HammerDB Time Profiling

This guide gives you an introduction to time profiling functions with HammerDB. You should already be familiar with running HammerDB workloads before capturing time profiling information.

Understanding Time Profiling

Time Profiling is a method by which you can capture the time it takes for individual procedures to run within HammerDB from the client running HammerDB as opposed to the SUT. (although depending on the database software running on the SUT you may be also able to time procedures there as well). The aim of time profiling is to measure client response times. Response times are an essential aspect of system performance in addition to system throughput measured as TPM, NOPM or Queries per minute. The response time is an indicator of user experience and should be measured at increasing degrees of system load. Figure 1 illustrates a comparison of response times for different levels of system load for the HammerDB OLTP test.

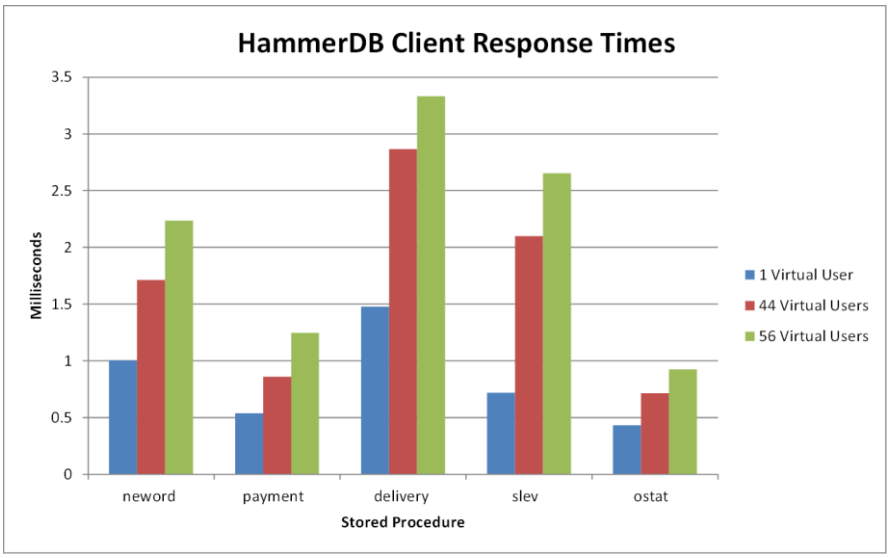

Figure 1 Client Response Times

It is important to understand the relationship of response times to the performance profile. Figure 2 shows an example of an actual performance profile comparison (unrelated to the response times in Figure 1.) There are a number of expectations that can be made from examining the performance profile. Firstly as shown in Figure 1 as the performance profile is examined from left to right with increased user load you would expect response times to increase to some degree, however by how much depends on the system especially the CPU performance and better performance is illustrated by the steepness of the profile. Additionally once the curve has either flattened or begins descending the system is processing the same (or if descending lower) number of transactions per minute for a higher number of virtual users and therefore it would be expected that response times would start to increase proportionally. Finally a flatter profile as

shown with the bottom curve in Figure 2 given a significantly large number of virtual users could eventually reach the same throughput (TPM, NOPM) as the steeper profile however the response times would be significantly greater at this point and therefore overall system TPM or NOPM should not be taken as the sole measurement.

