PERCONA LIVE ONLINE MAY 12 - 13th 2021

HammerDB

A Better Way to Benchmark Your Open Source Database

Hello!

Steve Shaw

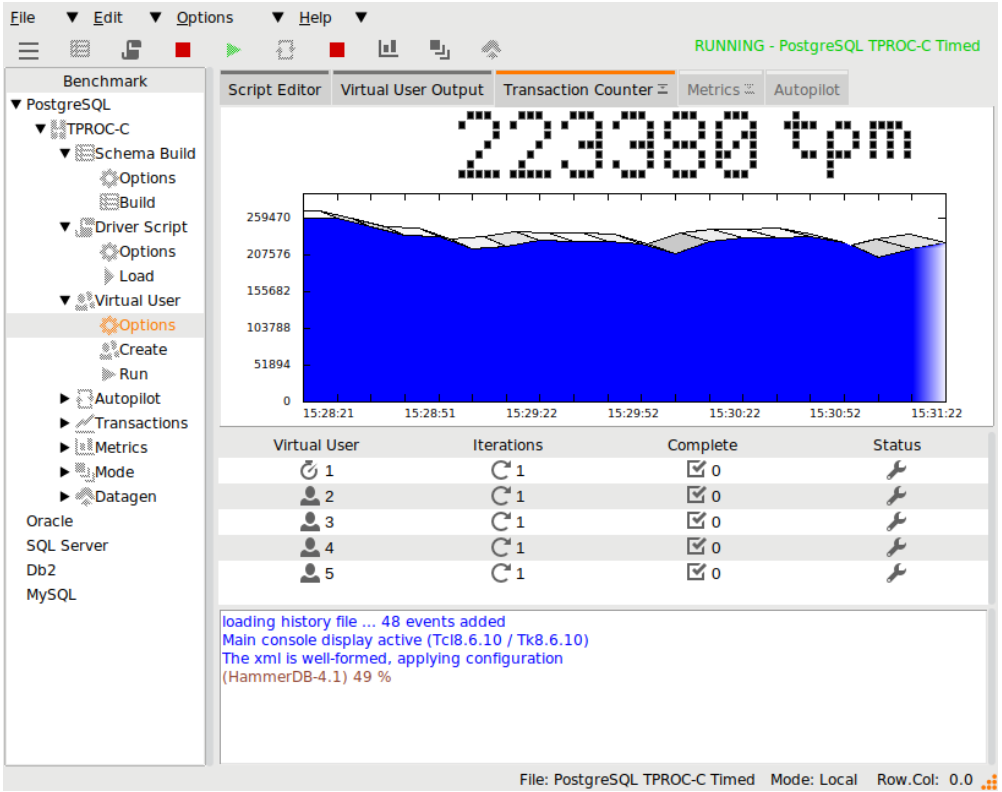
Open source database @ Intel

Main developer of HammerDB

Introduction

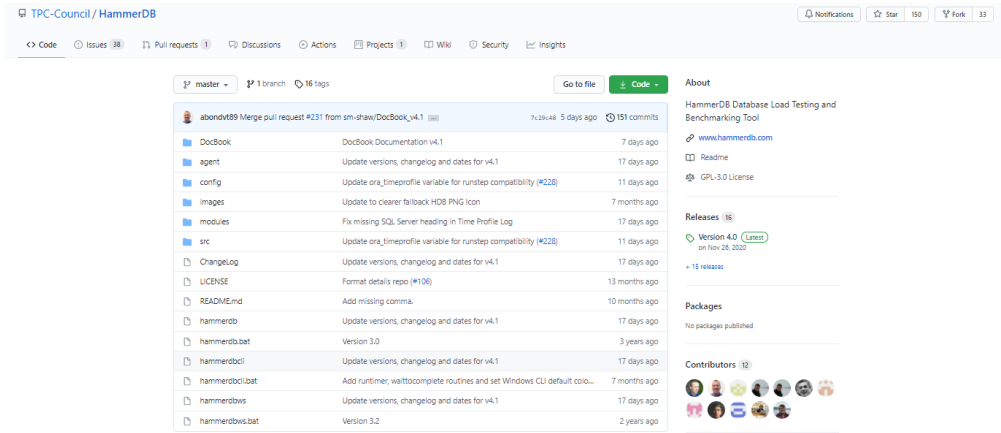
What is HammerDB?

- Not a database!
- Leading open source tool for benchmarking relational databases
- Interfaces
  - Graphical
  - Command Line
  - Web REST interfaces
- Industry standard benchmarks
- High performance and scalability



Open Source

- Hosted by TPC Council since 2019
  - Industry standard body for database benchmarks
- TPC-OSS subcommittee
  - Oversees and approves changes
- v4.1 Released on 22nd April 21
- Source code on GitHub
- Binaries @ GitHub Releases
  - https://www.hammerdb.com/download.html
- Client natively supports Linux and Windows on x64
  - GUI & CLI on both Linux and Windows
- GitHub Release Downloads @
  - https://www.hammerdb.com/stats.html
- Test databases on any platform



Supported Databases

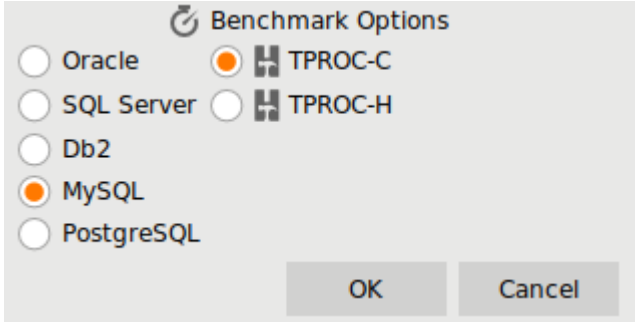
- HammerDB supports the most popular relational databases
- Commercial and open source
- Metrics enable comparison across database engines

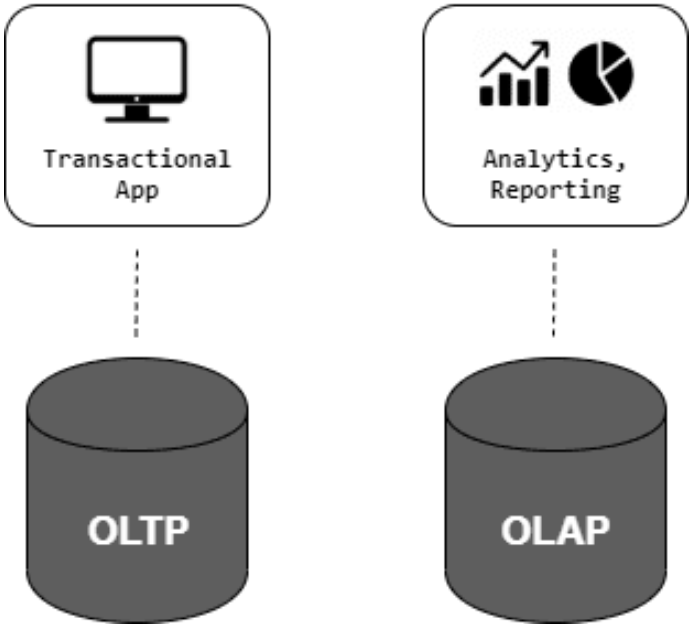
370 systems in ranking, April 2021

Rank			DBMS	Database Model	Score		
Apr 2021	Mar 2021	Apr 2020			Apr 2021	Mar 2021	Apr 2020
1.	1.	1.	Oracle	Relational, Multi-model	1274.92	-46.82	-70.51
2.	2.	2.	MySQL	Relational, Multi-model	1220.69	-34.14	-47.66
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	1007.97	-7.33	-75.46
4.	4.	4.	PostgreSQL	Relational, Multi-model	553.52	+4.23	+43.66
5.	5.	5.	MongoDB	Document, Multi-model	469.97	+7.58	+31.54
6.	6.	6.	IBM Db2	Relational, Multi-model	157.78	+1.77	-7.85

Supported Workloads

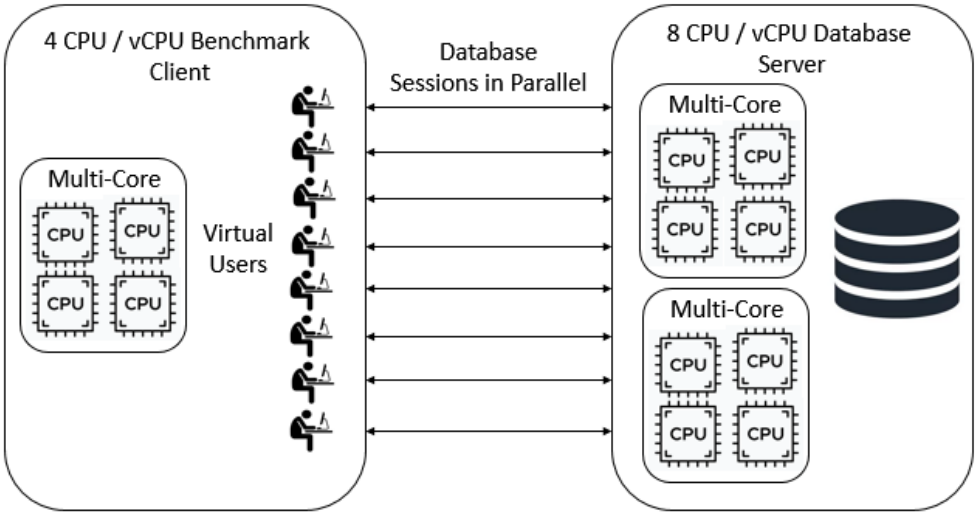
- TPROC-C = OLTP
  - Transactional workloads. Row oriented, high read and write throughput.
  - Derived from TPC-C
- TPROC-H = OLAP
  - Analytic, Decision Support
  - Focus on ETL
  - high bandwidth reads & minimal writes.
  - Derived from TPC-H
- Using TPCC/TPC-C, TPCH/TPC-H for derived workloads not permitted (trademark violation)





Key Database Benchmarking Concepts

- Parallel benchmarking software
  - Concurrency control must be in database, not in client
- Complex workloads designed to scale and test RDBMS concepts
  - Locking and latching
- Cross reference workloads across multiple database engines
  - Validate concepts
- HammerDB up to 6-7M NOPM on commercial database engines on 2 socket servers
  - High confidence levels that bottlenecks are in database software not HammerDB



HammerDB Programming Languages

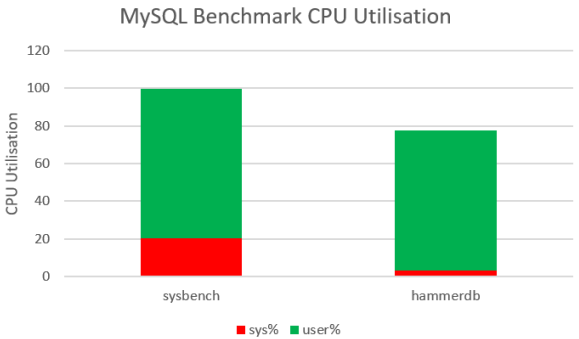
- Designed for High Performance and Scalability
- Database commands in SQL
- Application logic in stored procedures
- Database Interfaces in C
- More time in the database, less time in the ‘round trip’
- More system resources for the database, less resources for the benchmark client
  - 3% for HammerDB
  - 20% in sysbench in socket/network layer

Database	Programming Interface
Oracle	OCI
SQL Server	ODBC
Db2	CLI
PostgreSQL	Libpq
MySQL	MySQL Native Driver

HammerDB Database Interfaces

Database	Application Logic
Oracle	PL/SQL
SQL Server	T-SQL
Db2	SQL PL
PostgreSQL	PL/pgSQL
MySQL	stored program language

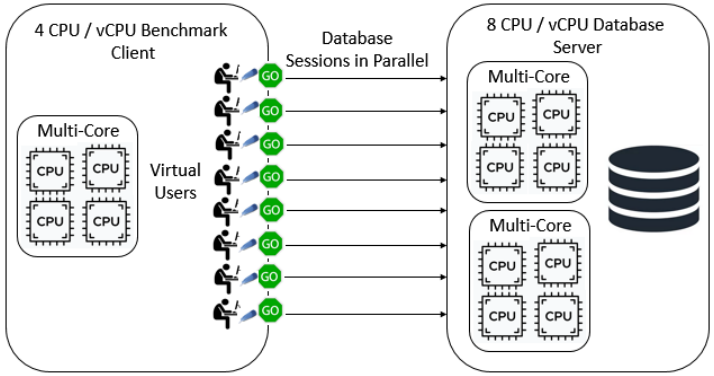
HammerDB Stored Procedures.



