Performance Evaluation of TPC-C Benchmark on Various Cloud Providers

Raghu Nandan Avula
Department of Electrical & Computer Engineering
University of Central Florida,
Orlando, USA
raghunandan@knights.ucf.edu

Cliff Zou
Department of Computer Science
University of Central Florida,
Orlando, USA
czou@cs.ucf.edu

***Abstract*— With the hardware costs becoming cheaper day by day and with industry giants focus, Cloud computing has exploded and reached leaps and bounds in the last 15 years. With the power of creating, using and destroying virtual machines in the cloud at the tip of mouse click, industries have started moving their core applications to the cloud. This has reduced the hassle for industries to maintain the hardware by themselves. Tech giants like Amazon, Microsoft and Google are head of the game and the fierce competition between them has led to astonishing innovation. With so many players in the cloud market, it is essential for cloud users to know how each of the services provided by these cloud service providers are performing against each other. In this paper we have evaluated the performance of famous OLTP benchmark TPC-C on these cloud providers. It is observed that Amazon's AWS has performed better than Microsoft Azure and Google Cloud Platform in terms of the number of transactions/orders per second, and I/O reads/writes. We have done the extended comparison with respect to transaction throughput, database throughput and Machine throughput.**

Keywords*— *Amazon Web Services, Microsoft Azure, Google Cloud Platform, TPC-C;

I. Introduction

Amazon Web Services (AWS), Microsoft Azure and Google Cloud Platform (GCP), the leading cloud providers, have a 75% combined market share and are the leaders in the cloud computing services. Users or Companies can spin up their instance in the cloud to run their web or database servers and scale with a minimal effort. These services have the "pay as you go model" where you pay for the services that you have used for. This will help the users or organizations not having to worry about the underlying hardware setup and cost, hardware security, and maintenance. However, choosing the right cloud service from many cloud providers is not an easy task for an individual or a company, and hence, there is a tremendous need to evaluate and compare the services provided by these providers. With AWS, Azure and GCP being the leading cloud providers, in this paper we have done our performance evaluation on them. We are very much interested to know given the similar configuration and specs how each of these cloud providers performs.

TPC benchmark C also know as TPC-C which is the leading on-line transaction processing (OLTP) benchmark has been used to perform the comparison [1]. In this paper we evaluate the cloud service performance based on the following four metrics: (1). Transaction Throughput, which is the number of transactions per minute; (2). Number of orders per minute; (3). Database throughput, which includes the number of statements executed per second along with Innodb reads and writes per second; (4). Machine throughput, which is CPU, memory utilization and disk I/O reads and writes. All of our performance evaluation is done on the virtual environments that are provided by the three cloud providers. As a consumer of these services we have no knowledge about the underlying hardware, its make and model or the exact location. However, we get to know important specs like CPUs, Memory in GB, Network bandwidths, Disk IOPS before we run the benchmark. Our aim is to compare instances with similar specs and configurations from each cloud provider to investigate how they perform in running a timed benchmark. The rest of this paper discusses about the related work followed by Instance types used in this project and then design, performance evaluation. Memsql [3] has done performance evaluation of Memsql in AWS environment by running the TPC-C benchmark. Azarudeen, Ganesh, Dinesh and Ramakrishnan [4] have used TPC-C to evaluate the performance of their architecture that guarantees the confidentiality of the data. To the best our knowledge, there were no previous survey conducted on the performance evaluation of major cloud providers by running TPC-C benchmark. Schad et al. [5] carried out a comprehensive study on performance variation of small and large instances on Amazon EC2.MapReduce application as well as micro-benchmarks including, Unix benchmark utility (Ubench) for CPU and Memory speed, Bonnie++ for Disk I/O and Iperf for network bandwidth, were used. . Folkerts et al.
[6] listed sample use-cases and proposed appropriate benchmarks for each use-case including TPC-C. They first defined the actors involved in cloud benchmarking, including their value network, and the system under test (SUT). Hill et al. [7] investigated the performance of computing, storage and database services offered by Microsoft Azure. They compared the SQL Server installed on the Same LAN and different LANs. As per our knowledge, we are first ones to compare the performance of leading cloud providers using TPC-C benchmark using HammerDB.

II. Cloud Providers

In this paper, we have run the TPC-C benchmark on 3 major cloud providers namely Amazon Web Services, Microsoft Azure, and Google Cloud Platform. AWS, Azure and GCP account for 70% of the combined cloud market share as of 2018 [2]. Gartner have placed these three cloud providers as leaders in their Quadrant for Cloud providers. In this section, we briefly describe each cloud provider and the cloud services we have used for testing.

A. AMAZON WEB SERVICES

Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud platform, offering over 175 fully featured services from data centers globally. Millions of customers—including

978-1-7281-9656-5/20/$31.00 ©2020 IEEE

the fastest-growing startups, largest enterprises, and leading government agencies—are using AWS to lower costs, become more agile, and innovate faster.[8]

Services used: We have used the following AWS services to run our TPC-C Benchmark using HammerDB.

a. AWS Virtual Private Cloud
b. AWS EC2 Instance
c. AWS EBS

B. MICROSOFT AZURE

Azure is an ever-expanding set of cloud computing services to help any organization meet its business challenges. Azure gives the freedom to build, manage, and deploy applications on a massive, global network using preferred tools and frameworks [9].

