Joris Poelmans
Juliën Hanssens

```
    public override string CreateSessionID(System.Web.HttpContext context)
    {
        // Generate a "regular" session id
        string sessionId = base.CreateSessionID(context);

        // Find the least busy state server
        StateServer leastBusyServer = null;
        float leastBusyValue = 0;
        foreach (StateServer stateServer in StateServers.List)
        {
            // Fetch first state server
            if (leastBusyServer == null) leastBusyServer = stateServer;

            // Fetch server's performance counter
            if (PerformanceCounterCategory.CounterExists("State Server Sessions Active", "ASP.NET",
stateServer.ServerName))
            {
                PerformanceCounter pc = new PerformanceCounter("ASP.NET", "State Server Sessions Active", "",
stateServer.ServerName);
                if (pc.NextValue() < leastBusyValue || leastBusyValue == 0)
                {
                    leastBusyServer = stateServer;
                    leastBusyValue = pc.NextValue();
                }
            }
        }

        // Modify session id to contain the server's id
        // We will change the first character in the string to be the server's id in the
        // state server list. Notice that this is only for demonstration purposes! (not secure!)
        sessionId = StateServers.List.IndexOf(leastBusyServer).ToString() + sessionId.Substring(1);

        // Return
        return sessionId;
    }
}
```

The class we created will have to be registered in *web.config*. Here's how:

```
<configuration>
  <system.web>
    <!-- ... -->
    <sessionState mode="StateServer"
                partitionResolverType="PartitionResolver"
                sessionIDManagerType="SessionIdManager" />
    <!-- ... -->
  </system.web>
</configuration>
```

You notice our custom *SessionIdManager* class is now registered to be the *sessionIDManager*. The *PartitionResolver* I blogged about is also present in a modified version.

3. Using the correct state server for a specific session id

In the previous code listing, we assigned a session to a specific server. Now for ASP.NET to read session state from the correct server, we still have to use the *PartitionResolver* class:

```
using System;


public class PartitionResolver : System.Web.IPartitionResolver
{

    #region IPartitionResolver Members

    public void Initialize()
    {
        // No need for this!
    }

    public string ResolvePartition(object key)
    {
        // Accept incoming session identifier
        // which looks similar like "2ywbtzez3eqxut45ukyzq3qp"
        string sessionId = key as string;

        // Since we defined the first character in sessionId to contain the
        // state server's list id, strip it off!
        int stateServerId = int.Parse(sessionId.Substring(0, 1));

        // Return the server's connection string
        return StateServers.List[stateServerId].ConnectionString;
    }

    #endregion

}
```

Kevin Dockx
RealDolmen Blogs

Recent posts

Windows Azure CDN updates Comments: 0 Rating: 5 / 2
Put your cloud on a diet (or: Windows Azure and scaling: why?) Comments: 4 Rating: 5 / 1
Authenticate Orchard users with AppFabric Access Control Service Comments: 4 Rating: 5 / 2
Thank you for getting me in Vegas! Comments: 3 Rating: 5 / 1
MvcSiteMapProvider 3.0.0 released Comments: 1 Rating: 5 / 2
ASP.NET MVC and the Managed Extensibility Framewok on NuGet Comments: 9 Rating: 5 / 1
Introducing the PHP on Azure Contest Comments: 5 Rating: 3.5 / 2
Viva, Las Vegas! Comments: 2 Rating: 5 / 2
Writing an Orchard widget: LatestTwitter Comments: 2 Rating: 5 / 2
ASP.NET MVC and jQuery Mobile Comments: 6 Rating: 5 / 2

Categories

ASP.NET
Azure
Book review
Books
C#
Debugging
Events
ExtJS
General
Hardware
ICT
Internet
JavaScript
jQuery
LINQ
Logging
MEF
MVC
MVP
NHibernate
NUnit
Object-Oriented concepts
Offtopic
OpenXML
Orchard
Personal
Pex
PHP
Podcasts
Presentations
Profiling
Projects
Publications
Quality code
Scalability
Screencasts
Security
SilverLight
Software
SQL Azure
Subversion
Testing
VS2010
VSTS
Webfarm
XML
Zend Framework