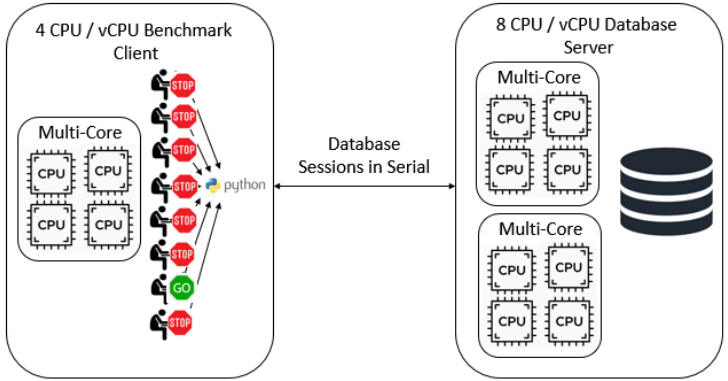
Tcl as Glue Language

- Python GIL limits to single-threading
- Tcl as glue language for truly parallel multithreading
- Tcl compiles into bytecode at runtime for high performance
- Co-routines used for event-driven scaling only to prevent bottleneck
- GUI & CLI on same codebase
- Native Tk GUI including 4k UHD scaling

Tcl Multithreading

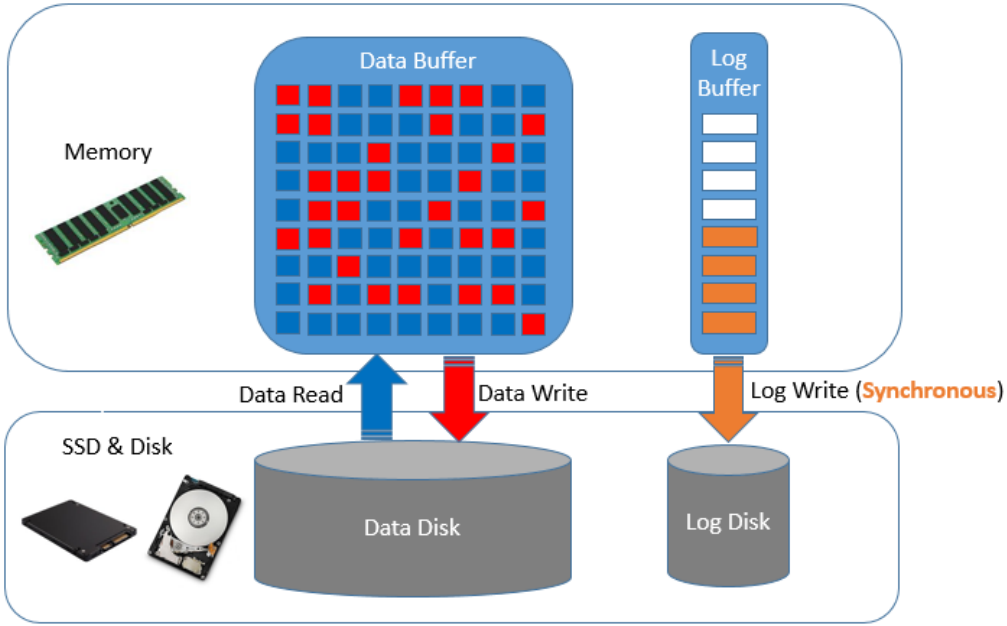


Python Multithreading with GIL



Cached vs Scaled Benchmarks

- Default Mode Cached workload
  - Testing goals
- Cached
  - Less than 5000 warehouses
  - All data in memory
  - 10s to hundreds of sessions
  - CPU & Memory Intensive
  - WAL & Redo Disk Write Intensive
  - Maximum performance at minimal configuration
- Scaled (Event driven scaling)
  - More than 5000 warehouses
  - Event driven scaling
  - Thousands of sessions
  - Middleware needed
  - Larger disk and networking requirement
  - Data Disk Read and Write Intensive
- Perfectly Scaled Configuration = Cached performance



System Configuration

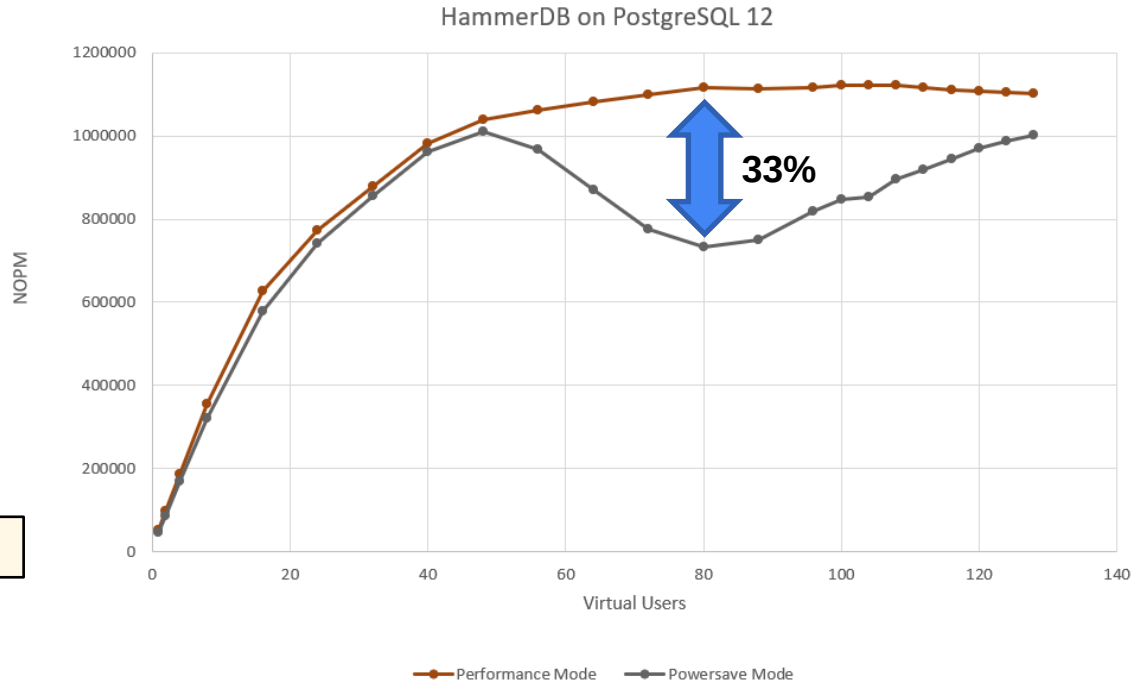
CPU

- Some Linux releases default to CPU powersave mode
  - 33% lower performance

sudo vi /etc/default/cpufrequtils
GOVERNOR="performance“

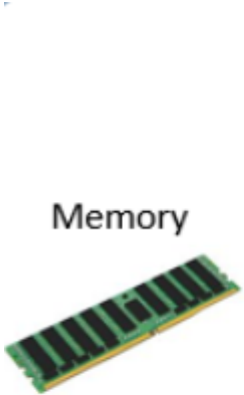
systemctl restart cpufrequtils
systemctl disable ondemand

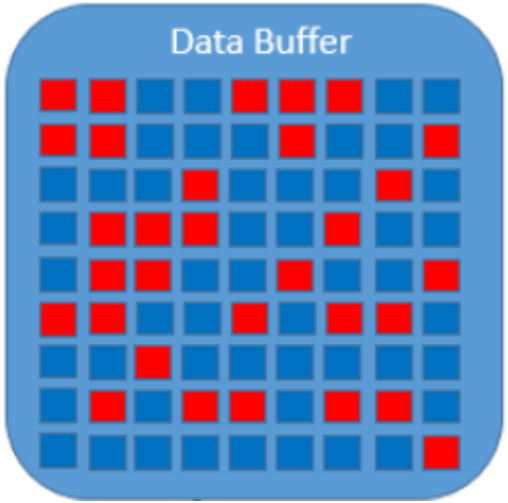
sudo ./cpupower frequency-info
analyzing CPU 0:
  driver: **intel_pstate**
…
  available cpufreq governors: performance powersave
  current policy: frequency should be within 800 MHz an 3.40 GHz.
  The governor "**performance**" may decide which speed to use
  within this range.
  current CPU frequency: Unable to call hardware
  current CPU frequency: 1.03 GHz (asserted by call to kern
  boost state support:
  Supported: yes
  Active: yes



Memory

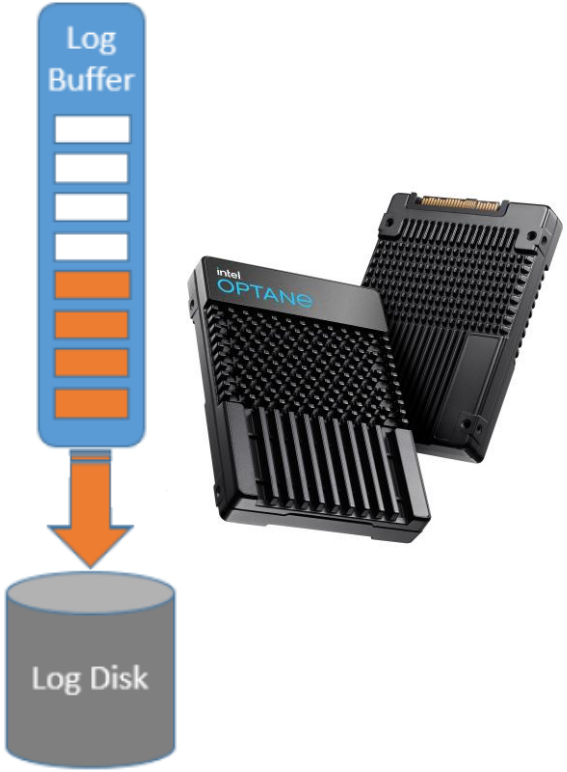
- For default cached workload
- Size Buffer Pool / Cache large enough to cache the TPROC-C schema
- Use 1GB Huge Pages for PostgreSQL
- MySQL/MariaDB InnoDB
  - innodb_buffer_pool_size=64000M
- PostgreSQL
  - shared_buffers = 64000MB
  - huge_pages = on





I/O WAL and Redo Performance

- Use Highest Performance SSDs for WAL/Redo
  - Intel Optane
  - Low latency writes
- Ensure partitions correctly aligned
- Use 1GB redo log / WAL segment size
- MySQL
  - innodb_log_file_size=1024M
  - innodb_log_files_in_group=32
- PostgreSQL
  - initdb -D ./data --wal-segsize=1024
- Synchronous Commit
  - What components are you testing?
- MySQL
  - innodb_flush_log_at_trx_commit=0/1
- PostgreSQL
  - wal_level = minimal/replica
  - synchronous_commit = off/on



Loading Database Client Libraries

- HammerDB needs access to client libraries to load interface
- CLI librarycheck command

```
hammerdb>librarycheck
Checking database library for MySQL
Success ... loaded library mysqltcl for MySQL
Checking database library for PostgreSQL
Success ... loaded library Pgtcl for PostgreSQL
```

- Export LIBRARY_PATH

```
export LD_LIBRARY_PATH=/opt/postgresql-13.2/pgsql/lib/:$LD_LIBRARY_PATH
```

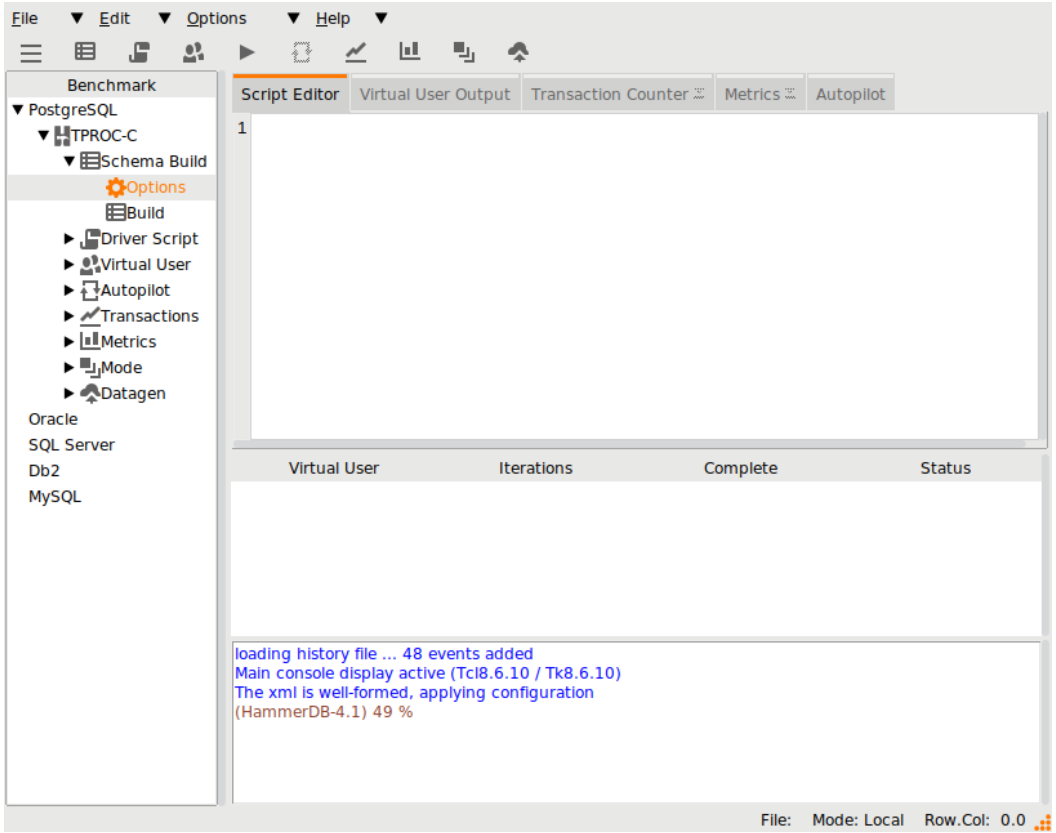
- Use ldd on HammerDB interfaces to verify library used
  - Up to v4.1 for MariaDB you need the MySQL client, from v4.2 MariaDB clients used

```
ldd libmysqltcl3.052.so
libmysqlclient.so.21 => /usr/lib/x86_64-linux-gnu/libmysqlclient.so.21
libpgtcl2.1.1.so
libpq.so.5 => /opt/postgresql-13.2/pgsql/lib/libpq.so.5 (0x00007f0e20ce5000)
```