Services used: We have used the following Azure services to run our TPC-C Benchmark using HammerDB.

a. Azure Virtual Network
b. Azure Virtual Machines
c. Azure Disks
d. Azure Blob Storage

C. GOOGLE CLOUD PLATFORM

GCP consists of a set of physical assets, such as computers and hard disk drives, and virtual resources, such as virtual machines (VMs), that are contained in Google's data centers around the globe. Each data center location is in a global region [10].

Services used: Below are the services that we have used to build and use benchmark that was running on MySQL server.

a. Google Virtual Private Cloud
b. Google Compute Engine
c. Google Persistent Disk
d. Google Cloud Storage

III. INSTANCE TYPES

In this project, to know the behavior of how MySQL is performing in various CPU and Memory combinations we have used three different instance types to run our TPC-C benchmark to investigate the cloud service performance under different focuses. Instance types that were used in this project are:

A. GENERAL PURPOSE:

General purpose instances provide a balance of computing, memory, and networking resources, and can be used for a variety of workloads. We want to investigate what performance each of the cloud provider provides for this type of instance.

Cloud Providers	Instance Type	VM Setup
AWS	M5.xlarge	CPU: 4 cores Memory: 16 GB N/w bandwidth: up to 10 Gbps
Azure	B4MS	
GCP	n2-standard-4 vm	

Table 1: General purpose cloud service instances and VM setup

B. MEMORY OPTIMIZED:

Memory-optimized instances are designed to deliver fast performance for workloads that process large datasets in memory.

Cloud Providers	Instance Type	VM Setup
AWS	C5.2xlarge	CPU: 8 cores Memory: 16 GB N/W bandwidth: up to 10 Gbps
Azure	F8s V2	
GCP	n2-highcpu-8	

Table 2: Memory Optimized cloud service instances and VM setup

C. COMPUTE OPTIMIZED:

Compute-optimized instances are ideal for compute-bound applications that benefit from high-performance processors.

Cloud Providers	Instance Type	VM Setup
AWS	R5.xlarge	CPU: 4 cores Memory: 32 GB N/W bandwidth: up to 10 Gbps
Azure	E4 V3	
GCP	n2-highmem-4	

Table 3: Computer Optimized cloud service instances and VM setup

IV. TPC-C BENCHMARK

A. Definition

TPC Benchmark C is a standard for on-line transaction processing (OLTP). Owing to its various interaction styles, increasingly dynamic databases and overall execution layout, TPC-C is more complicated than previous OLTP comparisons, such as TPC-A (TPC-A measures performance in update-intensive database environments typical in on-line transaction processing applications). TPC-C consists of a combination of five parallel transactions of varying styles and complexities either carried out on-line or in queues for delayed execution. The database consists of nine tables with a large variety of documents and population scales as shown in Figure 1. Although the benchmark defines a wholesale supplier's operation, TPC-C is not restricted to any market segment's activity but rather reflects any industry that must produce, deliver or supply a good or service [11].

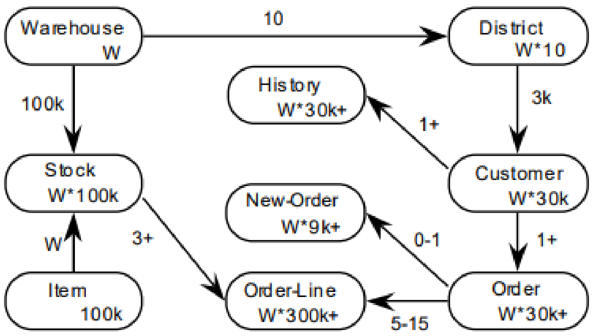

Figure 1: TPC-C Benchmark Database Design

B. Model

TPC-C Benchmark focuses on the core activities of an order-entry system. Transactions such as accessing shipments, logging sales, verifying the order status and tracking the stock amount at the warehouses. In this business model, a wholesale parts supplier operates several warehouses and their associated sales districts. TPC Benchmark is planned to scale as the company expands and new warehouses are built. Each warehouse in the TPC-C model must supply ten sales districts and each district serves three thousand customers. Each warehouse tries to maintain stock for the 100,000 items in the company's catalog and fill orders from that stock. An important metric to consider is the performance metric that is reported by TPC-C as number of orders that can be fully processed per minute and is expressed in tpm-C.

V. HAMMERDB

A. Definition

HammerDB is a load balancing and benchmarking tool for database loads. Loading it with data and simulating the workload of several virtual users against the database for both transactional and analytical scenarios to construct a test scheme. This workload can then be used to extract useful details about your system, such as comparisons of hardware performance and software configurations. The online transaction tracker records the amount of transaction taken from the database. With optimum configuration and the transaction rate would be constant and the transaction counter flat until most of the data is stored in memory. [12]

B. Setup

As a part of the benchmark setup, we have chosen to use 20 warehouses with users ranging from 100 to 1000. Below is the benchmark setup configuration.

a. Warehouses: 20
b. Users: 100, 500, 1000
c. Ramp-up: 3
d. Duration: 5

C. Build Schema

Our experimental setup is explained in the next section where HammerDB is run on a bastion host will connect to MySQL residing in General, Memory of Compute Optimized instance. As a first step, we have built the tpc-c benchmark by running the following commands in HammerDB.

a. Set the database to MySQL
    > dbset db mysql
b. Set the benchmark to tpc-c
    > dbset bm tpc-c
c. Set the warehouse count to 20
    > diset tpcc mysql_count_ware 20
d. Set the initial set of users
    > diset tpcc mysql_num_vu 5
e. Build the schema
    > buildschema

This will build the schema with 20 warehouses with 10 sale districts per warehouse and 100,000 items.