Archive

2011
March (2)
February (4)
January (6)
2010

December (1)
November (2)
October (6)
September (6)
August (3)
July (4)
June (2)
May (5)
April (4)
March (2)
February (6)
January (6)
2009
December (4)
November (6)
October (6)
September (4)
August (7)
July (13)
June (6)
May (7)
April (7)
March (5)
February (6)
January (9)
2008
December (12)
November (5)
October (4)
September (3)
August (2)
July (4)
June (3)
May (6)
April (5)
March (7)
February (4)
January (5)
2007
December (6)
November (4)
October (4)
September (2)
August (7)
July (4)
June (9)
May (3)
April (5)
March (4)
February (4)
January (5)
2006
December (2)
November (3)
October (4)
September (5)
August (11)

Disclaimer

The opinions expressed herein are my own personal opinions and do not represent my employer's view in any way.

kick it 3

Currently rated 5.0 by 11 people

Like Be the first of your friends to like this.

Categories: ASP.NET | C# | General | Software | Webfarm | XML
E-mail | Kick it! | DZone it! | del.icio.us
Permalink | Comments (32) | Post RSS

Related posts

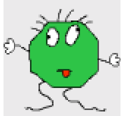
ASP.NET load balancing and ASP.NET state server (aspnet_state)
At one of our clients, we used to have only one server for ASP.NET applications (including web ser...

ASP.NET Session State Partitioning
After my previous blog post on ASP.NET Session State, someone asked me if I knew anything about AS...

Leveraging ASP.NET MVC 2 futures "ViewState"
Let's start this blog post with a confession: yes, I abused a feature in the ASP.NET MVC 2 fut...

Comments

Fenil Desai | Reply Friday, January 25, 2008 6:36 AM



There can't be a better post on Session Partitioning that this one.
Absolutely Amazing......

Scott Hanselman | Reply Thursday, January 31, 2008 8:18 PM

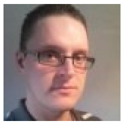
Fantastic series! One question, however, isn't there measurable overhead in the creation of that remote Performance Counter? How much optimization have you had to do in that CreateSessionID call?

hanselman.com | Reply Thursday, January 31, 2008 8:29 PM

Pingback from hanselman.com

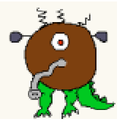
Scott Hanselman's Computer Zen - Troubleshooting Expired ASP.NET Session State and Your Options

maartenba | Reply Thursday, January 31, 2008 9:27 PM



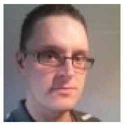
Thank's 🙂 The remote performance counter will definitely give some overhead, but I did not measure this in a real-life environment.

lianglisen | Reply Wednesday, December 30, 2009 8:37 PM



About your article "ASP.NET Session State Partitioning using State Server Load Balancing"
I want to ask something:I have download and test you code from maarten balliauw blog,is the webserver appication store the session on the state server one by one?e.g:1st connection,session store in state server A, 2nd connection,session store in state server B,....... cycle again and again。Hoping to get you help.Best Regards.

maartenba | Reply Wednesday, December 30, 2009 9:48 PM



Yes, that's correct. Unless the performance counter tells there's too much load, the server is skipped for storing state. Note that this is just an example and should not be used in production without further fine tuning.

weblogs.asp.net | Reply Friday, February 01, 2008 1:55 AM

Pingback from weblogs.asp.net

Maarten Balliauw on ASP.NET load balancing and the ASP.NET state server. - Joe On ASP.NET

blogs.msdn.com | Reply Friday, February 01, 2008 1:55 AM

Pingback from blogs.msdn.com

Joe Stagner - Frustrated by Design ! : Maarten Balliauw on ASP.NET load balancing and the ASP.NET state server.

geeks.ms | Reply Friday, February 01, 2008 2:48 AM

Pingback from geeks.ms

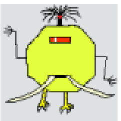
Maarten Balliauw on ASP.NET load balancing and the ASP.NET state server. - Noticias externas

radicaldevelopment.net | Reply Sunday, February 03, 2008 9:24 AM

Pingback from radicaldevelopment.net

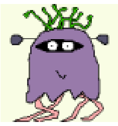
January 2008 Resources RoundUp - Radical Development

Yolion | Reply Wednesday, February 20, 2008 6:19 AM



Great

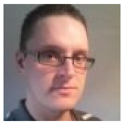
Elan | Reply Tuesday, March 18, 2008 5:05 PM



How does this address failover? Am I correct in assuming that when a server goes down, the if (PerformanceCounterCategory.CounterExists) doesn't blow the whole thing up and also prevents that [offline] server from being assigned to store this session state?