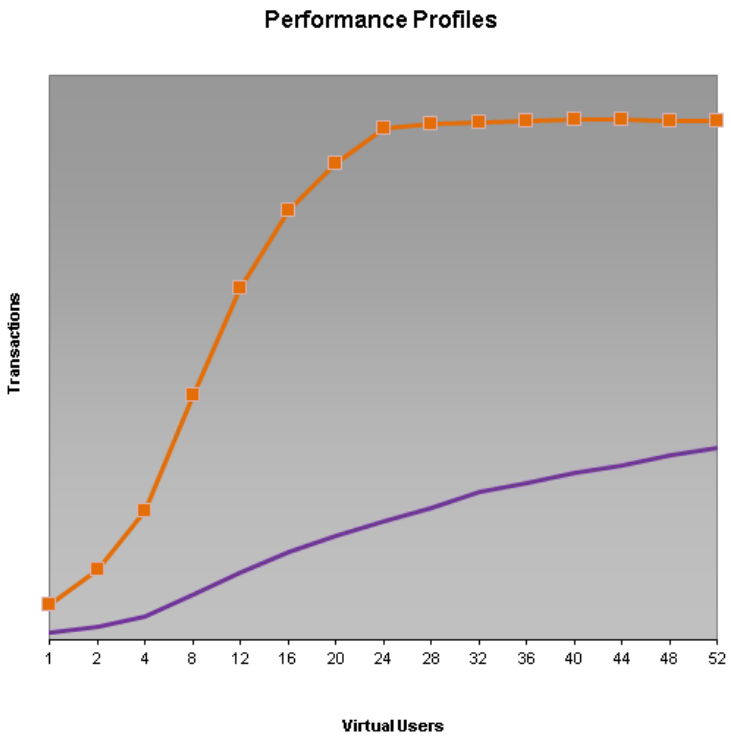

Figure 2 Performance Profiles

In addition to comparing systems time profiling can also be used to diagnose contention and bottlenecks in the database software. For example Figure 3 whereas the delivery procedure is experiencing contention resulting in a significant increase in virtual user response times.

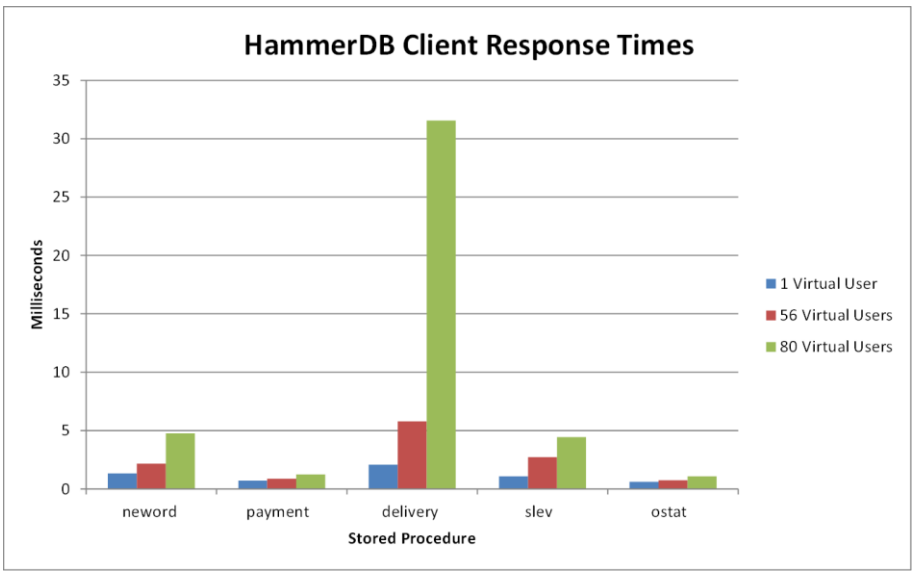

Figure 3 Client Response Times Contention

In this scenario this chart enables the drilling down into the data experiencing contention and in with this example partitioning the table in question significantly reduced response times at a higher system load and therefore also dramatically increased throughput.

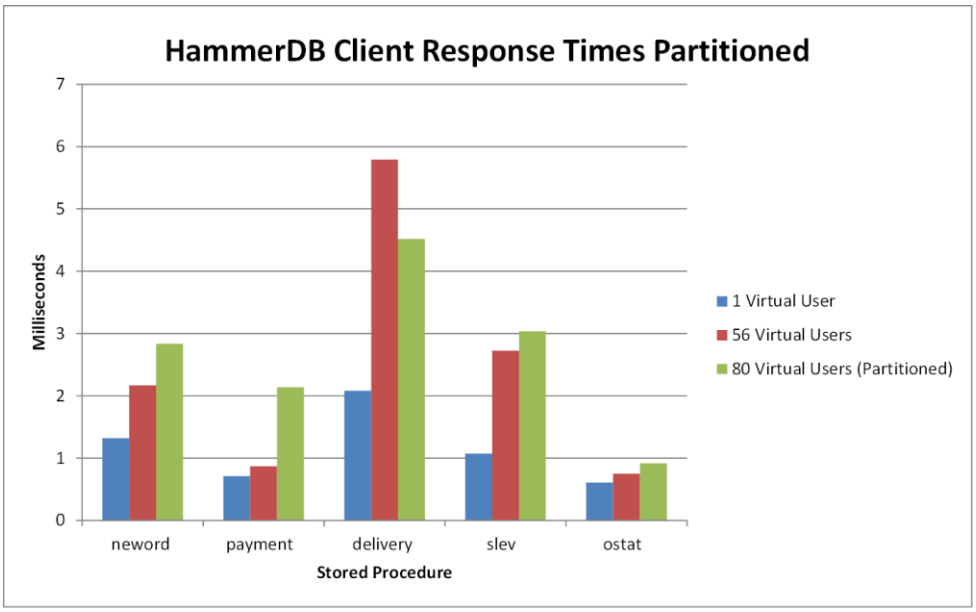

Figure 4 Client Response Times Resolved

Enabling Time Profiling

The following example shows how to enable time profiling for the OTLP workload on the Oracle database, however this procedure is identical for other databases and can also be adapted for other workloads with equal measure. For the OLTP workload as shown in Figure 5 configure the AWR Snapshot Driver Script or Timed Test Driver script for other databases. As detailed previously measuring Time Profiling should be done in conjunction with measuring transaction rates.

