# TmaxSoft OpenFrame and Microsoft Azure

## Mainframe Migration Sizing Study

## Table of Contents



TmaxSoft

Microsoft Azure

# Executive Summary

TmaxSoft and Microsoft recently completed a sizing study using the TmaxSoft mainframe modernization solution, OpenFrame, running on Microsoft Azure Infrastructure as a Service. TmaxSoft ran the CICS COBOL Reference application (zRef) workload, which was previously run on an IBM Z mainframe.

This paper presents the findings of the study. Based on these findings, we were able to draw the following conclusions:

- Under the same workload conditions, OpenFrame on Azure performed better than the z/OS and System z infrastructure and at a much lower cost.
- OpenFrame can deliver the same or better levels of service in terms of performance, batch/online transaction response time, and improving the overall SLAs of enterprise applications.
- OpenFrame can deliver the equivalent of several hundred (100s) to many thousands (1000s) of MIPS capacity using Azure infrastructure.
- This study demonstrated OpenFrame scale-up option to handle batch and Online workload. OpenFrame scale-out option will be demonstrated in future study.
- The cost of OpenFrame on Azure is dramatically lower than the cost of the equivalent z/OS and System z infrastructure.
- OpenFrame on Azure demonstrated better performance and scalability than System z infrastructure.

# OpenFrame: The TmaxSoft Mainframe Migration Solution

OpenFrame enables mainframe applications, resources and data to be migrated to a less expensive, high performance cloud environment while reducing TCO and minimizing risks, all in very short timeframes. Existing applications and data can be reused, which significantly decreases the cost, effort, and time of a conversion while conserving important application logic. OpenFrame customers can migrate mission-critical applications where continuity with existing applications is desired but the side-effects of a complete redevelopment are not wanted.

