Introduction to Transactional (TPC-C) Testing for all Databases

This guide gives you an introduction to transactional database load testing based on the TPC-C specification for all databases supported with HammerDB workloads.

What is a Transactional Workload?

A transactional or OLTP (online transaction processing) workload is a workload typically identified by a database receiving both requests for data and multiple changes to this data from a number of users over time where these modifications are called transactions. Each database transaction has a defined beginning point, manipulates and modifies the data within the database and either commits the changes or rollbacks the changes to the starting point. A database must adhere to the ACID (Atomicity, Consistency, Isolation, Durability) properties to ensure that the database remains consistent whilst processing transactions. Database systems that process transactional workloads are inherently complex in order to manage the user's sessions access to same data at the same time, processing the transactions in isolation whilst keeping the database consistent and recoverable. People will typically interact with OLTP systems on a regular basis with examples such as an online grocery ordering and delivery system or an airline reservation system. Performance and scalability are essential properties of systems designed to process transactional workloads and the TPC-C benchmark is a benchmark designed by the TPC to measure the performance of the software and hardware of a relational database system to process these workloads.

What is the TPC and TPC-C?

Designing and implementing a database benchmark is a significant challenge. Many performance tests and tools experience difficulties in comparing system performance especially in the area of scalability, the ability of a test conducted on a certain system and schema size to be comparable with a test on a larger scale system. When system vendors wish to publish benchmark information about database performance they have long had to access to such sophisticated test specifications to do so and the TPC is the industry body most widely recognized for defining benchmarks in the database industry recognized by all of the leading database vendors. TPC-C is the benchmark published by the TPC* for Online Transaction Processing and you can view the published TPC-C results at the TPC website.

As defined by the TPC *"*TPC benchmarks are industry standards. The TPC, at no charge, distributes its benchmark specifications to the public.*"* For this reason HammerDB includes an implementation of the specification of the TPC-C benchmark that can be run in supported database environments. Implementing the TPC-C specification has the significant advantage that you know that the test is reliable, scalable and tested to produce accurate, repeatable and consistent results.

*The TPC published an additional benchmark specification for transactional systems called TPC-E introduced in February 2007. However unlike TPC-C this benchmark has not been widely adopted by all of the commercial database vendors.

HammerDB Transactional TPC-C based workloads

HammerDB has different transactional workloads for the following databases.

Databases	SQL Server Oracle Oracle TimesTen MySQL PostgresQL EnterpriseDB Postgres Plus Advanced Server (Oracle compatible schema differs from PostgreSQL) Redis

The HammerDB workload is based on and intended to be as close as possible to the example published in the TPC-C specification. As such the implementation is intentionally non-optimized or biased towards any particular database implementation or system hardware (being open source you are free to inspect all of the HammerDB source code). The crucial element is to reiterate the point made in the previous section that the HammerDB workloads are designed to be ***reliable, scalable and tested to produce accurate, repeatable and consistent results***. In other words HammerDB is designed to measure relative as opposed to absolute database performance between systems. What this means is if you run a test against one particular configuration of hardware and software and re-run the same test against exactly the same configuration you will get exactly the same result within the bounds of the random selection of transactions which will typically be within 1%.

Any differences between results are directly as a result of changes you have made to the configuration (or management overhead of your system such as database checkpoints or user/administrator error). Testing has proven that HammerDB tests re-run multiple times unattended (see the autopilot feature) on the same reliable configuration produce performance profiles (see below) that will overlay each other almost identically. Figure 1 illustrates an example of this consistency and shows the actual results of 5 sequences of 18 tests (40 tests in all) run unattended one after another against one of the supported databases with the autopilot feature from 1 to 64 virtual users.