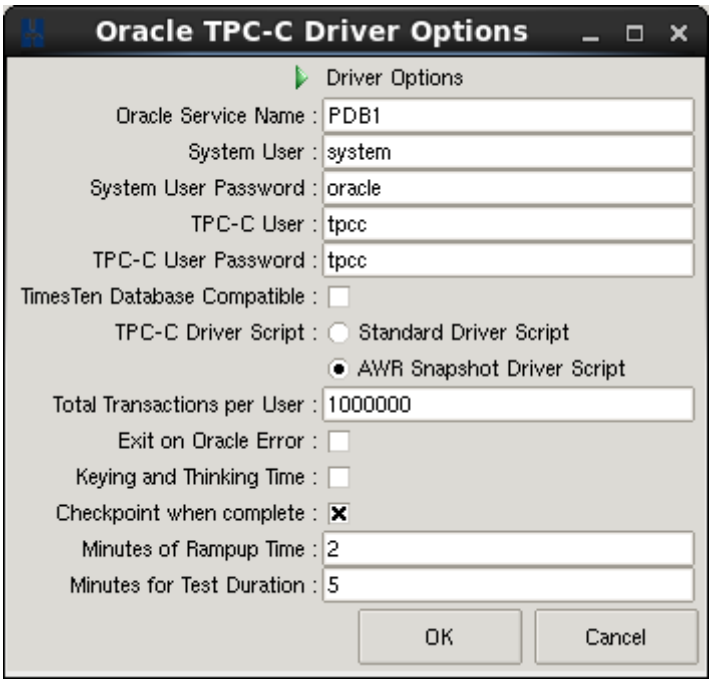

Figure 5 TPC-Driver Options

Load the AWR Snapshot or Timed Test Driver Script as shown in Figure 6. Ensure that you have Script

highlighting turned on (it is enabled by default and can be turned on or off with the last option under the edit menu), you are then ready to modify the script for time profiling.