D. Run Benchmark

Once we build the schema, the next step would be running the benchmark for set of users (100 or 500 or 1000) in each instance type. I have run the following commands to start the benchmark.

a. Set the database to MySQL
    > dbset db mysql
b. Set the benchmark to tpc-c
    > dbset bm tpc-c
f. Set the timed execution on MySQL server
    > diset tpcc mysql_driver timed
g. Set the ramp up time to 3.
    > diset tpcc mysql_rampup 3
h. Set the execution duration to 5 after the ramp up is completed.
    > diset tpcc mysql_duration 5
i. Now load the script that run the SQL command.
    > loadscript
j. Set the number of users to 100 or 500 or 1000. In the below script, I have set the number of user connections to be 100.
    > vuset vu 100
k. Create the above users and run the benchmark.
    > vucreate
    > vurun
l. To check the status of the job, I have used
    > vucomplete

HammerDB will display the number of transactions per minute and number of orders per minute at the end of the benchmark. We have used these number to plot TPC-C throughput metrics.

VI. DESIGN, SETUP & METRICS

A. Design

As all the instances are spun up in a cloud service provider, we have decided to run the instance with HammerDB in a separate instance different from MySQL Server 8. HammerDB is installed an instance/virtual machine which has 2cpus with 4 GB of RAM. None of the software is required in this instance. We have installed MySQL Server in GP or Memory Optimized or CPU Optimized residing in the same virtual network. Network Traffic is within this private network and reduced the network latency. In Figure 2 we depict the design for each type instance in each cloud provider. We connect to the Bastion host with has HammerDB and run the benchmark on HammerDB. HammerDB is setup to connect to the respective MySQL instance using MySQL user credentials.

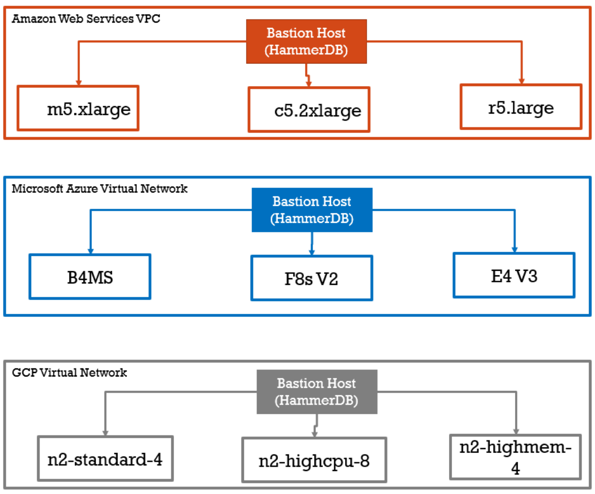

Figure 2: Experimental Design of Benchmark.

B. *Setup & Specs*

In this section we will discuss about the software setup in the MySQL Server. We have used Red Hat Linux as our Operating system as we have some issues with Ubuntu. Our MySQL Version is 8.0.0 Community Server. We have also installed *sysstat* package in RHEL to get the Linux performance metrics which will be discussed in the Metrics section.

As part of this project, we have collected the following three evaluation metrics: TPC-C Throughput, Database Throughput, and VM Throughput. We explain each metric in details below.

C. *TPC-C THROUGHPUT*

HammerDB provides the TPC-C Throughput at the end of each benchmark run. Statistics collected as a part of this throughput metric are:

- Number of order per minute
- Number of transactions per minute

The above metrics are displayed at the end of each benchmark run on the screen hence we don't have to do anything extra to get this information.

D. *DATABASE THROUGHPUT*

Database throughput is the important metric that we have collected in this project. These metrics are collected from MySQL Innodb Engine. We have followed the following procedure to collect these datapoints. Before we start running the benchmark, we will note down following metrics by running this command in MySQL:

> *SHOW ENGINE INNODB STATUS*;

The above command will display the following statistics collected so far in the machine.

- Pages read
- Pages written
- Select rows executed
- Insert rows executed
- Update rows executed
- Delete rows executed

Above command is run before and after running our benchmark. By finding the value differences between these two runs, we have calculated the average units per second for the following metrics:

- selects/sec.
- inserts/sec.
- updates/sec.
- deletes/sec.
- Innodb reads/sec.
- Innodb writes/sec

E. *MACHINE THROUGJPUT*

Machine throughput can give us an insight about performance of the machine while we are running the benchmark. We have followed the below steps for collecting these metrics.

a. Flush the cache
   >sudo sh -c "echo 1 > /proc/sys/vm/drop_caches"

b. Collect CPU metrics: Below command will give us about the CPU information with an interval of 10 seconds each.
   > *mpstat -d 10*

c. Collect Memory metrics: Below command will give us about the Memory information with an interval of 10 seconds each.
   > *vmstat -d 10*

d. Collect Disk metrics: Below command will give us about the disk information with an interval of 10 seconds each.
   > *iostat -d 10*

Hence the following statistics are collected as a part of the Machine throughput metric:

- CPU Utilization
- Memory Utilization
- Transfer per second
- Disk reads/second
- Disk writes/second

VII. EVALUATION

In this section, we will evaluate the metrics that we have collected and how each cloud service provider has behaved for each load and instance type.