Schema Build

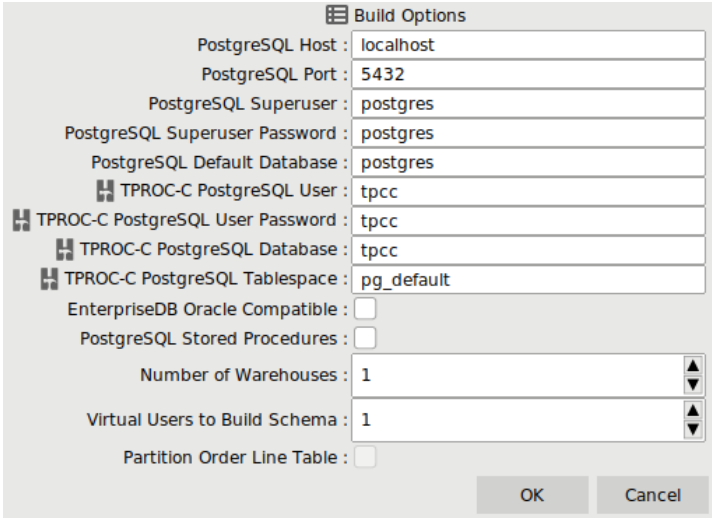
Schema Build

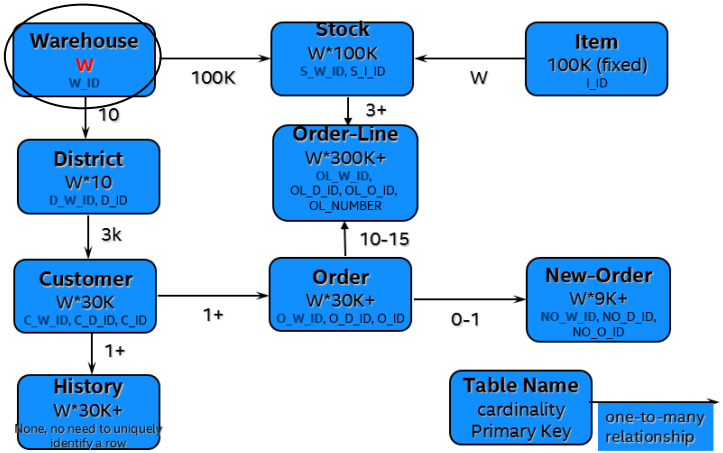
- Schema Build Options
  - Select options from menu
  - Configure with CLI commands
  - Same schema is built
- Options vary per database
  - MySQL Storage Engines
  - Partitioning
  - PostgreSQL stored procedures or functions
- Key Factors in Build Performance
  - CPU cores in client
  - I/O throughput on database



Schema Build Choices

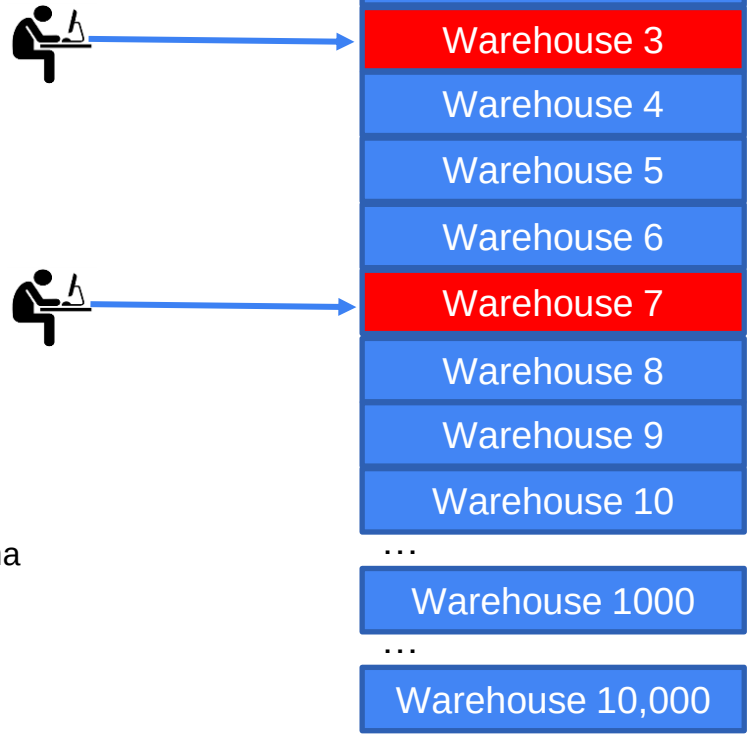
- Schema Build
  - Creates tables
  - Creates and loads data
  - Creates Indexes
  - Creates functions/stored procedures
  - Gathers statistics
- Number of Warehouses
  - Define according to system scale
  - Entire schema scaled based on warehouse count
- Stored Procedures
  - New Order
  - Payment
  - Delivery
  - Stock Level
  - Order Status
- Virtual Users to Build Schema
  - Schema creates and loads data in parallel
  - Use number of CPU cores/threads on HammerDB client





How many warehouses?

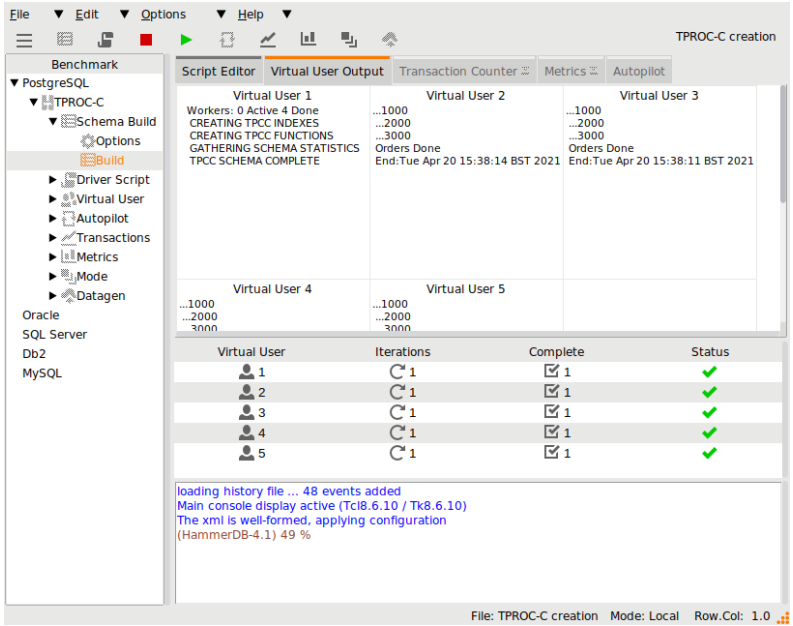
- Default Configuration
- Virtual Users chooses a home warehouse at random
- 90% of the workload satisfied from the home warehouse
  - Regardless of the number configured
  - Hot and cold data
- Configure enough warehouses to ensure an even spread of Virtual Users (eg 4X expected VU count)
- Overprovisioning warehouses will not increase performance or scalability
- Example on 2 socket 1000 warehouses
  - Takes 8-9 minutes to load
  - Depends on CPU and Disk and Virtual Users to build schema
- Warehouse Count Limits
  - 5000 warehouses in GUI
  - 30,000 in Datagen
  - No actual limit for advanced users, only interface limits



Running the Build

- Build Schema Command
  - GUI Build Option
  - CLI buildschema command

GUI



CLI Script

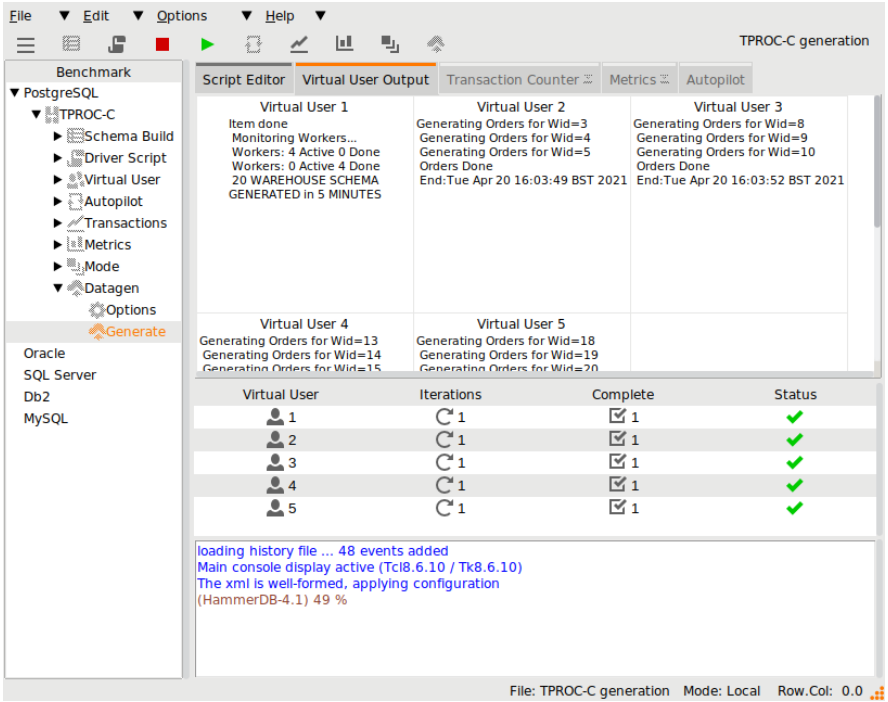
```
dbset db pg
dbset bm TPC-C
diset tpcc pg_count_ware 20
diset tpcc pg_num_vu 4
diset tpcc pg_superuser steve
diset tpcc pg_superuserpass postgres
buildschema
```

Run script

```
~/HammerDB-4.1$ ./hammerdbcli
HammerDB CLI v4.1
Copyright (C) 2003-2021 Steve Shaw
Type "help" for a list of commands
The xml is well-formed, applying configuration
hammerdb>source pgbuild.tcl
Database set to PostgreSQL
Benchmark set to TPC-C for PostgreSQL
Changed tpcc:pg_count_ware from 1 to 20 for PostgreSQL
Changed tpcc:pg_num_vu from 1 to 4 for PostgreSQL
Changed tpcc:pg_superuser from postgres to steve for PostgreSQL
Value postgres for tpcc:pg_superuserpass is the same as existing value postgres, no change made
Script cleared
Building 20 Warehouses with 5 Virtual Users, 4 active + 1 Monitor VU(dict value pg_num_vu is set to 4)
Ready to create a 20 Warehouse PostgreSQL TPROC-C schema
in host LOCALHOST:5432 under user TPCC in database TPCC?
Enter yes or no: replied yes
Vuser 1 created - WAIT IDLE
Vuser 2 created - WAIT IDLE
Vuser 3 created - WAIT IDLE
Vuser 4 created - WAIT IDLE
Vuser 5 created - WAIT IDLE
Vuser 1:RUNNING
Vuser 1:Monitor Thread
Vuser 1:CREATING TPCC SCHEMA
Vuser 1:CREATING DATABASE tpcc under OWNER tpcc
Vuser 2:RUNNING
Vuser 2:Worker Thread
Vuser 2:Waiting for Monitor Thread...
Vuser 1:CREATING TPCC TABLES
Vuser 1:Loading Item
Vuser 3:RUNNING
Vuser 3:Worker Thread
Vuser 3:Waiting for Monitor Thread...
Vuser 3:Loading 5 Warehouses start:6 end:10
Vuser 3:Start:Tue Apr 20 15:41:38 BST 2021
Vuser 3:Loading Warehouse
Vuser 3:Loading Stock Wid=6
Vuser 4:RUNNING
```

