z/OS Connect Enterprise Edition V3 Performance Evaluation using IMS

IMS Performance Evaluation Team (IBM Silicon Valley Laboratory)
October 2018

Contributors: Jasdeep Singh, Kevin Hite and Richard Antikoll

Abstract

This paper presents the performance characteristics of IBM® z/OS® Connect Enterprise Edition (z/OS Connect EE) V3.0.10, released in June 2018. The IMS Performance Evaluation Team at IBM Silicon Valley Laboratory evaluated z/OS Connect EE Server V3.0.10 using IMS™ Version 15 (IMS 15) and Java 8.

The following evaluations include assessing the performance capability of z/OS Connect EE invoking RESTful API with multiple services and an IMS application calling out to a RESTful API deployed on z/OS Connect EE.

The paper also describes the measurement methodology, workload and application details used for all measurements. z/OS Connect EE V3.0.10 and IMS 15 offer a great combination of performance for IT systems by enabling them to expose their z/OS assets to fully participate in the new API economy for mobile and cloud applications.

Note: Before you use this information and the product it supports, read the information in "Notices" on page 53.

© Copyright International Business Machines Corporation 2018.
US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp

Note: Performance is based on measurements and projections using IMS benchmarks in a controlled environment. The results that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, the amount of zIIP capacity available during processing, and the workload processed. Therefore, results may vary significantly and no assurance can be given that an individual user will achieve results similar to those stated here. Results should be used for reference purposes only.

The test scenarios (hardware configuration and workloads) used in this document to generate performance data are not considered 'best performance case' scenarios. Performance may be better or worse depending on the hardware configuration, data set types and sizes, and the overall workload on the system.

The information contained in this document has not been submitted to any formal IBM test and is distributed on an "AS IS" basis without any warranty either expressed or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into their operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

The information in this paper provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g, zIIPs, zAAPs, and IFLs) ("SEs"). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the "Authorized Use Table for IBM Machines" provided at
http://www-01.ibm.com/support/docview.wss?uid=isg3T1025368. No other workload processing is authorized for execution on an SE. IBM offers SE at a lower price than General Purpose Engines/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

The information provided in this paper was obtained at the IBM Silicon Valley Laboratory and is intended for migration and capacity planning purposes.

1 Introduction

1.1 Background

The IBM z/OS Connect EE is an integrated solution that enables developers to merge business applications into today's growing mobile, cloud and hybrid cloud application ecosystems. z/OS Connect EE combines IBM and industry state of the art technologies to deliver a performant, intuitive solution for defining services and APIs to access your IMS assets using industry standard REST technology.

The following three components in z/OS Connect EE provide API solutions for IMS:

- The z/OS Connect EE Server
- The IMS Service Provider (IMS SP)
- z/OS Connect EE API toolkit

Each of these components integrate seamlessly to provide a fast and reliable experience for developers as they build applications for mobile and cloud use cases where speed to market is critical.

z/OS Connect EE V2.0 introduced the API mapping feature and delivers RESTful APIs as a discoverable, first-class resource with Swagger 2.0 descriptions. It includes a new API package artifact that encapsulates the RESTful API with necessary detail to invoke the underlying services in the z/OS subsystems such as IMS.

The API mapping adds an abstraction layer between the API consumer and the underlying z/OS assets, allowing in-line manipulation of requests such as the mapping of HTTP headers, pass-through, redaction or defaulting of Java Script Object Notation (JSON) fields, and rearranging the order of JSON fields and data.

z/OS Connect EE V3.0 provides support for calling RESTful APIs from a z/OS application that is written in COBOL or PL/I. The z/OS application can be an application running in CICS, IMS, or batch. The RESTful API that needs to be called through z/OS Connect EE must be described using Swagger 2.0 in JSON format.

1.2 Objective

This paper describes the performance characteristics of z/OS Connect EE for the following scenarios:

- Invoking API in provider scenario
- Invoking API in requester or callout scenario

The number of users were scaled for the scenarios listed above to examine how throughput, client response time and CPU cost (Service time or processing time) per transaction behaves.

The provider scenario was evaluated with the following sizes for input and output messages:

- 1KB Request and Response Message Size
- 8KB Request and Response Message Size
- 30KB Request and Response Message Size

The requester scenario was evaluated with the following sizes for input and output messages:

- 1KB Request and Response Message Size
- 8KB Request and Response Message Size
- 30KB Request and Response Message Size

2 Executive Overview

The z/OS Connect EE performance evaluations using IMS 15 demonstrate that z/OS Connect EE V3.0.10 is capable of showing performance scalability in terms of transaction rate and CPU efficiency as the number of concurrent users increase.