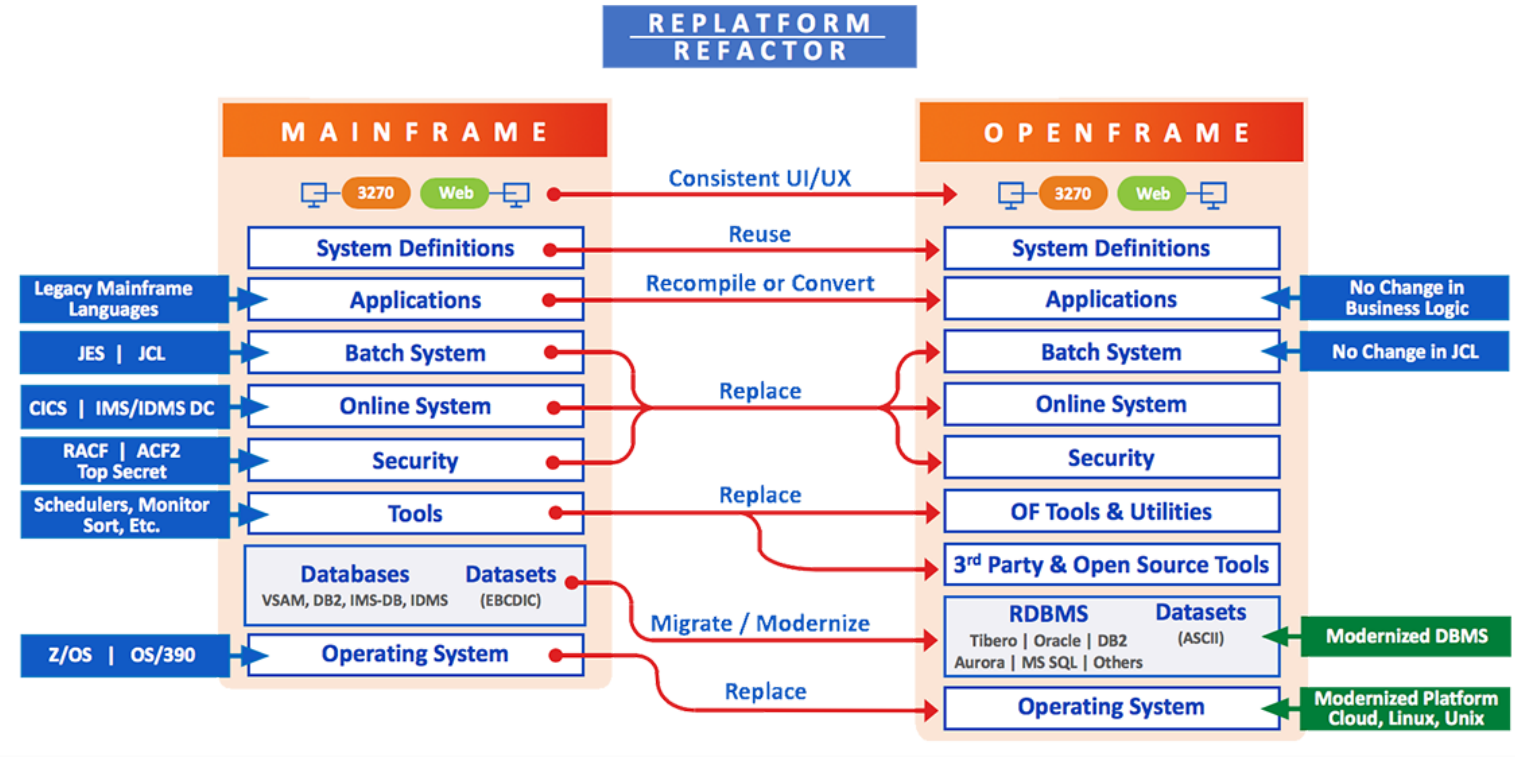


Figure 1: OpenFrame

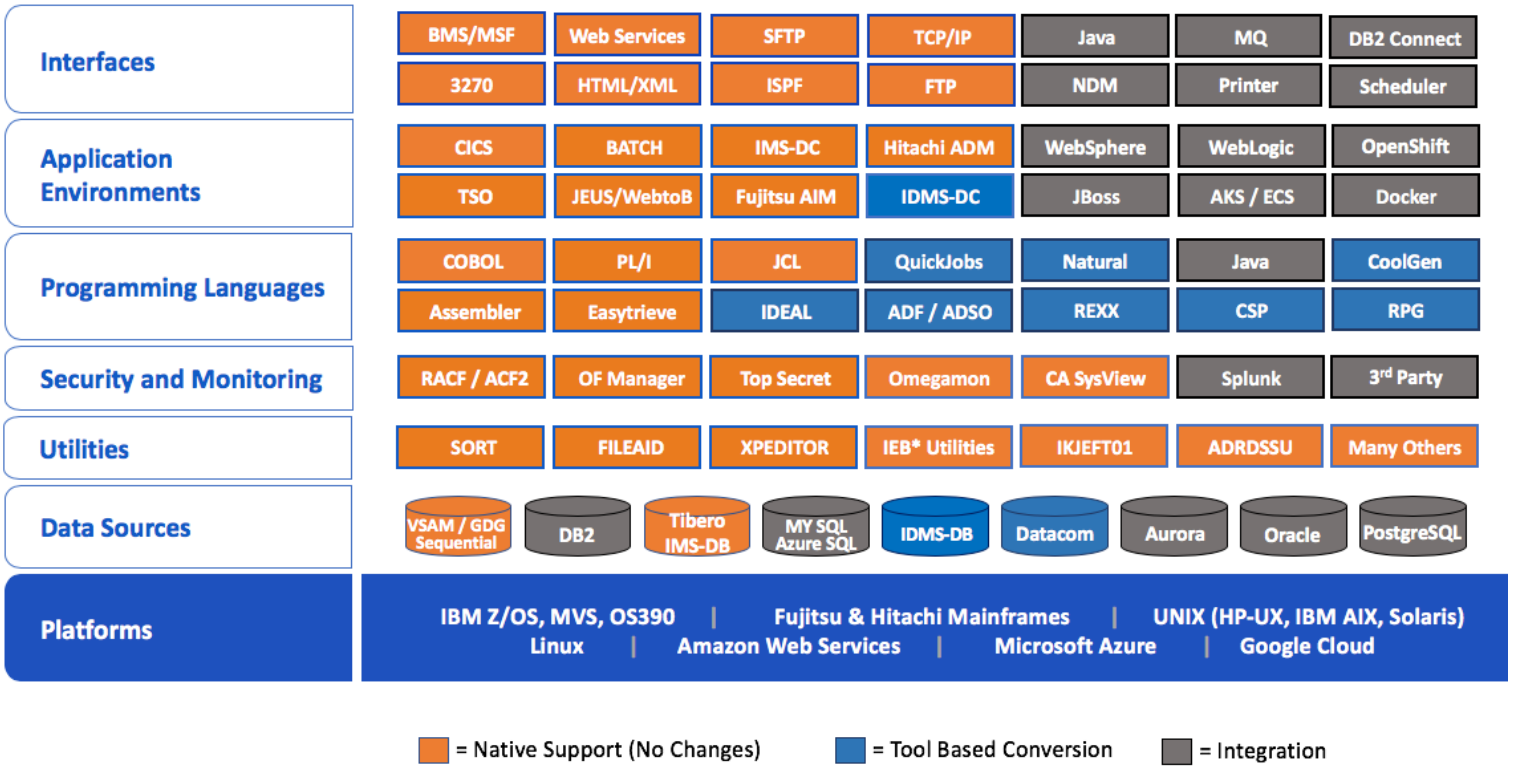


Figure 2: OpenFrame Supported Technologies

# OpenFrame on Azure Architecture

In this study OpenFrame on Azure is a single-node distributed architecture.

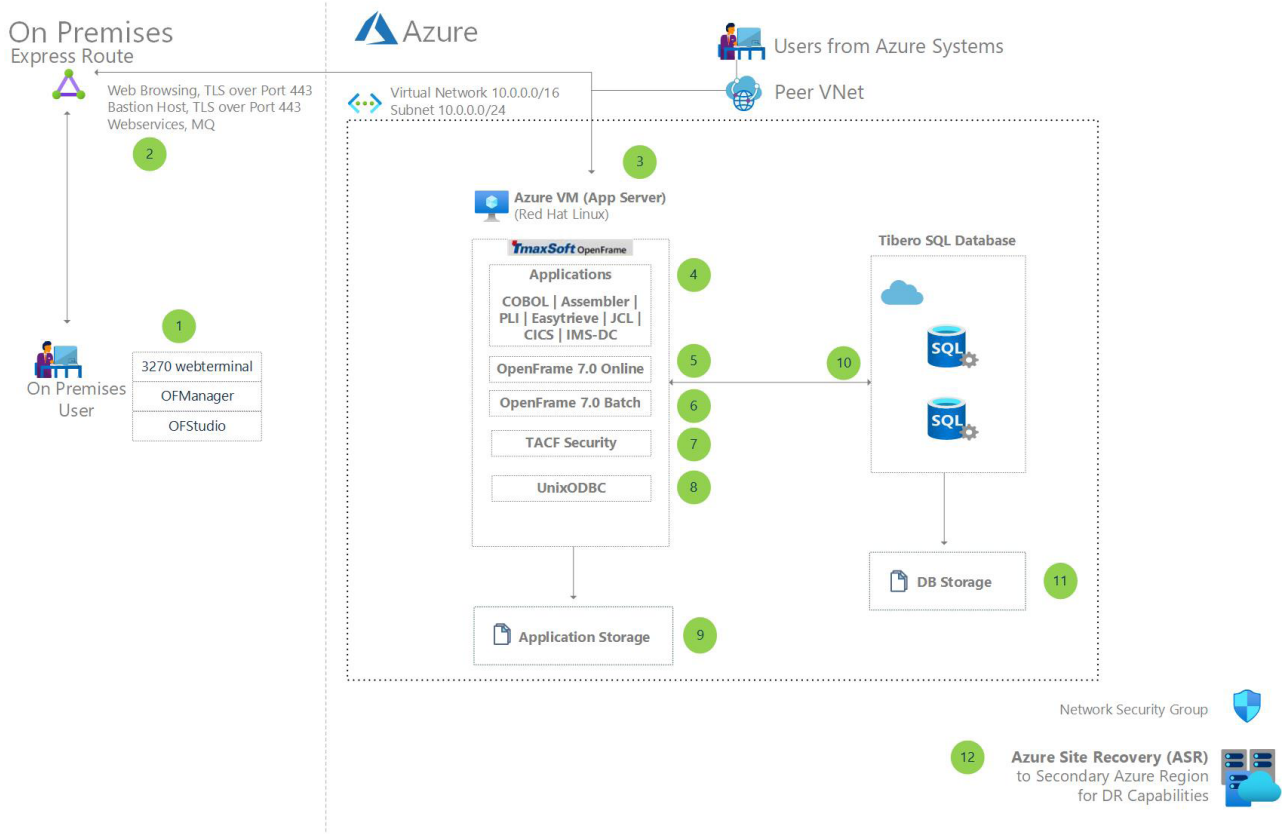


Figure 3: OpenFrame on Azure - Distributed Scale Up Architecture

In a future study OpenFrame on Azure multi-node architecture (scale out) will be presented.

