Blog

Subscribe

← Older Posts
Newer Posts →

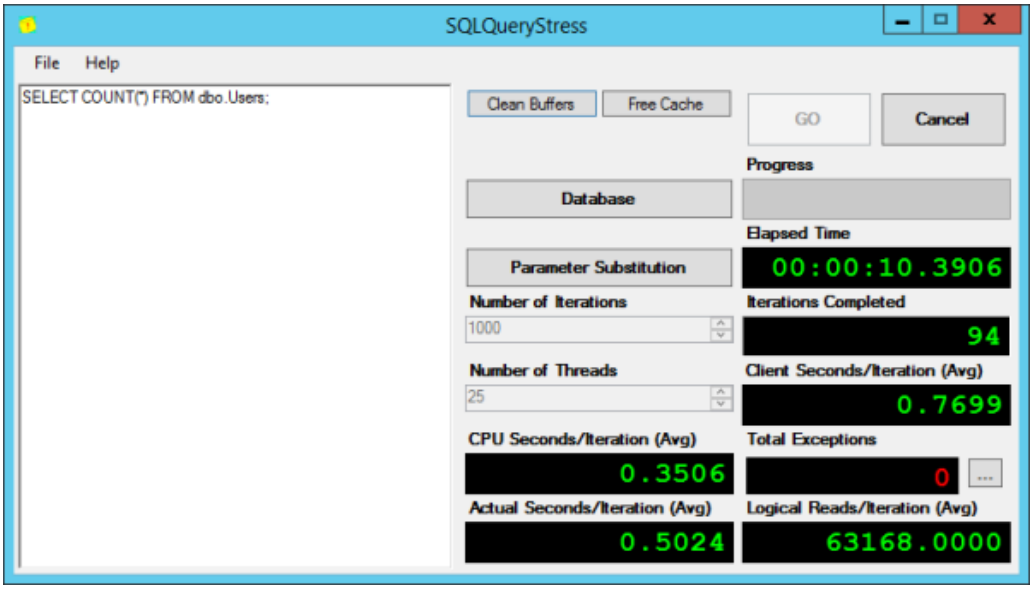
How to Fake Load Tests with SQLQueryStress

by Brent Ozar May 7, 2015
16 comments

Load testing – real, serious load testing – is hard.

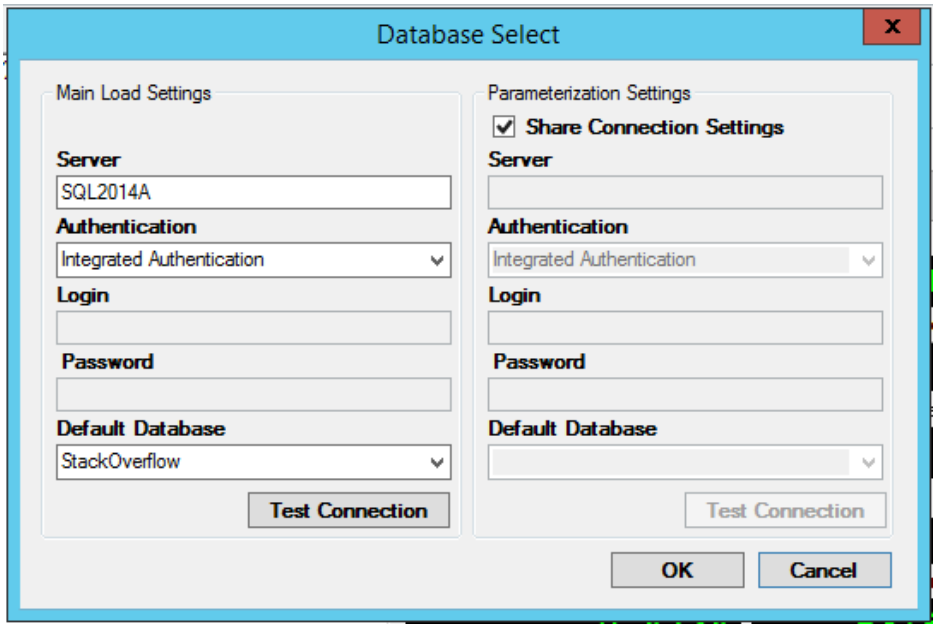
In a perfect world, you want to exactly simulate the kinds of queries that the end users will be throwing at your SQL Server. However, in the words of a timeless philosopher, ain't nobody got time for that.

