

Re: Using SQL to store aspx pages and memory problems

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2005-07/msg00345.html>

- *From:* "Klaus H. Probst" <usenet001@xxxxxxxxxxxxxx>
 - *Date:* Tue, 5 Jul 2005 01:36:17 -0700
-

The following suggestion may or may not work for you, but I'll chime in.

First off, to the folks that called the approach your existing app uses "idiotic", I've seen classic ASP applications design this way that work extremely well, are scalable and extremely manageable. However, the devil is in the details. Just putting "pages" in IMAGE or TEXT fields on a SQL database isn't some sort of magic pixie dust that solves all problems. It takes a lot of thought and experimentation to get just right, and the rewards in most cases offset the negative aspects of these designs.

Having said that, your approach is, as you've figured out by now, wrong. Others have already mentioned the perf hit you're taking by forcing ASP.NET to function in a way it wasn't designed to. But there are ways to get around the problem.

First off, I'll assume that you have more content than code. That is, **most** of your existing ASP/ASPX pages in the database contain stuff, like invoices and whatnot, that do not require intervention by the parser to load and display. If this is not the case then stop reading now =)

There are a lot of web applications nowadays that use their databases as primary content storage repositories. Blogs are a good example of this. However, they do not store code in the database, they store content. If you can adapt to this type of scenario then all you have to do is create a sort of "master page" or maybe an IHttpHandler implementation that looks at the request (even a path), pulls the **content** from the database and then displays it on a sort of templated page container. Think of a site like MSDN. Lots of articles, yet all the pages look the same. The content is what varies. The trick is to simply treat your "pages" as opaque chunks of content, and "stream" them into a placeholder page that is rendered the same way every time. You can pretty much do anything you want once you're hooked into the ASP.NET processing pipeline.

If you do this you'll get rid of your perf problem and you'll still have the flexibility and manageability of the database storage. This is web application design 101 – always separate content from function and structure.

Re: Using SQL to store aspx pages and memory problems

Of course getting from point A to point B could be tricky, but maybe this gives you some ideas =)

--

Klaus H. Probst, MVP

<http://www.simulplex.net/>

"matvdl" <matvdl@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message
news:DF375EB0-8373-4D91-B7F3-0400A8AE22E9@xxxxxxxxxxxxxxxxxxxx

> DLM,

>

> Thanks for your response.

>

> I don't believe that SQL is my problem – the time it takes to do the
remote

> calls to the external program has not presented any real limitations or

> caused any significant time-delays. The problem that I have is trying to

> deal with the quantity of files that I have and the combiling of those
pages

> once they have been returned. Any time issues that I have had are the
result

> of parsing and compilling the page.

>

> I understand your point – that asp.net was designed to have the pages

> pre-compiled before requests where made for the pages – I guess this is
one

> of the fundamental differances between asp and asp.net. From experiance
the

> design that has been implemented had no significant disadvantages when

> working with asp – in fact I believe that it had some very significant

> advantages – mainly due to the unlimited number of files it could manage.

>

> Question – is it practicale to have 10,000 pages in a single asp.net

> application. I wouldn't have thought so?

>

> I have no alternative but to deal with this many files as it is a legal

> requirment of the industry I work in is to be able to provide the original

> copies of invoices to clients – I can't go back to the original data and

> attempt to re-produce the same invoice just in case the data has changed.

>

> So – considering this is what I am stuck with – my problem mainly relates
to

> the memory increases – the slowness of the request could be dealt with by
the

> more grunt – but the ever increasing memory is more difficult. I believe

> that this is the result of the creation of a new class each time a new

page

> is displayed – although this page gets un-loaded once the info is

returned –

Re: Using SQL to store aspx pages and memory problems

> are the increases in memory the result of the new class definition in the
> asp.net application?
>
> If this is the case – is it possible to remove this class definition once
it
> has been created? I have tried to delete the file – but this does not
appear
> to fix the problem.
>
> Does any of this make sense.
>
>
> --
> matthew
>
>
> "DLM" wrote:
>
>> Go to this link:
>> <http://www.theserverside.net/books/inreview/ImprovingPerf/pag.tss>
>>
>> We all know that there is an initial performance 'hit' the first time a
ASP
>> .NET Page is requested, and this is true with the page sitting in it's
>> virtual directory on IIS. What you seem to be doing is increasing the
round
>> trip time for any given request/response for a page to include the time
it
>> takes to get the page from SQL Server. And this added time is database
access
>> time which is probably longer than any HTTP request.
>>
>> I have never seen this kind of architecture before. The whole idea is to
>> serve pages from a highly scalable architecture such as IIS from the
very
>> start, and not make the ASP .NET worker process wait for the files
before it
>> can dynamically compile pages. In a sense you are using SQL Server as a
web
>> server to your web server, which is highly unusual. You need to find a
way to
>> place all your pages on the web server and use SQL Server only for data
>> retrieval/update.
>>
>> TAKE A LOOK AT MICROSOFT PATTERNS & PRACTICES
>>
>>

.

- **Follow-Ups:**
 - ◆ **[Re: Using SQL to store aspx pages and memory problems](#)**
 - ◇ *From:* matvdl

- **References:**
 - ◆ **[Using SQL to store aspx pages and memory problems](#)**
 - ◇ *From:* matvdl
 - ◆ **[Re: Using SQL to store aspx pages and memory problems](#)**
 - ◇ *From:* Bob Lehmann
 - ◆ **[Re: Using SQL to store aspx pages and memory problems](#)**
 - ◇ *From:* matvdl
 - ◆ **[Re: Using SQL to store aspx pages and memory problems](#)**
 - ◇ *From:* TJS
 - ◆ **[Re: Using SQL to store aspx pages and memory problems](#)**
 - ◇ *From:* matvdl
 - ◆ **[Re: Using SQL to store aspx pages and memory problems](#)**
 - ◇ *From:* DLM
 - ◆ **[Re: Using SQL to store aspx pages and memory problems](#)**
 - ◇ *From:* matvdl

- Prev by Date: **[RE: ASP 2.0 Direct Connection Question](#)**
- Next by Date: **[Re: Deploying an IHttpModule in the GAC](#)**
- Previous by thread: **[Re: Using SQL to store aspx pages and memory problems](#)**
- Next by thread: **[Re: Using SQL to store aspx pages and memory problems](#)**
- Index(es):
 - ◆ **[Date](#)**
 - ◆ **[Thread](#)**