SQL Server Performance on AWS

October 2018

Notices

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Contents

Abstract

With so many instance types to choose from, it can be challenging for architects and customers to choose the best ones for migrating their SQL Server workloads to AWS. Detailed specifications are available for each instance type and storage options. However, all of this information may be difficult to map to real-world scenarios.

We introduce this SQL Server Benchmarking whitepaper to address these challenges. It provides information about how to benchmark instance types to test how they perform. It also shows the performance numbers from benchmarking popular combinations of compute and storage instance types.

The paper begins with an explanation of TPC-Benchmarking, introduces you to HammerDB, the industry default for database benchmarking, and then explains the scope of the tests. It gives details on the environment, workload, instance types, storage, and SQL Server configurations. The results of three tests are discussed:

- The first tests GP2 volumes only.
- The second tests GP2 volumes with the transaction log on the local NVMe disks.
- The third tests volumes from the instance store NVMe disks.

Final recommendations are provided.

A detailed appendix walks you through configuring HammerDB to run the TPC-C Hammer benchmark.

Introduction

Because there is such a large selection of instance families, generations, and sizes, it is difficult for architects and customers to select the right instance type. This is especially true when migrating SQL Server workloads to AWS. Although AWS publishes detailed specifications on the instance types and storage options, it can be challenging to map them to real-world workloads.

This paper provides guidance on how to benchmark instance types to see how they perform. It also provides performance numbers for popular combinations of compute and storage.

TPC-C Benchmarking

The TPC-C Benchmark is an online transaction processing (OLTP) benchmark. TPC-C involves a combination of five concurrent transactions of different types and complexity. The database is composed of nine types of tables with a wide range of record and population sizes. TPC-C is measured in transactions per minute (TPM).

While the benchmark portrays the activity of a wholesale supplier, TPC-C is not limited to the activity of any particular business segment. Rather, it represents any industry that must manage, sell, or distribute a product or service.

In this paper, we used HammerDB, an open-source, cross-platform, database load-testing and benchmarking tool. HammerDB is regarded as the industry 'default' for database benchmarking. We used it to generate TPC-C–like workloads.

Scope

We ran a TPC-C workload across a range of instance types. The SQL Server database was hosted on a variety of storage types. The performance numbers were logged and compared.

Environment

The tests are run in the US East (N. Virginia) Region. The HammerDB client machine and all SQL Server instances are launched in the same Availability Zone and placement group.

The AMI used for testing is Amazon's license-included SQL Server 2017 Enterprise (CU5) on Windows Server 2016.

- Windows_Server-2016-English-Full-SQL_2017_Enterprise-2018.04.11 (ami-ccb46cb3)

The AWS drivers on this AMI are:

- AWS PV driver 8.2
- AWS ENA driver 1.2.3.0
- AWS NVMe driver 1.0.0.146

HammerDB Workload

We use HammerDB to generate a large workload schema of 1,000 warehouses. This database is backed up to an EBS volume and then a snapshot is taken. The snapshot is used to create a new volume to attach to the SQL Server instances being tested. The database backup is restored on to the target instance.

For more information about the HammerDB installation and workload generation, see the appendix. In SQL Server Management Studio (SSMS), see the TPC-C schema with 1,000 warehouses.

dbo.STOCK	100,000,000	34,185,592	32,000,008	2,180,048	5,536
dbo.CUSTOMER	30,000,000	23,285,688	21,818,336	1,462,712	4,640
dbo.ORDER_LINE	300,002,025	19,729,072	19,672,384	54,152	2,536
dbo.HISTORY	30,000,000	1,777,040	1,776,640	80	320
dbo.ORDERS	30,000,000	1,552,880	979,712	568,608	4,560
dbo.NEW_ORDER	9,000,000	164,016	160,512	848	2,656
dbo.ITEM	100,000	12,840	9,560	464	2,816
dbo.DISTRICT	10,000	1,256	1,112	48	96
dbo.WAREHOUSE	1,000	296	112	48	136

TPC-C schema with 1,000 warehouses

Instance Types

A combination of eight General Purpose, Compute, Memory, and Storage-optimized instances are chosen, which are the common instance types that customers may consider for their high-performance workloads. All are Amazon EBS-optimized instances.