Instead, let's use the neato open source tool SqlQueryStress (latest exe) to fake it. This is an oldie but goldie app that will run any one query thousands of times (or more) from dozens of sessions (or more), all from the comfort of your desktop:



SQLQueryStress in action

After you install it on your desktop (or a VM in the data center, whatever, just not the SQL Server you're trying to load test), click the Database button to set up your connection string. In this instance, I'm pointing it at one of my Availability Groups, using Windows authentication. As soon as I set the server and auth methods, the database list gets populated so I can set my default database:



Setting up the connection string

Then it's time to pick the query to run.

But you want to test more than one query at a time, right? You want to test a variety of different queries running all at once.

Rather than calling a single query, call a "shell" stored procedure that runs other queries. Here's how it works:

1. Declare an integer, and set it to a random number
2. Based on the mod of that number, run a stored procedure
   (for example, if it's divisible by 3, run sp_C,
   else if it's divisible by 2, run sp_B,
   else run sp_A.)

Since SQLQueryStress will be calling this stored proc dozens of times at once, you'll end up with a variety of different queries running simultaneously.

Let's get a little more complex. Here's what mine looks like for one of my query tuning demos:

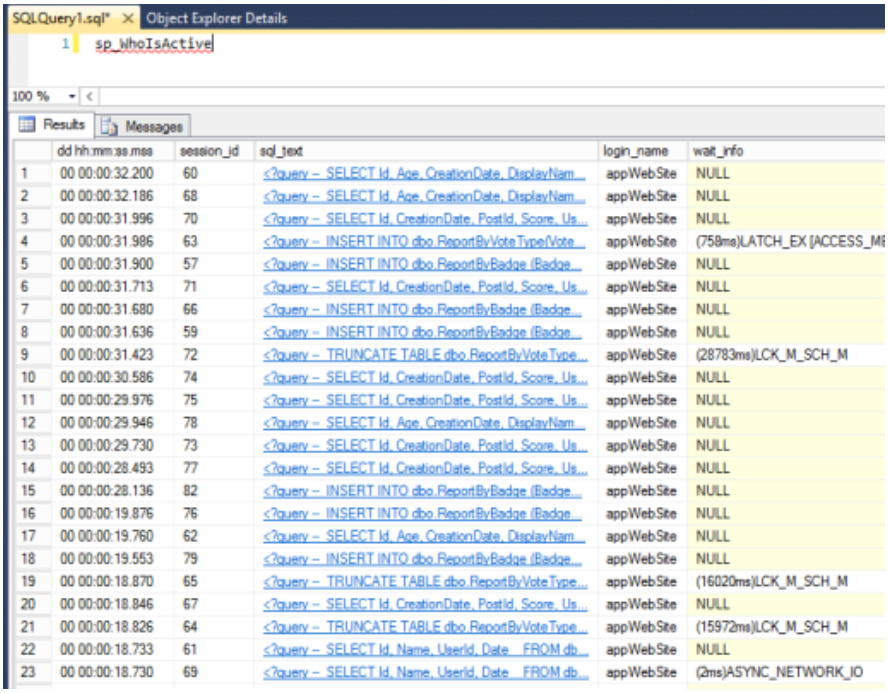
```
CREATE PROCEDURE [dbo].[usp_RandomQ] WITH RECOMPILE
AS
SET NOCOUNT ON
DECLARE @Id INT
SELECT @Id = CAST(RAND() * 10000000 AS INT)
IF @Id % 7 = 0
    EXEC dbo.Refresh_ReportByVoteType
ELSE IF @Id % 6 = 0
    EXEC dbo.Refresh_ReportByBadge
ELSE IF @Id % 5 = 0
    EXEC dbo.GetBadgesDetails @Id
ELSE IF @Id % 4 = 0
    EXEC dbo.GetCommentsDetails @Id
ELSE IF @Id % 3 = 0
    EXEC dbo.GetPostsDetails @Id
ELSE IF @Id % 2 = 0 AND @@SPID % 2 = 0
    EXEC dbo.GetUsersDetails @Id
ELSE
    EXEC dbo.GetVotesDetails @Id
GO
```

WITH RECOMPILE – I use this because I don't want the usp_RandomQ stored procedure to show up in my execution plan stats. The work involved with building this execution plan isn't significant, and it won't be the largest part of my workload. (Oh, I wish it were.) All of the stored procs it calls will still show up in the plan cache, though.

