

## SQL Server Instance-Level Benchmarks with DVDStore3

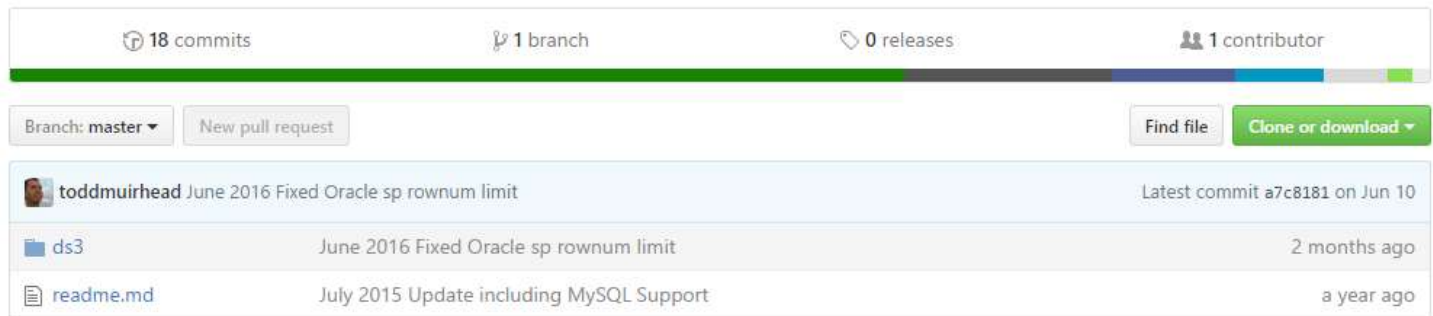
Dell developed a synthetic benchmark tool that can run benchmark tests against SQL Server, Oracle, MySQL, and PostgreSQL installations. It is open-sourced and freely available at <https://github.com/dvdstore/ds3/>. It generates a benchmark value in the form of 'orders placed' per test cycle, and tests the performance of the instance configuration and the infrastructure underneath it. You create a synthetic workload dataset with an included tool, load it into the instance via a new test database (DS3), and then run an included load testing tool to simulate orders of DVDs off of a web store.

This tool is used quite frequently by Heraflux to compare the relative performance between two SQL Server instances. Load the same test data into two different instances, run the same test on each server, and compare the results. Raw server performance is compared, and differences in the server performance can be isolated.

### Pre-Requisites

A base bundle is shipped for all database engines, and the current release is located on the main GitHub page. You can download the entire package by clicking the 'Clone or download' button.

DVD Store version 3



Extract this bundle to the drive of your choice, and then move the inner folder ds3-master/ds3 to the root of the drive to keep things simple.

It will create a file structure that is similar to the following.

```
E:\ds3>dir
Volume in drive E is Storage
Volume Serial Number is C4D4-43BF

Directory of E:\ds3

08/14/2016  03:55 PM  <DIR>          .
08/14/2016  03:55 PM  <DIR>          ..
08/14/2016  03:49 PM             5,308  CreateConfigFile.pl
08/14/2016  03:55 PM  <DIR>          data_files
08/14/2016  03:55 PM  <DIR>          drivers
08/14/2016  03:49 PM      10,103  ds2_change_log.txt
08/14/2016  03:49 PM           404  ds3_change_log.txt
08/14/2016  03:49 PM     29,396  ds3_Documentation.txt
08/14/2016  03:49 PM      1,608  ds3_faqs.txt
08/14/2016  03:49 PM      2,283  ds3_readme.txt
08/14/2016  03:49 PM      7,206  ds3_schema.txt
08/14/2016  03:49 PM     18,013  gpl.txt
08/14/2016  03:56 PM     36,861  Install_DVDStore.pl
08/14/2016  03:55 PM  <DIR>          mysqlds3
08/14/2016  03:55 PM  <DIR>          oracleds3
08/14/2016  04:12 PM  <DIR>          sqlserverds3
07/25/2016  10:18 AM           274  testsqlserver.bat
```

The creation script will prompt you to specify the size of the target database. It will create CSV files on the local file system containing data with this much space. Make sure you have enough space on the drive you place them on to create the workload files.