Type	Family	vCPU	Memory	Instance Storage	Network Performance	EBS-optimized: Max Bandwidth	EBS-optimized: Throughput	EBS-optimized: Max 16K IOPS
General Purpose	m5.xlarge	4	16	EBS only	High	2,120 Mbps	265.0 MB/s	16,000 IOPS
Storage Optimized	i3.2xlarge	8	61	1,900 GiB	Up to 10 Gigabit	1,700 Mbps	212.5 MB/s	12,000 IOPS
Storage Optimized	i3.4xlarge	16	122	3,800 GiB	Up to 10 Gigabit	3,500 Mbps	437.5 MB/s	16,000 IOPS
Memory Optimized	r4.4xlarge	16	122	EBS only	Up to 10 Gigabit	3,500 Mbps	437.5 MB/s	18,750 IOPS
Memory Optimized	r5d.4xlarge	16	128	600 GiB	Up to 10 Gigabit	3,500 Mbps	437.5 MB/s	18,750 IOPS
Memory Optimized	r4.8xlarge	32	244	EBS only	10 Gigabit	7,000 Mbps	875.0 MB/s	37,500 IOPS
Memory Optimized	x1e.8xlarge	32	976	960 GiB	Up to 10 Gigabit	3,500 Mbps	437.5 MB/s	20,000 IOPS
Memory Optimized	r4.16xlarge	64	488	EBS only	25 Gigabit	14,000 Mbps	1750.0 MB/s	75,000 IOPS
Memory Optimized	m5.24xlarge	96	384	EBS only	25 Gigabit	10,000 Mbps	1250.0 MB/s	65,000 IOPS

Storage

General Purpose SSD (GP2) volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver:

- Single-digit millisecond latencies
- The ability to burst to 3,000 IOPS for extended periods of time
- A baseline performance of 3 IOPS/GiB up to a maximum of 10,000 IOPS (at 3,334 GiB).

GP2 volumes can range in size from 1 GiB to 16 TiB. Customers can achieve higher IOPS and throughput by striping multiple GP2 volumes together.

Some instance types include fast local instance store disks. These disks offer great performance when properly configured.

SQL Server Configurations

The following SQL Server configuration settings ensure consistency and comparability of results.

Max Degree of Parallelism

A few initial test runs showed TPM dropping due to inefficient parallel query plans. We stabilized this by setting MAXDOP to 1 to prevent any parallel query plans. For real workloads, you would examine those query plans and potentially set MAXDOP to a more suitable number, for example, the number of vCPUs.

Max SQL Server Memory

We tried to keep the SQL settings as close as possible to the default install settings. In the case of Max Server Memory, we opted to ensure that at least 4 GB was reserved for the operating system; that is, Max Server Memory = (Total Server Memory – 4 GB).

Simple Mode

To limit the impact of unexpected transaction log growth and log space issues, we put the workload database in SIMPLE mode.

Results

In this section, we walk through the configuration and performance numbers for the three instance-type scenarios across which we ran TPC-C workloads.

Run 1 – GP2 Only

We started by testing GP2 volumes only.

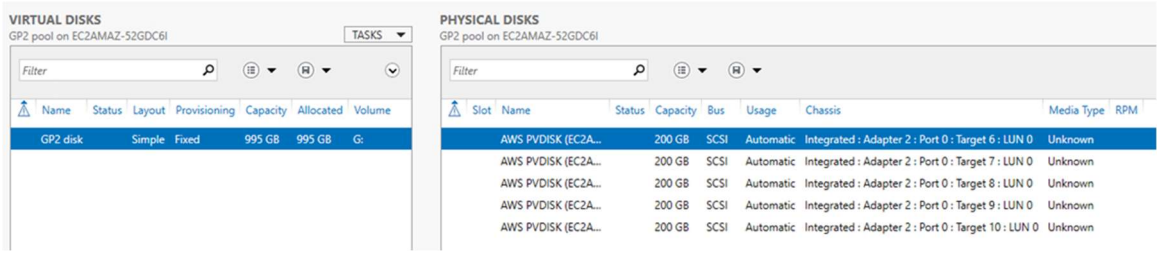
Overview

GP2 volumes are the most cost-effective choice for non-demanding database workloads requiring a high degree of durability. For the benchmark, we created five 200-GB GP2 volumes.

