Query To Simulate CPU Load

Quite often when working testing different performance measures from the Database Health Reports to testing Resource Governor configuration I end up in a situation where I need to emulate a high CPU load on my test SQL Server. In this case it is usually my desktop or a Hyper-V virtual machine, where I want to see how things will work if I had a nearly 100% CPU load. In the real world you would just have to get millions of users to visit your website that has a SQL Server backend, that its not that easy in my development / test lab.

Here is that I came up with for a plan. Keeping in mind the goal here is to write queries that will use up as much CPU as possible. This is counter-intuitive, completely against everything that I practice on a daily basis, but here goes.

First create a table with poor design. Using UNIQUEIDENTIFIERS for a primary key and a foreign key (parent_id) is probably ugly enough.

Next we fill the table up with lots and lots of rows, specifically 1,000,000 rows, remember here the goal is to simulate CPU load. If this isn't enough I often times run this script several times. Keep in mind the GO statement followed by a number says to execute the batch that many times.

Next, this part makes me just feel nasty. Create a CLUSTERED index on the table that we just filled up, and cluster on BOTH columns that were UNIQUEIDENTIFIERS.

At this point is is a bit ugly, but it still doesn't use much memory. You are probably wondering why I called the table split thrash. I gave it this name so that updating the UNIQUEIDENTIFER would cause as many page splits or new page allocations as possible. So we update the parent_id which is part of the clustered index

This update statement causes chaos in the page structure for the table as updating the unique identifiers causes quite a bit of processor work.

On my wimpy VM for this development environment I need to repeat this entire process creating 4 or 5 tables, and doing the update in 4 or 5 SSMS windows in order to use up all of the CPU on the database.

Don't try this on a production system, but it is a great test to run on a development server.

See Also:

If you like this you might also like my article on how to use up all of your memory with a common table expression.