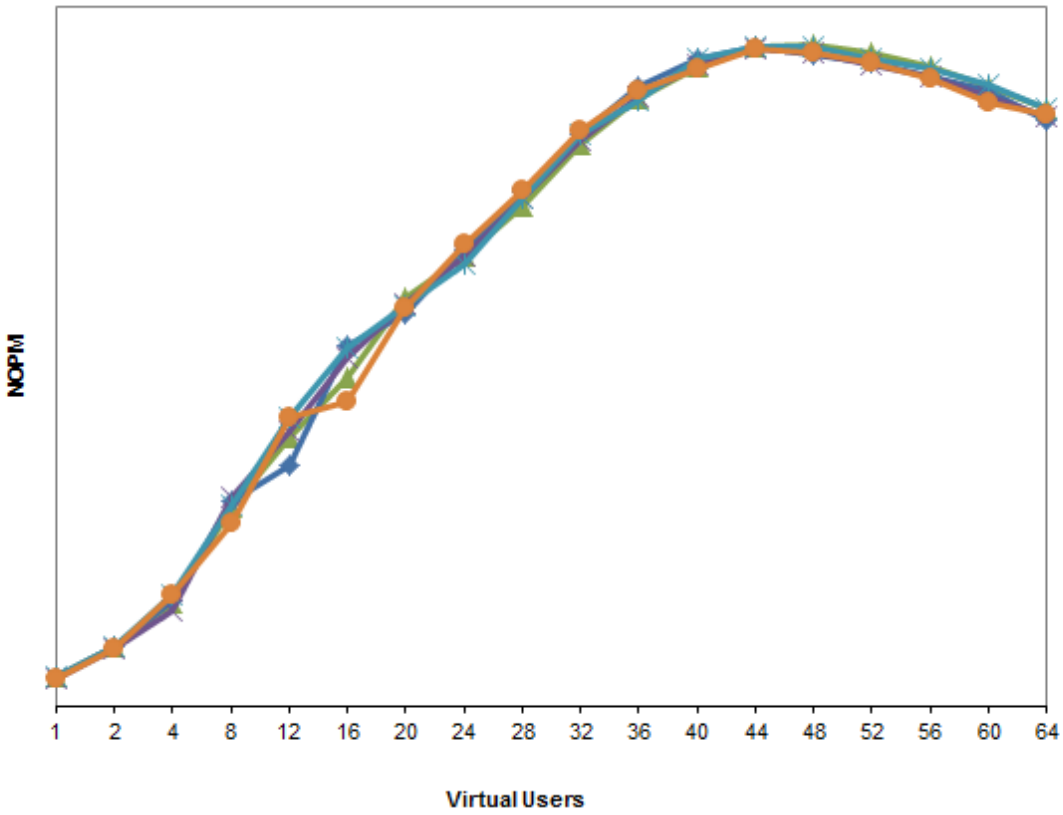

Figure 1 Consistent Results

In other words ***HammerDB will give you the same results each time***, if your results vary you need to focus entirely on your database, OS and hardware configuration. The transaction counter can assist you with this configuration. For example figure 2 illustrates the contrast between optimal and sub-optimal configurations and their impact on performance.