Datagen

- GUI or CLI
  - Bulk Loads to bypass database logging and network overhead



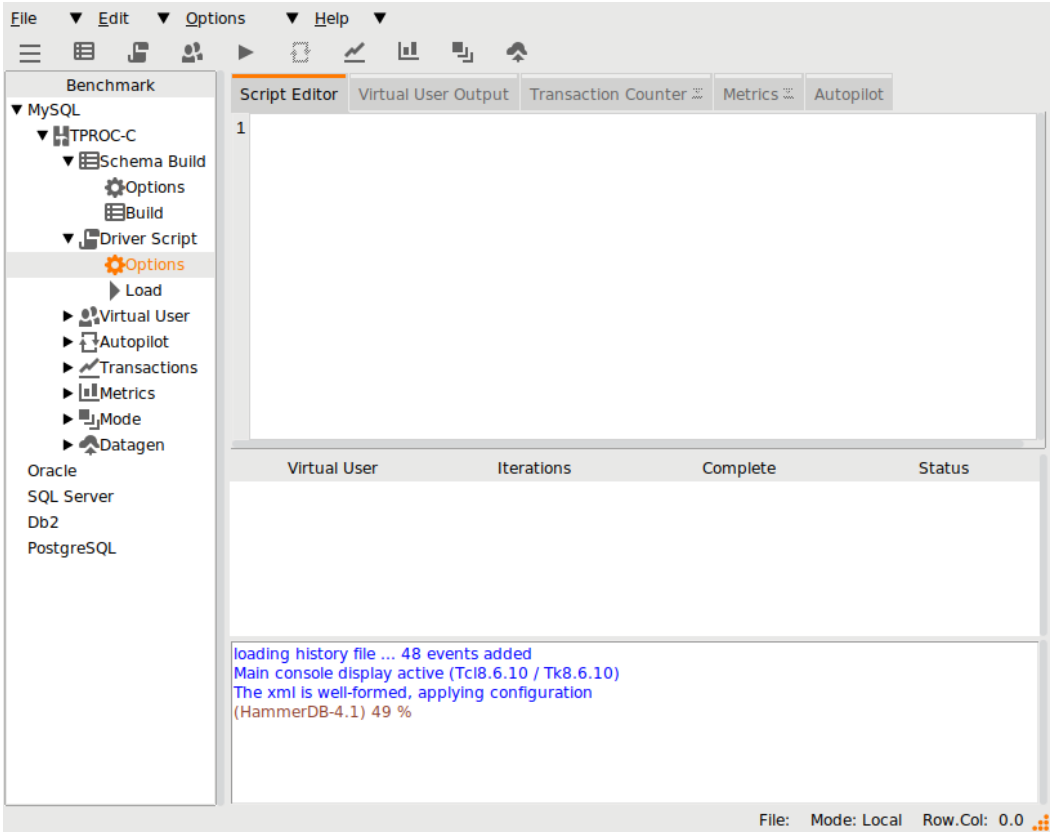
Schema Data in text files

```
84868629 Apr 20 16:02 customer_1.tbl
84876579 Apr 20 16:02 customer_2.tbl
85005196 Apr 20 16:02 customer_3.tbl
84998551 Apr 20 16:02 customer_4.tbl
    4557 Apr 20 16:01 district_1.tbl
    4569 Apr 20 16:01 district_2.tbl
    4606 Apr 20 16:01 district_3.tbl
    4541 Apr 20 16:01 district_4.tbl
 8523405 Apr 20 16:02 history_1.tbl
 8585538 Apr 20 16:02 history_2.tbl
 8823033 Apr 20 16:02 history_3.tbl
 8825046 Apr 20 16:02 history_4.tbl
 7556823 Apr 20 15:58 item_1.tbl
  409500 Apr 20 16:03 new_order_1.tbl
  418500 Apr 20 16:03 new_order_2.tbl
  454500 Apr 20 16:03 new_order_3.tbl
  454500 Apr 20 16:03 new_order_4.tbl
98030250 Apr 20 16:03 order_line_1.tbl
98708359 Apr 20 16:03 order_line_2.tbl
101014415 Apr 20 16:03 order_line_3.tbl
101028751 Apr 20 16:03 order_line_4.tbl
 5997569 Apr 20 16:03 orders_1.tbl
 6027850 Apr 20 16:03 orders_2.tbl
 6147526 Apr 20 16:03 orders_3.tbl
 6147400 Apr 20 16:03 orders_4.tbl
152901209 Apr 20 16:01 stock_1.tbl
152996867 Apr 20 16:01 stock_2.tbl
153408316 Apr 20 16:01 stock_3.tbl
153395417 Apr 20 16:01 stock_4.tbl
      429 Apr 20 16:01 warehouse_1.tbl
      471 Apr 20 16:01 warehouse_2.tbl
      441 Apr 20 16:01 warehouse_3.tbl
      419 Apr 20 16:01 warehouse_4.tbl
```

Running the Test

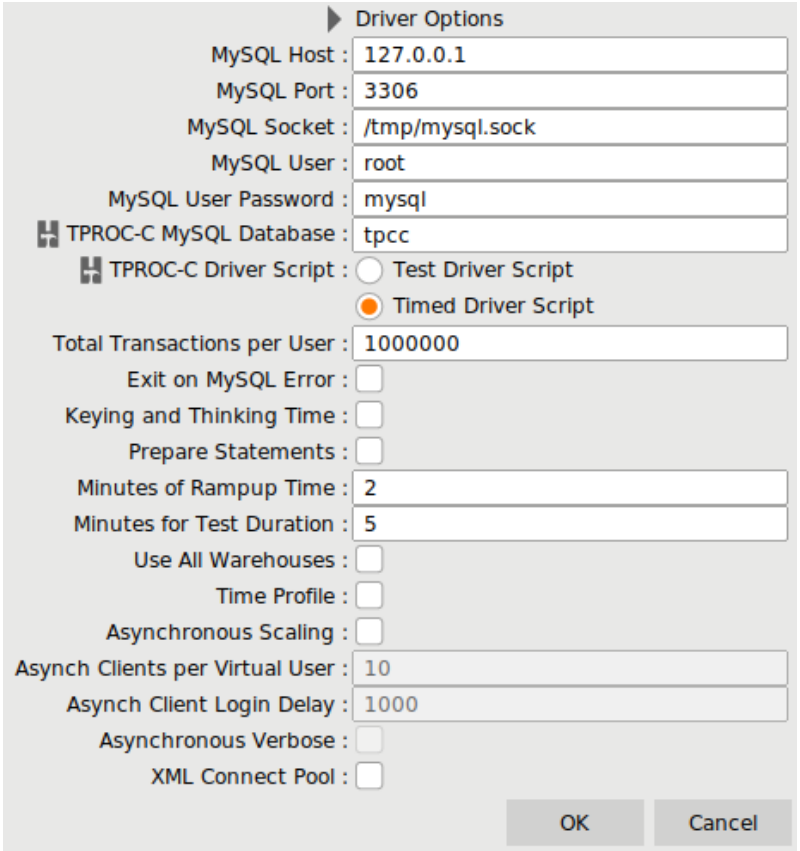
Running the Test

- Driver Script Options
  - Test Loads a driver script
  - Options modify script loaded
- Test Script
  - Simple run
  - Small number of Virtual Users
  - Verify Schema Build
- Timed Script
  - Measured Test
  - Small to larger number of Virtual Users
  - Suppressed Output



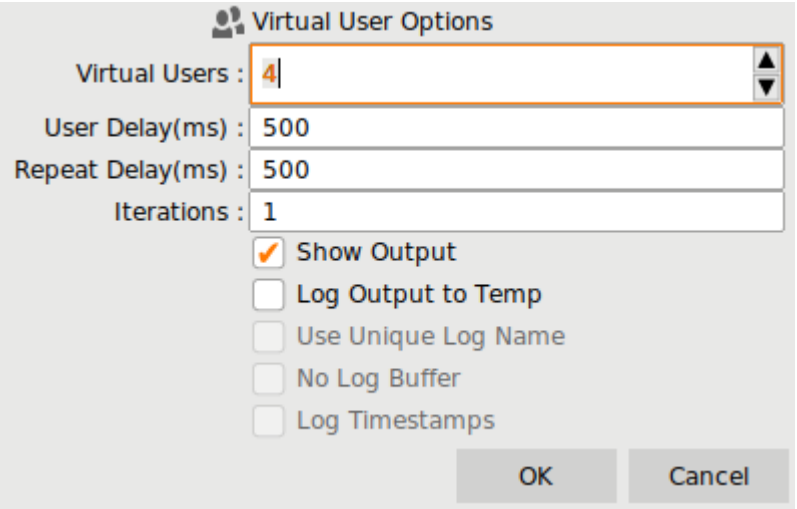
Driver Script Options

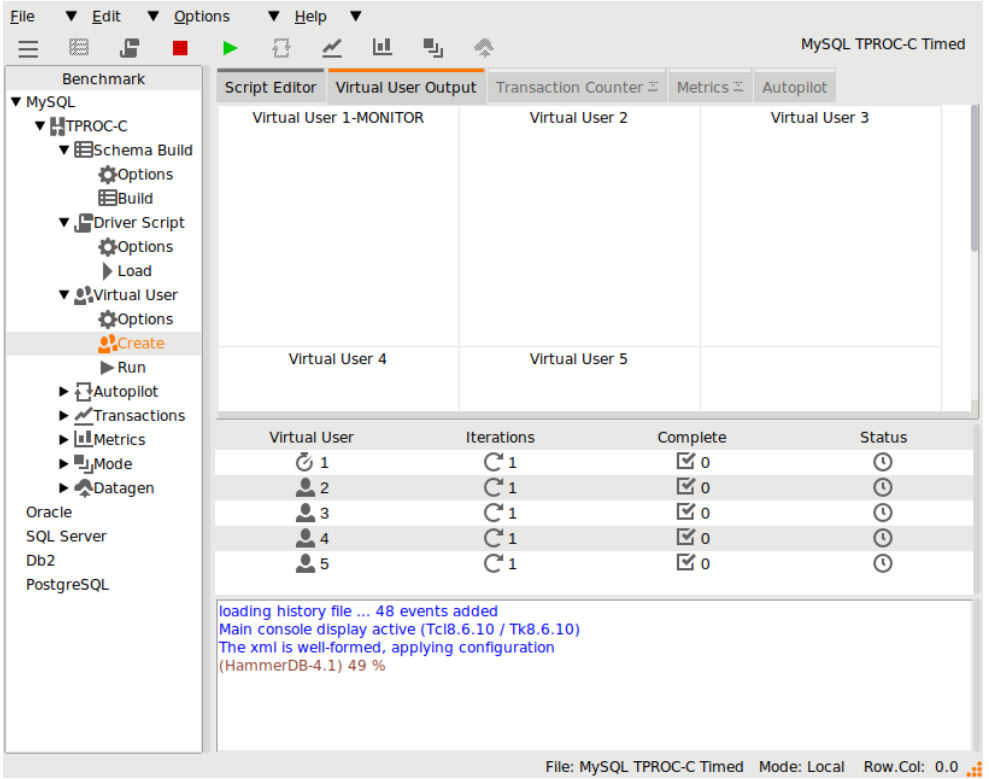
- Connection Parameters
- Driver Script
- Total Transactions
  - Sets an upper limit for number of transactions for each Virtual User to run
- Rampup Time
  - Time for data to load into cache
- Test Duration
  - Timed period of test
- Advanced Options



Virtual Users

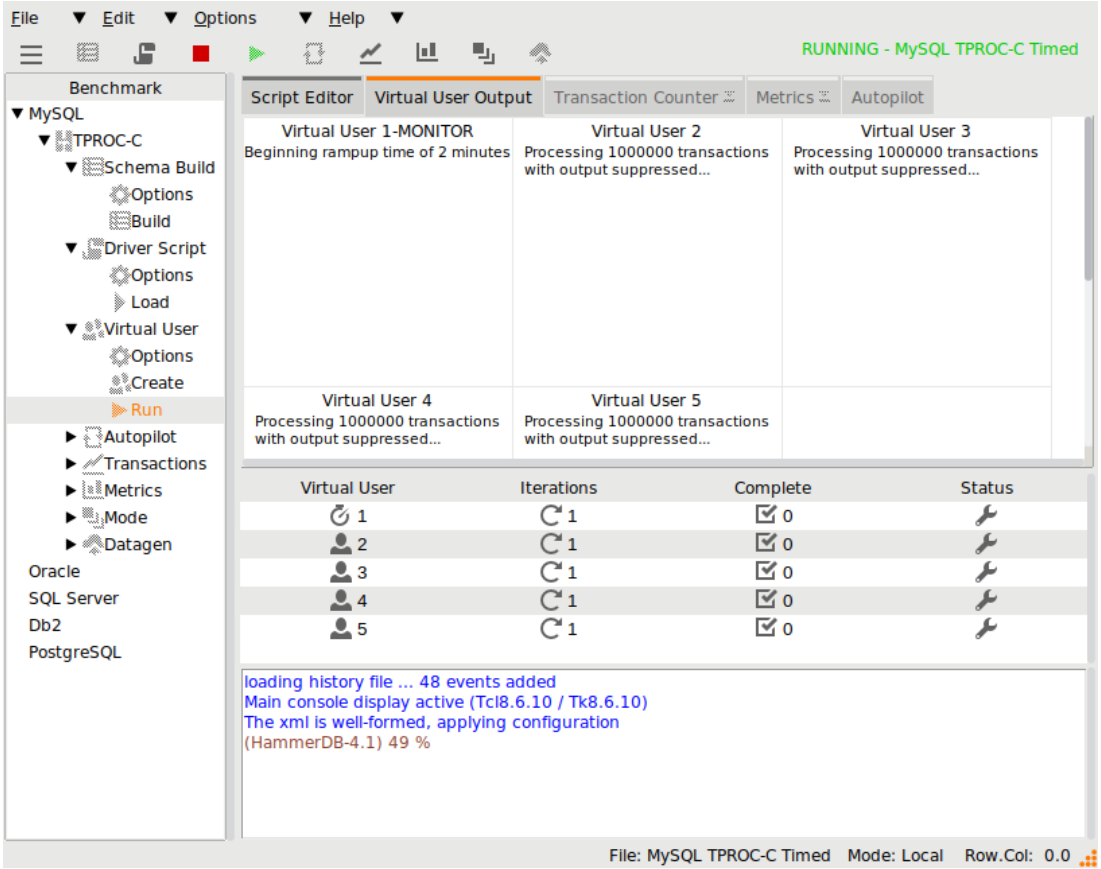
- Configure and Create Virtual Users
  - Virtual Users run in parallel
  - Each Virtual User is OS thread
  - Runs independently





Running the Test

- Click Run to start
- Transaction Mix
  - New Order 45%
  - Payment 43%
  - Delivery 4%
  - Stock Level 4%
  - Order Status 4%
- Status shown of Virtual Users
  - Running
  - Complete
  - Error Status
- Press Stop to terminate Virtual Users



Running a Test with the CLI

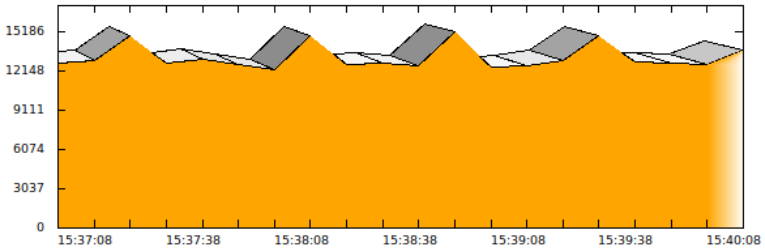
- Choose Options
- Load Script
- Workload is identical to that run by GUI (driver script is the same)