While performance in specific production environments will vary, results of IBM internal testing in a controlled laboratory environment revealed that the z/OS Connect EE V3.0.10 is capable of performing as described below:

API Provider Workload Performance:

- Invoking one API which further randomly invoked one of the 25 services deployed on z/OS Connect EE demonstrated a peak transaction rate of 21,000 transactions per second with 1KB message sizes.
- Invoking one API which further randomly invoked one of the 25 services deployed on z/OS Connect EE demonstrated a peak transaction rate of 8,900 transactions per second with 8KB message sizes.
- Invoking one API which further randomly invoked one of the 25 services deployed on z/OS Connect EE demonstrated a peak transaction rate of 3,500 transactions per second with 30KB message sizes.

API Requester Workload Performance:

- IMS application invoking a RESTful API on z/OS Connect EE demonstrated a peak transaction rate of 5,100 transactions per second with 1KB message sizes.
- IMS application invoking a RESTful API on z/OS Connect EE demonstrated a peak transaction rate of 3,800 transactions per second with 8KB message sizes.
- IMS application invoking a RESTful API on z/OS Connect EE demonstrated a peak transaction rate of 2,100 transactions per second with 30KB message sizes.

The following performance characteristics were observed in both provider and requester scenarios:

- Increasing the number of users showed little to no change in CPU cost per transaction.
- The CPU percent utilization increased linearly as the number of users increased.

- The transaction rate increased linearly as the number of users increased before stabilizing.

- z/OS Connect EE, a Java based product, processing is approximately 99% eligible for offloading onto zIIP processors.

3 Measurement Methodology

The performance evaluation cycle is shown in Figure 1 below, where a test environment is created or customized for a specific measurement, performance tests are run, and data is analyzed. All testing was done in an isolated and stable environment to produce consistent and repeatable performance measurement results.

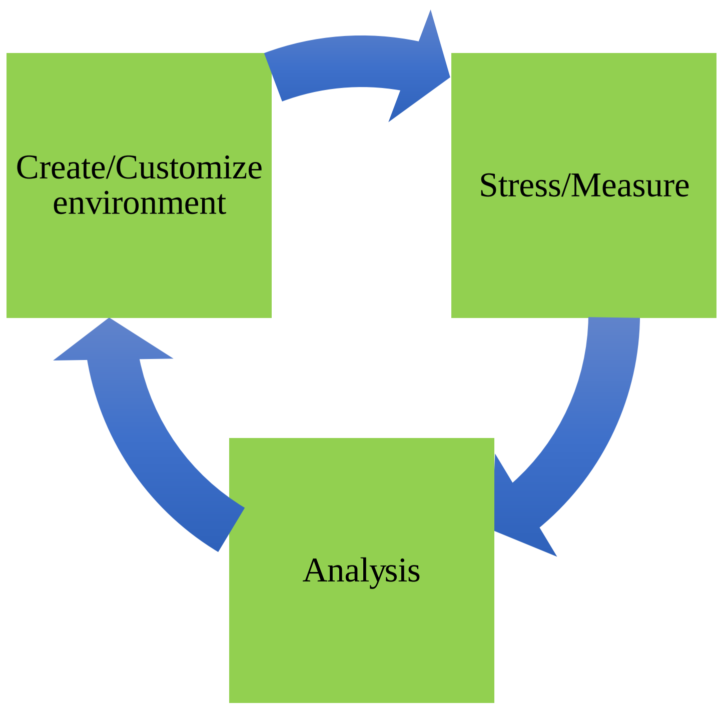

Figure 1: IBM z/OS Connect EE Performance Evaluation Cycle

3.1 Pre-Measurement Procedure

The following generic procedure was used to setup the measurement environment before the start of the measurement procedure to evaluate z/OS Connect EE:

1. Allocate IMS system data sets including Online Log Data Sets (OLDS), Write Ahead Data Sets (WADS) and Recovery Control Data Sets (RECONs)

2. Initialize IMS RECONs for database recoverability

3. Start Structured Call Interface (SCI), Operations Manager (OM) and Internal Resource Lock Manager (IRLM)

4. Initialize and 'Cold Start' IMS

5. Start the IMS Dependent regions, also known as MPP (Message Processing Program)

6. Start the IMS Connect address space

7. Start the z/OS Connect EE server

8. Start the web server, ICSF (Integrated Cryptographic Service Facility) and policy agent (PAGENT) address space for requester scenario

3.2 Measurement Procedure

The following generic steps were taken to measure the performance of both provider and requester workloads. The measurement procedure captures key performance data about the overall z/OS system as well as data specific to IMS and z/OS Connect EE. The following procedure was used to capture the measurement:

1. Start Java REST client driver address space to:

    1.1. Continuously send HTTPS GET requests to z/OS Connect EE for invoking the API in the provider scenario

    1.2. Drive the IMS application which issues REST API requests to the web server in the requester scenario

2. Wait four minutes for the workload to stabilize

3. Start Resource Management Facility (RMF) Monitor I – Two minute RMF intervals were used

4. Start RMF Monitor III – Two minute RMF intervals were used

5. Issue the /SWITCH OLDS command to force an IMS OLDS switch

6. Issue the /CHECKPOINT STATISTICS command to request that IMS performance records be created and written to the IMS log