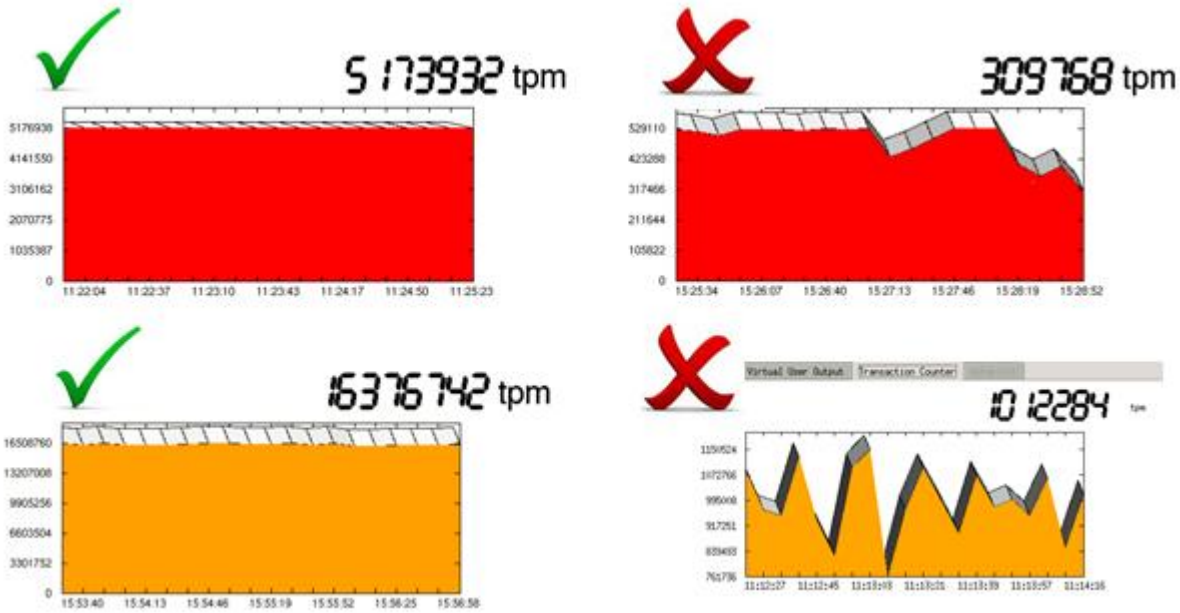

Figure 2 Consistent and Inconsistent Transaction Rates

Consistent performance with HammerDB measured by transaction rates is achievable for all databases irrespective of the number of virtual users, with a higher number of virtual users the transaction rate will be higher but always remain consistent as shown in figure 3.