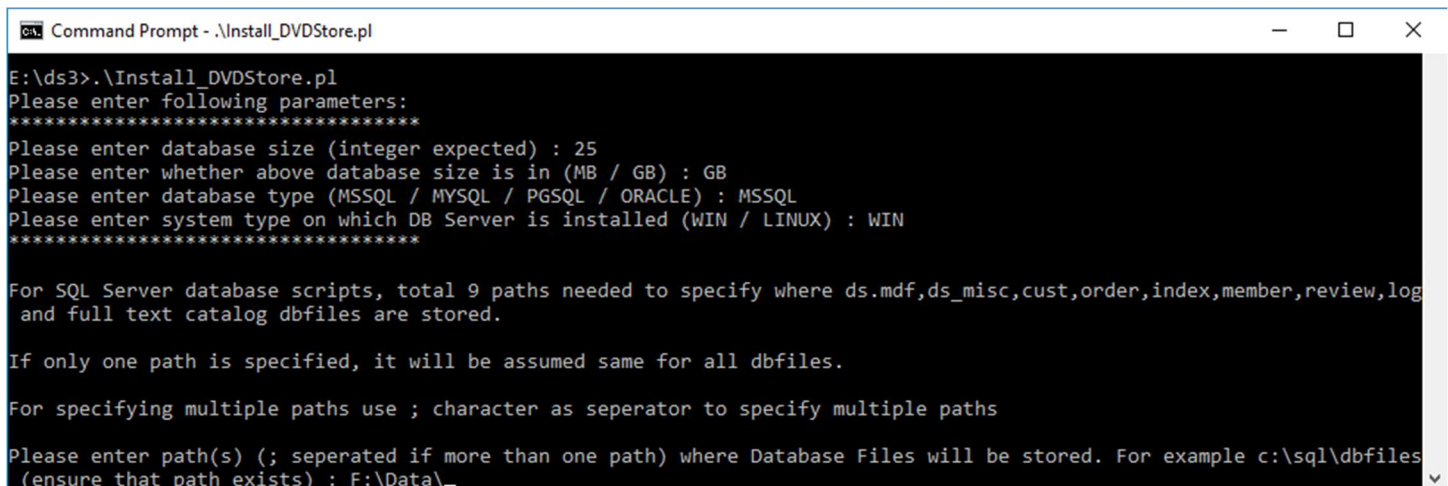
Next, you need to install a Perl for Windows package so the workload files can be successfully generated. Commonly used is ActiveState's ActivePerl community edition, available at <http://www.activestate.com/activeperl/downloads>. Download the version that matches your Windows platform and install it.

### Setup

Next, create the workload files. You should be doing this on the database server itself so you do not have to copy the workload files across the network after you create them. How large should you make the workload dataset? Strive for targeting the average size of the database that you are looking to migrate between platforms, or just a good size that represents average data volumes in your environment. Again, remember that you need this amount of space on the drive where you are building the workload files.

For this example's sake, generate a 25GB workload dataset here. Open an elevated command prompt and change directory into the new DS3 directory that you created. Run `Install_DVDStore.pl` to begin the process.

Enter the database size (integer value), then enter the measurement of space (usually GB). Enter the database platform (MSSQL) and the platform (WIN).



```
Command Prompt - .\Install_DVDStore.pl
E:\ds3>.\Install_DVDStore.pl
Please enter following parameters:
*****
Please enter database size (integer expected) : 25
Please enter whether above database size is in (MB / GB) : GB
Please enter database type (MSSQL / MYSQL / PGSQL / ORACLE) : MSSQL
Please enter system type on which DB Server is installed (WIN / LINUX) : WIN
*****
For SQL Server database scripts, total 9 paths needed to specify where ds.mdf,ds_misc,cust,order,index,member,review,log
and full text catalog dbfiles are stored.
If only one path is specified, it will be assumed same for all dbfiles.
For specifying multiple paths use ; character as separator to specify multiple paths
Please enter path(s) (; seperated if more than one path) where Database Files will be stored. For example c:\sql\dbfiles
(ensure that path exists) : F:\Data\
```

Next, enter the location where the new database objects will be placed on the SQL Server's file system itself. It does this as part of the create database script. These locations can be changed later if necessary. Please remember to place a trailing backslash at the end of the path because the build script does not do this for you. This example uses the `F:\Data\` directory.