7. Issue the /F DRIVER,APPL=PRINT command to print statistics from the Java REST client drivers

8. Wait for two minutes

9. Issue the /F DRIVER,APPL=PRINT command to print statistics from the Java REST client drivers

10. Issue the /CHECKPOINT STATISTICS command to request that IMS performance records be created and written to the IMS log

11. Issue the /SWITCH OLDS command to force an IMS OLDS switch

12. Stop the Java REST client driver address space (/P DRIVER)

13. Shutdown z/OS Connect EE, IMS Connect, web server and IMS address space

3.3 Post Measurement Procedure

After completing the measurement, the following generic steps were used to capture the performance data to evaluate z/OS Connect EE:

- Run the SMF dump program (IFASMFDP) to allow for post processing

- Run RMF post processing (ERBRMFPP) against the dumped SMF data to produce various RMF reports detailing z/OS system activity

- Run IMS Performance Analyzer (IMSPA) against all of the IMS OLDS processed between steps 5 and 11 of Section 3.2 to produce various reports detailing IMS activity

3.4 Measurement Metrics and Analysis

The results of each performance evaluation include many different data points from:

- RMF providing information about z/OS and hardware resources such as CPU utilization, memory consumption, and I/O rates
- IMSPA providing IMS internal statistics such as transaction rate, logging rate, and latch contention rates

All data is captured and saved for future research and analysis.

There are some basic metrics that apply to all workload measurements for z/OS Connect EE as shown in Table 1 below.

Metric	Description
ETR (Tran/sec)	External Transaction Rate is the observed average transaction rate in transactions per second (TPS) over the measurement interval captured from the RMF report.
Client Response Time (ms)	The total client response time from the Java workload driver.
Total IMS CPU Service Time (sec)	The amount of CPU time consumed by the processors for all IMS address spaces captured from the RMF report. IMS address spaces include Control region, DL/I region, DBRC region, SCI, OM and IRLM. It is reported on a per transaction basis by dividing the total CPU service time consumed during an interval by the number of transactions processed in that interval.
IMS Connect CPU Service Time (sec)	The amount of CPU time consumed by the processors for the IMS Connect address space captured from the RMF report. It is reported on a per transaction basis by dividing the total CPU service time consumed during an interval by the number of transactions processed in that interval.
IMS Dependent Regions CPU Service Time (sec)	The amount of CPU time consumed by the processors for IMS dependent region address space captured from the RMF report. It is reported on a per transaction basis by dividing the total CPU service time consumed during an interval by the number of transactions processed in that interval.
z/OS Connect EE CPU Service Time (sec)	The amount of CPU time consumed by the processors for the z/OS Connect EE address space captured from the RMF report. It is reported on a per transaction basis by dividing the total CPU service time consumed during an interval by the number of transactions processed in that interval.
Projected z/OS Connect EE zIIP Service Time (sec)	The amount of zIIP time, if zIIPs were available, consumed by the processors on an LPAR captured from the RMF report. It is reported on a per transaction basis by dividing the total zIIP service time consumed during an interval by the number of transactions processed in that interval.

Table 1: Performance Metrics for IBM Z **Processor Comparison**

4 System Software and Hardware

Table 2 lists the software components and version level information below.

Component	Details
IBM z/OS	z/OS® Version 2 Release 2 (5650-ZOS)
IMS	IMS™ Version 15
z/OS Connect Enterprise Edition with IMS SP	z/OS Connect EE V3.0.10
z/OS API tool kit	V3.0.0.0
IRLM	IRLM Version 2.3 (5635-A04)
IBM Enterprise COBOL for z/OS	Version 4 Release 2
z/OS Resource Measurement Facility (RMF)	Version 2 Release 2
z/OS Resource Access Control Facility (RACF)	Version 2 Release 2
IMS Performance Analyzer for z/OS (IMSPA)	Version 4 Release 4
Java	Java 8 Service Release 5 Fixpack 15

Table 2: Software Components and Version Levels

The data sets required for z/OS, IMS and measurement data were allocated and distributed on IBM System Storage® DS8000® series latest model DS8886 with 16 FICON® channel paths with 64 real volumes and 128 alias volumes per LCU using dynamic Hyper Parallel Access Volumes (HyperPAVs) to ensure that z14 and DS8886 have the optimal bandwidth to sustain high data rates.

5 Provider Workload

When z/OS resources are exposed as RESTful APIs for mobile and cloud applications to consume, z/OS Connect EE acts as RESTful API provider.

Using the z/OS Connect API toolkit, a workstation-based eclipse tool, developers with or without z/OS skills can easily create RESTful APIs from traditional z/OS based assets. These APIs are then deployed to z/OS Connect EE allowing them to be accessible from mobile and cloud applications.