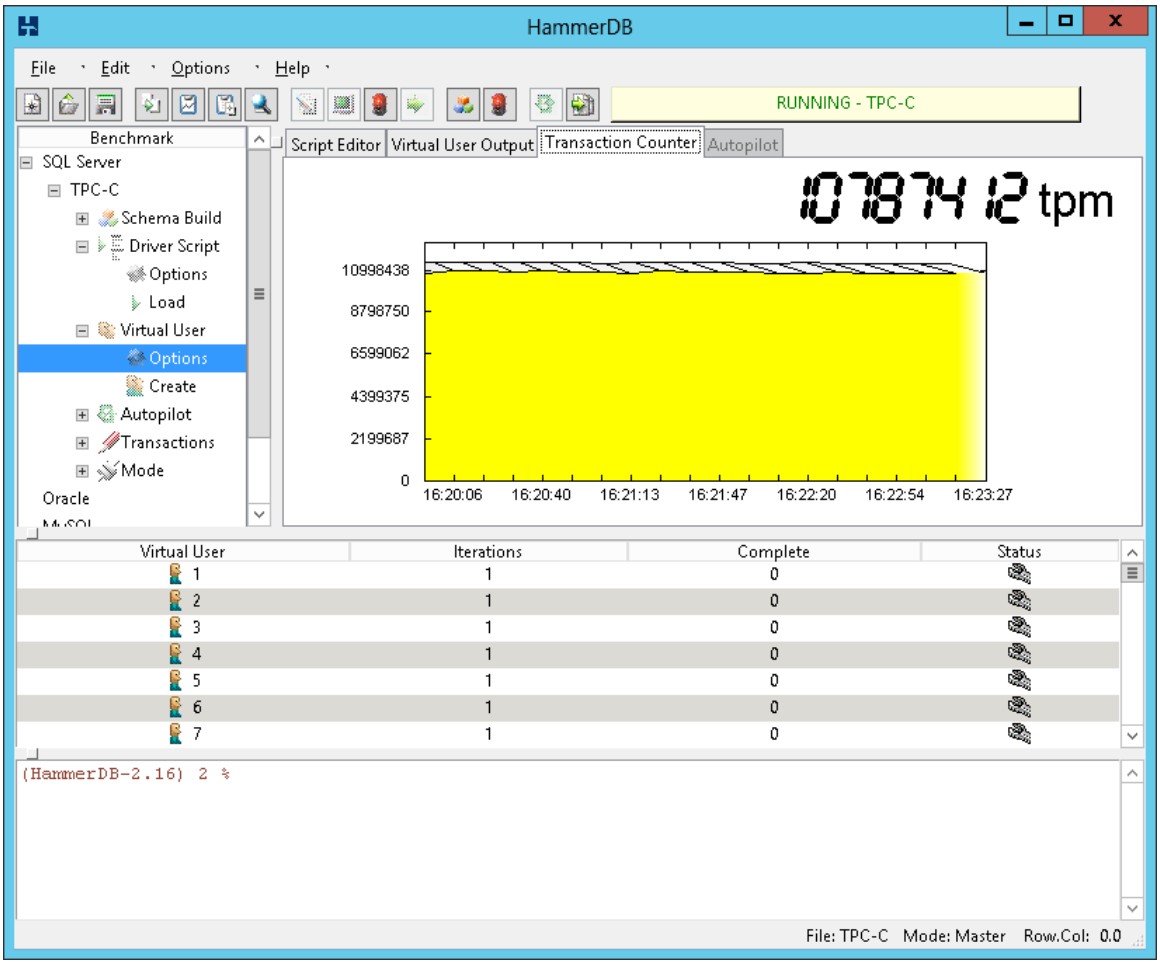

Figure 3 Consistent SQL Server Performance

When you have achieved consistent performance you can use HammerDB metrics to assess whether you are reaching the full capabilities of your system. On both Windows and Linux at full system performance you will observe close to maximum CPU utilisation as shown in figure 4.

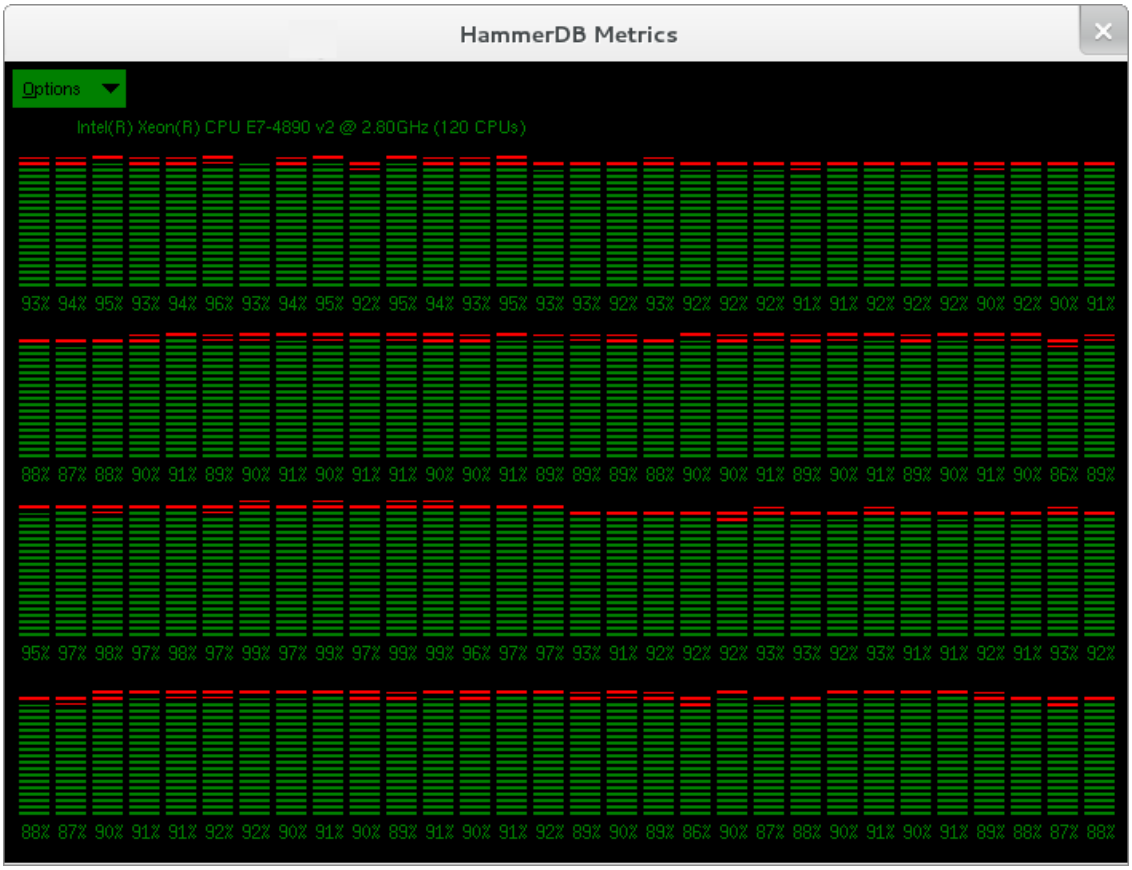

Figure 4 Maximum CPU utilisation

Note however that the converse statement is not necessarily true, maximum CPU utilisation is not necessarily an indication of full system performance. You need to observe the functions utilising the CPU to be assured that they are contributing to the database workload. Nevertheless if your CPU is not fully utilised and increasing the virtual user load does not increase utilisation further then you have a bottleneck or limitation, either in your hardware configuration of memory, I/O or network, database or schema configuration or the database software release itself. You should also note potential limitations on the HammerDB client in the form of CPU utilisation, memory and network bandwidth. The HammerDB architecture enables high levels of scalability up to the most demanding systems.