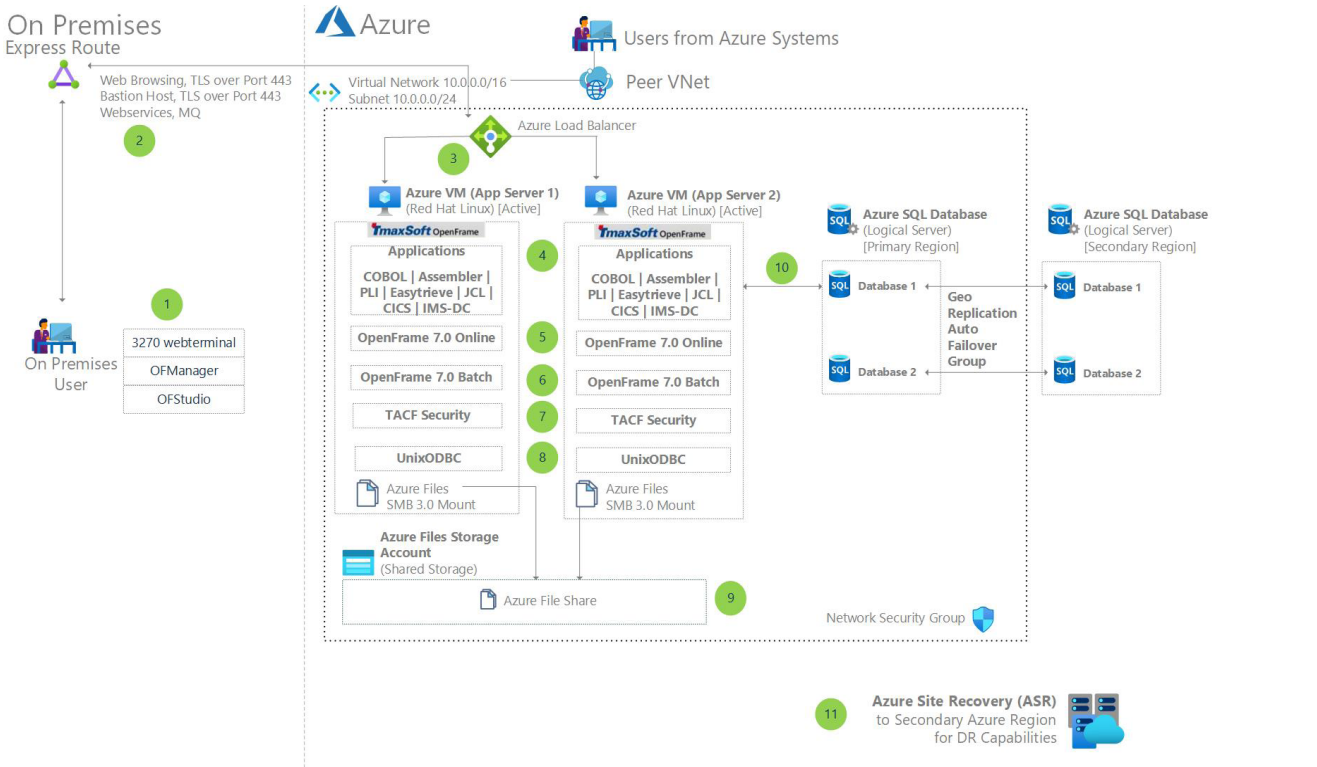


Figure 4: OpenFrame on Azure - Distributed Scale Out Architecture (High Availability and Scalability)

# zRef Study Overview

The z/OS Reference Application (zRef) used in this study was a representation of a z/OS batch and CICS application written in COBOL that uses DB2 and simulates a real financial application.

## Characteristics of zRef

zRef is based on the TPC-E benchmark, but it was not built to *"achieve the highest throughput possible"* benchmark. **zRef was programmed to be as much like a real-world "mainframe" as possible** so it could deliver a relative throughput comparison on various alternative platforms running typical mainframe applications.

For example, SQL calls to Db2 are simple and many, often using a position cursor and retrieving a single table row at a time. COBOL was programmed in the style of mainframe applications commonly found in production. Many mainframe programs are evolutions of prior work, written with coding habits that date back to the 1980s, and therefore they don't use high performance techniques.

## zRef Database

The zRef database consists of 33 tables with a wide range of columns, cardinality, and scaling properties.

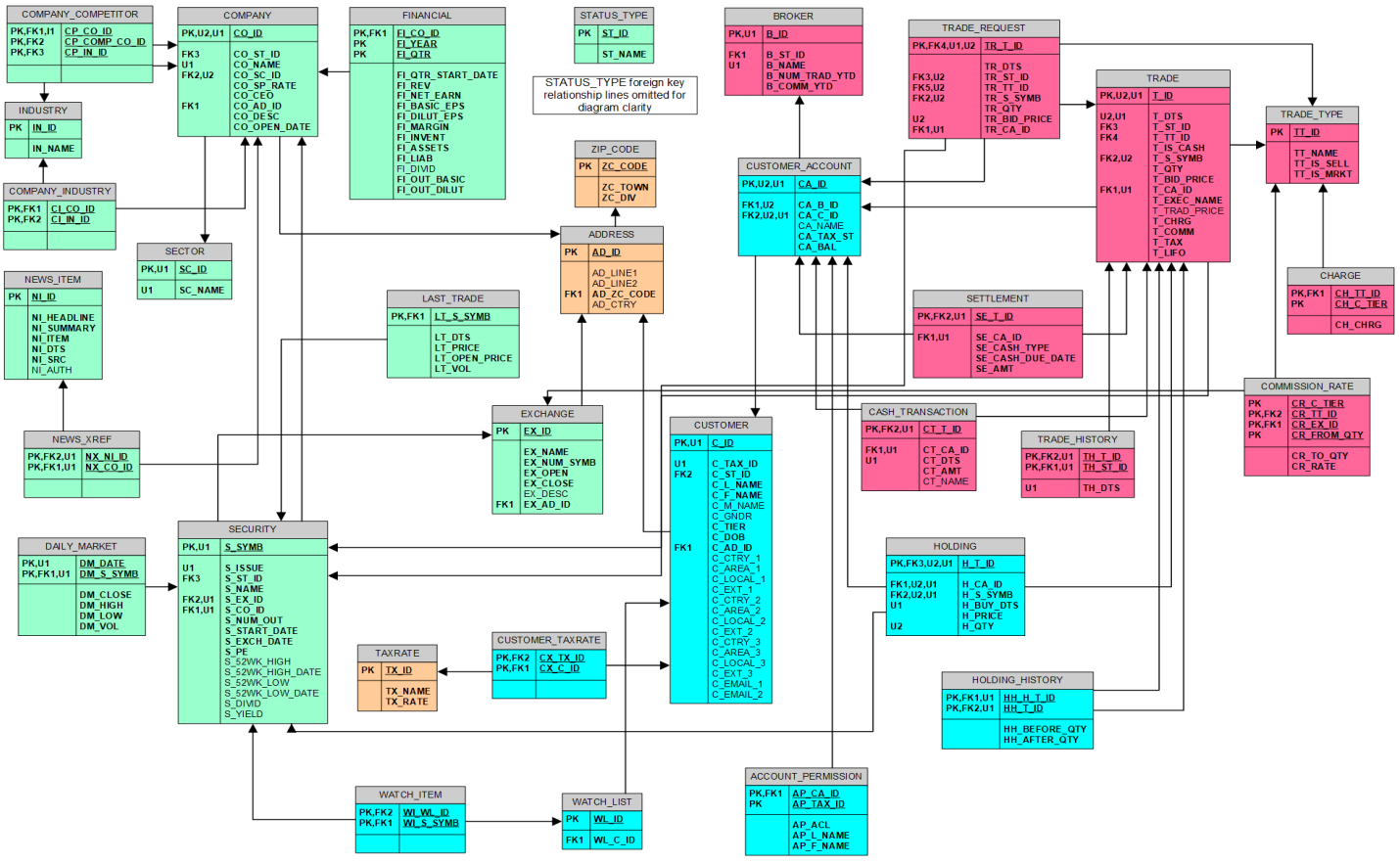


Figure 5: zRef Database from TPC Benchmark Specification

## zRef Batch

The zRef application includes two batch jobs that are complementary in function to the transactional (CICS) programs.

The Batch Reporting JOB **(zRefBR)** is read-only on the zRef database and represents typical periodic point-in-time status reporting (e.g., sales report, accounts status). This is the type of report run to produce weekly, monthly, and quarterly summaries. zRefBR starts with read-only extract STEPs from the zRef database, followed by STEPs of classic sort/merge sequential file processing.

The Transactional Batch Update **(zRefBU)** JOB runs the same business "transactions" as in the CICS transactional workload. In this case, however, it is driven from an input file and not a terminal. The zRef Batch Update job simulates, for example, the processing of business transactions that arrive via file transfer, such as FTP.

These zRef batch jobs can be run one at a time with multiple copies concurrent in parallel and also concurrent with zRef CICS transaction processing.

## zRef CICS

The CICS portion of zRef **(zRef CICS)** consists of 10 CICS transactions processed by more than 30 executable CICS programs that read, update, and insert data into a DB2 database and optionally log audit information (write) to a VSAM ESDS file. These transactions differ in type and complexity and are submitted by "users," which is TN-3270 in the mainframe case.

