For every enterprise level application the key to make that application successful is the responsiveness of the application. ASP.NET offers a great deal of the features for developing web based enterprise applications, but sometimes, due to adhering to best practices, the application is not as fast as it should be. Here are the some useful suggestions to make your application super fast.

1. Always set debug=”false” in web.config production environment.
2. Always set trace=”false” in web.config production environment
3. If you are using asp.net 2.0 or higher version then always use precompiled version of your code and also prefer web application project over website. If you are using website then always publish it and then upload that precompiled version of your site in production environment.
4. Always compile your project in Release Mode before uploading application to production environment.
5. Decrease your html kb as much as you can for that use tables less html using div’s and if possible then do not give big name to your control it will increase your html kb as asp.net uses client Id to differentiate all the controls. If you are creating custom controls then you can overwrite your clientid and uniqueId.
6. Use cache api as much as possible it will decrease your server roundtrip and boost application performance. ASP.NET 2.0 or higher version provides functionality called sqlcachedependancy for your database caching. It will validate cache with your database operation like insert,update and delete and if not possible with it then use the file base caching.
7. Remove blank spaces from your html it will increase your kb. You can use regular expression to remove white spaces. I will post the code for removing white spaces next posts.
8. For asp.net 2.0 and higher version use master pages. It will increase your performance.
9. Prefer database reader over dataset unless and until you have specific reason to use database.
10. Use ADO.NET asynchronous calls for ado.net methods. asp.net 2.0 or higher version is supporting your performance. If you are using same procedure or command multiple time then use ADO.NET Prepare command it will increase your performance.
11. Do IIS performance tuning as per your requirement.
12. Disable view state for your controls if possible. If you are using asp.net 2.0 or higher version then use asp.net control state instead of view state. Store view state in session or database by overriding the default methods for storing view state.
13. User Server.Transfer instead of response.redirect.
14. Always use inproc session state if possible.
15. Use Ajax for your application wisely. Lots of Ajax calls for a page will also decrease your performance.
16. Measure your application performance with tools like redgate profiler,firebug and whyslovw from yahoo.
17. User System.Text.StringBuilder for string concatenation its 4 times more faster then the normal strings.
18. Right JavaScript in .Js files and place it as bottom of the application.
19. Use Separate CSS files for styling of application.
20. User database paging over normal paging while displaying huge amount of data.
21. Call web service from java script instead of server side. Use asynchronous calls to call a web method from web service.

That’s it!!…..Happy Programming...

Here is the free tool for the stress testing in asp.net application.

<http://blogs.msdn.com/alikl/archive/2008/03/09/stress-test-asp-net-web-application-with-free-wcat-tool.aspx>

Another is called asp.net performance engineering which will tell how we can stress application.

<http://blogs.msdn.com/alikl/archive/2008/05/05/asp-net-performance-engineering-stress-test-your-architecture-design-and-code.aspx>

Also go through the following post:

<http://stackoverflow.com/questions/340564/best-way-to-stress-test-a-website>

From my experience before moving to the production environment please take of following things.

1) set debug=false into the web.config

2) set trace enabled=false into the web.config

3) Always use precompiled version of your code.

4) Compile your project into the release mode.

5) Publish your code if you are using asp.net 2.0 or higher version

6) User caching api as much as possible.

7) Decrease your html kb.

8) remove blank spaces from the asp.net html code.

9) Use stylesheet as external .css file

10) USE IIS Compression if poosible.

11) Put your javascript file in .js files

12) Use Server.Transfer instead of Response.redirect

13) Use Inproc Session State if possible.

14) Use Viewstate efficiently- Use controlstate instead of viewstate which is newer feature in asp.net 2.0

15) Avoid giving big name to controls it will increase your html kb.

16) Use Div instead of tables it will decrease your size.

17) Do IIS Performance tuning as per your requirement

Here is the good link that teaches us good way of deployment in production environment.

<http://www.vbdotnetheaven.com/UploadFile/dsdaf/111222006014732AM/1.aspx>