Once you hit enter, it will begin to generate the CSV workload dataset files.

```

C:\> Command Prompt - .\Install_DVDStore.pl

For specifying multiple paths use ; character as separator to specify multiple paths

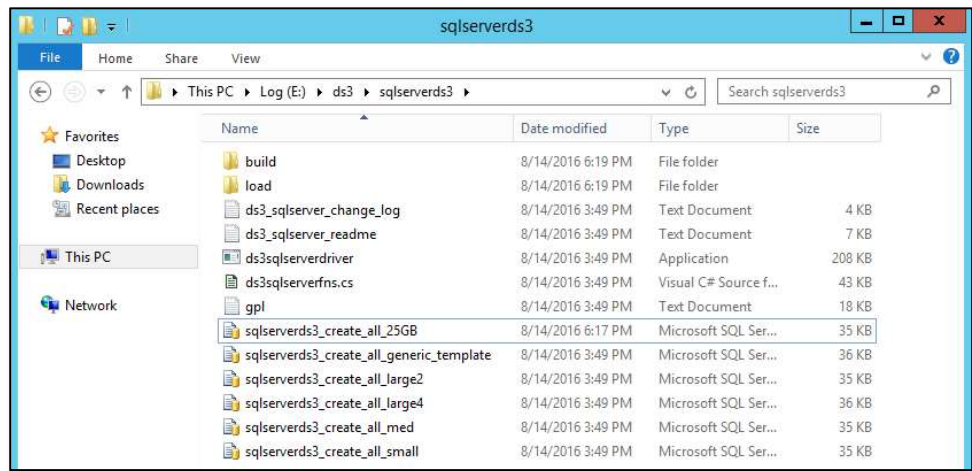
Please enter path(s) (; seperated if more than one path) where Database Files will be stored. For example c:\sql\dbfiles
(ensure that path exists) : F:\Data\
*****
Initializing parameters...
*****
Database Size: 25
Database size is in GB
Database Type is MSSQL
System Type for DB Server is WIN
File Paths : F:\Data\
*****

Calculating Rows in tables!!
Large size database ( greater than 1 GB)
Ratio calculated : 0.25
Customer Rows: 50000000
Order Rows / month: 2500000
Product Rows: 250000

Creating CSV files....
Starting to create CSV data files....
For larger database sizes, it will take time.
Do not kill the script till execution is complete.

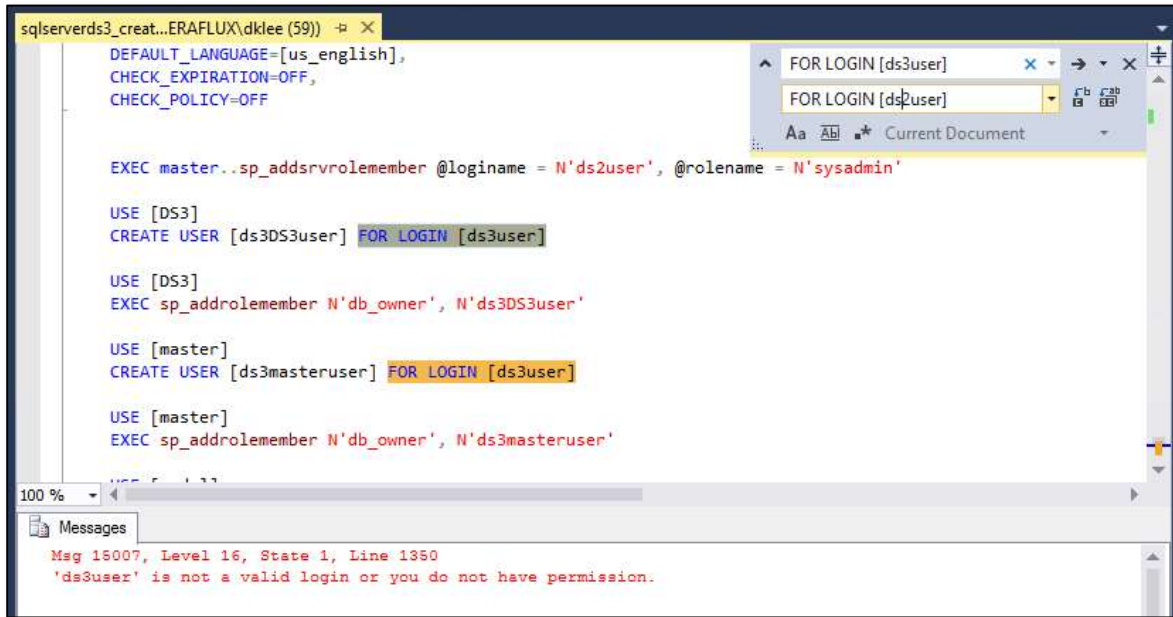
Creating Customer CSV files!!!
1 25000000 US L 1
  