Name	Size	Volume Type	IOPS	Availability Zone	State	Volume Status
GP2 1000 1 OF 5	200 GiB	gp2	600 / 3000	us-east-1b	in-use	Okay
GP2 1000 2 OF 5	200 GiB	gp2	600 / 3000	us-east-1b	in-use	Okay
GP2 1000 3 OF 5	200 GiB	gp2	600 / 3000	us-east-1b	in-use	Okay
GP2 1000 4 OF 5	200 GiB	gp2	600 / 3000	us-east-1b	in-use	Okay
GP2 1000 5 OF 5	200 GiB	gp2	600 / 3000	us-east-1b	in-use	Okay

Five 200-GB GP2 volumes

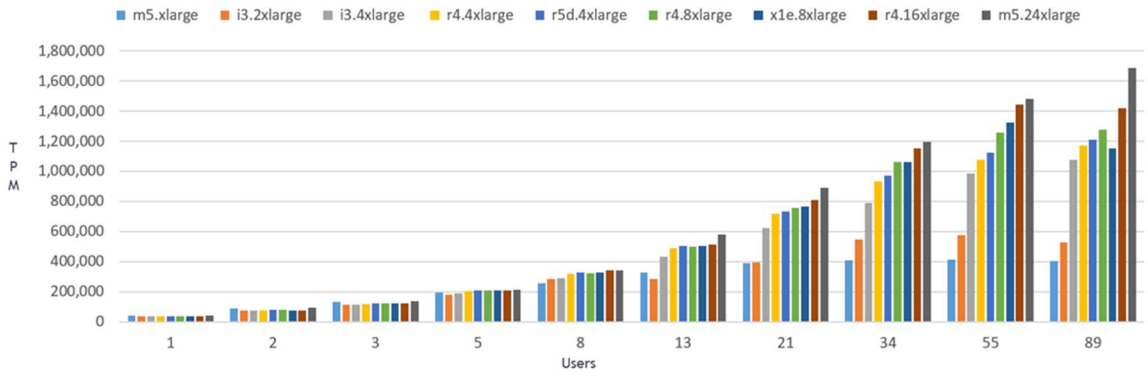
We then added these five volumes to a pool on the instance using Windows storage spaces and created a single disk from this pool. Striped together, these GP2 volumes provide a single disk with 15,000 max burst IOPS (16K) and a throughput of 800 MB/sec.



Five 200-GB GP2 volumes striped together to create a single disk

We then ran HammerDB for ten minutes on each instance type and recorded the TPM for each batch of users.

Chart



TPM for each batch of users by instance type

Conclusions

Performance is similar across instance types with single-digit users. As we scaled up the number of concurrent users, we saw that the larger instance types delivered higher TPM. Smaller instance types, like m5.xlarge and i3.2xlarge, are limited by Amazon EBS throughput and hit a ceiling around 400,000 TPM.

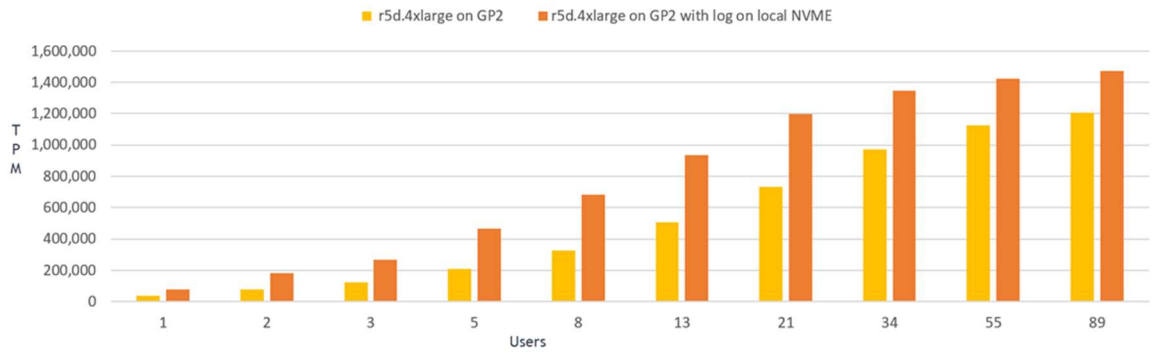
Run 2 – GP2 with Transaction Log on Local NVMe Disk

In our next scenario, we tested GP2 volumes with the transaction log on the local NVMe disks.

Overview

For this test, we placed the transaction log on the local NVMe disks, which are included with R5d instance types. We compared the results to those from where the database is only on the GP2 volumes.

