

## Re: .Net Scalability problem

**Source:**

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.adonet/2004-08/0381.html>

---

**From:** Cowboy \((Gregory A. Beamer\) [MVP] (*NoSpamMgbworld\_at\_comcast.netNoSpamM*)

**Date:** 08/05/04

Date: Thu, 5 Aug 2004 07:56:45 -0500

Depending on how you set up tests, you may find 2000 virtual users can be equivalent to hundreds of thousands of users. The test client simulates a much heavier load. For example, on a default setup, Mercury Interative's LoadRunner will peak out a server with a few virtual users.

To get an idea of load, consider your clients. Assume all X clients are logged in at once. Then, determine average lag time between requests and calculate out the number of hits to the server per minute (or second, if you wish). That is the load you have to simulate.

A load test with 2000 clients can potentially tag a server thousands of times per second. If the average person fills out a form (30 seconds), queries the serve and then reads the resulting report (1-2 mintues) and then does it again, even 2000 per second (120,000 per minute) is a load heavier than your users, which will average out to 1 server hit every 2 minutes or so. At 100,000 users, that is 50,000 hits per minute, at peak.

You can load up ACT on multiple machines to get to the proper load, if you need a heavier load.

The biggest point to get is you have to run through the app and see how long it takes you to go through the steps in your application. Remember that you will be faster than the average user, as you know the app very well, but you can use your lag times to determine a hypothetical max load. Multiply the number of hits you have per minute filling out forms, reading reports. (NOTE: You can get a user and watch his times to get a better idea), times the number of users. That is your theorhetical max load (actual max load will likely be much less). Fire off the test client and watch the number of hits per minute. If the load is above your theorhetical max load, you are golden.

Load is all about math and nothing more. Now, ACT allows you to add in lag, et al, but ultimately load is an issue of requests over time.

--

Gregory A. Beamer

MVP; MCP: +I, SE, SD, DBA

\*\*\*\*\*

## microsoft.public.dotnet.framework.adonet: Re: .Net Scalability problem

Think Outside the Box!

\*\*\*\*\*

"Refky Wahib" <wahib@iws-australasia.com> wrote in message  
news:c37196b4.0408042018.11f5080f@posting.google.com...

> Hi  
> Actually I need Technical Support  
> I finished a Great project using .Net and SQL Server and .Net Mobile  
> Control  
> My Business case is to implement this Program to accept about 1  
> Million concurrent users  
> So I designed the project as master Node that has all administration  
> project and about 10 client Nodes  
> Those subscribers will login and I configured the replication at  
> Server that can publish any change to the Nodes  
> And I tried this technique and every thing is perfect.  
> NOW I am at final test  
> I want to proof that my project at every Node is scalable up to 100  
> 000 concurrent users  
> My problem is  
> 1- I have not Any Test tool to generate this Huge connections  
> 2- I do not Know How to configure the tool to let me got reasonable  
> Result  
> 3- I do not know if My Environment is suitable for real test or not  
> 4- Finally I do not know where is the hole at my program that made a  
> bottle neck in this project  
>  
> The Problem in detail  
>  
> I tried to Download ANTS program and tried it but unfortunately it  
> gave me only 5 client to test I do not know if it possible to increase  
> this value at another test program or not?  
> I tried to using Microsoft Test Center but I do not know How to  
> configure the parameter  
> for Example the Only parameter that gave me is No of client and No of  
> Titrations?  
> No of client is very limited to 2000 and you can specify huge no of  
> titration  
> but I still do not understand is it possible that 100 000 concurrent  
> users login at same time ?  
> is the network latency and bandwidth is it a huge bottle neck that  
> will permit the concurrent user to login in different time  
> I think that the average no of request depend on the bandwidth of  
> channel that established between client and server  
> So I think that the MTC generate concurrent connection and per  
> connection try to do it several time  
> Real Test  
> I tried with 2000 concurrent user and 1000 Iteration  
> at NT2003 Server with MS SQL Server 1 G.B RAM and 3GHz  
> and the result was very very strange  
> the program at first connection establish a pool with minimum 20 and  
> max is 100 {I do not know is it suitable for my project or not}  
> I see the performance monitor and I found that the connection begin  
> with 20 connections and increase linearly up to 1000 connections  
> and the result of the test is fail and the error was different and I  
> could not understand the result  
> after that I tied manually to run the application at I could not I  
> found the curser is busy and no reply  
> in the program it was configured that after the db connection time out  
> it reply with error it never happen  
> I wait more than 2 minutes and still busy  
> So I assumed that problem happen in db I restart SQL Server and I  
> found that still problem exist the site too busy and nothing happen