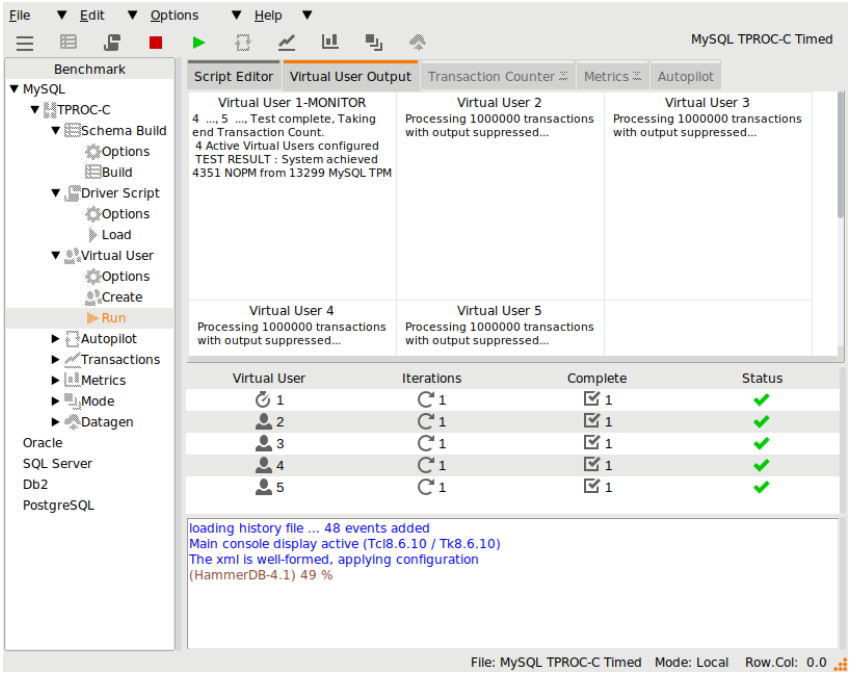
```
dbset db mysql
diset connection mysql_host localhost
diset connection mysql_port 3306
diset connection mysql_socket /tmp/mysql.sock
diset tpcc mysql_user root
diset tpcc mysql_pass mysql
diset tpcc mysql_driver timed
diset tpcc mysql_rampup 2
diset tpcc mysql_duration 5
loadscript
vuset vu 4
vucreate
vurun
runtimer 500
vudestroy
```

```
HammerDB CLI v4.1
Copyright (C) 2003-2021 Steve Shaw
Type "help" for a list of commands
The xml is well-formed, applying configuration
hammerdb>source myrun.tcl
Database set to MySQL
Changed connection:mysql_host from 127.0.0.1 to localhost for MySQL
Value 3306 for connection:mysql_port is the same as existing value 3306, no change made
Value /tmp/mysql.sock for connection:mysql_socket is the same as existing value /tmp/mysql.sock, no change made
Value root for tpcc:mysql_user is the same as existing value root, no change made
Value mysql for tpcc:mysql_pass is the same as existing value mysql, no change made
Clearing Script, reload script to activate new setting
Script cleared
Changed tpcc:mysql_driver from test to timed for MySQL
Value 2 for tpcc:mysql_rampup is the same as existing value 2, no change made
Value 5 for tpcc:mysql_duration is the same as existing value 5, no change made
Script loaded, Type "print script" to view
Vuser 1 created MONITOR - WAIT IDLE
Vuser 2 created - WAIT IDLE
Vuser 3 created - WAIT IDLE
Vuser 4 created - WAIT IDLE
Vuser 5 created - WAIT IDLE
5 Virtual Users Created with Monitor VU
Vuser 1:RUNNING
Vuser 1:Beginning rampup time of 2 minutes
Vuser 2:RUNNING
Vuser 2:Processing 1000000 transactions with output suppressed...
Vuser 3:RUNNING
Vuser 3:Processing 1000000 transactions with output suppressed...
Vuser 4:RUNNING
Vuser 4:Processing 1000000 transactions with output suppressed...
Vuser 5:RUNNING
Vuser 5:Processing 1000000 transactions with output suppressed...
```

Review Results

- Test Result Printed when compete
  - GUI
  - CLI
- Review Engine Throughput with Transaction Counter



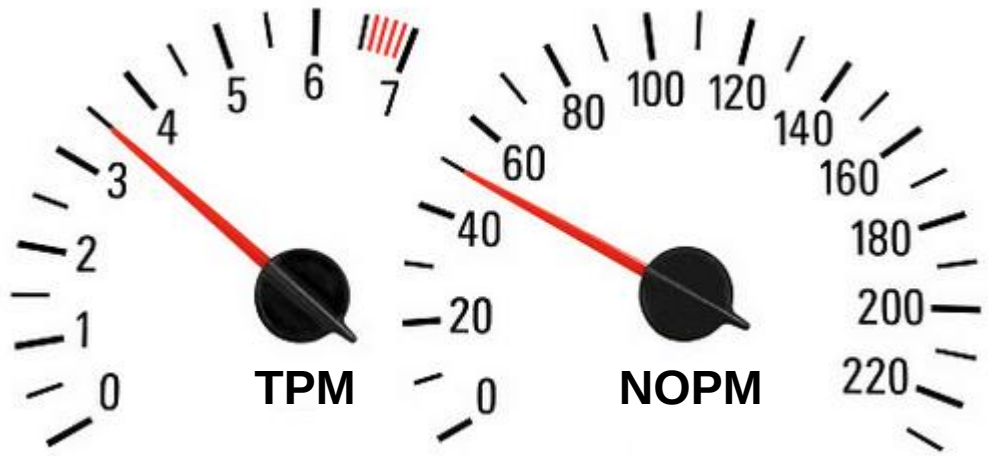



```
Vuser 1:Test complete, Taking end Transaction Count.
Vuser 1:4 Active Virtual Users configured
Vuser 1:TEST RESULT : System achieved 4335 NOPM from 13058 MySQL TPM
Vuser 1:FINISHED SUCCESS
Vuser 4:FINISHED SUCCESS
Timer: 7 minutes elapsed
Vuser 2:FINISHED SUCCESS
Vuser 5:FINISHED SUCCESS
Vuser 3:FINISHED SUCCESS
ALL VIRTUAL USERS COMPLETE
runtimer returned after 424 seconds
vudestroy success
```

Understanding Results: NOPM vs TPM

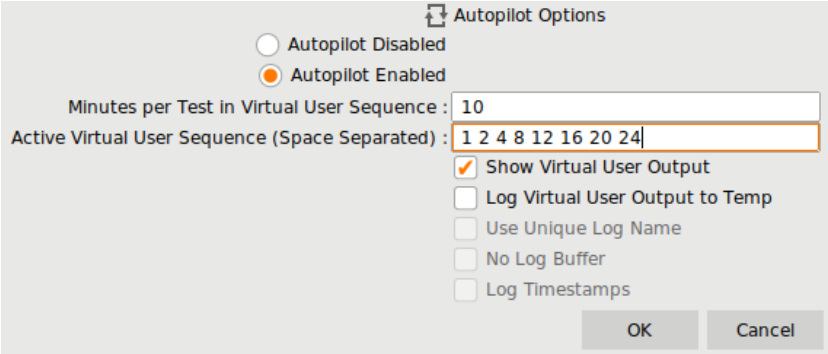
```
Vuser 1:Test complete, Taking end Transaction Count.
Vuser 1:140 Active Virtual Users configured
Vuser 1:TEST RESULT : System achieved 1722391 NOPM from 5216847 MySQL TPM
```

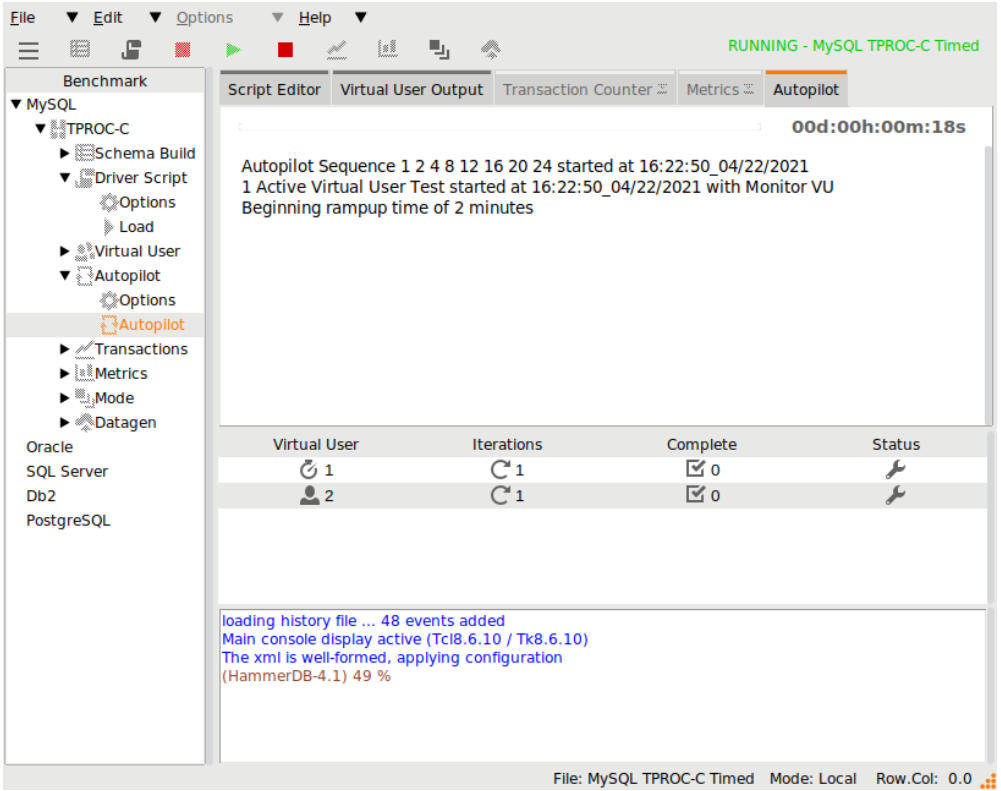
- NOPM
  - **How fast you are going**
  - Close relation to official tpmC
- TPM
  - **How hard your engine is working**
- Comparing performance
  - NOPM can be compared between engines
  - TPM can only be compared across the same engine
  - TPM useful engineering metric to compare statistics



GUI Automation: Autopilot

- GUI Automation
- Run Unattended Test Sequence
- Define sequence of tests
  - Increased Virtual User Count
- Log Output





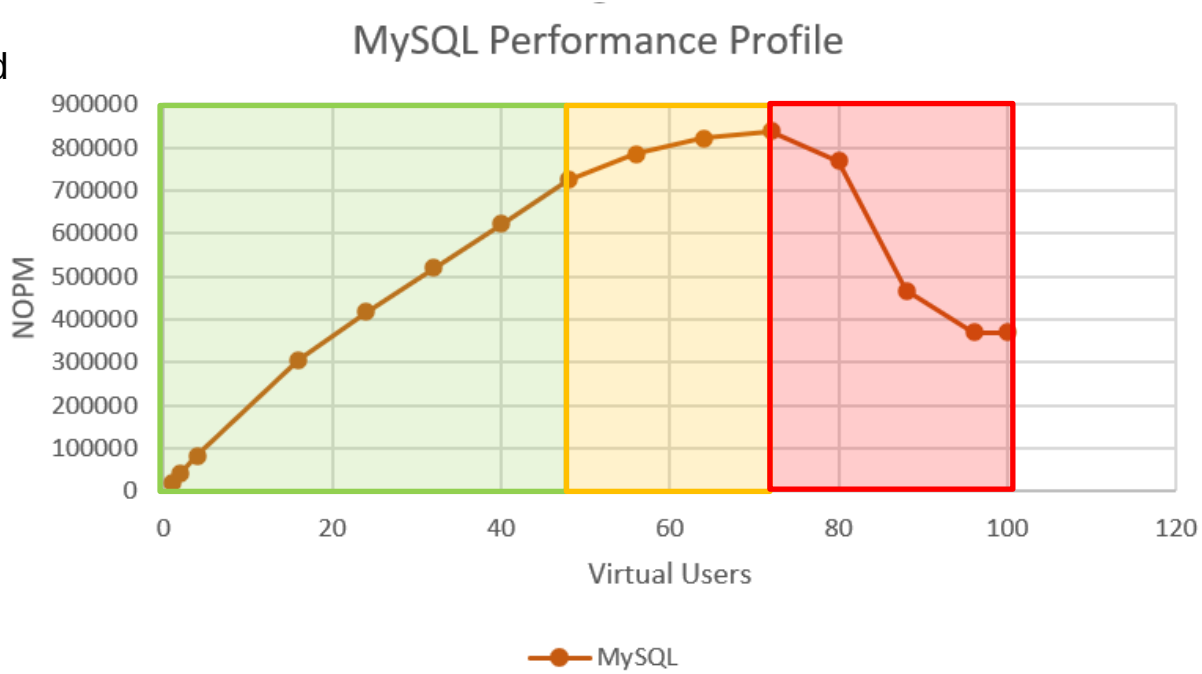
CLI Automation: Scripting

- CLI supports full TCL syntax
- Simple foreach loop for test sequence
- Can modify any parameters desired
- Log output

```
dbset db mysql
diset connection mysql_host localhost
diset connection mysql_port 3306
diset connection mysql_socket /tmp/mysql.sock
diset tpcc mysql_user root
diset tpcc mysql_pass mysql
diset tpcc mysql_driver timed
diset tpcc mysql_rampup 2
diset tpcc mysql_duration 5
loadscript
foreach z {1 2 4 8 12 16 20 24} {
puts "$z VU TEST"
vuset vu $z
vucreate
vurun
runtimer 500
vudestroy
}
```