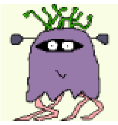
But then what happens if a user's session state had been assigned to the server that just went down? Does the PartitionResolver throw an error? Perhaps to make this more robust, the exception handling can be set to reset and reassign the session? Then the worst that happens to the client is a lost session - but no error messages.

maartenba | Reply Tuesday, March 18, 2008 5:08 PM



This example code does NOT include any checks on remote server existance. You would have to check if the state server is running in the code prior to assigning a session to it without failure massages.

Elan | Reply Tuesday, March 18, 2008 6:58 PM

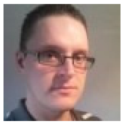


if (PerformanceCounterCategory.CounterExists("State Server Sessions Active", "ASP.NET", stateServer.ServerName))

Isn't that an inherent check on whether the State Server is operating? Or will you error out entirely if the machinename reference leads to a server that is down?

If it works properly as is, then couldn't you include this check in the PartitionResolver class and handle appropriately if your session is pointing at a server that can no longer be found?

maartenba | Reply Tuesday, March 18, 2008 7:58 PM



As far as I can see there are 2 things that can go wrong:
1) The performance counter is unavailable
2) State server is unavailable

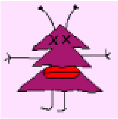
Possible solutions:
1) Check for Exceptions on PerformanceCounterCategory.CounterExists(...), respond to Exceptions by trying the next server. If it fails > X times, remove it from the list of servers.
2) Respond to possible Exceptions. If it fails > X times, remove it from the list of servers.

blogs.msdn.com | Reply Tuesday, July 08, 2008 9:01 AM

Pingback from blogs.msdn.com

JoeOn.net In Japanese : Windows Workflow Foundation ãƒãƒ¥ãƒ¼ãƒˆãƒªã‚¢ãƒ« ã‚·ãƒªãƒ¼ã‚º

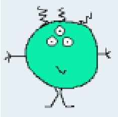
sss 🇺🇸 | Reply Wednesday, July 16, 2008 12:55 PM



Pingback from blogs.msdn.com

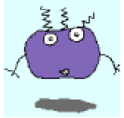
JoeOn.net In Japanese : Windows Workflow Foundation ãƒãƒ¥ãƒ¼ãƒˆãƒªã‚¢ãƒ« ã‚·ãƒªãƒ¼ã‚º

Chris 🇬🇧 | Reply Tuesday, September 02, 2008 3:44 PM



This sounds absolutely fab!! I have tried implementing it but have failed. How do I implement this in IIS? thanks

Echte Rotterdamer 🇬🇧 | Reply Saturday, November 01, 2008 8:50 AM



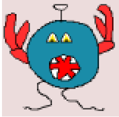
Lets try it again. Your code is flawed in 2 places:

1.SessionIdManager - if you have two state servers in the list one will never get hit.

2.ParftitionResolver - When a state server goes down all live sessions for that server will raise an exception. If that exception is handled in Application_Error handler in Global.asax by redirecting to a custom error page (common scenario) and that page try to access session it will hit PartitionResolver again creating a nasty loop. If you have enough live sessions for the dead state server it might bring your application down.

To overcome the problem you have to catch that exception, create a new sessionID, recreate session object for the current context with that new sessionID. Sure user will loose the session but the application will recover gracefully and they can start a new session.

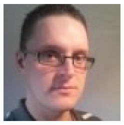
SamR 🇺🇸 | Reply Tuesday, November 18, 2008 9:30 PM



Hey All,

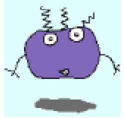
We have a website that is running on 2 boxes, and every other click we lose our sessions in production. This doesn't happen in local machine at all. Should I try switching to SQL Server or partitioning.
Thank u.

maartenba 🇧🇪 | Reply Wednesday, November 19, 2008 7:48 AM



Sam, I guess you should consider both options: SQL server will be more reliable but a little slower, state server / partitioning will be a bit faster but there's still possible loss of session data.