## microsoft.public.dotnet.framework.adonet: Re: .Net Scalability problem

```
> after that I restart the IIS and nothing Happen
> this result very strange because it means the Server feel a very high
> traffic contention at his resource but this resource is not DB or IIS
> sessions
> So what type of this resource??
> Suggestions
> Is this problem because .Net with every hit Create many object and
> leave it to Garbage collection to release and the connection was very
> fast so it makes a problem before triggering the garbage collection??
> is this problem because I did not use the COM+ for pooling ?{but I
> read an article at .NET help called that no need to use COM+ at
> pooling because
> .NET with SQLClient is automatically use the pooling and He advises to
> do not complicate program with using this technique use it only with
> .net with OLEDB like oracle or any db Actually I tried this manually
> and it was work but with few no of connections and observing NT
> Performance monitor with SQL server connections}
> Actually I have no clue How to pin the COM+ to OS and dose I need to
> special tool or I can do it using .NET?
> Is this problem is due to limitation of SQL Server and Windows 2003
> Server as Development tool limitation Actually this server was
> completely installed with toolkit that shipped with .NET enterprise
> this toolkit have SQL Server2000 and windows 2003 windows 2000 and
> exchange server and ecommerce,.
> So is it for Development purpose did not support a huge connection??
> Note: I tried to limit the IIS for only 1000 concurrent connections
> and I found the MTC still send connection and it was more than 1000
> connection and still work without error ? SO I have doubt of the
> program.
> is the problem that I load a lot of information from web.config with
> each connection to program like get the db connection?
>
> I NEED IF IT Possible
> 1-Good test program with the minimum configuration to reach my target
> 100 000 concurrent users
> 2-Article about the bottle neck at any project with .NET and SQL
> Server and .Net Mobile Controls that may be cause a problem?
> 3-If is Microsoft Test Center is good How Can I configure this test to
> reach my target 100 000 concurrent user
> And how can I test it?
> 3- Finally I will write the source code of the main 3 Part
> the Source Code
>
> connection to database
> public Message Connect()
> {
>     UTIL util=new UTIL(session);
>
> if(myConnection!=null)if(myConnection.State==System.Data.ConnectionState.Ope
n)
>
>         {
>             return new
> Message(true,"Is Already opened",0);
>         }
>
>     try
>     {
>
>         myConnection = new SqlConnection();
>
> if(util.ConnectionString.Trim()==String.Empty)util.ConnectionString=UTIL.Use
rApp.GetFromUserPart("DATABASENAME");
>
>
```

## microsoft.public.dotnet.framework.adonet: Re: .Net Scalability problem

```
if(util.ContentDataConnectionString.Trim()==String.Empty)util.ContentDataCon
nectionString=UTIL.UserApp.GetFromUserPart("CONTENTDATABASENAME");
>
>         myConnection.ConnectionString =
> util.ConnectionString;
>         if( myConnection.State.ToString().ToUpper()
> !="OPEN" )      myConnection.Open();
>                 return new Message(true,"",0);
>         }//try
>         catch(Exception ex)
>         {
>                 return new Message(false,ex.Message,-1);
>
>         }
>
>     }
>
> Retrieve data From db
> public Message Retreive(string sql,DataSet dataResult)
>     {
>         long val;
>
>         try
>         {
>             Connect();
>             SqlDataAdapter myDataAdapt=new
> SqlDataAdapter(sql,myConnection);
>             if(dataResult!=null)dataResult.Clear();
>             myDataAdapt.Fill(dataResult,"myData");
>             /*result.l_No*/
>             val=(long)dataResult.Tables["myData"].Rows.Count;
>             Disconnect();
>             return new Message(true,"",val);
>         }
>         catch(Exception e)
>         {
>             Disconnect();
>             return new Message(false,e.Message,-1);
>         }
>     }
>
> }
```