The workload used to evaluate the provider scenario focused on stressing the z/OS Connect EE IMS SP code and consists of message only transactions without any database activity.

5.1 Measurement Environment

The z/OS Connect EE IMS SP evaluation was executed on z14 in a two LPAR configuration as shown in Figure 2 for the provider scenario:

- LPAR 1 hosts z/OS Connect EE V3.0.10 with IMS SP, IMS 15 with 350 MPPs, and IMS Connect 15 with nine General Purpose Engines
- LPAR 2 hosts a Java-based workload driver with 10 General Purpose Engines

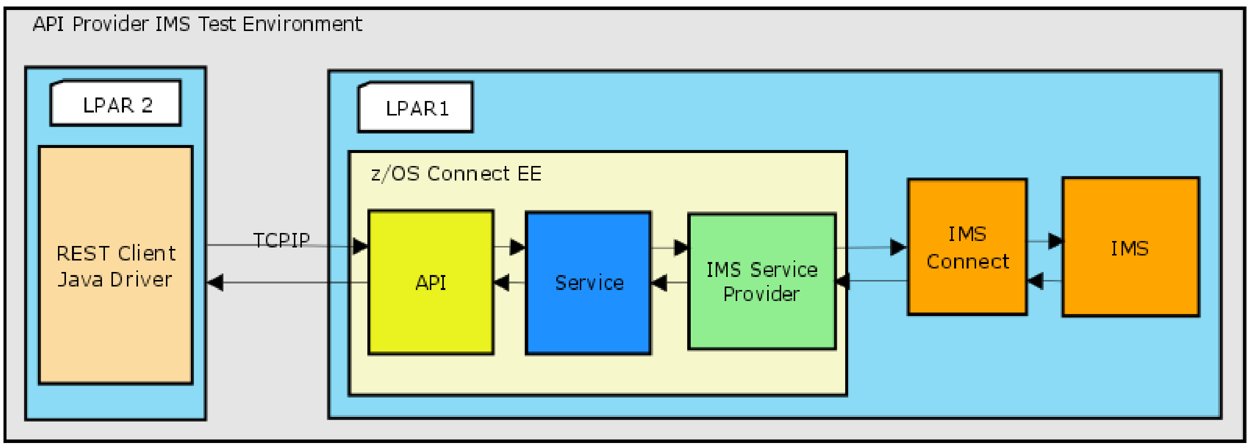

Figure 2: API Provider Environment Configuration

5.2 IMS

5.2.1 Transactions

The workload consists of 25 message-only transactions with transaction codes VARTX001 to VARTX025. All transactions run in MPP regions. Figure 3 shows the transaction attributes from issuing IMS command /DIS TRAN ALL below.

TRAN	CLS	ENQCT	QCT	LCT	PLCT	CP	NP	LP	SEGSZ	SEGNO	PARLM	RC
VARTX001	1	0	0	65535	65535	1	1	1	0	0	0	0
VARTX002	2	0	0	65535	65535	1	1	1	0	0	0	0
VARTX003	3	0	0	65535	65535	1	1	1	0	0	0	0
VARTX004	4	0	0	65535	65535	1	1	1	0	0	0	0
VARTX005	5	0	0	65535	65535	1	1	1	0	0	0	0
VARTX006	6	0	0	65535	65535	1	1	1	0	0	0	0
VARTX007	7	0	0	65535	65535	1	1	1	0	0	0	0
VARTX008	8	0	0	65535	65535	1	1	1	0	0	0	0
VARTX009	9	0	0	65535	65535	1	1	1	0	0	0	0
VARTX010	10	0	0	65535	65535	1	1	1	0	0	0	0
VARTX011	11	0	0	65535	65535	1	1	1	0	0	0	0
VARTX012	12	0	0	65535	65535	1	1	1	0	0	0	0
VARTX013	13	0	0	65535	65535	1	1	1	0	0	0	0
VARTX014	14	0	0	65535	65535	1	1	1	0	0	0	0
VARTX015	15	0	0	65535	65535	1	1	1	0	0	0	0
VARTX016	16	0	0	65535	65535	1	1	1	0	0	0	0
VARTX017	17	0	0	65535	65535	1	1	1	0	0	0	0
VARTX018	18	0	0	65535	65535	1	1	1	0	0	0	0
VARTX019	19	0	0	65535	65535	1	1	1	0	0	0	0
VARTX020	20	0	0	65535	65535	1	1	1	0	0	0	0
VARTX021	21	0	0	65535	65535	1	1	1	0	0	0	0
VARTX022	22	0	0	65535	65535	1	1	1	0	0	0	0
VARTX023	23	0	0	65535	65535	1	1	1	0	0	0	0
VARTX024	24	0	0	65535	65535	1	1	1	0	0	0	0
VARTX025	25	0	0	65535	65535	1	1	1	0	0	0	0

Figure 3: IMS Service Provider Workload Transaction Attributes

5.2.2 Application

The application was designed to receive and return messages with variable lengths based on the incoming message's field values. If the incoming message specified 1KB in its input message size field, the response message would have a size of 1KB. The I/O messages contain simple character fields. The evaluations in this paper focused on I/O message sizes of 1KB, 8KB and 30KB.