Echte Rotterdamer 🇬🇧 | Reply Monday, November 24, 2008 4:45 PM



Sam

Maarten's post was the great starting point and I am really greatful to him for helping me kick-start my own project. I have done some further work and now I have the solution we are going to use in our production environment. You are free to use it if you wish, you can read about if you visit the following link:

en.aspnet-bhs.info/.../...Server-Partitioning.aspx

RealDolmen blogs | Reply Wednesday, December 31, 2008 3:43 PM

Trackback from RealDolmen blogs

Top blog posts in 2008

answerspluto.com | Reply Monday, July 13, 2009 9:38 PM

Pingback from answerspluto.com

list of urls 4 « Answers Pluto

Microsoft RealDolmen blogs | Reply Monday, January 04, 2010 8:55 AM

Byebye 2009, welcome 2010!

Byebye 2009, welcome 2010!

code-inside.de | Reply Sunday, March 21, 2010 10:14 PM

Pingback from code-inside.de

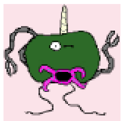
HowTo: Session in ASP.NET & ASP.NET Session State Server | Code-Inside Blog

msjoe.com | Reply Saturday, October 09, 2010 4:38 PM

Pingback from msjoe.com

Maarten Balliauw on ASP.NET load balancing and the ASP.NET state server. : MS-Joe

raees | Reply Thursday, October 28, 2010 6:31 PM



Well everything is fine in this post.

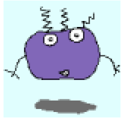
But from the sample application when i host my application on Server A, and use State Server in B and C and try to do load balancing i added

```
stateServers.Add(new StateServer("tcpip=192.168.1.11:42424", "."));
stateServers.Add(new StateServer("tcpip=192.168.1.12:42424", "."));
```

Then from my webserver whoose ip is 192.168.1.10 i accessed the information and everytime the nextvalue was 0 only and it was picking the first state server. I accessed the same machine from 10 different machine and all were coming to first state server and nextvalue was always 0 only when i printed that.

Please help how can i do this.

Echte Rotterdamer | Reply Friday, October 29, 2010 7:52 AM

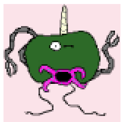


Here is full solution with all the issues resolved:

en.aspnet-bhs.info/.../...Server-Partitioning.aspx

It is built upon this excelent post by Maarten.

raees | Reply Friday, October 29, 2010 4:43 PM



Hi Echte

I tried executing your code. Everything appears to be fine. but i get this error

Access is denied
Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.ComponentModel.Win32Exception: Access is denied

Source Error:

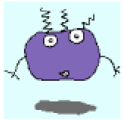
```
Line 77:
Line 78:                ' Fetch server's performance counter
Line 79:                If PerformanceCounterCategory.CounterExists("State Server Sessions Active", "ASP.NET", stateServer.ServerName) Then
Line 80:
Line 81:                    Dim pc As New PerformanceCounter("ASP.NET", "State Server Sessions Active", "", stateServer.ServerName)
```

Source File: D:\Projects\App_Code\BLL\SessionIdManager.vb   Line: 79

On debugging further i found that in the web.config when i change the servername from "." to my machine name it gives this error, where as when i use the name as just "." it works fine. But in that case i doubt it is picking up multple state server it just picks up the first state server.

Can you let me know the behavious of this.

Echte Rotterdamer | Reply Saturday, October 30, 2010 1:11 AM



raees,

That may be specific to your environment, it is obviously a security issue. You may wish to read the following post: http://west-wind.com/weblog/posts/1572.aspx

I tested it a few year ago on a couple of virtual machines running on the single box, most likely there were no security limitations of that kind.

Maarten Balliauw {blog} | Reply

Saturday, January 01, 2011 3:29 PM

Byebye 2010, welcome 2011!

Byebye 2010, welcome 2011!

Add comment

Name*	
E-mail*	
Website	
Country	United States

Comment | Preview | b i u quote

□ Notify me when new comments are added

Type the two words:

Save comment