@Id parameter – note that some of the stored procs take an @Id. For example, the stored proc GetBadgesDetails takes @Id, and uses that to look up a particular badge number's details. This is handy because each of my stored procedures don't have random number generators – they're designed to mimic more real-world stored procs that have input values. If you wanted to get really fancy and test procs with lots of parameters, you'll need to generate those in usp_RandomQ. You don't want to hard-code the same values because then that relevant table data will end up getting cached in memory.

@@SPID – some of my workload queries simulate blocking. Due to the wonders of random number generation and very fast queries, if a blocking chain starts on any two sessions, then eventually the rest of the sessions will also call the stored proc that's susceptible to blocking. After a few seconds, the only symptom my server will have is blocking – and that's no fun. Instead, by using @@SPID %2 before calling GetVotesDetails (which gets blocked in my scenario), I make sure that no more than half of my sessions will get blocked at once.

The end result is beautiful – well, at least if you want something that looks like a production server getting hammered with all kinds of different queries:


SQLQuery1.sql* | Object Explorer Details

```
sp_WhoIsActive
```

100 %

Results | Messages

	dd hh:mm:ss.mss	session_id	sql_text	login_name	wait_info
1	00 00:00:32.200	60	<?query -- SELECT Id, Age, CreationDate, DisplayNam...	appWebSite	NULL
2	00 00:00:32.186	68	<?query -- SELECT Id, Age, CreationDate, DisplayNam...	appWebSite	NULL
3	00 00:00:31.996	70	<?query -- SELECT Id, CreationDate, PostId, Score, Us...	appWebSite	NULL
4	00 00:00:31.986	63	<?query -- INSERT INTO dbo.ReportByVoteType(Vote...	appWebSite	(758ms)LATCH_EX [ACCESS_M
5	00 00:00:31.900	57	<?query -- INSERT INTO dbo.ReportByBadge (Badge...	appWebSite	NULL
6	00 00:00:31.713	71	<?query -- SELECT Id, CreationDate, PostId, Score, Us...	appWebSite	NULL
7	00 00:00:31.680	66	<?query -- INSERT INTO dbo.ReportByBadge (Badge...	appWebSite	NULL
8	00 00:00:31.636	59	<?query -- INSERT INTO dbo.ReportByBadge (Badge...	appWebSite	NULL
9	00 00:00:31.423	72	<?query -- TRUNCATE TABLE dbo.ReportByVoteType...	appWebSite	(28783ms)LCK_M_SCH_M
10	00 00:00:30.586	74	<?query -- SELECT Id, CreationDate, PostId, Score, Us...	appWebSite	NULL
11	00 00:00:29.976	75	<?query -- SELECT Id, CreationDate, PostId, Score, Us...	appWebSite	NULL
12	00 00:00:29.946	78	<?query -- SELECT Id, Age, CreationDate, DisplayNam...	appWebSite	NULL
13	00 00:00:29.730	73	<?query -- SELECT Id, CreationDate, PostId, Score, Us...	appWebSite	NULL
14	00 00:00:28.493	77	<?query -- SELECT Id, CreationDate, PostId, Score, Us...	appWebSite	NULL
15	00 00:00:28.136	82	<?query -- INSERT INTO dbo.ReportByBadge (Badge...	appWebSite	NULL
16	00 00:00:19.876	76	<?query -- INSERT INTO dbo.ReportByBadge (Badge...	appWebSite	NULL
17	00 00:00:19.760	62	<?query -- SELECT Id, Age, CreationDate, DisplayNam...	appWebSite	NULL
18	00 00:00:19.553	79	<?query -- INSERT INTO dbo.ReportByBadge (Badge...	appWebSite	NULL
19	00 00:00:18.870	65	<?query -- TRUNCATE TABLE dbo.ReportByVoteType...	appWebSite	(16020ms)LCK_M_SCH_M
20	00 00:00:18.846	67	<?query -- SELECT Id, CreationDate, PostId, Score, Us...	appWebSite	NULL
21	00 00:00:18.826	64	<?query -- TRUNCATE TABLE dbo.ReportByVoteType...	appWebSite	(15972ms)LCK_M_SCH_M
22	00 00:00:18.733	61	<?query -- SELECT Id, Name, UserId, Date FROM db...	appWebSite	NULL
23	00 00:00:18.730	69	<?query -- SELECT Id, Name, UserId, Date FROM db...	appWebSite	(2ms)ASYNC_NETWORK_IO

AAAAAHHHH, THE SERVER'S ON FIRE

I love using this quick-load-generation technique in our performance tuning classes. It's a great way to show a server that looks like home, and gets students to figure out which queries are causing problems – and which ones are just harmless background noise.