The following diagram is an overview of the zRef application used in this study.

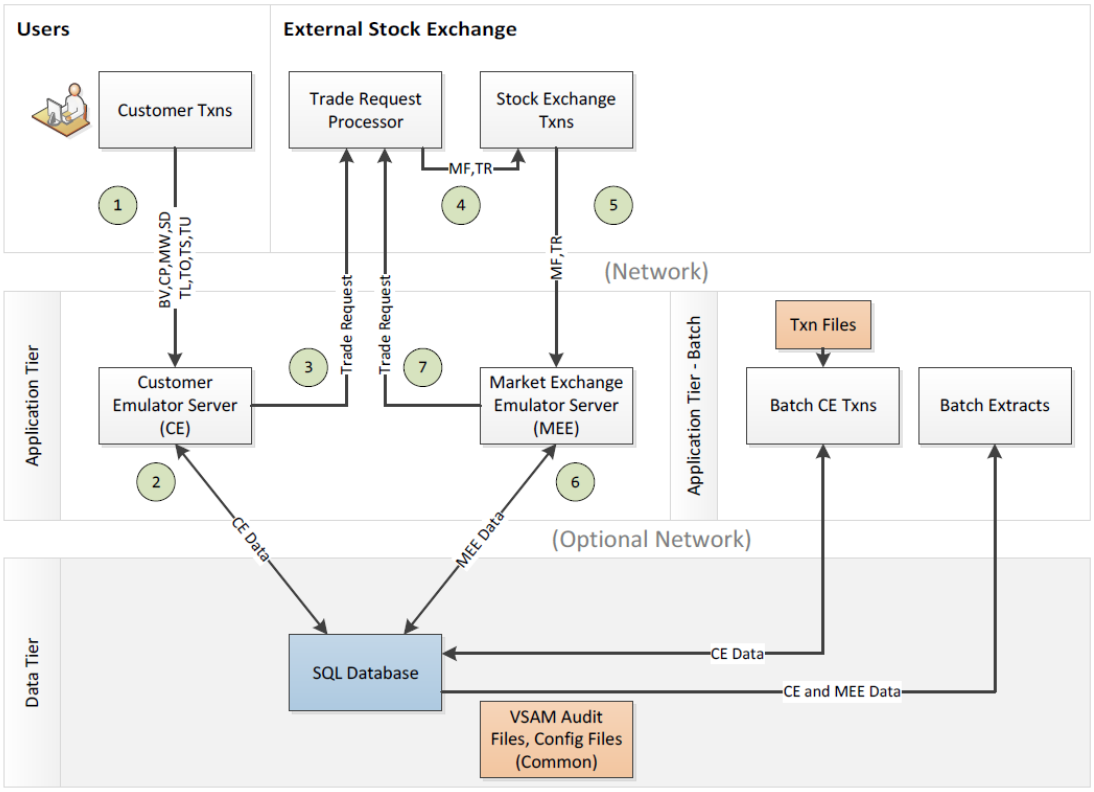


Figure 6: zRef Overview from zRef Manual

# OpenFrame Azure Mainframe Sizing Study

The study ran zRef under OpenFrame, hosted in Microsoft Azure. There was no change in code or business logic. The programs were simply compiled and deployed on OpenFrame platform using OpenFrame native tools, technologies, and language compilers. The database considered for this study is TmaxSoft's Tibero, a high-performance distributed RDBMS.

## zRef on OpenFrame run scenarios

These were the scenarios run for zRef on OpenFrame:

1. Batch workload:
   a. zRefBR: Records were extracted from the database to sequential file and then processed through multiple sort and merge steps. Multiple runs were conducted to get an average runtime and CPU consumption
   b. zRefBU: 30,000 business transactions were processed from the input transaction file. Multiple runs were made to obtain average runtime and CPU consumption
2. Batch concurrency:
   a. Many jobs were run concurrently to simulate mainframe batch workload
   b. Five zRefBU and 3 zRefBR running concurrently on OpenFrame
3. CICS workload
4. Batch workload that ran parallel to CICS online transaction processing

## The Systems Tested

**Study 1**

- OpenFrame Application (transaction) server ran on Microsoft Azure F-series VM (Standard F16s_v2 (16 vCPU, 32 GB memory))
- Tibero database server on E-series VM (E16s v3 (16 vCPU, 128 GB memory))
- Small distributed system tested with 2,000 simulated users workload

**Study 2**

- OpenFrame Application (transaction) server ran on Microsoft Azure F-series VM (Standard F32s_v2 (32 vCPU, 64 GB memory))
- Tibero database server on E-series VM (Standard E32s v3 (32 vCPU, 256 GB memory))
- Scaled up distributed system tested with 4,500 simulated users workload

**CICS Concurrent Load**

- Generated through py3270, a Python interface to x3270 terminal emulator for two zRef CICS regions.
- The load was submitted to OpenFrame application server from client machine - Standard B20ms (20 vCPU, 80 GB memory) Azure Linux machine.

**Additionally**

- Relational data services were provided by TmaxSoft's RDBMS Tibero 6.
- The OS was RedHat Linux 7.7 for application and database servers.
- Data was stored on an Azure Premium Persistent Disk.
- Application server storage was single 200GB (Azure premium SSD), 1,100 provisioned IOPS.
- Database server storage was single 1TB (Azure premium SSD), 7,500 provisioned IOPS.
- Accelerated networking was enabled for Azure servers.
- Azure insights was enabled for detailed server monitoring and data points.

## Batch Comparative Results

zRefBR is a batch reporting workload, and zRefBU makes the same set of complex updates to the zRefDB that are done by zRef CICS transaction programs but in an input file-driven batch process.

The standalone batch (zRefBR and zRefBU) results are shown in Figure 7.