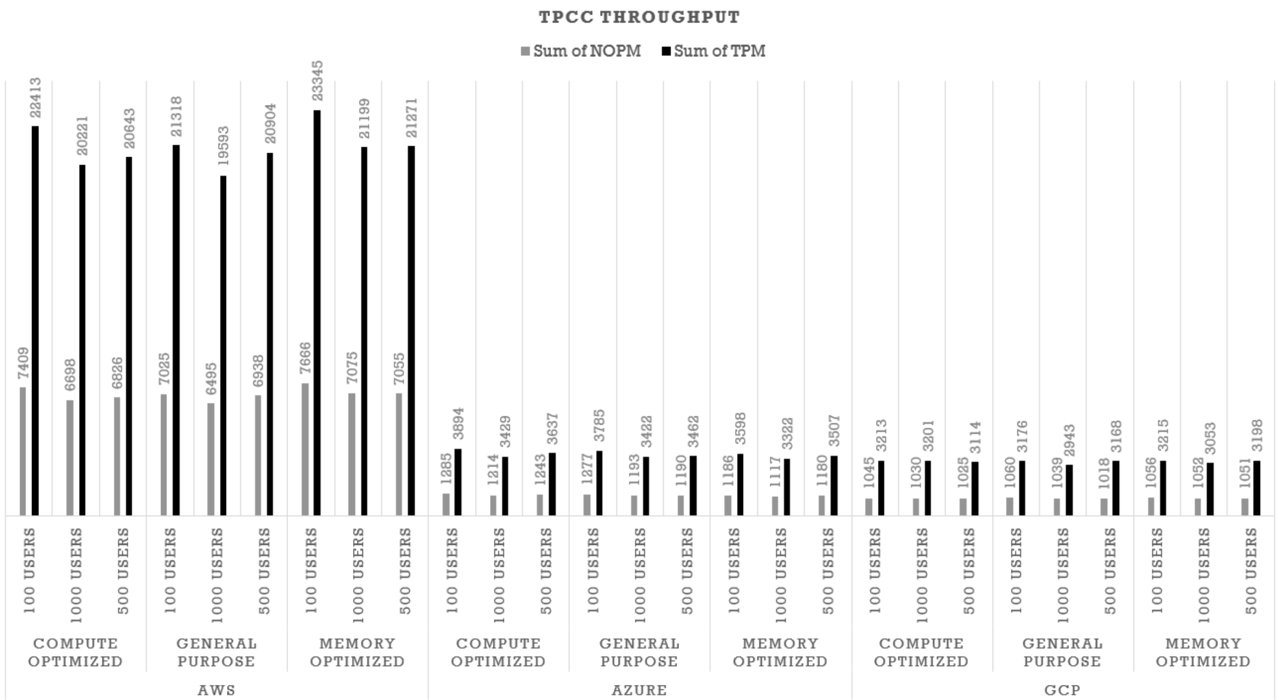

Figure 3: TPC-C Throughput

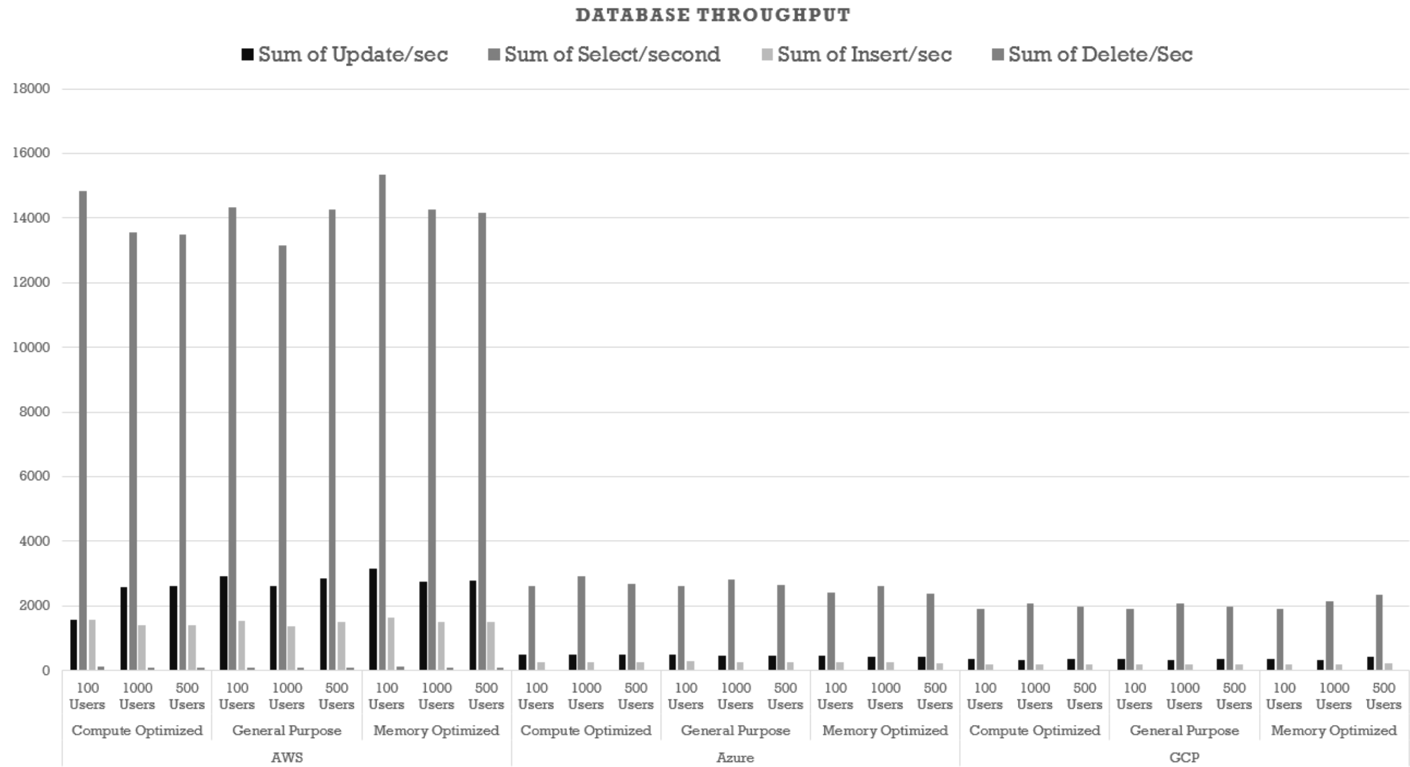

Figure 4: Database Throughput

A. TPC-C THROUGHPUT

AWS has an average 22,000 transactions per minute and 7000 orders per minute irrespective of the instance type that we have used. Where as Azure clocked in around 3500 transactions per minute & 1250 orders per minute where GCP performed least of all with only 3150 transactions per minute and 1050 orders minute. AWS clearly outperformed Azure and GCP with respect to TPC-C throughput. It is observed from the **Figure 3** in all the cloud providers that TPM and NOPM have decreased with the increase of number of users. Increasing the Memory or CPU from the general-purpose instance doesn't have any impact on the TPM and NOPM.

B. DATABASE THROUGHPUT

We have split Database throughput into two figures as mentioned below. **Figure 4** is the graph plotted for statements per second whereas **Figure 5** is the graph plotted for Innodb reads/second and Innodb writes/second. AWS has recorded over 14000 selects per sec, 1500 insert per sec, 2650 updates per sec and 97 deletes per second. Azure

has recorded over 2630 selects per sec, 260 inserts per sec, 17 deletes per sec. GCP has recorded over 2060 selects per sec, 198 inserts per sec, 360 updates per sec and 13 deletes per sec. Clearly AWS is the winner performing over 5 times as that of Azure and GCP.

It is also observed that Azure has slightly better than (20 – 30%) GCP when it comes to completing the number of statements per sec. Note that we don't really have any information about the underlying hardware specs by the three cloud providers. However, based on the information provided in their website AWS, Azure and GCP used standard SSDs with around 3000 IOPS. But still AWS outperformed significantly.