```

A short period of time later, and now you are ready for the next step. If you navigate to the sqlserverds3 directory, you will now see a script called sqlserverds3\_create\_all\_25GB.sql. If you have multiple scripts, it is the script containing the same size specification as what you just entered into the data build script.



Do not execute this script yet. DVDStore for SQL Server has a small bug in it. Edit the highlighted file with your favorite editor. SQL Server Management Studio will work quite well.

Scroll to approximately line 1349. You will see it is a create user command with it specifying the login as ds3user. Do a find and replace in the entire file and replace ds3user with ds2user, as this is a holdover from the previous version of the benchmarking tool. Save the file.



At this time, verify that full text search is enabled on your SQL Server instance.



If you see a 1 as the result, Full Text Search is installed and enabled. If not, work with your DBA to install Full Text Search into this SQL Server instance.

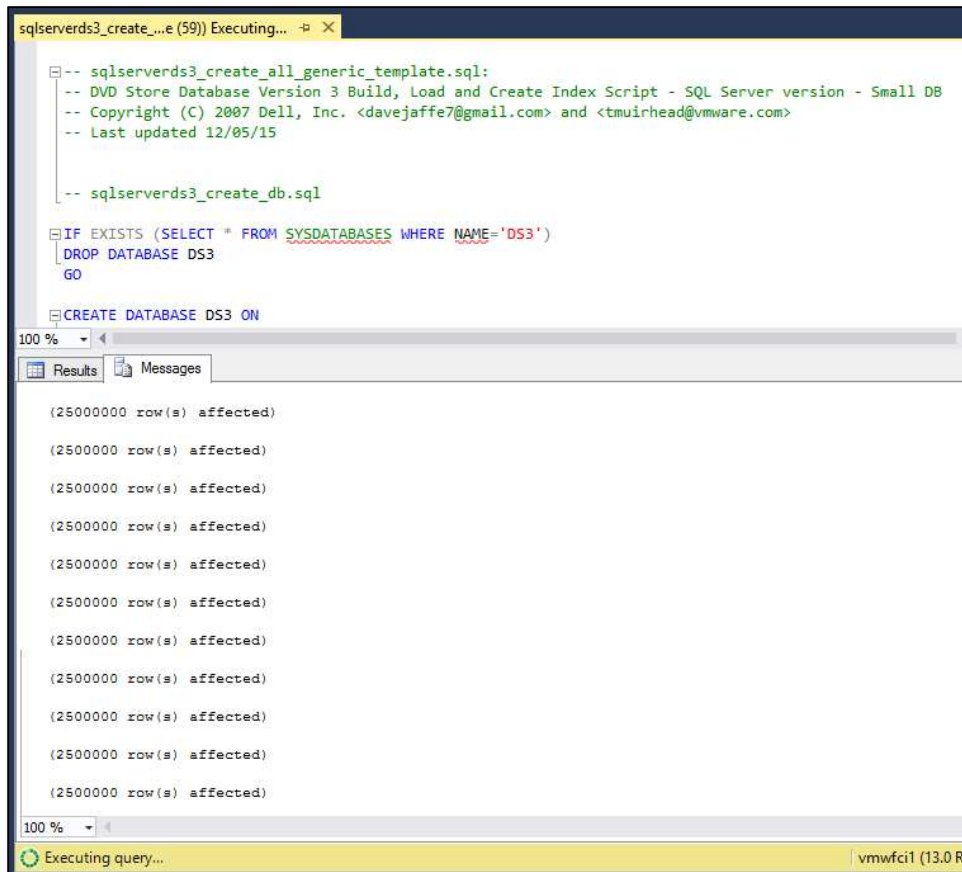
Now, review the script. Some DBAs wish to keep the database transaction log files on a different drive (good practice in most circumstances). The portions at the top of the script specify the locations of the various database components. Review these locations and make any changes as necessary.