Once you have maximised your configuration taking a baseline of a database system with a HammerDB workload is an ideal method to ensure that optimal database configurations are engineered put into production.

Comparing HammerDB results

HammerDB implements a workload based on the TPC-C specification however does **NOT** implement a full specification TPC-C benchmark and the transaction results from HammerDB cannot be compared with the official published TPC-C benchmarks in any manner. Official Audited TPC-C benchmarks are extremely costly, time consuming and complex to establish and maintain. The HammerDB implementation based on the specification of the TPC-C benchmark is designed to capture the essence of TPC-C in a form that can be run at low cost on any system bringing professional, reliable and predictable load testing to all database environments. For this reason HammerDB results cannot and should **NOT** be compared or used with the term tpmC in any circumstance. You are permitted however to observe for your own benefit whether a

correlation exists between the ratios of HammerDB results conducted on different systems and officially, audited and published results.

HammerDB workloads produce 2 statistics to compare systems called TPM and NOPM respectively. TPM is the specific database transactional measurement typically defined as the number of user commits plus the number of user rollbacks. Being database specific TPM values cannot be compared between different database types. On the other hand the NOPM value is based on a metric captured from within the test schema itself. As such NOPM (New Orders per minute) is a performance metric independent of any particular database implementation and is the recommended primary metric to use.

Understanding the TPC-C workload

TPC-C implements a computer system to fulfil orders from customers to supply products from a company. The company sells 100,000 items and keeps its stock in warehouses. Each warehouse has 10 sales districts and each district serves 3000 customers. The customers call the company whose operators take the order, each order containing a number of items. Orders are usually satisfied from the local warehouse however a small number of items are not in stock at a particular point in time and are supplied by an alternative warehouse. Figure 5 shows this company structure.