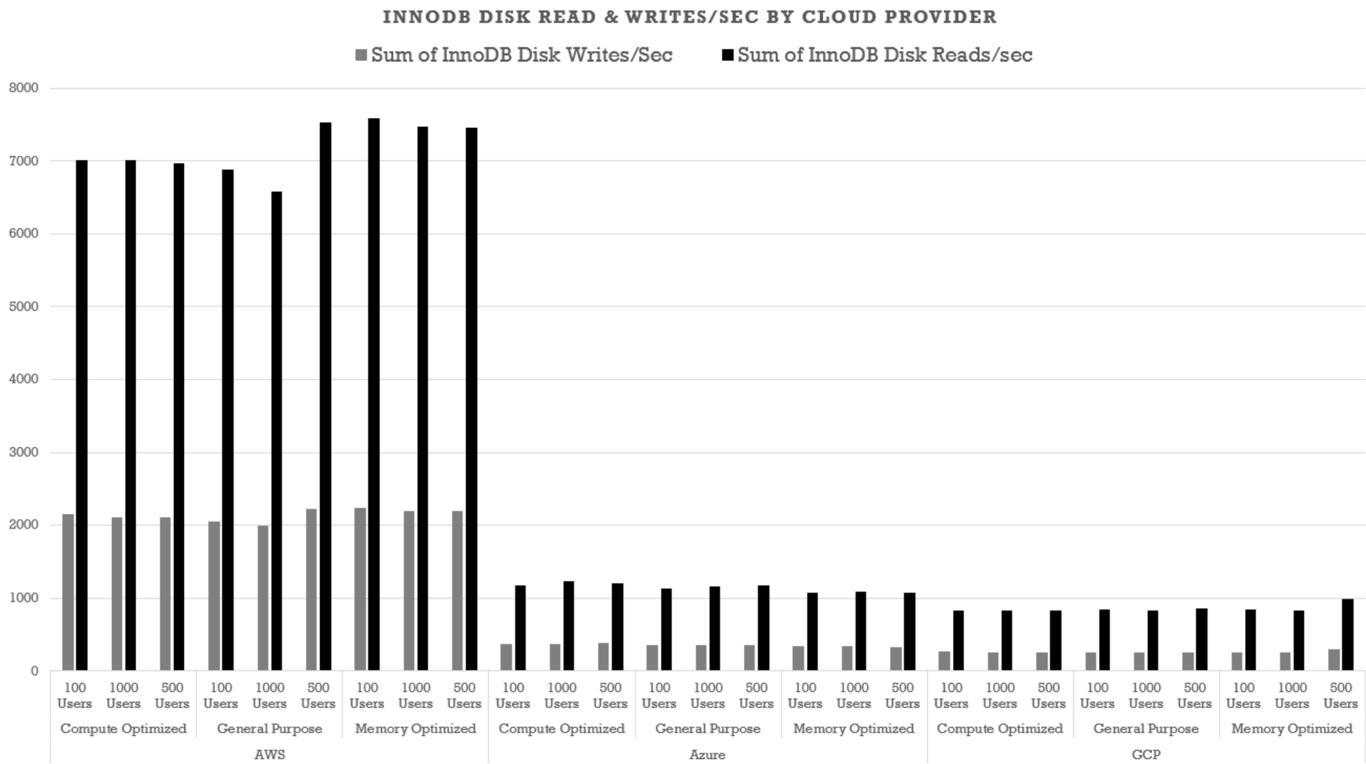

Figure 5: InnoDB reads/sec and writes/sec

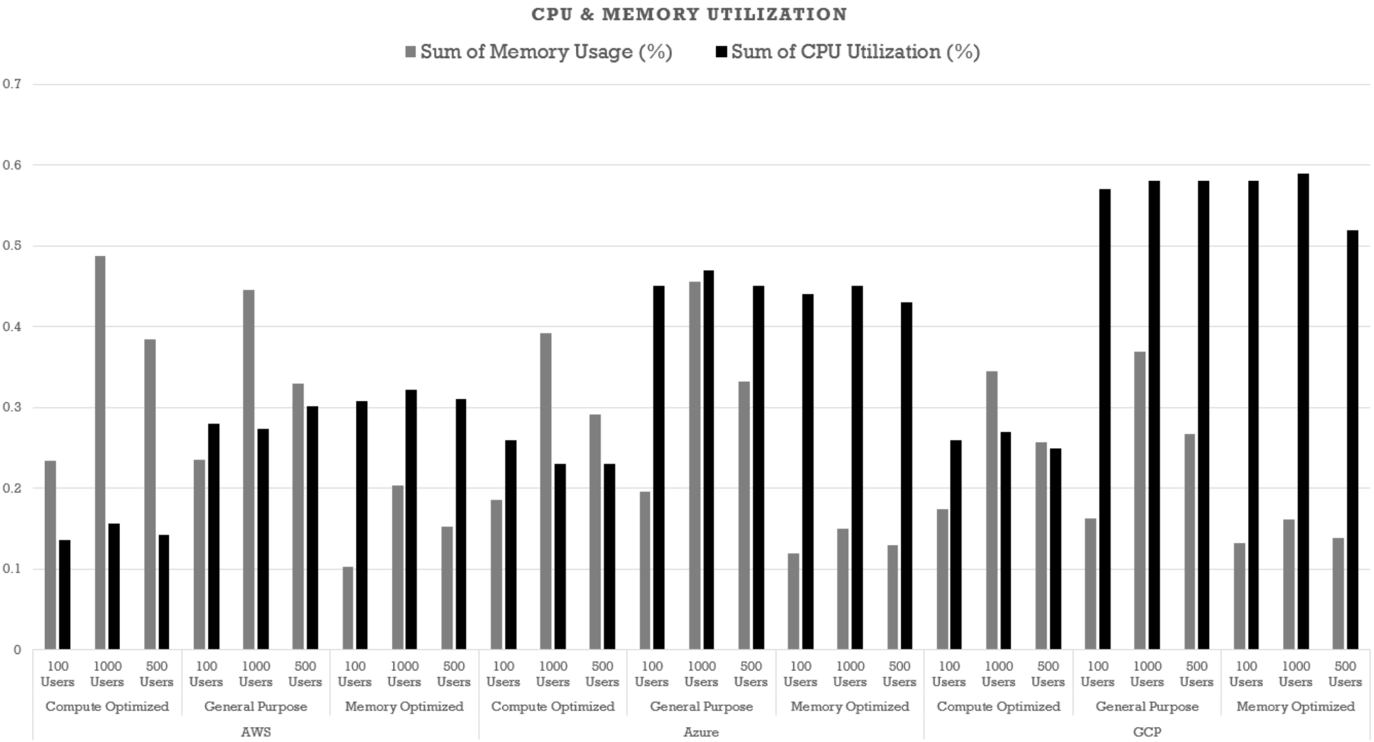

Figure 6: CPU & Memory Utilization

Coming to the innodb engine metrics, Innodb is the underlying MySQL storage engine. Hence Innodb clearly depicts the performance of the I/O operations by the MySQL Server. In this project we have collected Innodb disk reads and write per second and plotted a graph as shown in the below **Figure 5.**AWS recorded around 7160 reads per second, 2141 writes per second. Azure has recorded around 1150 reads per second and 350 writes per second. GCP has recorded 865 reads per second and 265 writes per second. It is observed that AWS has clearly outperformed Azure and GCP in terms of number of reads and write per second by times. Whereas Azure has slightly better than (30%) GCP. GCP performed the worst even though they have similar configuration and specs.

C. MACHINE THROUGHPUT

In this section, we evaluated CPU, Memory and Disk metrics as shown in the **Figure 6 & 7.** AWS has recorded around 25% average CPU Utilization, 28.63% memory utilization. Azure has recorded over 34% average CPU Utilization and 30% average memory utilization. GCP has recorded over 57% average CPU Utilization, 20% average memory utilization. It is observed that with the increase of number of users' memory utilization has been increased in cloud providers. However, this pattern was not found with respect to the CPU Utilization. CPU Utilization remained the same with increase of users. GCP recorded has highest average CPU Utilization where Azure which is slightly above AWS has highest average Memory Utilization. Coming to the disk writes/second once again AWS & Azure has outperformed GCP so let's go into the details. AWS has recorded over 71500 disk writes/sec, 2230 disk reads/sec. Azure has recorded over 72000 disk writes/sec, 7400 reads/sec. Whereas GCP has recorded over 9000 disk writes/sec and 2290 disk reads/sec. It is clearly observed that both AWS, Azure has performed 8 times better