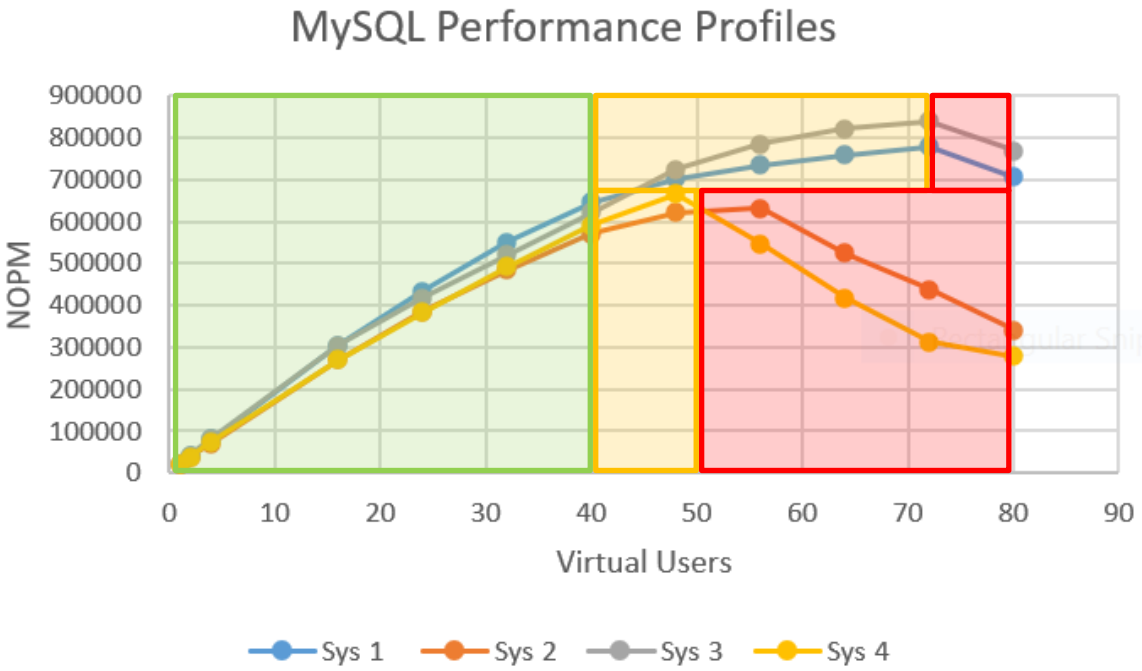
Performance Profiles

- Run Multiple Tests
  - Increasing Virtual User Load
- Example 56 (2 x 28) cores
- (Near) Linear Scale
  - Up to CPU cores/threads
  - Dependence on Database software
- Performance Plateau
  - Capture Peak Performance
  - Highest CPU Utilisation
- Contention
  - Increasing response times
  - Flat to lower performance



Comparing Performance

- Different Systems have different profiles
  - Not predictable on CPU Count
  - Database engines differ
- MySQL example
  - Linear scale is the same
  - Sys 1 + 3 extended performance plateau
  - Sys 2 + 4 show contention earlier
- Plan for differing levels of capacity



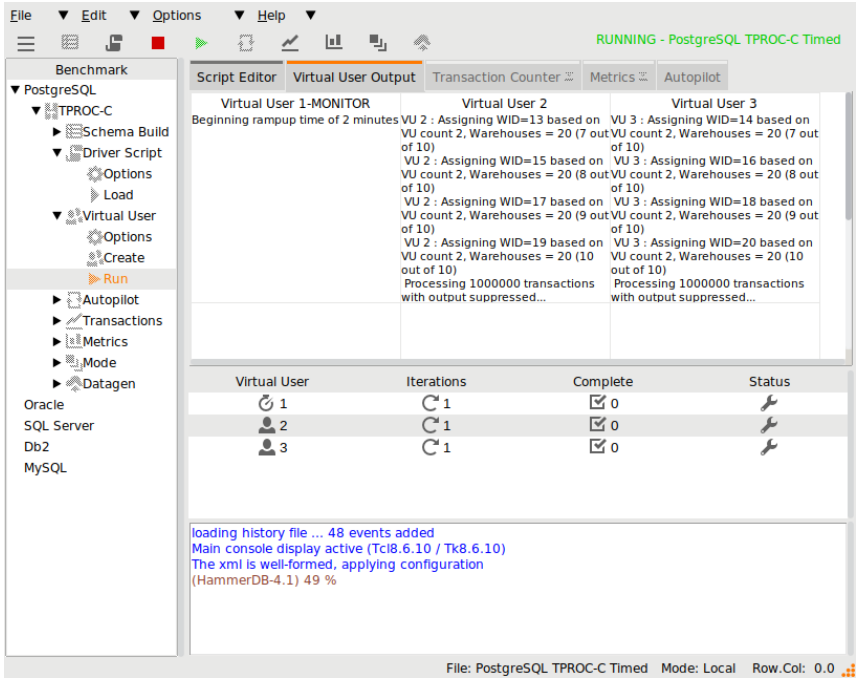
Advanced Testing Features

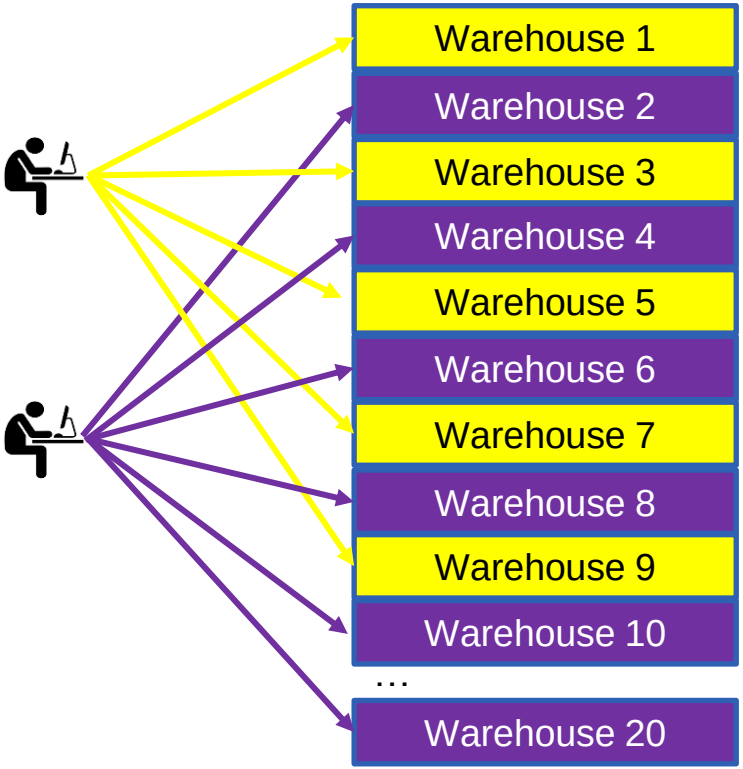
Advanced Testing Features

- Use All Warehouses
  - Increase physical I/O to the data area
- Connect Pooling
  - Direct parts of the workload to different nodes in the same cluster
  - For example read/write and read-only nodes
- Event Driven Scaling
  - Co-routine based
  - Implements keying and thinking time
  - Scales to thousands of sessions
- Time Profiling
  - Capture Virtual User response times
- Step Workloads
  - Variable throughput by adding and removing Virtual Users
- Advanced Features not mutually exclusive
  - Can use some or all of the advanced features at the same time

Use All Warehouses

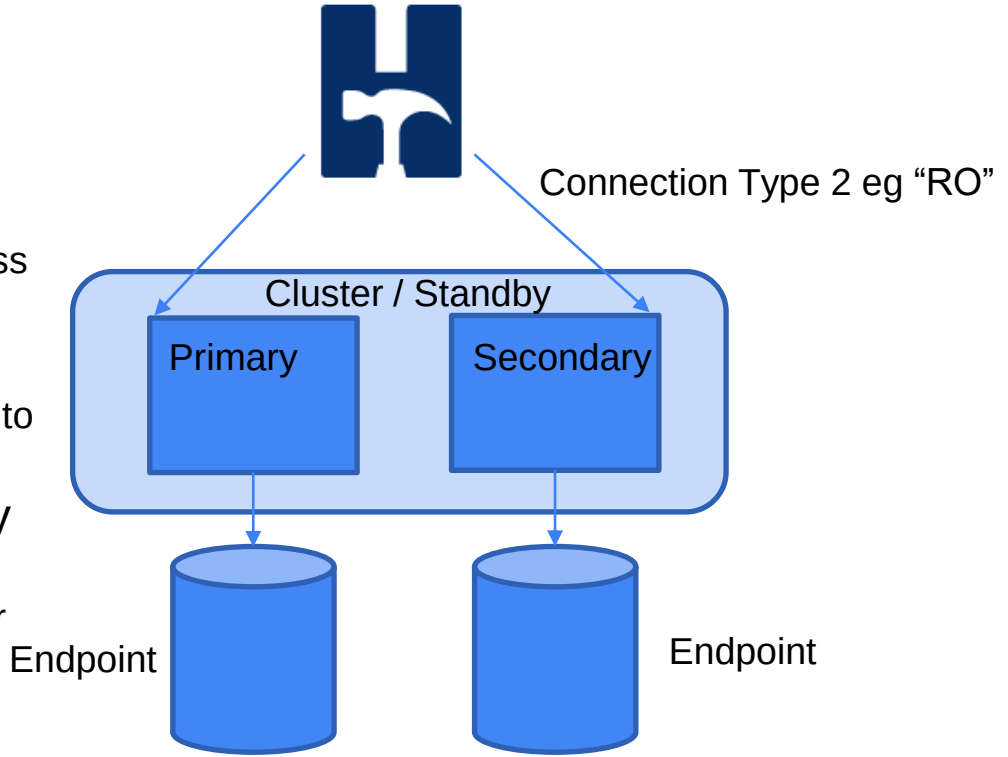
- All Warehouses divided between Virtual Users
  - New warehouse selected per transaction
  - More physical I/O





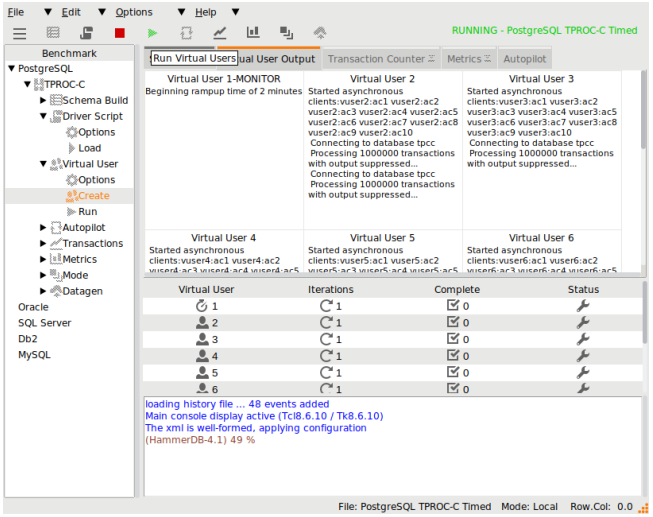
Connect Pooling for Clusters

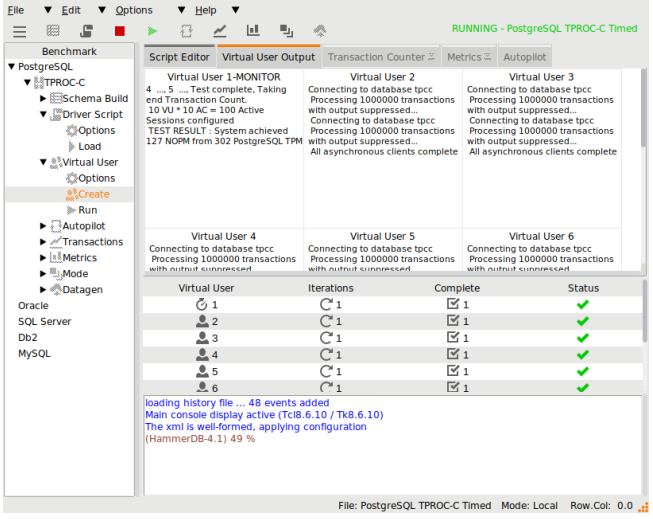
- Define in XML Configuration
  - Multiple connections instances in cluster
  - Which transactions are directed to which nodes
  - Policy on how to allocate transactions across pool of connections eg round robin
- Example RW/RO nodes
  - Define RW transactions to primary and RO to standby
- Reports NOPM and TPM from Primary
  - Also reports client side TPM
  - Detailed view of transactions processed per node



Event Driven Scaling

- Default Workload is Cached
- Scaled Workloads
  - Large Session Counts
  - Fixed Throughput
  - Keying and Thinking Time delays
- Requires larger storage and networking
  - Requires middleware
  - HammerDB connects to middleware
  - Middleware connects to database
- Multiple Sessions per Virtual User
  - Uses co-routines to make key and think asynchronous
  - Appx 1 NOPM per session
- Example 1000 warehouses
  - 10,000 Sessions
  - 10,000 NOPM