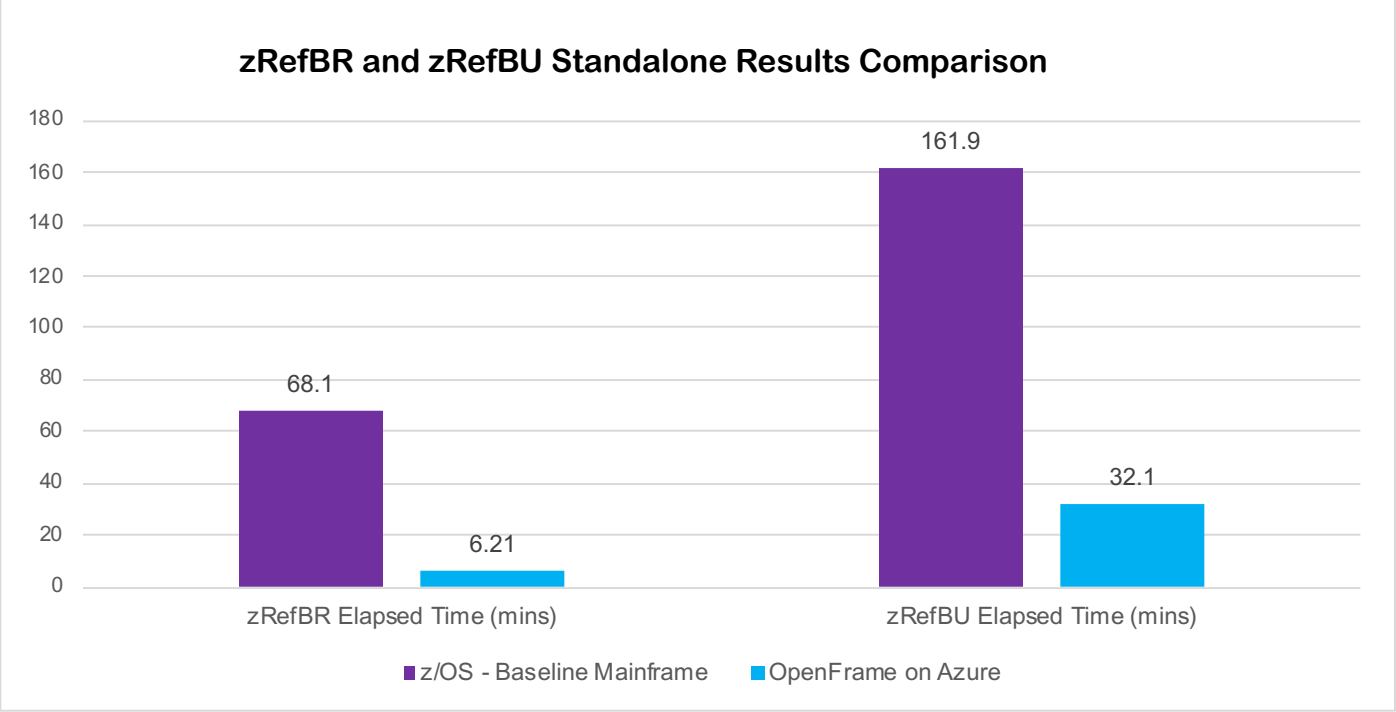


Figure 7: zRefBR and zRefBU Standalone Results

For batch concurrency, five zRefBU and three zRefBR jobs were run concurrently to simulate a typical mainframe batch workload. The system being tested ran these without any overhead, and all jobs finished successfully with some increase in elapsed time for zRefBR. The zRefBR elapsed time was approximately **13 mins**, and the zRefBU elapsed time was approximately **32 mins**. You can see the details in the following screen shot.

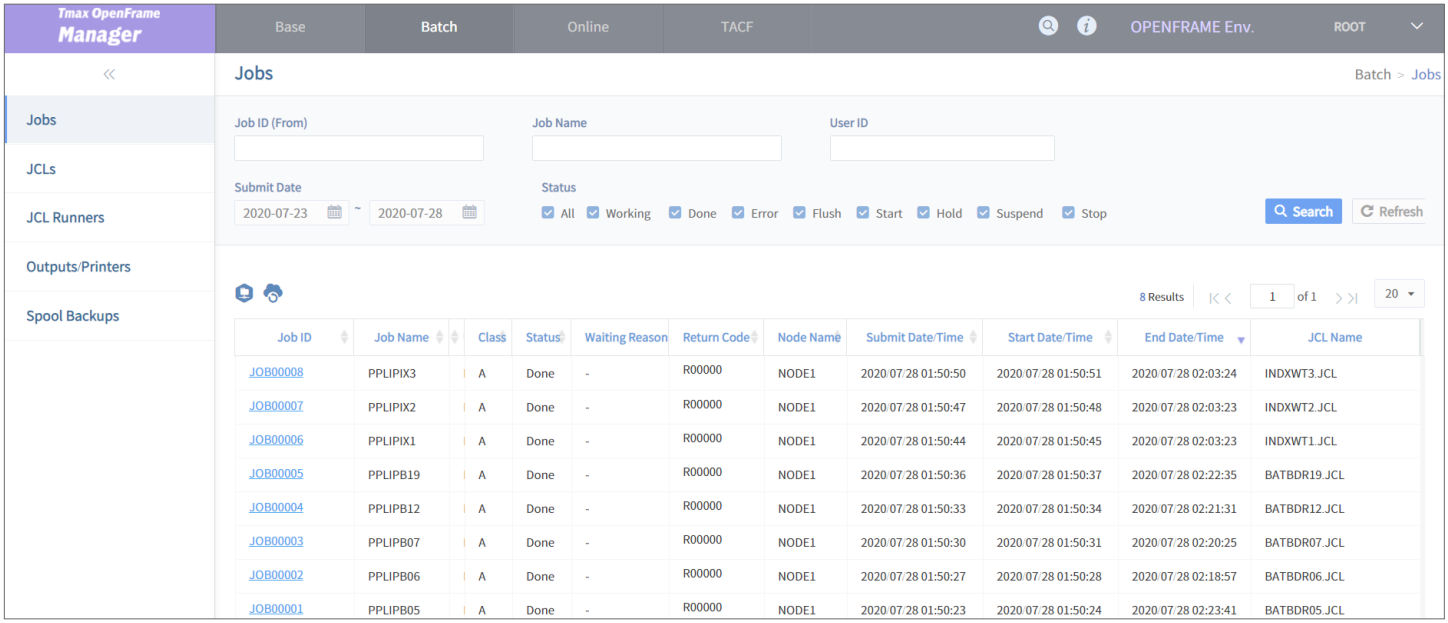


Figure 8: zRefBU and zRefBR Concurrency

# CICS Workload: 1,672 Transactions Per Second (TPS) Throughput

Small distributed system tested with 2,000 simulated users workload delivered 13,676 MIPS equivalent throughput.

## The System Tested

- OpenFrame Application (transaction) server ran on Microsoft Azure F-series VM (Standard F16s_v2 (16 vCPU, 32 GB memory))
- Tibero database server on E-series VM (E16s v3 (16 vCPU, 128 GB memory)

## Workload Test Results

- **2,000** simulated users workload submitted for one hour resulted total 6,018,830 online transactions during this time
- Achieved **1,672 zRef TPS** (transactions per second) with avg CPU utilization of 79% and 92% for Application and Database server
- Average online transactions response time recorded was **207 ms**
- 1,672 zRef TPS represents 13,676 MIPS equivalent throughput delivered by this test