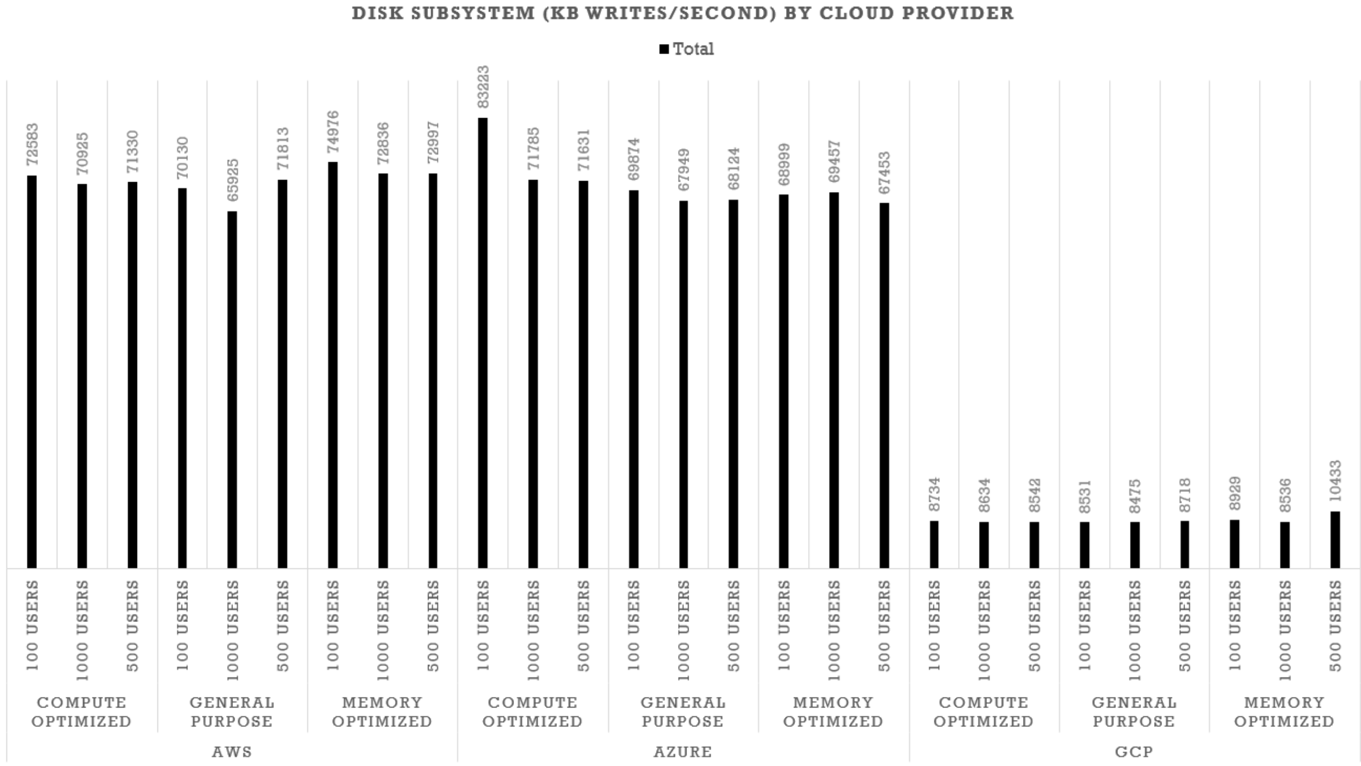

Figure 7.a: Disk writes/second

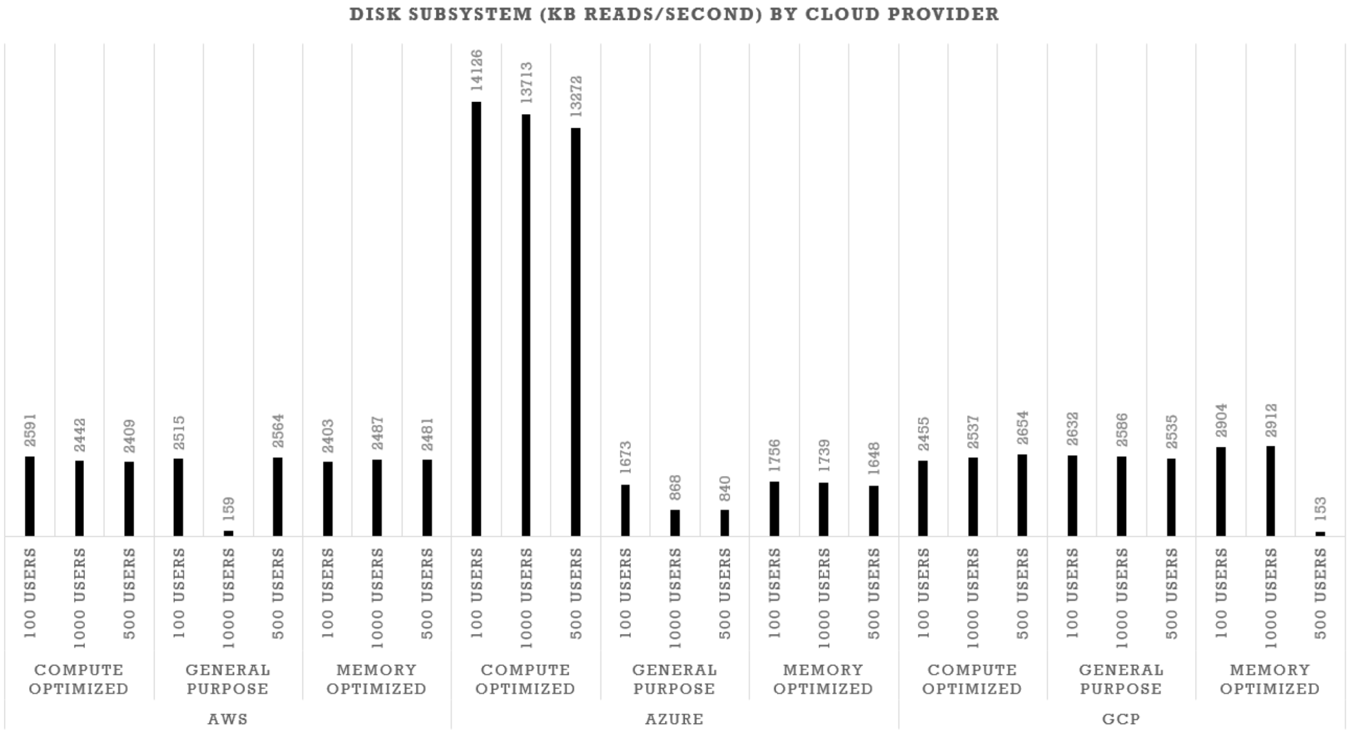

Figure 7.a: Disk writes/second

than in terms of disk writes/sec. Azure has performed 3 times better than AWS and GCP with respect to number of disk reads/sec. One of the reasons for this anomaly is Azure has recorded a very high amount of disk reads when running on Compute Optimized instance. In computer technology, transfers per second and its more common secondary terms gigatransfers per second (abbreviated as GT/s) and megatransfers per second (MT/s) are informal language that refer to the number of operations transferring data that occur in each second in some given data-transfer channel. It is also known as sample rate, i.e. the number of data samples captured per second, each sample normally occurring at the clock edge. Coming to the disk transfers/second as shown in the Figure 10, AWS and Azure performed better than GCP. AWS has recorded over 3300 transfers per second. Azure has recorded over 3400 transfers/sec, 7400 reads/sec. Whereas GCP has recorded over 450 transfers per second.
It is clearly observed that both AWS, Azure has performed 8 times better than GCP in terms of disk Transfer/sec.

VIII. CONCLUSION

In this paper, we have performed the performance evaluation of various instance types both in terms of CPU, Memory Utilization IOPS per second in the top three cloud providers, AWS, Azure and GCP. We have used TPC-C, which is the industry standard OLTP benchmark, to conduct the performance evaluation and used a load-testing tool called HammerDB. We have recorded Innodb metrics provided by MySQL storage engine Innodb. We used *sysstat* package to collect machine related metrics, and HammerDB to collect TPC-C throughput. It is observed that AWS outperformed both Azure and GCP in most of the areas with multiple folds. Though instances from cloud providers share the same configurations and specifications, we have seen this anomaly clearly especially in GCP. As a future work we want to implement in Cluster Environment with Master and Slave setup. with 100 – 200 warehouses. Each benchmark will yield 50 – 75 GB of data and will more concrete metrics to evaluate more datapoints. We also want to compare the performance of VM Instances against services provided by Cloud Providers (AWS Relational Database Service vs EC2 Instance).