## Data Load

Now, load and execute the script while logged into the instance as a sysadmin. It will create the new 'DS3' database and a new login called 'ds2user'. This login does not have the ability to set or change the password because of the way the workload files were coded, so for security purposes disable this login when your tests are complete.

It then loads the CSV workload files with bulk insert commands. It could take a bit of time, so be patient. You can monitor the progress in the query's Messages window.



When completed, run whatever index maintenance job that you have available on this new database. I recommend the Ola Hallengren database maintenance solution, freely available at <http://ola.hallengren.com>. Ensure that the statistics update job is updating all statistics and not those associated with indexes.

Finally, take a backup of this database so that you can restore it on this or other instances to replay this test to get relative performance differences, if necessary.

## Test Execution

Now, it's time for the actual test. Create a new batch file in the \DS3 directory. A sample and easily identifiable name could be 'runstresstest.bat'. In the batch file, add the line:

```
.\sqlserverds3\ds3sqlserverdriver.exe --target=localhost --n_threads=8 --run_time=5
--ramp_rate=10 --db_size=25GB --warmup_time=0 --think_time=0.085 --n_searches=3
--search_batch_size=5 --n_line_items=5 --windows_perf_host=agdb01a --detailed_view=Y
--out_filename=results.txt
```

In this string, please change your target if you intend to run this across a network or if you have a named SQL Server



instance. Set your database size according to how you built the workload files. Adjust your number of threads to set the expected load. If you need more threads than 64, I suggest running a second copy of the batch file to get more threads, and would set the test configuration at increments of the total thread count required. Set the run time to the number of minutes that you wish to run the test. At a minimum, I suggest at least five minutes, and probably longer if you are performing an infrastructure load test.

The think\_time setting is the amount of time that a simulated user would ‘think’ before clicking again. I have the time set very low so I can stress test the instance. I store the output to a ‘results.txt’ file so I can record the results later.

Now, schedule a time that works best for your test and execute it. If this is a production instance with background production tasks, this test will throttle the CPUs and memory, so be careful not to disrupt production tasks. Make sure to have whatever monitoring tool or Perfmon running so you can capture the performance metrics – i.e. CPU usage, memory PLE, I/O activity, etc. – so you know how hard this machine is being throttled.

You may need to execute this command as administrator since it requires registry access to query the target server with WMI to capture CPU load.