Chart



TPM for each batch of users for R5d and R5d with the transaction log on local NVMe

Conclusions

Performance can be significantly improved by placing the transaction log on a local NVMe store. In this case, the R5d instance includes local NVMe storage.

Warning: Local NVMe SSD storage is not automatically replicated like Amazon EBS. If the instance is stopped or terminated, all data on the local SSD may be lost. For more information, see Amazon EC2 Instance Store.

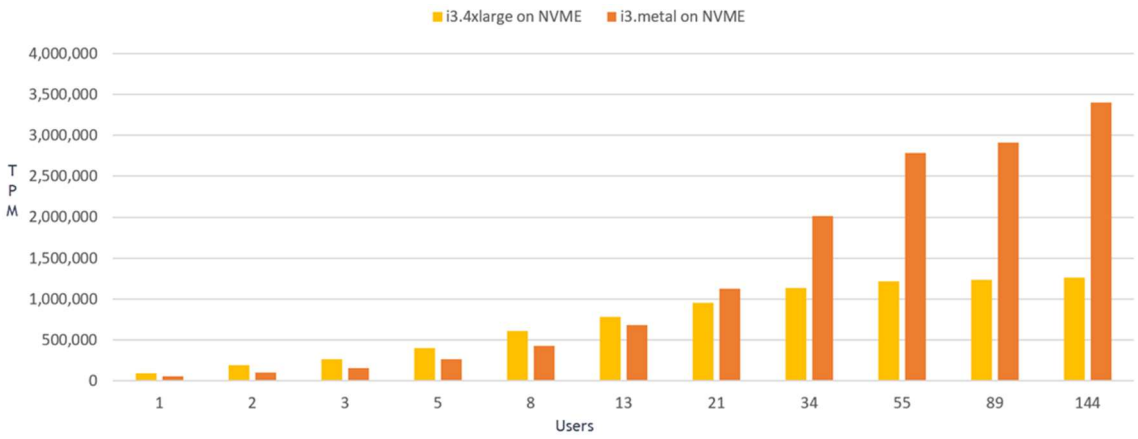
Run 3 – All NVMe

Our last scenario tests volumes from the instance store NVMe disks

Overview

In this test, we created volumes from the instance store NVMe disks and placed all data and log files on them.

Chart



TPM for each batch of users for i3.4xlarge on NVMe and i3.metal on NVMe

Conclusions

We were able to achieve over 3 million TPS with i3.metal. NVMe can be a good choice for low-latency OLTP workloads.

Warning: Local NVMe SSD storage is not automatically replicated like Amazon EBS. If the instance is stopped or terminated, all data on the local SSD may be lost. For more information, see Amazon EC2 Instance Store.

Final Conclusions and Recommendations

As expected, the larger instance types performed better with larger workloads. GP2 is a great storage choice for enterprise workloads, provided the volumes are configured correctly. Local NVMe disks provide flexibility where ultra-low latency is required.

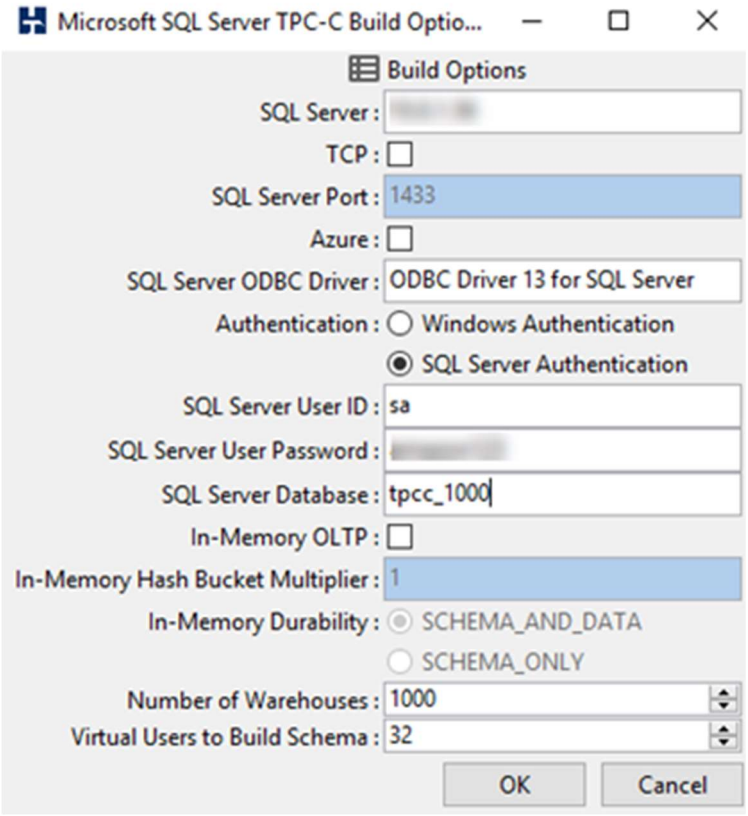
Appendix

The following procedure sets up HammerDB.