5.2.3 Message Structures

The z/OS Connect EE IMS SP provides data transformation between JSON and the binary format that the IMS transaction expects. This is done using message metadata imported from COBOL copybooks or PL/I includes. At runtime, the data and message metadata structures are dynamically inspected and converted.

Figures 4, 5 and 6 show the COBOL copybook structures for 1KB, 8KB and 30KB message sizes used in evaluating the API performance.

```
 01   INPUT-MSG.
  02  IN-LL                   PIC S9(4) COMP-5.
  02  IN-ZZ                   PIC S9(4) COMP-5.
  02  IN-TRAN                 PIC X(8).
  02  IN-ALTPCB               PIC X(8).
  02  IN-LINKNAME             PIC X(8).
  02  IN-NUM-SEGS-IN          PIC X(8).
  02  IN-SIZE-PER-SEG-IN      PIC X(5).
  02  IN-NUM-SEGS-OUT         PIC X(8).
  02  IN-SIZE-PER-SEG-OUT     PIC X(5).
  02  IN-WAIT-TIME            PIC X(8).
  02  IN-DATA                 PIC X(962).

01 OUTPUT-MSG-SEG.
  02  OUT-LL-SEG              PIC S9(4) COMP-5.
  02  OUT-ZZ-SEG              PIC S9(4) COMP-5.
  02  OUT-DATA-SEG            PIC X(1020).
```

Figure 4: COBOL Copybook 1KB Message Size

```
01   INPUT-MSG.
  02  IN-LL                   PIC S9(4) COMP-5.
  02  IN-ZZ                   PIC S9(4) COMP-5.
  02  IN-TRAN                 PIC X(8).
  02  IN-ALTPCB               PIC X(8).
  02  IN-LINKNAME             PIC X(8).
  02  IN-NUM-SEGS-IN          PIC X(8).
  02  IN-SIZE-PER-SEG-IN      PIC X(5).
  02  IN-NUM-SEGS-OUT         PIC X(8).
  02  IN-SIZE-PER-SEG-OUT     PIC X(5).
  02  IN-WAIT-TIME            PIC X(8).
  02  IN-DATA                 PIC X(8130).

01 OUTPUT-MSG-SEG.
  02  OUT-LL-SEG              PIC S9(4) COMP-5.
  02  OUT-ZZ-SEG              PIC S9(4) COMP-5.
  02  OUT-DATA-SEG            PIC X(8188).
```

Figure 5: COBOL Copybook 8KB Message Size

```
01   INPUT-MSG.
  02  IN-LL                   PIC S9(4) COMP-5.
  02  IN-ZZ                   PIC S9(4) COMP-5.
  02  IN-TRAN                 PIC X(8).
  02  IN-ALTPCB               PIC X(8).
  02  IN-LINKNAME             PIC X(8).
  02  IN-NUM-SEGS-IN          PIC X(8).
  02  IN-SIZE-PER-SEG-IN      PIC X(5).
  02  IN-NUM-SEGS-OUT         PIC X(8).
  02  IN-SIZE-PER-SEG-OUT     PIC X(5).
  02  IN-WAIT-TIME            PIC X(8).
  02  IN-DATA                 PIC X(30658).
```

```
01 OUTPUT-MSG-SEG.
  02  OUT-LL-SEG              PIC S9(4) COMP-5.
  02  OUT-ZZ-SEG              PIC S9(4) COMP-5.
  02  OUT-DATA-SEG            PIC X(30716).
```

Figure 6: COBOL Copybook 30KB Message Size

5.3 z/OS Connect Enterprise Edition

To establish communication with IMS Connect and execute transactions on IMS, several resources need to be modified on the z/OS Connect EE install configuration. This section covers the types of resources that needs to be modified.

5.3.1 Connection Profile

An IMS connection profile defines a connection between z/OS Connect EE and IMS Connect. The connection profile must be updated according to the host name and port number defined in the IMS Connect configuration member (HWSCFGxx). This can be done manually or using the IBM z/OS Explorer tool, which connects to the z/OS Connect EE server installed on the z/OS system. The following changes were made to ims-connections.xml:

- For XML element imsmobile_imsConnection: connectionFactoryRef and id attributes were changed to PERFCONN1_CF and PERFCONN1 respectively.
- For XML element connectionFactory: id attribute was changed to PERFCONN1_CF.
- For XML element properties.gmoa: hostName and portNumber attributes were changed according to HWSCFGxx.