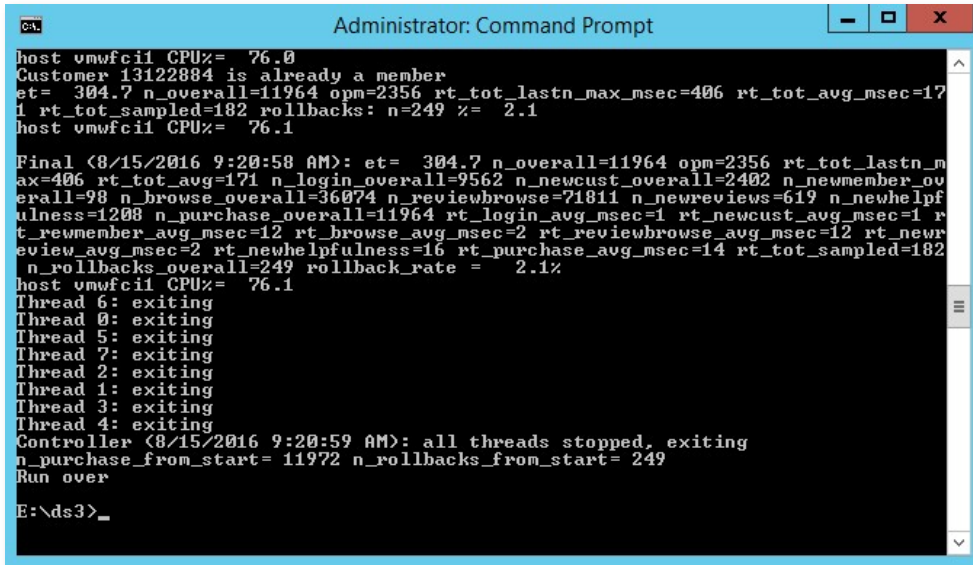
It looks like the test is not doing much at all in the command window. The window shows periodic updates with orders per minute (OPM) values.

```

Administrator: Command Prompt - runstresstest.bat
E:\ds3>runstresstest.bat
E:\ds3>.\\sqlserver\ds3\sqlserverdriver.exe --target=vmwfcil --n_threads=8 --run_time=300 --ramp_rate=10 --db_size=25GB --warmup_time=0 --think_time=0.085 --n_searches=3 --search_batch_size=5 --n_line_items=5 --windows_perf_host=vmwfcil --detailed_view=Y --out_filename=testresults.txt
Total number of Threads to be Spawned across multiple servers are n_threads: 8
target= vmwfcil n_threads= 8 ramp_rate= 10 run_time= 300 db_size= 25GB warmup_time= 0 think_time= 0.085 pct_newcustomers= 20 pct_newmembers= 1 n_searches= 3 search_batch_size= 5 n_reviews=3 pct_newreviews=5 pct_newhelpfulness=10 n_line_items=5 virt_dir= ds3 page_type= php windows_perf_host= vmwfcil detailed_view= Y linux_perf_host=
Using .NET DateTime for measuring response time
Performance Counter Category Test and Counter MaxRT exist
Thread 1: created for User 1
Thread 3: created for User 3
Thread 2: created for User 2
Thread 7: created for User 7
Thread 5: created for User 5
Thread 4: created for User 4
Thread 0: created for User 0
Thread 6: created for User 6
Thread 3: connected to vmwfcil
Controller (8/15/2016 8:56:19 AM): all threads running
Controller: n_threads_connected = 1 : ConnectionTimeout remaining 60
Thread 1: connected to vmwfcil
Controller: n_threads_connected = 2 : ConnectionTimeout remaining 59
Controller: n_threads_connected = 2 : ConnectionTimeout remaining 58
Thread 6: connected to vmwfcil
Controller: n_threads_connected = 3 : ConnectionTimeout remaining 57
Thread 4: connected to vmwfcil
Controller: n_threads_connected = 4 : ConnectionTimeout remaining 56
Controller: n_threads_connected = 4 : ConnectionTimeout remaining 55
Thread 2: connected to vmwfcil
Controller: n_threads_connected = 5 : ConnectionTimeout remaining 54
Thread 7: connected to vmwfcil
Thread 0: connected to vmwfcil
Controller: n_threads_connected = 7 : ConnectionTimeout remaining 53
Controller: n_threads_connected = 7 : ConnectionTimeout remaining 52
Thread 5: connected to vmwfcil
Controller (8/15/2016 8:56:28 AM): all threads connected - issuing Start
  
```

You get an entry in the output file every ten seconds, with an orders per minute (OPM) value recorded.

Finally, the test will end, and you should have a final ordered placed value for the test.