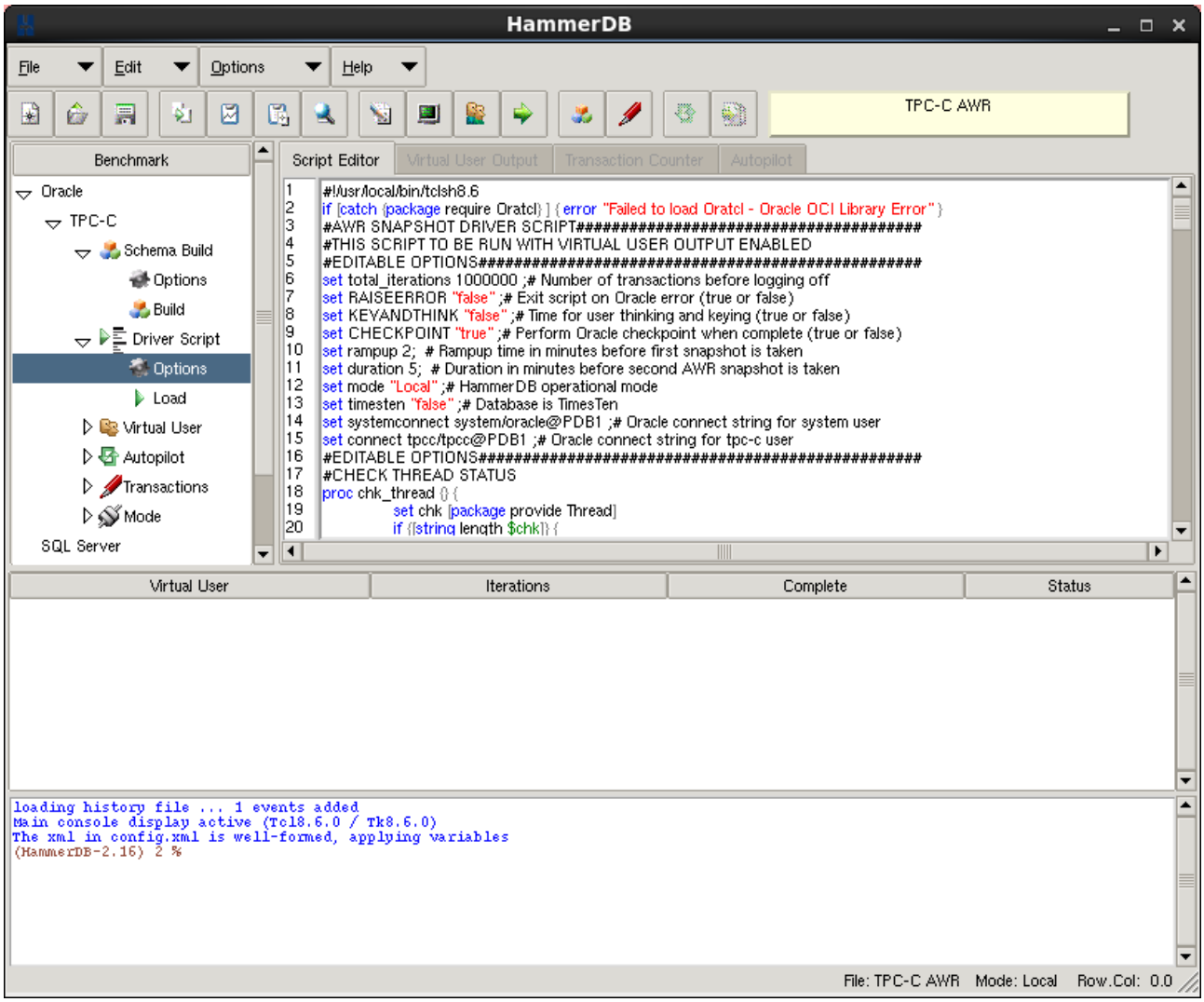

Figure 6 TPC-Driver Options

Time profiling is done using the etprof package that is included with all HammerDB installations. It is important to note that all time profiling has an overhead associated with capturing timing information and therefore the aim is to keep this overhead to a minimum. Consequently best practice is not to profile all virtual users at the same but instead to select one or two virtual users to profile as a sample for the time it takes for response times for an individual user out of a larger number of virtual users. To observe how individual virtual users are identified scroll to line 73.

```
73 switch $myposition {
74 1 {
```

Figure 7 switch statement

You can observe that a switch statement on the variable $myposition is used and virtual user 1 does the timing and taking of snapshots (if this functionality is enabled in the database software). Scroll down to line 181 and you can observe that the default action hence for virtual users running the workload begins here as shown in Figure 8, therefore if we want to enable time profiling for a particular virtual user it should be done here.

```
181 default {
```

Figure 8 default action

To do this directly under the default switch add the following statement to load the etprof package for your chosen virtual user or users as shown in Figure 9.

```
181 default {
182 if { $myposition eq 2 } {
183 package require etprof
184 }
```

Figure 9 Load etprof

Note that the package has been modified from the default open source to initialize on loading and therefore additional initialisation code is not required. Then scroll down to line 414 and add the following statement to print the results.

```
414 ::etprof::printLiveInfo
415 oraclose $curn_no
```

Figure 10 Print results

These are the only steps required to capture time profiling for a virtual user. Configure the virtual user options as shown in Figure 11. Note that more than 1 virtual user needs to be configured as in this example we have selected virtual user 2 to profile. Also it is important to choose “Log Output to Temp” to ensure that you capture the Time Profiling information that is generated.

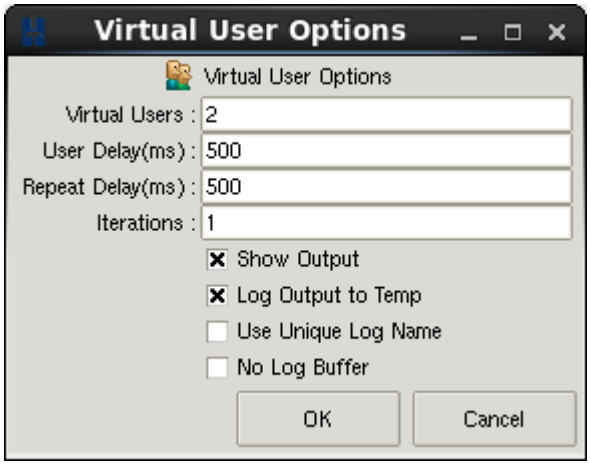

Figure 11 Virtual User Options

Start the test running, you should receive a prompt as shown in Figure 12 to show that logging is active.