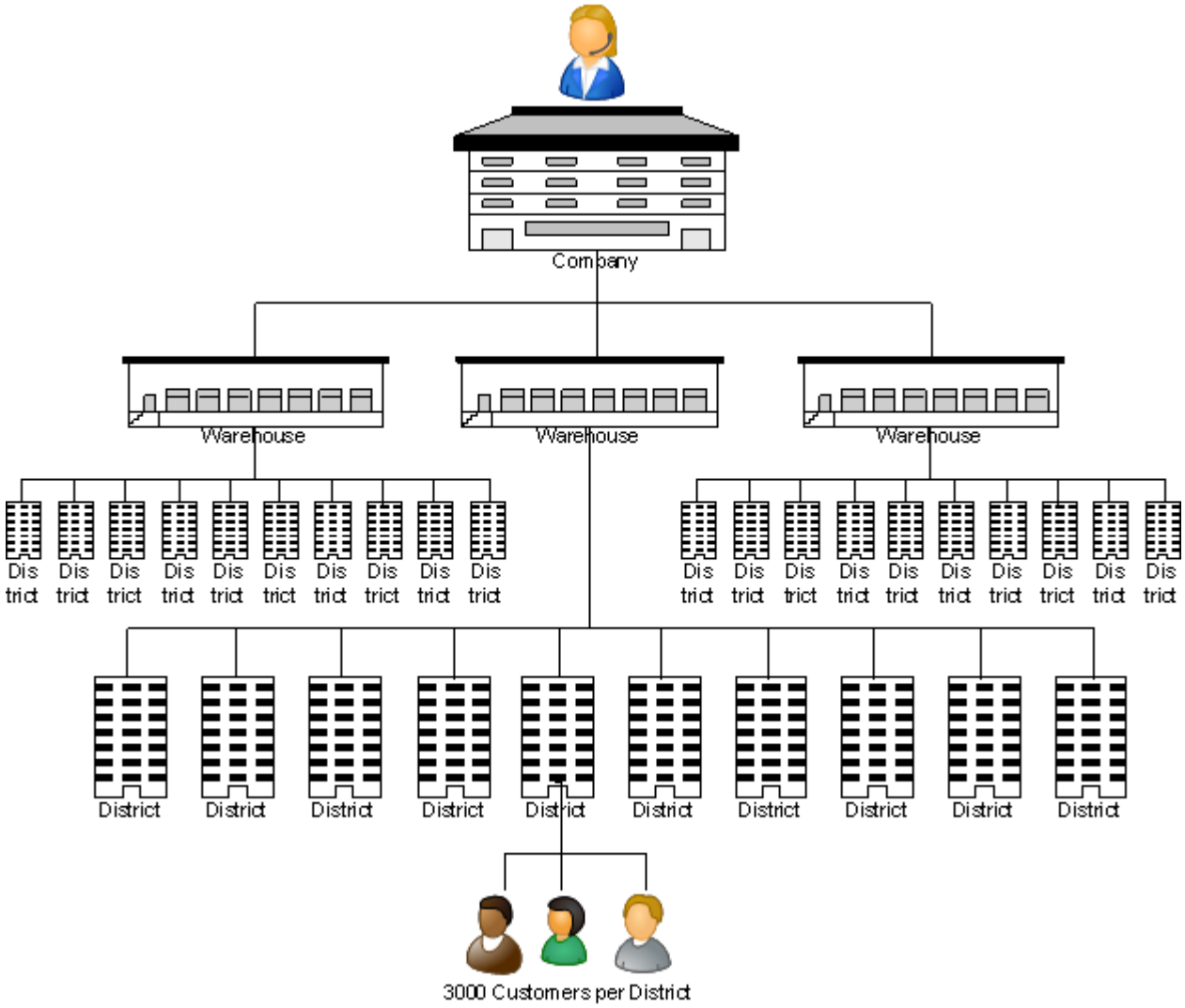

Figure 5 TPC-C Company Structure

It is important to note that the size of the company is not fixed and can add Warehouses and sales districts as the company grows. For this reason your test schema can be as small or large as you wish with a larger schema requiring a more powerful computer system to process the increased level of transactions. Figure 6 shows the TPC-C schema, in particular note how the number of rows in all of the tables apart from the ITEM table which is fixed is dependent upon the number of warehouses you choose to create your schema.