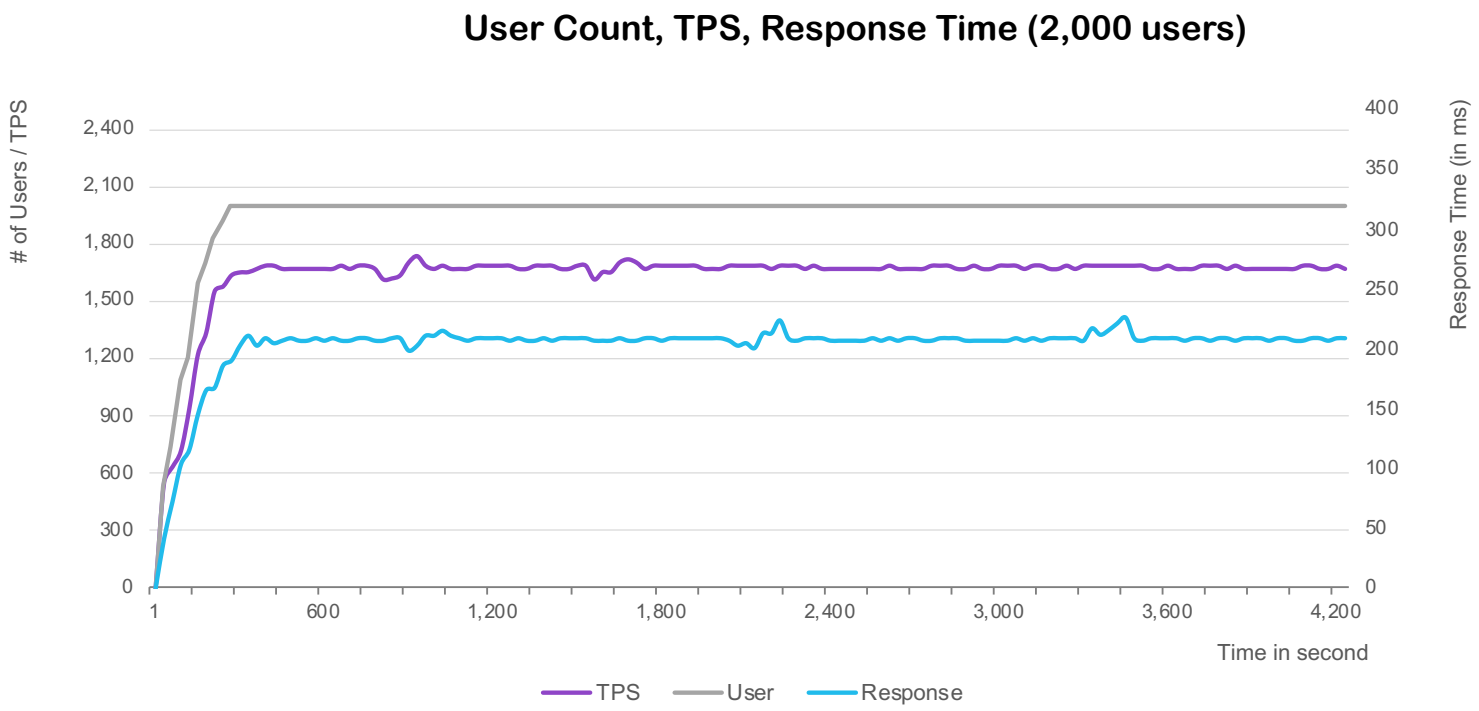


Figure 9: Users, TPS, and Response Time for 13,676 MIPS

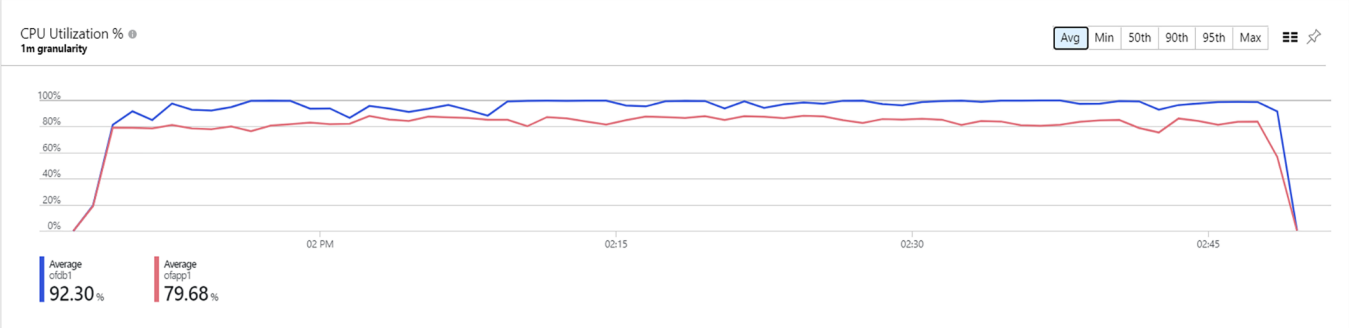


Figure 10: CPU Utilization Database and Application Servers

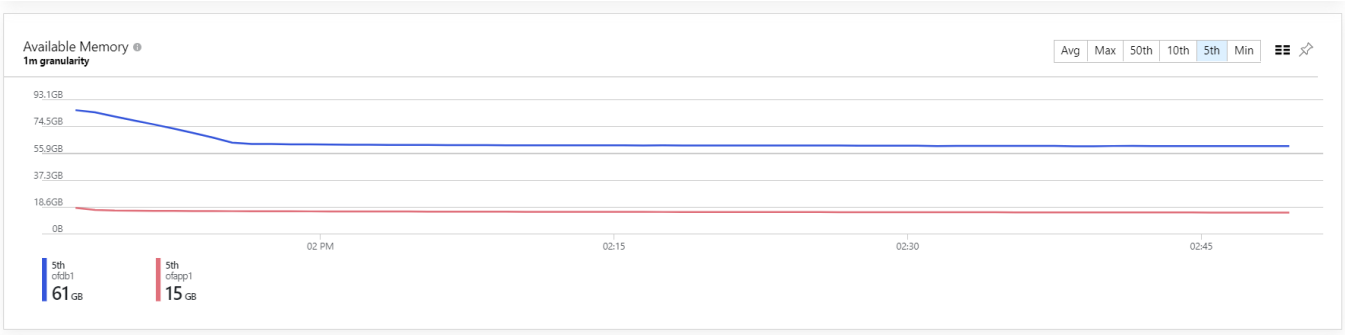


Figure 11: Available Memory Database and Application Servers

## Cost

The annual Azure infrastructure cost of the tested system would be approximately $17,641 which includes:

- Azure Linux VMs (Application and Database servers)
- Azure Premium Storage
- Virtual Network and VPN Gateway
- Load Balancer
- Static IP Addresses
- Azure Backup
- Azure Site Recovery (DR)
- Azure Monitor
- Azure Support

**The Azure costs were based on one year of reserved pricing and estimated using the Azure Pricing Calculator (https://azure.microsoft.com/pricing/calculator/).**

- **The OpenFrame product license depends on the current mainframe technology stack used**
- **Complete the OpenFrame questionnaire in order for us to generate an accurate quote (https://tinyurl.com/yaka5myt)**

# CICS Workload – 2,600 Transactions Per Second (TPS) Throughput

Scaled-up distributed system tested with 4,500 simulated users workload delivered 21,268 MIPS equivalent throughput.

## The System Tested

- OpenFrame Application (transaction) server ran on Microsoft Azure F-series VM (Standard F32s_v2 (32 vCPU, 64 GB memory))
- Tibero database server on E-series VM (Standard E32s v3 (32 vCPU, 256 GB memory))

## Workload Test Reults

- 4,500 simulated users workload submitted for one hour and 10 mins resulted in a total of 10,921,221 online transactions during this time
- Achieved **2,600 zRef TPS** (transactions per second) with average CPU utilization of 76% for the application server and 84% for the database server
- Average online transactions response time recorded was **242** ms
- 2,600 zRef TPS represents 21,268 MIPS equivalent throughput delivered by this test