Figure 7 shows an example of the connection profile file.


```
<server>
<imsmobile_imsConnection comments="IMS Gateway Connection" configSchemaVersion="3" connectionFactoryRef="PERFCONN1_CF" connectionTimeout="-1"
connectionType="IMSCONNECT" id="PERFCONN1" idleTimeout="-1" modifiedBy="omvsadm" modifyTime="2017-04-19 19:58:59.872" passwordExists="false"
traceAppend="false" traceLevel="0" usesTrace="false"/>
<connectionFactory id="PERFCONN1_CF">
<properties.gmoa hostName="1.2.3.4" portNumber="1111"/>
</connectionFactory>
</server>
```

Figure 7: IMS Connections Profile ims-connections.xml example

5.3.2 Interaction Properties Profile

An interaction properties profile specifies how z/OS Connect EE invokes an IMS transaction. The following changes were made to ims-interactions.xml:

- For XML element imsmobile_interaction: id and imsDatastoreName attributes were changed to PERFPROPS and datastore ID from HWSCFGxx respectively

Figure 8 shows an example of interactions profile file.


```xml
<server>
<imsmobile_interaction ackNakProvider="0" comments="" commitMode="32" configSchemaVersion="1" id="PERFPROPS" imsConnectCodepage="cp1047" imsConnectTimeout="0" imsConnectUserMessageExitIdentifier="*SAMPL1*" imsDatastoreName="FR01" inputMessageDataSegmentsIncludeLlzzAndTrancode="true" interactionTimeout="-1" interactionTypeDescription="SENDRECV" ltermOverrideName="" modifiedBy="omvsadm" modifyTime="2017-04-19 19:59:00.974" outputMessageDataSegmentsIncludeLlzz="true" propertyType="TRAN" purgeUndeliverableOutput="false" rerouteName="reroute" rerouteUndeliverableOutput="false" responseIncludesLlll="false" resumeTpipeAlternateClientId="" resumeTpipeProcessing="16" returnMfsModname="false" syncLevel="0" useCM0AckNoWait="false" propagateNetworkSecurityCred="false"/>
</server>
```

Figure 8: IMS Interaction Properties Profile ims-interactions.xml

5.3.3 Creating Services and REST API

To access a resource on z/OS using z/OS Connect EE via APIs, the following two-step approach is required:

- A service needs to be created that defines how the JSON schemas for the request and response messages map to the resource

- A REST API needs to be defined on how an HTTP action such as GET, PUT, POST or DELETE would act on a service

5.3.3.1 Services and Service Archive File

A service in z/OS Connect EE is used by the REST API to act on a z/OS resource through connections and data transformation provided through a service provider. Information about service is contained in a service archive (.sar) file, which includes information about the request and response JSON schemas required by the service.

Using the z/OS Connect API Toolkit, services corresponding to the IMS transactions are created by importing the COBOL copybooks described in Section 5.2.3, which explain the input and output message structure for 1KB, 8KB and 30KB message sizes. In the service interface definition window, specific fields can be excluded, if needed, by unchecking the box next to the field for the request service interface.

Figure 9 shows an example of the request service interface for 1KB message size.

Fields	Include	Interface rename	Default Field Value	Data Type	Field Length	Start Byte
vartx001Request						
Segment 1						
INPUT_MSG						
IN_LL	☐	IN_LL		SHORT	2	1
IN_ZZ	☐	IN_ZZ		SHORT	2	3
IN_TRAN	☐	IN_TRAN		CHAR	8	5
IN_ALTPCB	☐	IN_ALTPCB		CHAR	8	13
IN_LINKNAME	☐	IN_LINKNAME		CHAR	8	21
IN_NUM_SEGS_IN	☑	IN_NUM_SEGS_IN		CHAR	8	29
IN_SIZE_PER_SEG_IN	☑	IN_SIZE_PER_SEG_IN		CHAR	5	37
IN_NUM_SEGS_OUT	☑	IN_NUM_SEGS_OUT		CHAR	8	42
IN_SIZE_PER_SEG_OUT	☑	IN_SIZE_PER_SEG_OUT		CHAR	5	50
IN_WAIT_TIME	☑	IN_WAIT_TIME		CHAR	8	55
IN_DATA	☑	IN_DATA		CHAR	962	63

Figure 9: Request Service Interface Definition for Input Message with 1KB Message Size

Similarly, Figure 10 shows an example of the response service interface for 1KB message size.

Fields	Include	Interface rename	Default Field Value	Data Type	Field Length	Start Byte
vartx001Response						
Segment 1						
OUTPUT_MSG_SEG						
OUT_LL_SEG	☐	OUT_LL_SEG		SHORT	2	1
OUT_ZZ_SEG	☐	OUT_ZZ_SEG		SHORT	2	3
OUT_DATA_SEG	☑	OUT_DATA_SEG		CHAR	1020	5

Figure 10: Response Service Interface Definition for Output Message with 1KB Message Size

Next, the connection and interaction properties profiles are identified in the configuration window as shown below in Figure 11.