To configure HammerDB

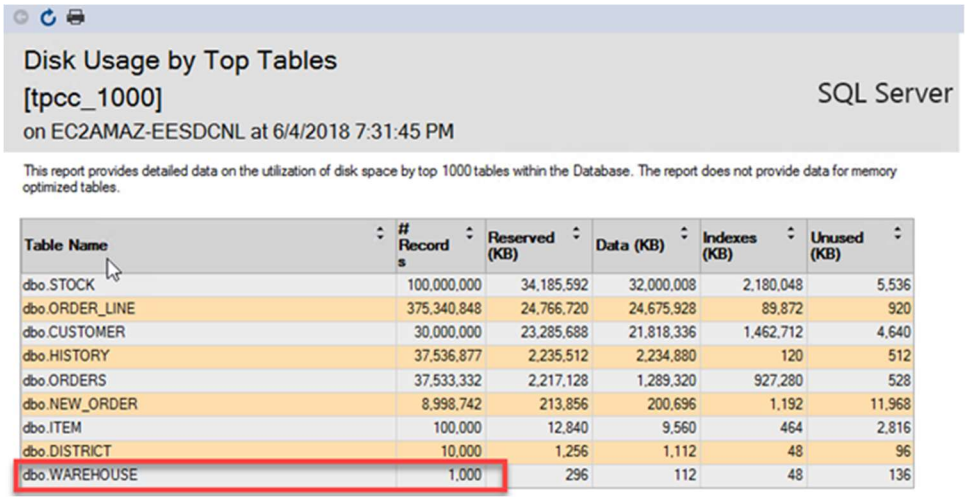
1. Navigate to http://www.hammerdb.com/, choose **Download,** and select the latest version for your Windows OS. As of 8/2018, it is Release 3.0.
2. Copy the downloaded file to the remote machine from which you plan on running the test.

   Note: This test uses a pre-created database (empty) set to SIMPLE RECOVERY. This mode has a SQL-authenticated account (with SysAdmin permissions), and pre-grown DB data and log files.
3. Navigate to C:\Program Files\HammerDB-3.0 and open hammerdb.bat.
4. In the **Benchmark** panel, choose **SQL Server**.
5. For **Benchmark Options**, ensure that both **SQL Server** and **TPC-C** are select and choose **OK**.
6. Choose **OK**.
7. Choose **TPC-C**, **Schema Build**, and **Options**.
8. In the **Options** menu, fill out the form with the following information and then choose **OK**:
   - Your IP address (the SQL Server target against which the test runs).
   - User ID (if you've designated a SQL-authenticated account).
   - The password for the account.
   - The number of warehouses. This example uses 1000, but you can use as many as you like. However, a higher number of warehouses takes more time to create.
   - Increase the **Virtual Users to Build Schema** value to a number closer to the vCPUs on the machine against which you are running the test (SQL Server).



TPC-C Build Options menu

9. In the **Benchmark** panel, choose **Build**. To confirm, choose **Yes**.
10. On the **Virtual User Output** tab, view messages about objects being created and loaded.
11. To view the schema build operation, view the **Disk Usage by Top Tables** report. Choose **SSMS**, **Reports**, **Standard Reports**, and **Disk Usage by Top Tables**. Open the context (right-click) menu on the database.



Disk Usage by Top Tables
[tpcc_1000]
on EC2AMAZ-EESDCNL at 6/4/2018 7:31:45 PM

SQL Server

This report provides detailed data on the utilization of disk space by top 1000 tables within the Database. The report does not provide data for memory optimized tables.