Figure 12 Logging Activated

When the test is complete the virtual user profiled will show profiling information in its monitoring output as shown in Figure 13.

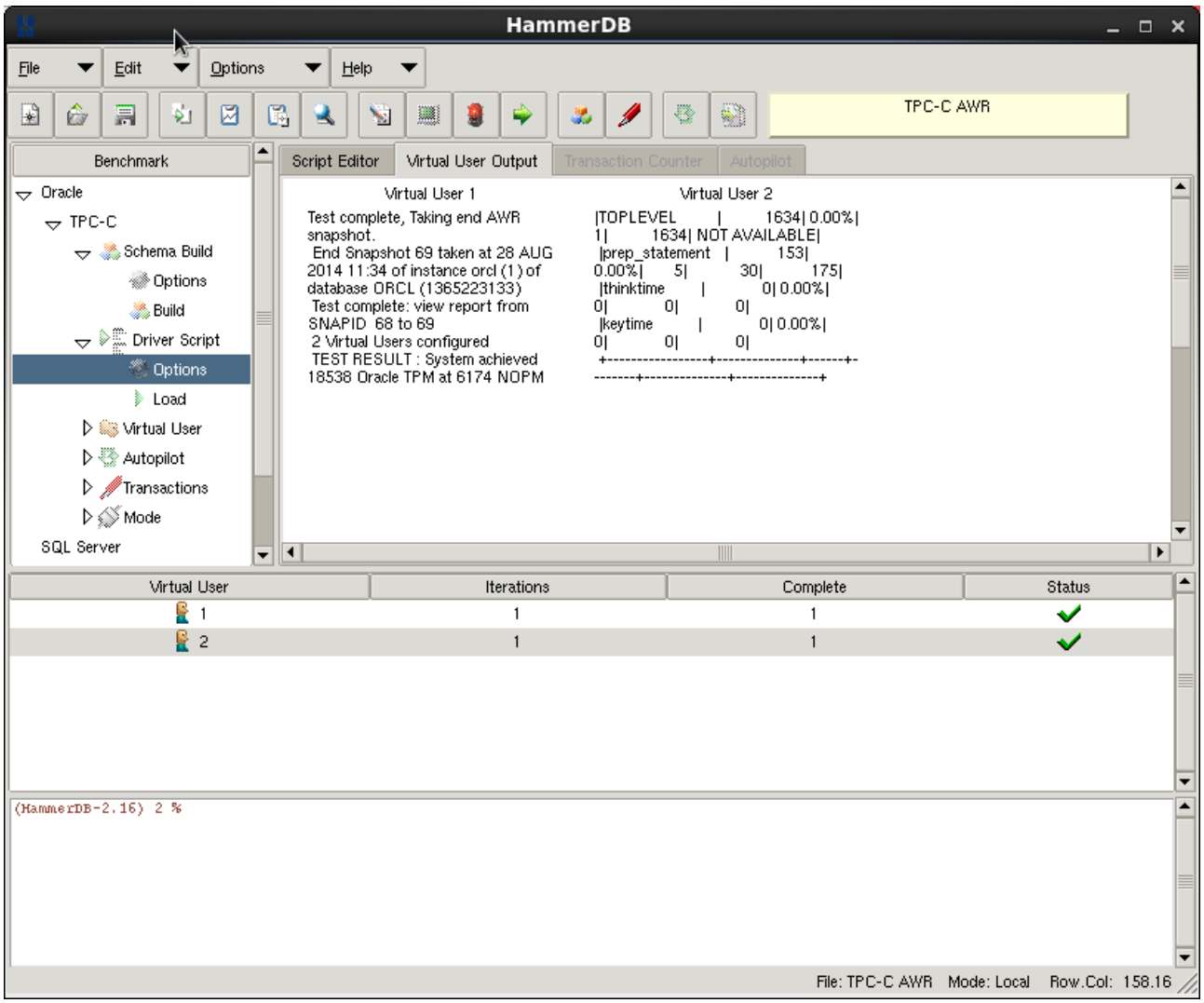

Figure 12 Profiling Complete

You can then retrieve your profiling information from the logfile. The log file will show information such as the number of times a procedure was called and of particular importance the average time per call measured in microseconds as highlighted below.

```
Vuser 2:+----------------+-------------+------+--------+-------------+-------
-------+
Vuser 2:|PROCNAME | EXCLUSIVETOT| %| CALLNUM| AVGPERCALL| CUMULTOT|
Vuser 2:+----------------+-------------+------+--------+-------------+-------
-------+
Vuser 2:|neword | 138544928|50.24%| 137976| 1004| 161920197|
Vuser 2:|payment | 74125353|26.88%| 138004| 537| 114281403|
Vuser 2:|delivery | 20044915| 7.27%| 13580| 1476| 21901271|
Vuser 2:|RandomNumber | 16348208| 5.93%| 1835073| 8| 18644899|
Vuser 2:|slev | 9881174| 3.58%| 13715| 720| 10122934|
Vuser 2:|ostat | 5961937| 2.16%| 13814| 431| 7830964|
Vuser 2:|gettimestamp | 5355590| 1.94%| 289560| 18| 32729388|
Vuser 2:|NURand | 3445515| 1.25%| 151818| 22| 8771516|