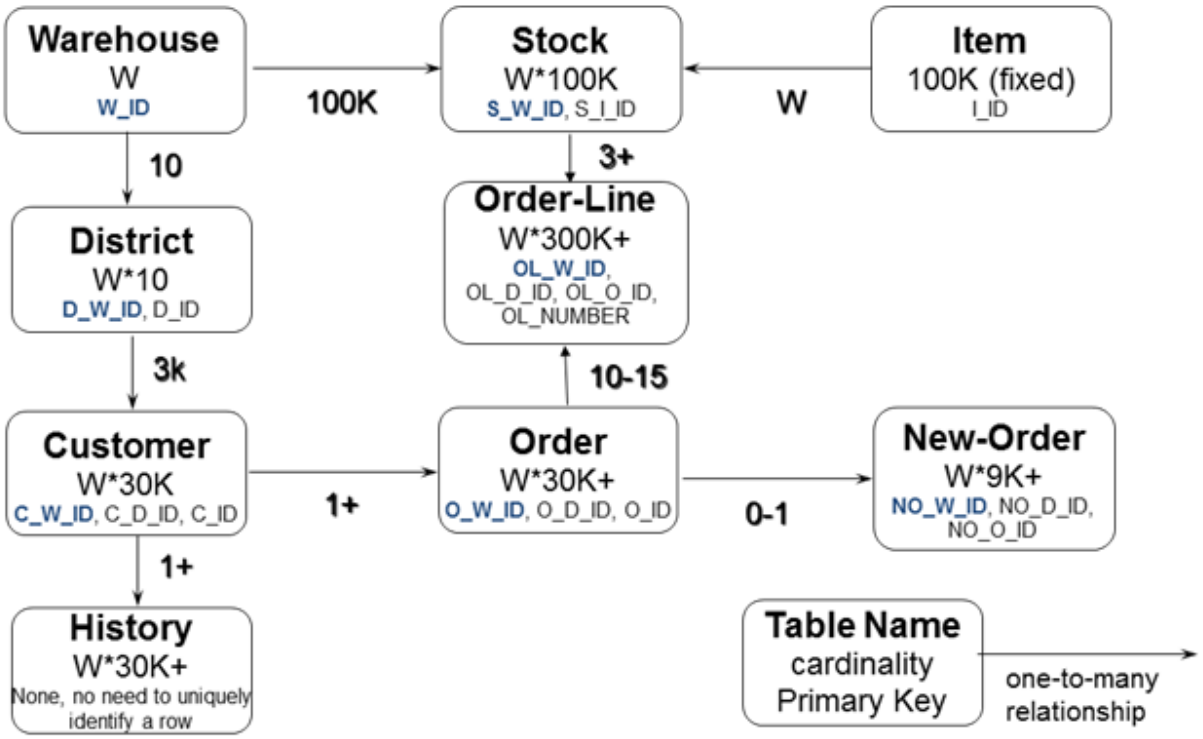

Figure 6 TPC-C Schema

For additional clarity please note that the term Warehouse in the context of TPC-C bears no relation to a Data Warehousing workload, as you have seen TPC-C defines a transactional based system and not a decision support (DSS) one.

In addition to the computer system being used to place orders it also enables payment and delivery of orders and the ability to query the stock levels of warehouses. Consequently the workload is defined by a mix of 5 transactions selected at random according to the balance of the percentage value shown as follows:

- **New-order**: receive a new order from a customer: 45%
- **Payment**: update the customers balance to record a payment: 43%
- **Delivery**: deliver orders asynchronously: 4%
- **Order-status**: retrieve the status of customer's most recent order: 4%
- **Stock-level**: return the status of the warehouse's inventory: 4%

Generating Performance Profiles

For an official audited TPC-C benchmark the result of the tests is detailed as tpmC which represents the number of New Orders processed. This measurement is published as a single data point representing peak performance. A particular advantage of HammerDB is is the ability to generate a performance profile as the load increases on your system (see the Autopilot feature for doing this in an unattended manner). Therefore

whereas an official TPC-C benchmark gives you a single data-point and a typical single-threaded test, consider the graph shown in figure 7.