Go get SQLQueryStress now, and get the usp_RandomQ scripts for the top queries from Data.StackExchange.com.

Brent Ozar

I make Microsoft SQL Server faster and more reliable. I love teaching, travel, and laughing.

I'm mostly a figurehead here at Brent Ozar Unlimited. My true skills are menu advice, interpretive dance, and reading Wikipedia.

↑ Back to top

16 comments

Leave a comment

1. *Karthik* May 7, 2015 | 11:49 am

   This is awesome! I will download and test it right away!! Thanx a lot!!

   Reply

   - 

     Brent Ozar May 7, 2015 | 2:36 pm

     You're welcome. It's a great tool.

Reply

2. 

Eyob May 9, 2015 | 8:50 am

Thanks, it looks great. I can't wait to test it.

Reply

3. 

Gyan May 9, 2015 | 8:58 pm

This is awesome tool ..

Reply

4. 

Andy CW June 19, 2015 | 11:39 am

Just downloaded thsi tool, great, just what we need to run some stress testing. Thanks for this little gem!

Reply

5. 

Donny December 30, 2015 | 11:03 am

The link no longer works, anyone got an update link or url where I can download this tool?

Reply

- 

Brent Ozar December 30, 2015 | 11:55 am

Ooo, unfortunately looks like he's taken it down.

Reply

6. 

Arifin January 7, 2016 | 1:15 am

Can somebody re-upload it to codeplex or somewhere ?

Reply

- 

Hotsauce February 19, 2016 | 4:53 pm

Its on GitHub

https://github.com/ErikEJ/SqlQueryStress

Reply

7. 

   Arifin January 7, 2016 | 1:26 am

   Hey, I got it here.

   http://sqlquerystress.software.informer.com/

   Reply

8. 

   Douglas Schmitt January 14, 2016 | 3:34 pm

   Check out Adam's recent post about SQLQueryStress:

   http://sqlblog.com/blogs/adam_machanic/archive/2016/01/04/sqlquerystress-the-source-code.aspx

   Reply

9. 

   sdhdfghdfg February 19, 2016 | 1:23 pm

   http://wayback.archive.org/web/20151028024821/http://www.datamanipulation.net/SQLQueryStress/sqlquerystress_0_9_7.msi

   Reply

10. 

    basava March 15, 2016 | 1:09 pm

    Hi Brent, Can you elaborate little more on your true skills like menu advice,interpretive dancing… Well I suppose your hidden true skills are Database Consulting,Correcting Mis-Interpreted SQL Query Optimizer

    Cheers,
    Basava

    Reply

    - 

      Brent Ozar March 15, 2016 | 4:24 pm

      No, no, I'm not really all that good at consulting. Interpretive dance, though, I'm the bomb.

      Reply

11. 

    Asim Paracha April 13, 2016 | 5:36 pm

    Great, thanks. What is the best way to kill all the sessions that were started from this tool?

    Reply

    - 

Asim Paracha April 13, 2016 | 5:40 pm

I think I figured it out myself. Just Exited the tool and looks like that terminated all associated sessions.

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

Post Comment

Notify me of followup comments via e-mail. You can also subscribe without commenting.

← Older Posts
Newer Posts →

- SQL Server on VMware and SANs

Popular Topics

- Query Execution Plans
- SQL Server Clusters
- Table Partitioning
- SQL Server 2012 AlwaysOn
- SQL Server Indexing
- Performance Tuning
- SQL Server on SANs
- SQL Server on VMware
- SQL Interview Questions
- SQL Server Setup Checklist
- sp_Blitz®: Free Health Check

Archives by Author

- Brent Ozar
- Angie Rudduck
- Doug Lane
- Erik Darling
- Tara Kizer

Recent Posts by Date

- August 2016
- July 2016
- June 2016
- May 2016
- April 2016
- March 2016
- February 2016
- January 2016
- December 2015
- November 2015
- October 2015
- September 2015

Get SQL news before it actually happens! Enter your email address here

Subscribe

- Home
- Problems We Solve
- Training
- First Aid
- Blog
- Team
- Contact