

# Re: Async Pages and Thread Pools

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework/2008-02/msg00165.html>

---

- *From:* [stcheng@xxxxxxxxxxxxxxxxxxxxxx](mailto:stcheng@xxxxxxxxxxxxxxxxxxxxxx) (Steven Cheng[MSFT])
  - *Date:* Mon, 11 Feb 2008 02:28:44 GMT
- 

Hi Chris,

The following article also show you some benefits and results of using Async page. One thing I think good there is it also mentioned what you need to take care when determine to use async page. That's whether the operation you want to run in async page is executing on a custom thread rather than thread-pool thread. If the task itself also occupy thread-pool thread, async page won't help much:

#Async Pages in ASP.NET 2.0 – some results  
<http://www.pluralsight.com/blogs/fritz/archive/2004/10/19/2892.aspx>

You can also get more info about ASP.NET threading model through the following webcast materials:

#Support WebCast: Microsoft ASP.NET Threading  
<http://support.microsoft.com/kb/820913>

Sincerely,

Steven Cheng

Microsoft MSDN Online Support Lead

=====

Get notification to my posts through email? Please refer to  
<http://msdn.microsoft.com/subscriptions/managednewsgroups/default.aspx#notifications>.

Note: The MSDN Managed Newsgroup support offering is for non-urgent issues where an initial response from the community or a Microsoft Support Engineer within 1 business day is acceptable. Please note that each follow up response may take approximately 2 business days as the support

Re: Async Pages and Thread Pools

professional working with you may need further investigation to reach the most efficient resolution. The offering is not appropriate for situations that require urgent, real-time or phone-based interactions or complex project analysis and dump analysis issues. Issues of this nature are best handled working with a dedicated Microsoft Support Engineer by contacting Microsoft Customer Support Services (CSS) at <http://msdn.microsoft.com/subscriptions/support/default.aspx>.

=====

This posting is provided "AS IS" with no warranties, and confers no rights.

-----

From: "Chris Marsh" <cjmarsh@xxxxxxxxxxxxxxxxxxxx>  
References: <OKjj3GkaIHA.4180@xxxxxxxxxxxxxxxxxxxx>

<ux9hP0kaIHA.536@xxxxxxxxxxxxxxxxxxxx>

Subject: Re: Async Pages and Thread Pools  
Date: Fri, 8 Feb 2008 14:04:19 -0000

Kevin

Many thanks for the response.

"Kevin Spencer" <unclechutney@localhost> wrote in message  
<news:ux9hP0kaIHA.536@xxxxxxxxxxxxxxxxxxxx>

The advantage of Asynch ASP.Net pages is seen when the number of concurrent requests exceeds the number of available request-processing threads in the thread pool. The size of the thread pool is limited, and

if

exceeded, successive requests can be delayed, or if there are enough requests, some may eventually time out waiting for a thread, resulting

in

a "503 Server Unavailable" error from the server. Asynch ASP.Net pages return the request-processing thread to the thread pool immediately,

using

another thread to do the actual processing. When the processing is completed, another request-processing thread is grabbed to return the Response to the client.

## Re: Async Pages and Thread Pools

So, if we have an IIS thread pool size of 25 and 25 requests are made that involve intensive IO, other requests (for inexpensive resources that \*should\* be served very quickly) will not be served until at least one of the other threads is freed up. If we use asynchronous page processing, we hand off processing of the intensive IO operations to threads outside the IIS thread pool, thus freeing up threads from the IIS thread pool to

service

other requests.

What I don't understand is what benefit using asynchronous page processing has over simply increasing the number of threads in the IIS thread pool.

Why

does it matter whether threads are taken from the IIS thread pool or not?

[..]

<http://msdn.microsoft.com/msdnmag/issues/05/10/WickedCode/>

A useful resource, thanks.

--

Regards

Chris Marsh