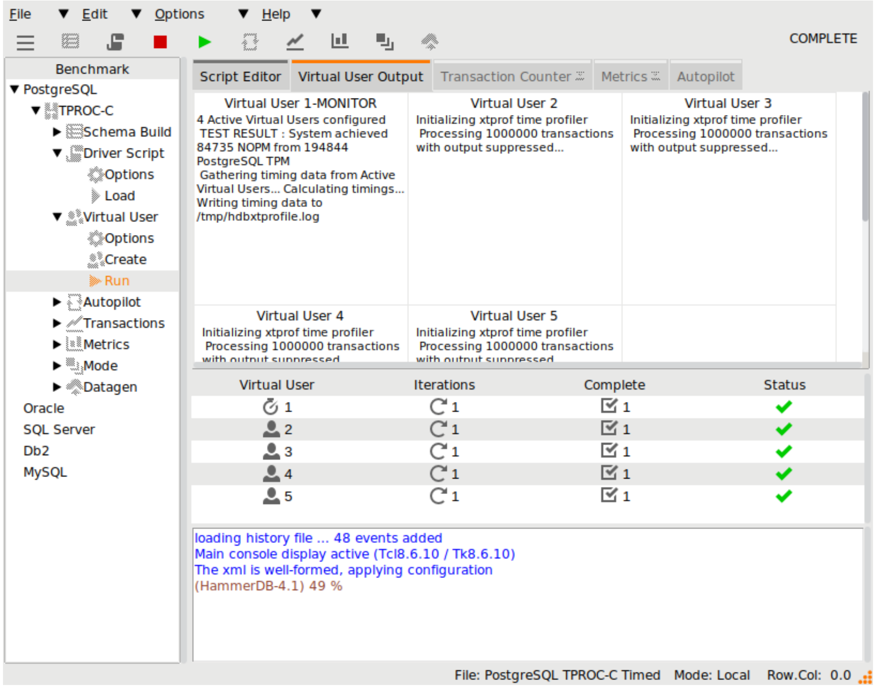




Time Profiling for Response Times

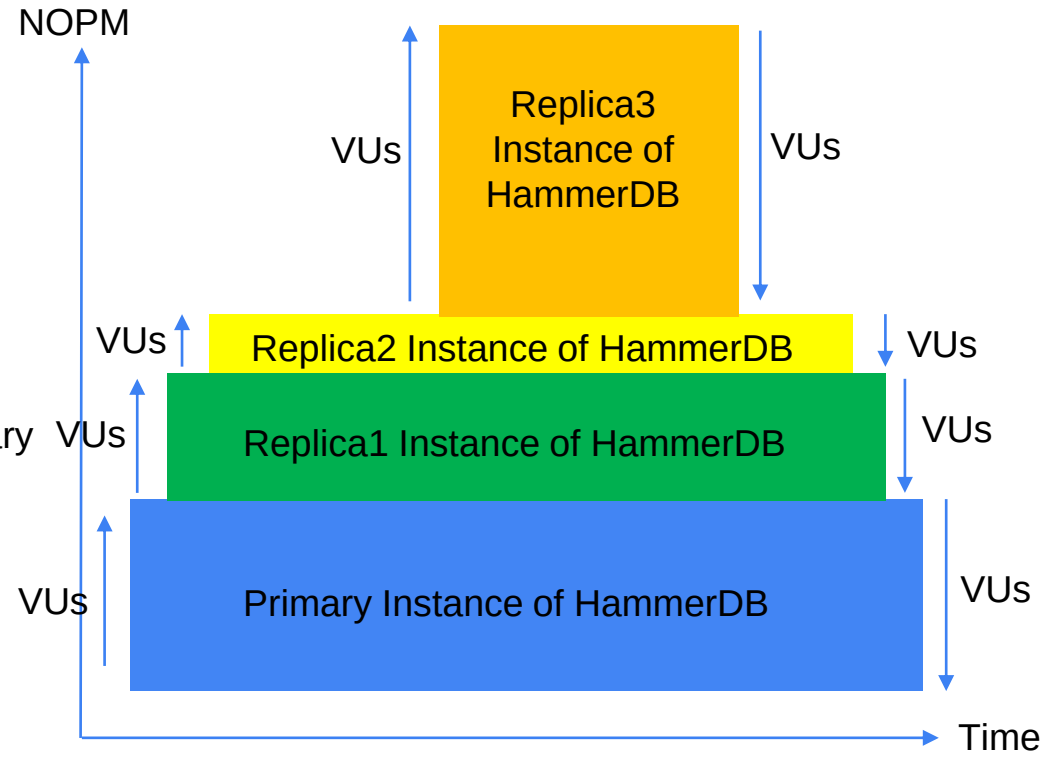
- 2 Time Profiling Packages
  - Xtprof – all virtual users
  - Etprof – first active virtual user
- Xtprof
  - Profile of all virtual user response times
  - Summary of all virtual users

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
>>>>> SUMMARY OF 4 ACTIVE VIRTUAL USERS : MEDIAN ELAPSED TIME : 179324ms
>>>>> PROC: NEWORD
CALLS: 253013    MIN: 0.414ms    AVG: 1.375ms    MAX: 121.556ms   TOTAL: 347988.879ms
P99: 4.628ms     P95: 2.553ms    P50: 1.092ms    SD: 1951.917     RATIO: 48.514%
>>>>> PROC: PAYMENT
CALLS: 252926    MIN: 0.183ms    AVG: 0.822ms    MAX: 123.252ms   TOTAL: 208144.667ms
P99: 3.319ms     P95: 1.780ms    P50: 0.544ms    SD: 1855.444     RATIO: 29.018%
>>>>> PROC: DELIVERY
CALLS: 25081     MIN: 0.589ms    AVG: 2.663ms    MAX: 34.680ms    TOTAL: 66813.642ms
P99: 7.967ms     P95: 4.789ms    P50: 2.352ms    SD: 1244.199     RATIO: 9.315%
>>>>> PROC: SLEV
CALLS: 25412     MIN: 0.699ms    AVG: 1.782ms    MAX: 41.734ms    TOTAL: 45289.974ms
P99: 4.685ms     P95: 3.030ms    P50: 1.594ms    SD: 811.589      RATIO: 6.314%
>>>>> PROC: OSTAT
CALLS: 25286     MIN: 0.149ms    AVG: 0.696ms    MAX: 123.539ms   TOTAL: 17610.867ms
P99: 3.018ms     P95: 1.556ms    P50: 0.400ms    SD: 2373.871     RATIO: 2.455%
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```



Variable Step Workloads

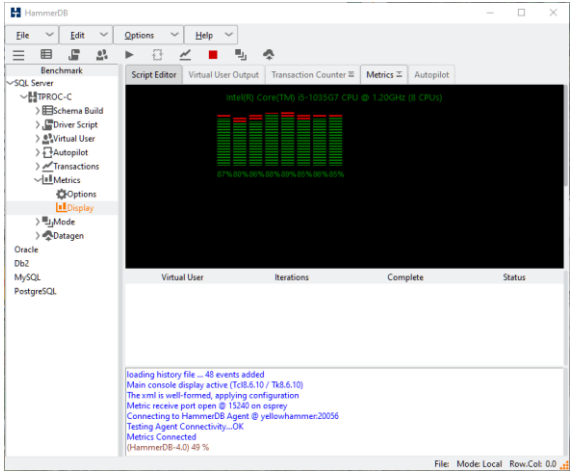
- Creates Variable Load
  - Define in XML
  - Pyramid of HammerDB Instances
- Runs in CLI only
  - Primary Instance of HammerDB
  - Replica instances created automatically and connect to primary
  - Timed delay of replica starts
- Start with “steprun” command
- Measure response times
  - Variations in performance

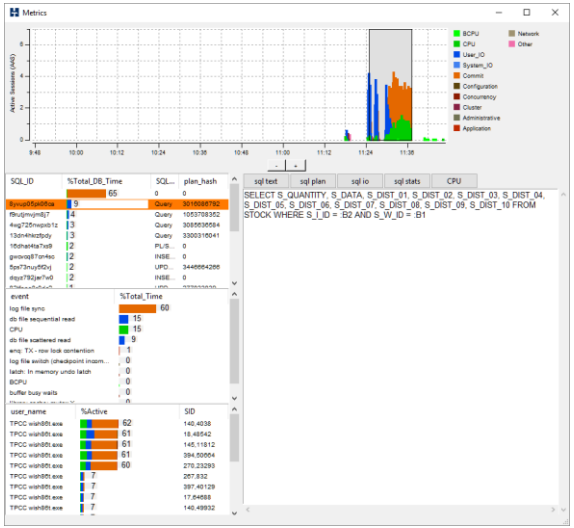


Performance Monitoring

CPU & Database Metrics

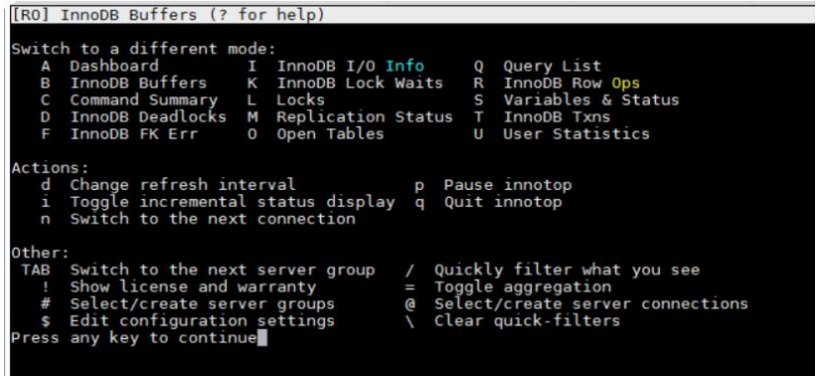
- HammerDB Graphical CPU Monitor
- Visualization of CPU load
- eg 50% CPU Average
  - Could be 50% of all cores at 50%
  - Could be 50% of cores at 100% and 50% at 0
- Detect CPU imbalance
- System & User CPU Utliization
- Identify Interrupt bottlenecks on individual cores
- GUI Database Metrics for PostgreSQL in progress

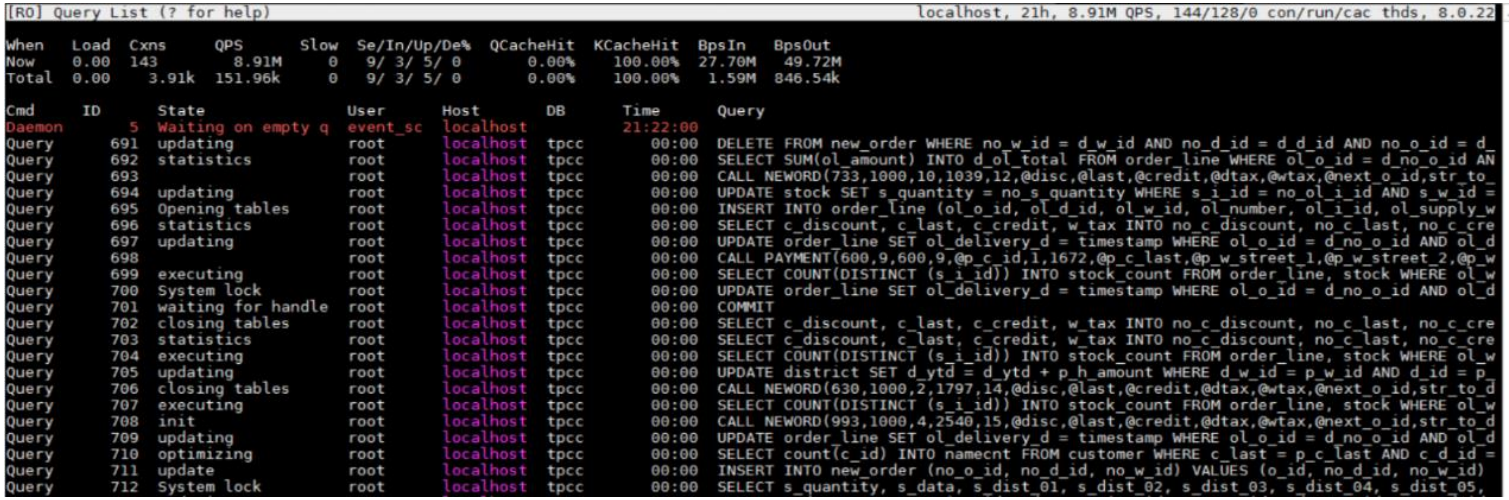




MySQL

- MySQL 8.0.20+ recommended
  - Improved Lock Scheduling
- Monitor InnoDB storage engine
- Innotop





PostgreSQL

- PostgreSQL 13+ recommended
  - Improved throughput
- pg_stat_statements / pg_sentinel

```
%  |  AAS  |  backend_type  | wait_event_type |      wait_event
----+-------+----------------+-----------------+---------------------
48 | 28.00 | client backend | CPU             | CPU
12 |  6.82 | client backend | LWLock          | XactSLRU
11 |  6.18 | client backend | LWLock          | WALInsert
9 |  5.41 | client backend | IPC             | ProcArrayGroupUpdate
6 |  3.71 | client backend | Client          | ClientRead
6 |  3.65 | client backend | IPC             | XactGroupUpdate
5 |  2.82 | client backend | Lock            | extend
2 |  0.94 | client backend | LWLock          | ProcArray
1 |  0.35 | client backend | IPC             | CheckpointDone
```