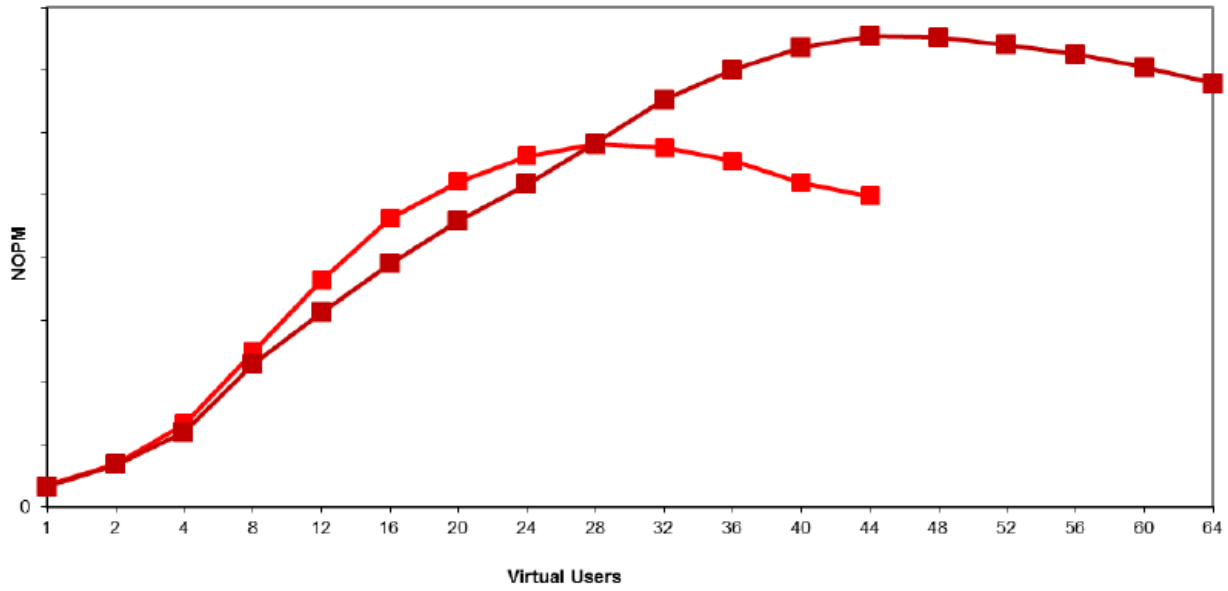

Figure 7 Performance Profile Example

This graph shows the relative performance of real tests on different database configurations, this example in fact shows an identical database and system configuration whilst modifying a processor multithreading feature. Observed as a single data point disabling this feature illustrates a marginal performance gains for a lower number of virtual users, however the performance profile clearly shows a crossover point and for a more scalable workload enabling this feature provides benefits for a larger number of virtual users. Your chosen configuration would then be determined by the level of system utilisation expected and the scalability required. It should be clear that your testing goal for transactional systems should be to measure the performance profile of your system across all levels of utilisation starting with 1 virtual user through to peak utilisation.

Generating Time Profiles

In addition to performance profiles based on throughput you should also take note of transaction response times. Whereas performance profiles show the cumulative performance of all of the virtual users running on the system, response times show performance based on the experience of the individual user. When comparing systems both throughput and response time are important comparative measurements. HammerDB includes a time profiling package called *etprof* that enables you to select an individual user and measure the response times for the transactions that the user processes as shown in Table 1.

```
Vuser 2:+-----------------+-------------+------+--------+-------------+-------------+
Vuser 2:|PROCNAME      | EXCLUSIVETOT | %    | CALLNUM| AVGPERCALL | CUMULTOT     |
Vuser 2:+-----------------+-------------+------+--------+-------------+-------------+
Vuser 2:|neword        |152375421     |46.14%| 53760  | 2834        | 167684090    |
Vuser 2:|payment       |114514026     |34.68%| 53541  | 2138        | 140651925    |
Vuser 2:|delivery      | 23805429     | 7.21%| 5268   | 4518        | 25027780     |
Vuser 2:|slev          | 16299664     | 4.94%| 5368   | 3036        | 16464540     |
Vuser 2:|RandomNumber | 10992363     | 3.33%| 712992| 15          | 12433866     |
Vuser 2:|ostat         | 4896241      | 1.48%| 5336   | 917         | 6121910      |
Vuser 2:|gettimestamp | 3609806      | 1.09%| 112569 | 32          | 21402133     |
Vuser 2:|NURand        | 2320574      | 0.70%| 58877  | 39          | 5753442      |
Vuser 2:|randname      | 1402532      | 0.42%| 58877  | 23          | 1517314      |
Vuser 2:|TOPLEVEL      | 1852         | 0.00%| 1      | 1852        | NOT AVAILABLE|
Vuser 2:|prep_statemen| 228          | 0.00%| 5      | 45          | 235          |
Vuser 2:|thinktime     | 0            | 0.00%| 0      | 0           | 0            |
Vuser 2:|keytime       | 0            | 0.00%| 0      | 0           | 0            |
Vuser 2:+-----------------+-------------+------+--------+-------------+-------------+
```

Table 1 Time Profile Example

The output from *etprof* taken from each system should be used in context with the overall performance profile to break down the overall system throughput to the timing of the individual transactions themselves.

Publishing database performance results

Commercial software typically includes a clause (commonly known as a DeWitt clause) in the license agreement that prohibits the user from publishing program benchmarking results that have not been approved by the software vendor. This clause may exist in your database software license and/or your operating system and virtualization software system license. Typically this clause does not exist in the open source databases supported by HammerDB, namely MySQL, PostgreSQL and Redis. Consequently as a general guideline you are free to publish performance data generated by HammerDB against the open source databases in any domain, for the commercial databases you are recommended to consult the legal guidelines in your region. HammerDB is released under the GPL license and cannot accept responsibility for published data or offer legal advice on doing so.

Support and Questions

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