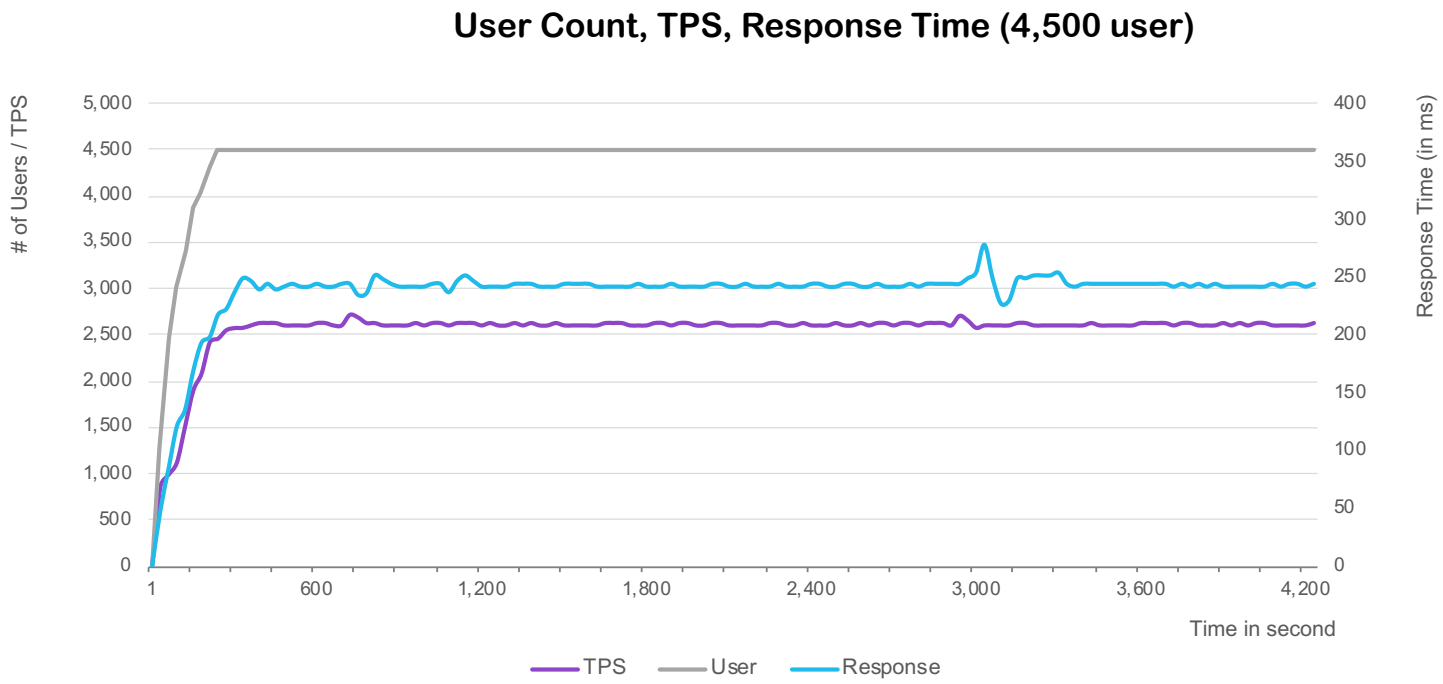


Figure 12: Users, TPS, and Response Time for 21,268 MIPS

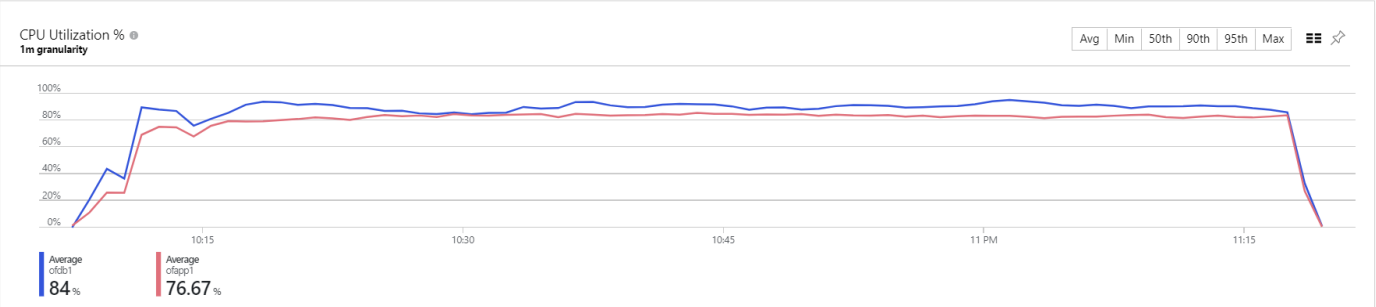


Figure 13: CPU Utilization Database and Application Servers


Figure 14: Available Memory Database and Application Servers

## Cost

The annual Azure infrastructure cost of tested system would be approximately $26,624 which includes:

- Azure Linux VMs (Application and Database servers)
- Azure Premium Storage
- Virtual Network and VPN gateway
- Load Balancer
- Static IP addresses
- Azure Backup
- Azure Site Recovery (DR)
- Azure Monitor
- Azure Support

**The Azure costs were based on one year of reserved pricing and estimated using the Azure Pricing Calculator (https://azure.microsoft.com/pricing/calculator/).**

- **The OpenFrame product license depends on the current mainframe technology stack used.**
- **Complete the OpenFrame questionnaire in order for us to generate an accurate quote (https://tinyurl.com/yaka5myt)**

# Batch and CICS Workload Running Parallel – 1,409 Transactions Per Second (TPS) Throughput

Multiple zRef Batch Reporting (zRefBR) jobs were run parallel to CICS workload to measure the batch effect on online transactions. Purpose is to simulate the real mainframe application scenario where online and batch workloads are running in parallel.

## The System Tested

- The OpenFrame Application (transaction) server ran on Microsoft Azure F-series VM (Standard F16s_v2 (16 vCPU, 32 GB memory))
- Tibero database server on E-series VM (E16s v3 (16 vCPU, 128 GB memory))

## Workload Test Results

- Three zRefBR jobs were submitted along with **1,500** simulated users' CICS workload
- CICS workload was sustained for one hour which resulted in a total of 5,074,330 online transactions during this time
- zRefBR jobs completed in 16 mins (i.e. 3 mins longer when compared to no CICS workload)
- Achieved 1,409 zRef TPS (transactions per second) with average CPU utilization of 68% for the application server and 72% for the database server
- The average online transactions response time recorded was **99 ms**
- 1,409 zRef TPS represents **11,525 MIPS equivalent throughput** delivered by this test

