

Too many objects loaded at the same time and cause OutOfMemory exception.

Too many objects loaded at the same time and cause OutOfMemory exception.

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2007-01/msg03442.html>

- *From:* "elaine" <elain.sun@xxxxxxxxx>
 - *Date:* 29 Jan 2007 16:17:53 -0800
-

I'm working on a .net web application. The architect of this web application is quite different than other web applications i worked before. Since we use a set of tools to generate most of the basic code. Every table in the database related to an object in the application. There is no stored procedures in database; In every object of the application, the tool generates methods for basic database operations like insert, delete, update, selete, and methods to compose a select query inside the table(object)

But it definitely is a bad design, since we have more and more users, big performance issues coming out when generating reports for users. For example, if i use this architect for a school, i can easily create objects of grade, class, student, lesson, semester, now let's say i have 6 grades, each grade has 10 class, each class has 20 students, each student has 5 lessons. If i want to generate a score report for all students's score in fall of 2006, i have to first load grade object collection, loop into grade collection; load class object collection, loop into class collection; load student object collection, loop into student collection; load lesson object collection, loop into lesson collection. I don't know how .net garbage collection will work on this situation, whether or not it does load all $6 \times 10 \times 20 \times 5 = 6000$ objects in memory at the same time. I noticed a huge memory overhead for aspnet_ws process, then i got Service Unavailable error in my browser, and the memory of aspnet_ws back to normal.

Does anyone know how .net framework works on my situation? Any resolution to this OutOfMemory issue except using stored procedure? I don't think asynchronized process will help, what do you think? What's the best way to scale out for the current system?

thank you for your help,
-Elaine

.