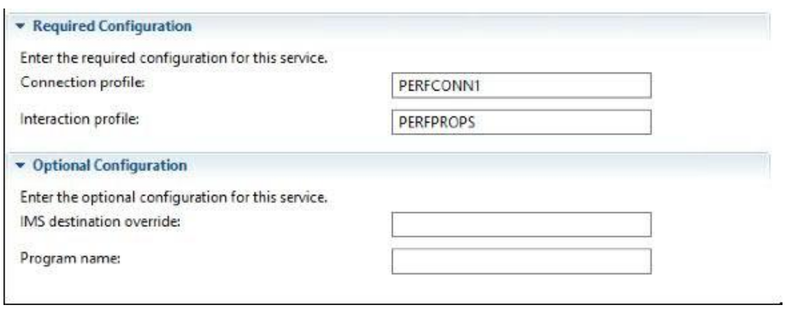

Figure 11: Connection and Interaction Properties Profile Configuration

The service can now be exported as a .sar file and deployed to the z/OS Connect EE server.

5.3.3.2 REST API for accessing Services

The REST API defines how HTTP actions such as GET, PUT, POST or DELETE would act on services defined to z/OS Connect EE. Information about the REST APIs for a service are contained in an API archive (.aar) file that can be deployed to z/OS Connect EE.

A new z/OS API project is created in the z/OS Connect EE API editor as shown in Figure 12.

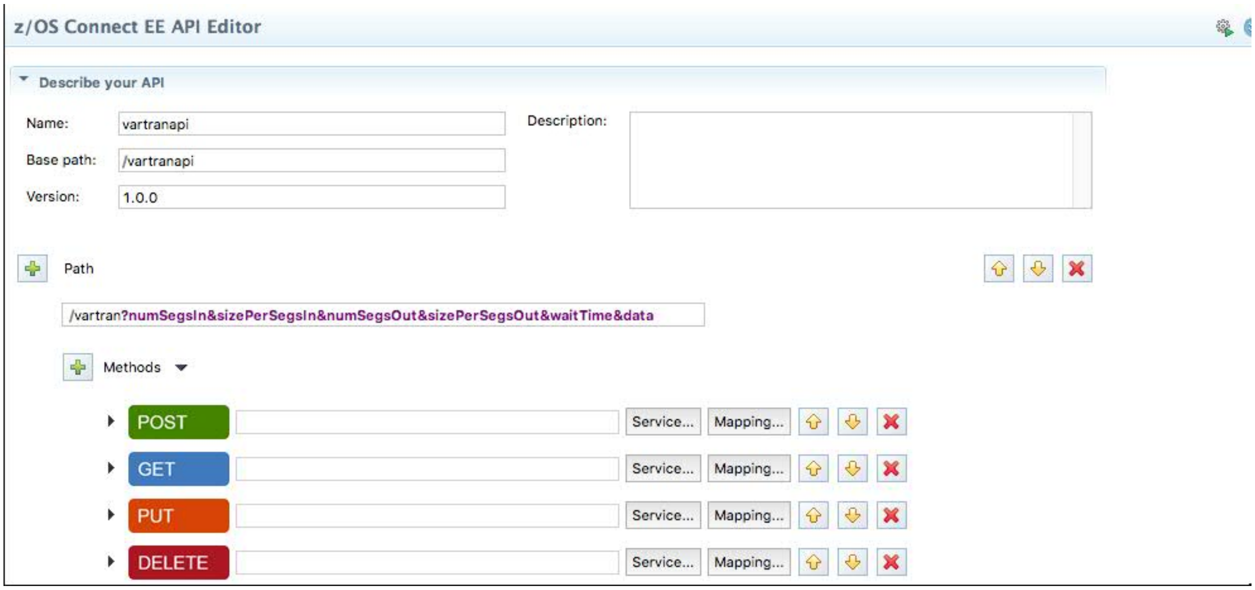

Figure 12: z/OS Connect EE API Editor

The API Editor, introduced in z/OS Connect EE V2.0, is used to perform HTTP-to-JSON mappings as shown in Figure 13 and 14 for request and response messages. This mapping capability adds a powerful abstraction layer between the API consumer and underlying z/OS assets.