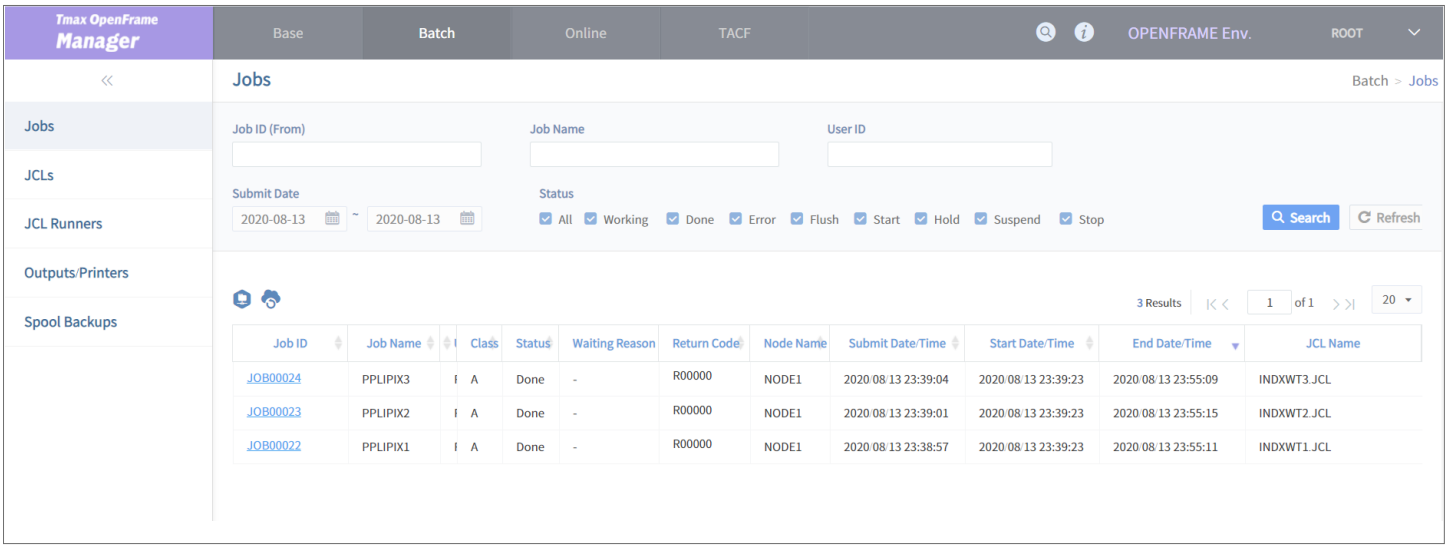


Figure 15: Batch and CICS Workload Running Parallel

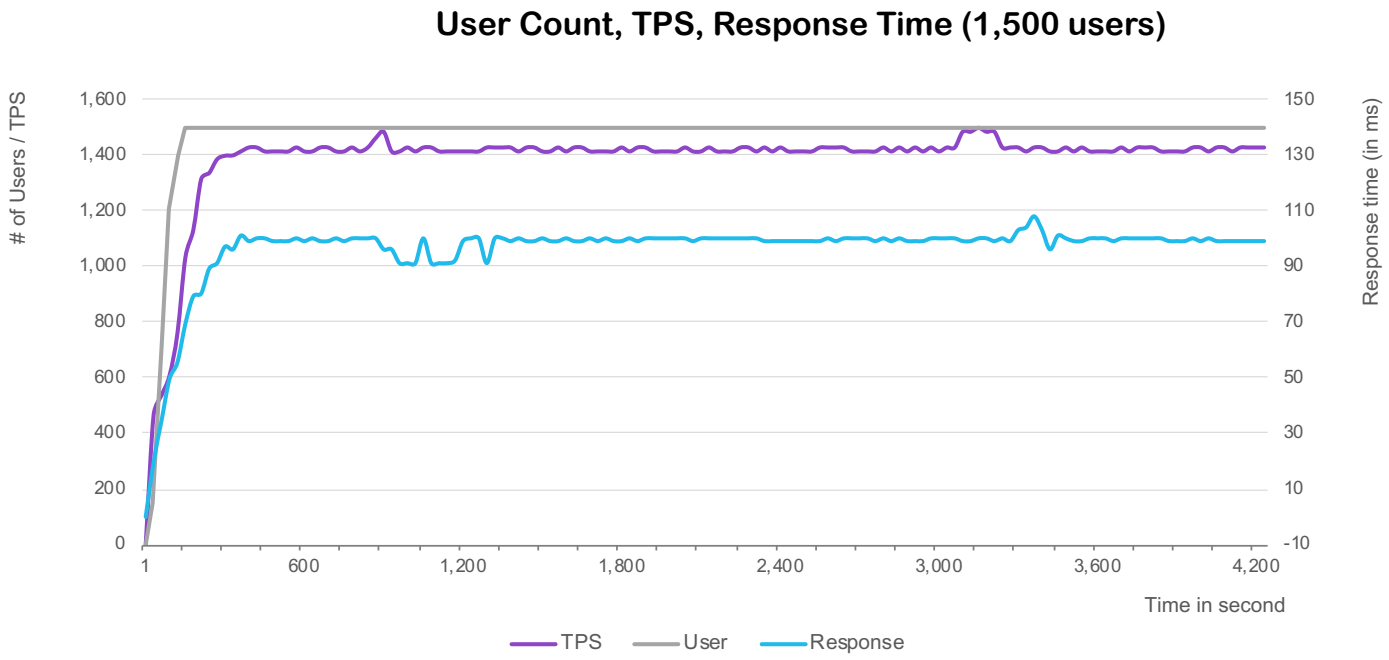


Figure 16: Users, TPS, Response time for 11,525 MIPS

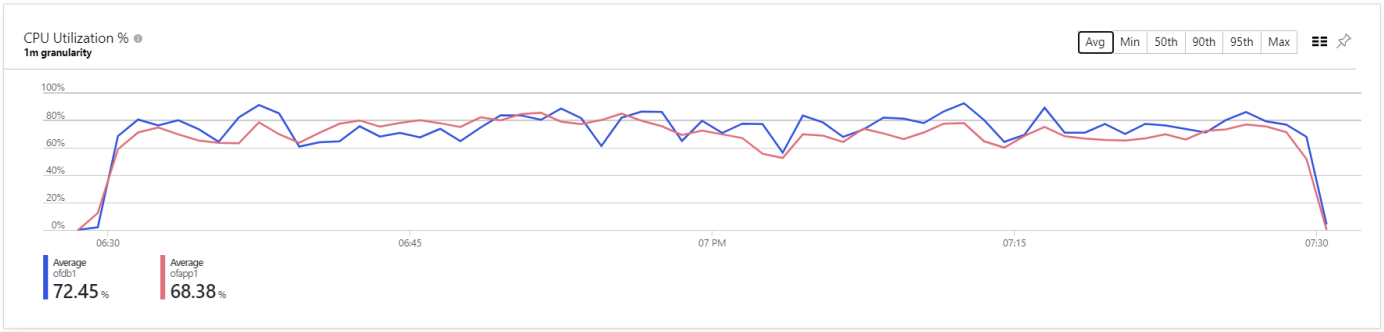


Figure 17: CPU Utilization Database and Application Servers

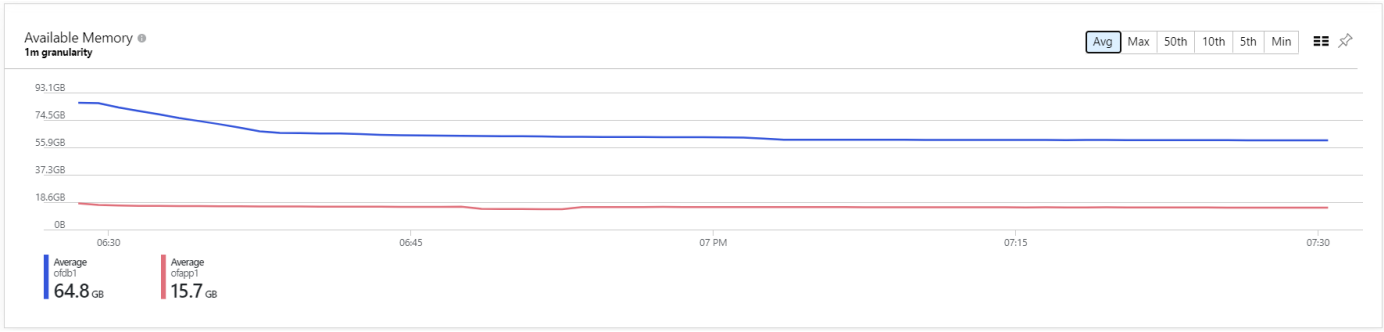


Figure 18: Available Memory Database and Application Servers

# Conclusion

The results of our studies are conclusive proof that OpenFrame on Azure can perform better than a IBM Z mainframe and at a much lower cost. When running on Azure, OpenFrame can deliver the equivalent of thousands of MIPS. In addition, this study demonstrated OpenFrame scale-up option to handle batch and online workload.

**Disclaimer:** Performance results may vary for each application. Sizing to be determined on a case-by-case basis.

TmaxSoft

For all office locations, visit **www.tmaxsoft.com/about/locations**