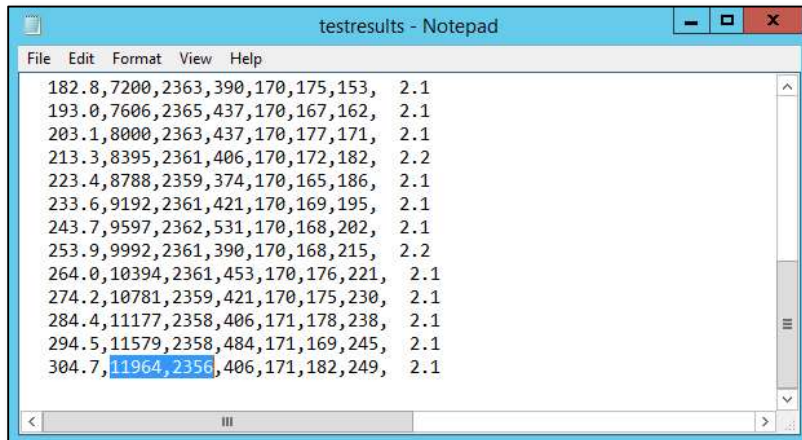
```
Administrator: Command Prompt

host umwfcil CPU%= 76.0
Customer 13122884 is already a member
et= 304.7 n_overall=11964 opm=2356 rt_tot_lastn_max_msec=406 rt_tot_avg_msec=17
1 rt_tot_sampled=182 rollbacks: n=249 %= 2.1
host umwfcil CPU%= 76.1

Final (8/15/2016 9:20:58 AM): et= 304.7 n_overall=11964 opm=2356 rt_tot_lastn_m
ax=406 rt_tot_avg=171 n_login_overall=9562 n_newcust_overall=2402 n_newmember_ov
erall=98 n_browse_overall=36074 n_reviewbrowse=71811 n_newreviews=619 n_newhelpf
ulness=1208 n_purchase_overall=11964 rt_login_avg_msec=1 rt_newcust_avg_msec=1 r
t_rewmember_avg_msec=12 rt_browse_avg_msec=2 rt_reviewbrowse_avg_msec=12 rt_newr
eview_avg_msec=2 rt_newhelpfulness=16 rt_purchase_avg_msec=14 rt_tot_sampled=182
n_rollbacks_overall=249 rollback_rate = 2.1%
host umwfcil CPU%= 76.1
Thread 6: exiting
Thread 0: exiting
Thread 5: exiting
Thread 7: exiting
Thread 2: exiting
Thread 1: exiting
Thread 3: exiting
Thread 4: exiting
Controller (8/15/2016 9:20:59 AM): all threads stopped, exiting
n_purchase_from_start= 11972 n_rollbacks_from_start= 249
Run over
E:\ds3>_
```

This number is the relative score for the instance's performance on this particular test.

Open the testresults.txt output file when the test completes.



```
testresults - Notepad

File Edit Format View Help

182.8,7200,2363,390,170,175,153, 2.1
193.0,7606,2365,437,170,167,162, 2.1
203.1,8000,2363,437,170,177,171, 2.1
213.3,8395,2361,406,170,172,182, 2.2
223.4,8788,2359,374,170,165,186, 2.1
233.6,9192,2361,421,170,169,195, 2.1
243.7,9597,2362,531,170,168,202, 2.1
253.9,9992,2361,390,170,168,215, 2.2
264.0,10394,2361,453,170,176,221, 2.1
274.2,10781,2359,421,170,175,230, 2.1
284.4,11177,2358,406,171,178,238, 2.1
294.5,11579,2358,484,171,169,245, 2.1
304.7,11964,2356,406,171,182,249, 2.1
```

The two primary benchmark values are total orders placed per test cycle (second column in the output file) and orders per minute (third column in the output file).

Now, you are free to re-run this test in any permutation you see fit for your purposes. Restore your backup so you are working from the exact same dataset before each test. Document what you change if you are testing infrastructure or instance changes, and try to only make one change at a time.

You can now test infrastructure, instance, or database configuration changes and see the raw performance differences in one simple to read number.