Vuser 2:|randname | 2059505| 0.75%| 151818| 13| 2250218|
Vuser 2:|TOPLEVEL | 1764| 0.00%| 1| 1764| NOT AVAILABLE|
Vuser 2:|prep_statement | 247| 0.00%| 5| 49| 253|
Vuser 2:|thinktime | 0| 0.00%| 0| 0| 0|
Vuser 2:|keytime | 0| 0.00%| 0| 0| 0|
Vuser 2:+----------------+-------------+------+--------+-------------+-------
-------+
```

Take this data and use it to generate a comparison at different user loads, typically the microsecond data can be converted to milliseconds for graphing as shown in Figure 13.

1 vuser		
neword	1004	1.004
payment	537	0.537
delivery	1476	1.476
slev	720	0.72
ostat	431	0.431
44 vusers		
neword	1712	1.712
payment	859	0.859
delivery	2866	2.866
slev	2100	2.1
ostat	714	0.714
56 vusers		
neword	2236	2.236
payment	1246	1.246
delivery	3332	3.332
slev	2653	2.653
ostat	925	0.925

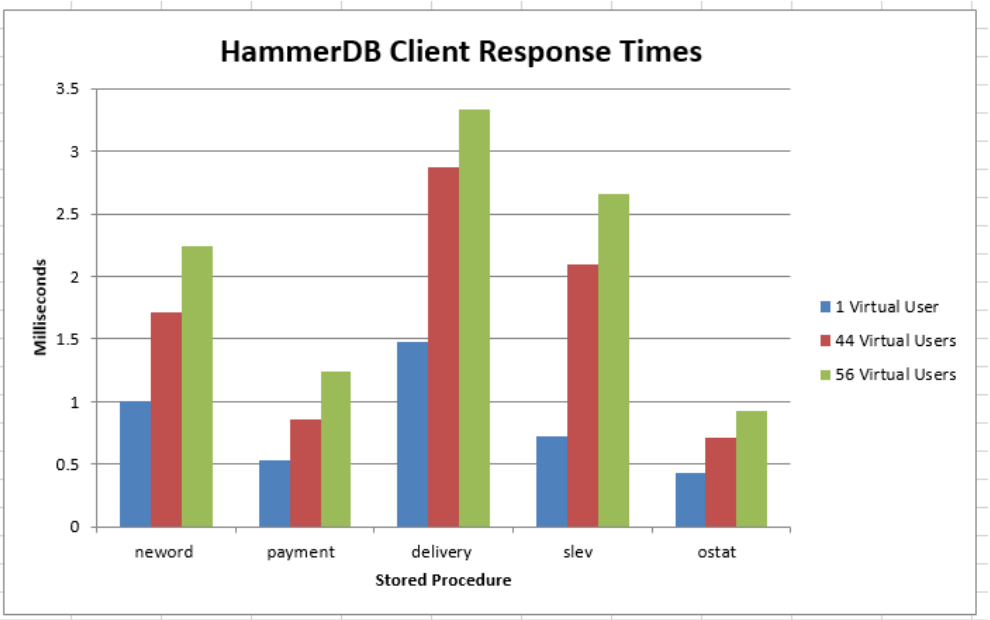

Figure 13 Profiling Comparison

It is important to reiterate that Time Profiling is not limited to the built-in workloads and can be added as described in this document to any workload in order to capture client-side response times.

Support and Questions

For help use the HammerDB Sourceforge forum available at the HammerDB sourceforge project.