HPE LinuxKI

- LinuxKI Toolset (Trace-based performance analysis tool)
- System level analysis
  - Beyond database only statistics

MySQL - HammerDB

1.3.3 Trace Events of Top 10 Processes

[Prev Subsection][Next Subsection]---[Prev Section][Next Section][Table of Contents]

Analyzing Pid: 3180 Trace Records: 4327871 cmd: /usr/sbin/mysqld (mysqld)

Freq	Percent	Trace_type	64bit	ElapsedT	Max	Ave	Errors
1898956	43.88%	sys_enter					
1307788	30.22%	sched_yield	1	0.713	0.0044	0.000001	0
469256	10.84%	futex	1	3.808	0.0037	0.000008	1417
340574	7.87%	sched_wakeup					
188886	4.36%	sched_switch					
121911	2.82%	nanosleep	1	9.256	0.0067	0.000076	0
499	0.01%	hardclock					

MySQL - Sysbench

1.3.3 Trace Events of Top 10 Processes

[Prev Subsection][Next Subsection]---[Prev Section][Next Section][Table of Contents]

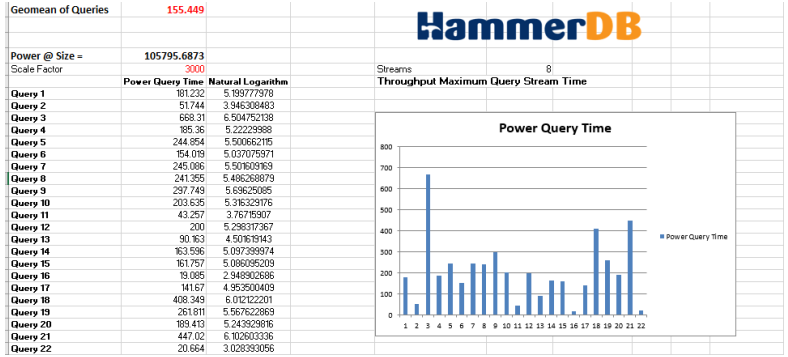
Analyzing Pid: 6134 Trace Records: 1786621 cmd: /usr/sbin/mysqld (mysqld)

Freq	Percent	Trace_type	64bit	ElapsedT	Max	Ave	Errors
742430	41.55%	sys_enter					
444924	24.90%	recvfrom	1	0.578	0.0008	0.000001	148226
151720	8.49%	sched_wakeup					
149126	8.35%	sched_switch					
148348	8.30%	sendto	1	0.365	0.0005	0.000002	0
148226	8.30%	ppoll	1	5.559	0.0055	0.000038	0
914	0.05%	hardclock					
821	0.05%	futex	1	0.085	0.0054	0.000104	146
111	0.01%	sched_yield	1	0.002	0.0004	0.000021	0

Next Steps

Analytic Testing

- TPROC-H for Analytics
- Cloud Queries
- Stream of 22 Complex Queries
- PostreSQL Parallel Query
- Columnstores
- More complex skillset required

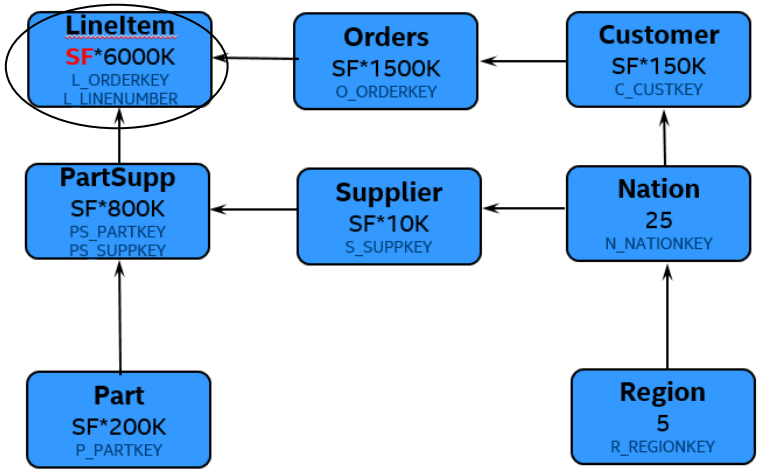


Geomean of Queries	155.449	
Power @ Size =	105795.6873	
Scale Factor	3000	
	Power Query Time	Natural Logarithm
Query 1	181.232	5.199777978
Query 2	51.744	3.946308483
Query 3	668.31	6.504752138
Query 4	185.36	5.22229988
Query 5	244.854	5.500662115
Query 6	154.019	5.037075971
Query 7	245.086	5.501609169
Query 8	241.355	5.486268879
Query 9	297.749	5.69625085
Query 10	203.635	5.315329176
Query 11	43.257	3.76715907
Query 12	200	5.298317367
Query 13	90.163	4.501619143
Query 14	163.596	5.097399974
Query 15	161.757	5.086095209
Query 16	19.085	2.948902686
Query 17	141.67	4.953500409
Query 18	408.349	6.01212201
Query 19	261.811	5.567622869
Query 20	189.413	5.243929816
Query 21	447.02	6.102603336
Query 22	20.664	3.029393056

HammerDB

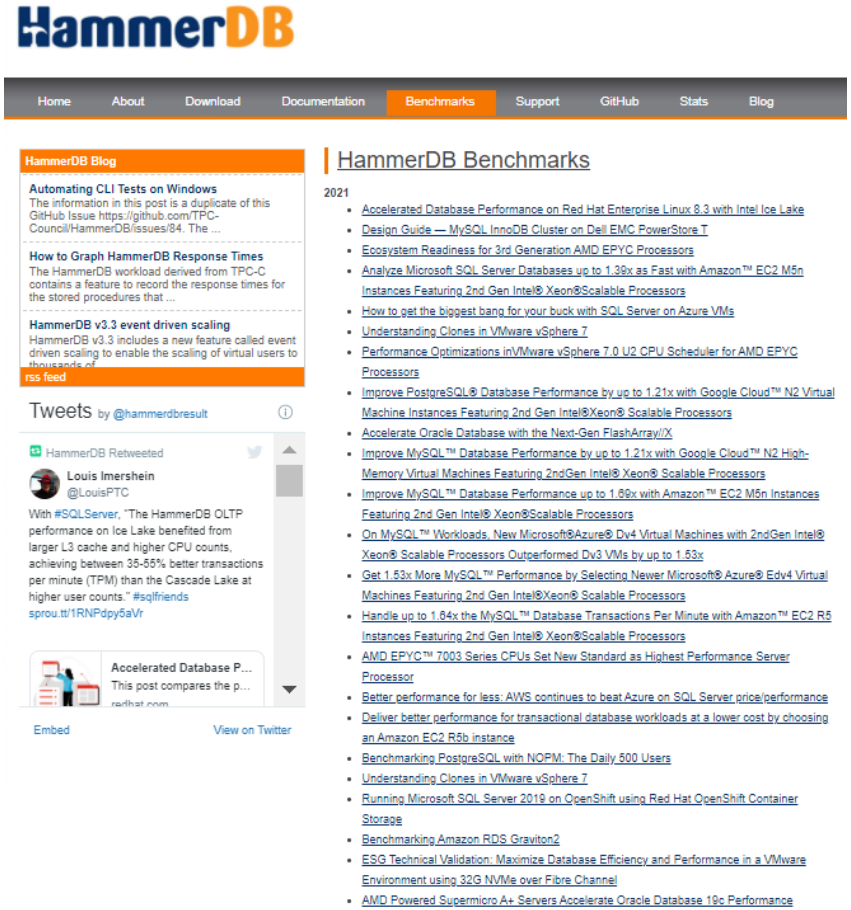
Streams 8

Throughput Maximum Query Stream Time



Published Benchmarks

- Has someone already done a performance study you can use?
  - https://www.hammerdb.com/benchmarks.html
- Have you published your findings for other people to use?
- Making database performance data open source benefits all



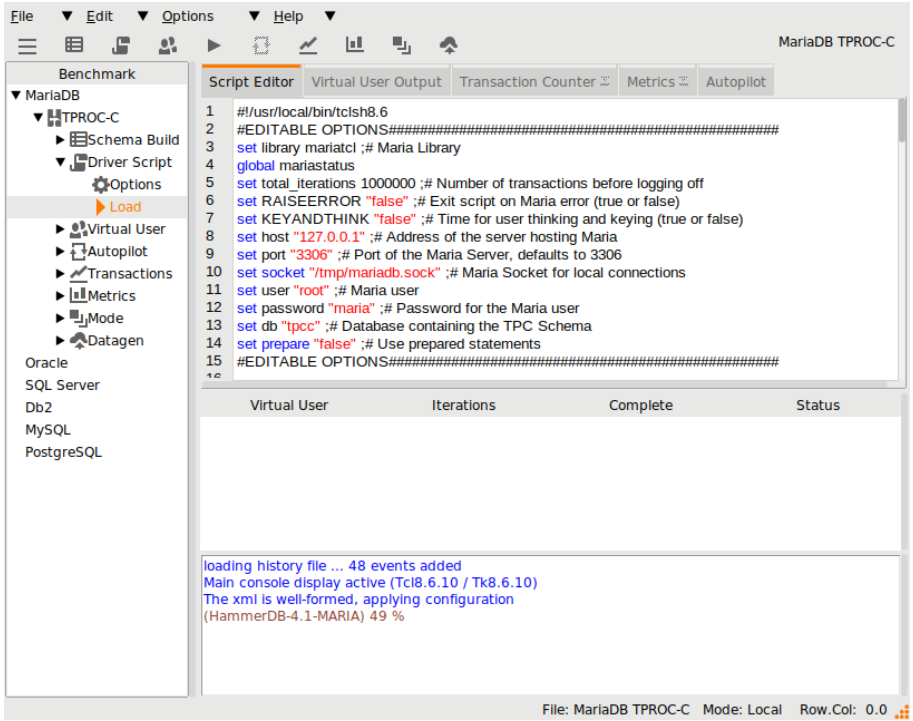
Up Next v4.2

Add MariaDB as a separate database #54 #234

New issue

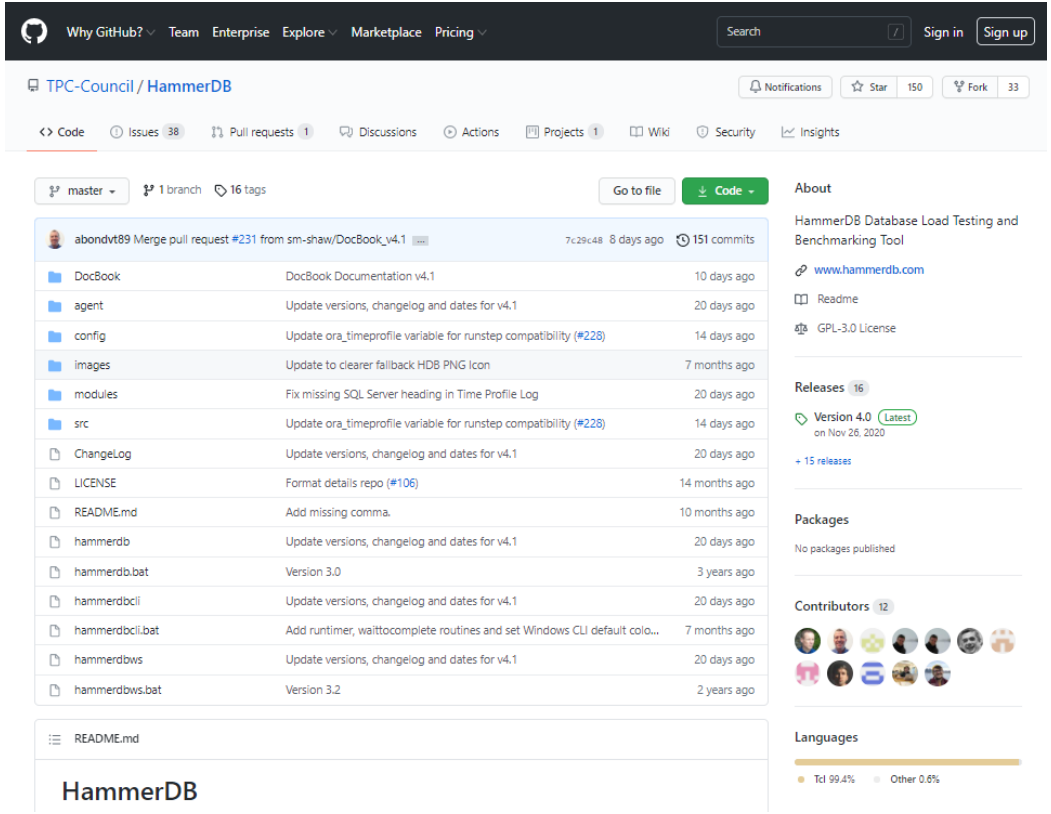
Open Jiang-Hua wants to merge 10 commits into `TPC-Council:master` from `Jiang-Hua:master`

- MariaDB as separate database
  - TPROC-C and TPROC-H
  - Current support requires MySQL client library
  - Future support MariaDB client
  - Opportunity to diversify workload



Contribute to HammerDB on GitHub

- Contribute to HammerDB
- All source code open source
- Documentation open source
  - Docbook format
  - Edit with any XML editor
- Issues
- Discussions
- Binary releases



Thanks!

Any questions?

THANK YOU !

PERCONA
LIVEONLINE
MAY 12 - 13th
2021