

Re: Good Design Practice Question

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2005-05/msg00645.html>

- *From:* "Dave" <NOSPAM-dave@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 4 May 2005 04:46:23 -0400
-

Databases are meant to be used for reliable, scalable and fast data-access. When you read that it is best to keep the database hits to a minimum there is a discrepancy about what actually is considered "minimum". If your site is not expecting alot of hits, say under 100 concurrently, then accessing the database will not be your bottle neck on a server that is of a descant caliber, especially when the amount of data being sent and retrieved can be done in a single batch. Also, using stored procedures will greatly increase the effeciency of your database hits.

- 1) Use stored procedures to read/write the data from the dataset
- 2) Keep the "hits" to a minimum by making the "least" amount of calls to the database
- 3) Keep the effeciency maxed by only pulling/saving what is needed in context. i.e., don't just SELECT * FROM customers if you only need one customer. (if you forsee requiring more customers in a single context, then it may be better to make one hit immediately to get all of them. This is a trade-off for db hits --> data transfer speed and memory usage.)
- 4) Data Access Appication Blocks for .NET show how parameter caching can add to the effeciency of using a database.
- 5) Ensure that your database server is configured to properly handle the largest load you expect. Isolating this server is always better, when possible. (i.e. Keep the web server and database servers on physically seperate machines)

Session state is usefull for minimizing database hits and reducing the amount of times you have to "munge" the data before it's usable directly from the Db.

- 1) Use session state when the data being read from/written to the database is being changed dramattically on the client (which in this case, is the web server). This aids in performance of your application since applying business rules to data, when this is a heavy process, will only need to occur once.
- 2) Data stored in the session can be located in a few different places. Choose the best config that meets the needs of your app. (check out session state inproc, stateserver and sqlserver)

There are plenty of points I have not listed above. The list was meant to show you how there can't be a general-purpose answer for

Re: Good Design Practice Question

your question. It really depends on a lot of factors, all though you will hear people argue one way or the other without knowing all of the facts I've mentioned above.

Making assumptions can be dangerous, so using ACT or other testing tools will also be in your favor.

Wade in the facts for a bit, choose what you believe is the best solution, and test it until it crumbles. You'll get an idea of what solution will be best in your particular situation.

Dave Sexton
dave@xxxxxxxxxxxxxxxxxxxxxx

"Calvin" <calvin@xxxxxxxxxxxxxxxxxxxxxx> wrote in message
news:BE9DE5AE.18D2%calvin@xxxxxxxxxxxxxxxxxxxxxx
> I'm fairly new to programming, that said, I know how to 'program', but my
> question is about design.
>
> How big and what sort of objects should be stored in Session.
>
> I'm doing an e-commerce website, and I've created an xsd based around the
> shopping cart, customers details and the order's tables. The resulting
> dataset is then populated accordingly and stored in session.
>
> Is this bad news? I ask because the articles I read suggest limiting the
> number of database hits... but then also say that the session object should
> not be used excessively.
>
> It's obviously about getting a balance, but how do I tell? Look at the task
> manager to see how much memory the aspnet_wp worker process is using. About
> 50,000kbs... kind of feels too much
>
>
>

• **Follow-Ups:**

◆ **Re: Good Design Practice Question**

◇ From: Calvin

• **References:**

◆ **Good Design Practice Question**

◇ From: Calvin

- Prev by Date: **SandBox – How?**
- Next by Date: **Web application, redirection, restore values**
- Previous by thread: **Good Design Practice Question**

Re: Good Design Practice Question

- Next by thread: ***Re: Good Design Practice Question***
- Index(es):
 - ◆ ***Date***
 - ◆ ***Thread***