IX. ACKNOWLEDGEMENT

This work is supported by the National Science Foundation under grant DGE-1915780.

X. REFERENCES

[1] TPC-C, Available: http://www.tpc.org/tpcc/

[2] Gartner Says Worldwide IaaS Public Cloud Services Market Grew 31.3% in 2018, Available: https://www.gartner.com/en/newsroom/press-releases/2019-07-29-gartner-says-worldwide-iaas-public-cloud-services-market-grew-31point3-percent-in-2018

[3] We Spent a Bunch of Money on AWS And All We Got Was a Bunch of Experience and Some Great Benchmark Results, Available: https://www.memsql.com/blog/memsql-tpc-benchmarks/ , Accessed: 2019-09-30

[4] A.Azarudeen, N.Ganesh, R.Dinesh, R.Ramakrishnan, “Secure Storage using TPC-C in Cloud.” 2015 IJSRSET

[5] J. Schad, J. Dittrich, and J.-A. Quiane-Ruiz, “Runtime measurements in ´ the cloud: Observing, analyzing, and reducing variance,” Proc. VLDB Endow., vol. 3, no. 1-2, pp. 460–471, Sep. 2010.

[6] 179. [10] E. Folkerts, A. Alexandrov, K. Sachs, A. Iosup, V. Markl, and C. Tosun, “Benchmarking in the cloud: What it should, can, and cannot be,” in Selected Topics in Performance Evaluation and Benchmarking, ser. Lecture Notes in Computer

Science. Springer Berlin Heidelberg, 2013, vol. 7755, pp. 173–188.

[7] Z. Hill, J. Li, M. Mao, A. Ruiz-Alvarez, and M. Humphrey, “Early observations on the performance of Windows Azure,” in Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing, ser. HPDC ’10, 2010, pp. 367–376.

[8] What is AWS, Available: https://aws.amazon.com/what-is-aws/

[9] What is Azure. Available - https://azure.microsoft.com/en-us/overview/what-is-azure/

[10] GCP Overview, Available - https://cloud.google.com/docs/overview /

[11] Overview of the TPC-C Benchmark, Available: http://www.tpc.org/tpcc/detail.asp

[12] HammerDB Avout, Available - https://www.hammerdb.com/about.html