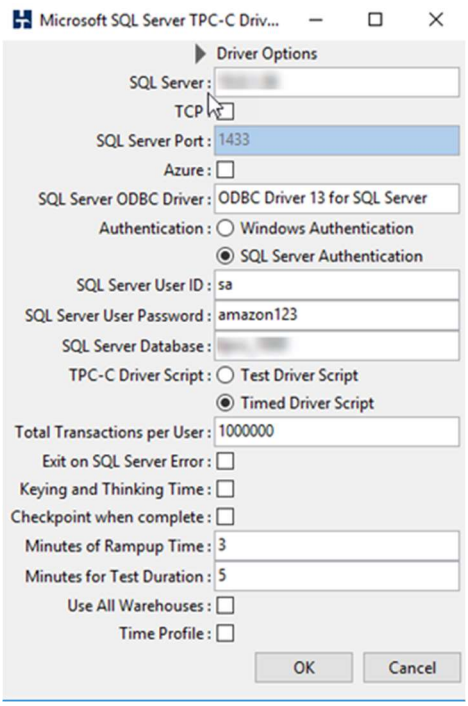
Table Name	# Records	Reserved (KB)	Data (KB)	Indexes (KB)	Unused (KB)
dbo.STOCK	100,000,000	34,185,592	32,000,008	2,180,048	5,536
dbo.ORDER_LINE	375,340,848	24,766,720	24,675,928	89,872	920
dbo.CUSTOMER	30,000,000	23,285,688	21,818,336	1,462,712	4,640
dbo.HISTORY	37,536,877	2,235,512	2,234,880	120	512
dbo.ORDERS	37,533,332	2,217,128	1,289,320	927,280	528
dbo.NEW_ORDER	8,998,742	213,856	200,696	1,192	11,968
dbo.ITEM	100,000	12,840	9,560	464	2,816
dbo.DISTRICT	10,000	1,256	1,112	48	96
dbo.WAREHOUSE	1,000	296	112	48	136

Disk Usage by Top Tables report

12. Verify that the schema build is complete before moving on to the **Driver Script** configuration. The following is an example of a completed build.

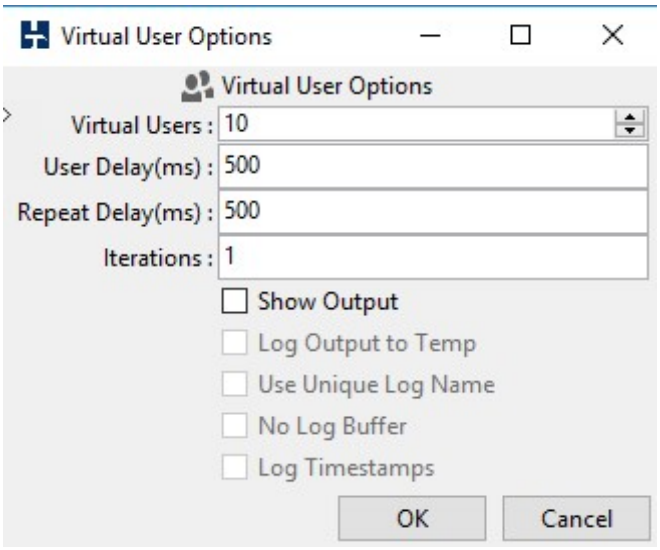
Schema Build Complete message

13. In the **Benchmark** pane, choose **Driver Script**, **Options**.
14. Enter the SQL Server IP address, your authentication method, SQL Server database, and the type of TPC-C Driver Script option, and then choose **OK**.



TPC-C Build Options menu

15. In the **Benchmark** pane, choose **Load**.
16. To run a manual test (untimed), choose **Virtual User**, **Options**. This example uses ten virtual users for the first test. Change this when increasing the number of virtual users after each test is complete. Uncheck **Show Output**, and choose **OK**.



Virtual Users Options window

17. Choose **Create** so that the tool can create the user connections that the HammerDB tool is establishing.
18. Choose **Run** and select the graph icon ().
19. Choose **Transaction Counter** to view the TPM metric. The metric takes a few seconds to display.
20. To end the test, choose the red **Stop** icons (one stops the transactions while the other stops the load).
21. To change the number of virtual users against which to run the TPC-C HammerDB benchmark, repeat steps 16–20.

Contributors

The following individuals and organizations contributed to this document:

- Bini Berhe, solutions architect, AWS Solutions Architecture
- Alan Cranfield, systems engineer, AWS Windows Business

Document Revisions

Date	Description
August 2018	First publication