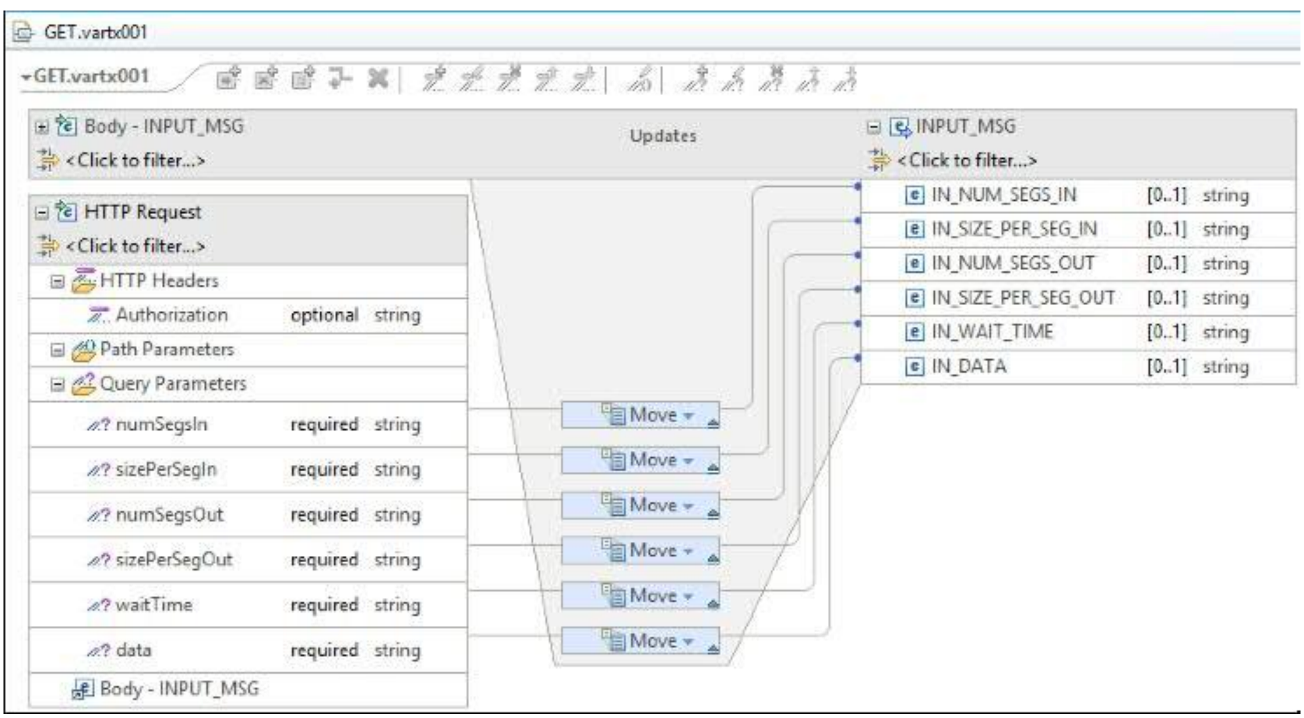

Figure 13: z/OS Connect EE API Editor 1KB Request API Mapping

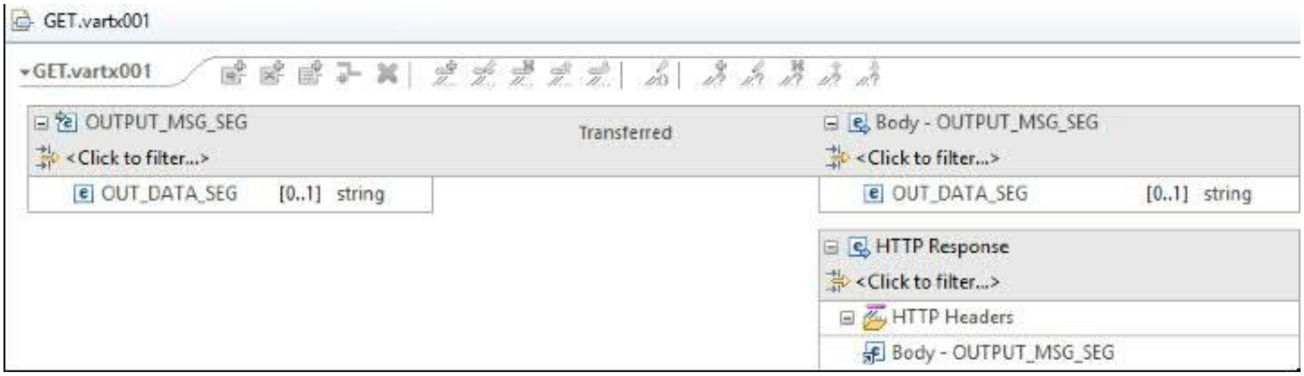

Figure 14: z/OS Connect EE API Editor 1KB Response API Mapping

5.4 REST Driver

A stand-alone Java application was used to drive the workload by sending concurrent REST API requests across several threads to simulate multiple users with a think time (i.e. the delay between sending requests) of zero. Each request calls a REST API using HTTPS GET and randomly invokes one of the 25 services deployed on the z/OS Connect EE server.

5.4.1 I/O Payload

The API is invoked by sending HTTPS GET requests to the z/OS Connect EE IMS SP. HTTPS is required in order to communicate with z/OS Connect EE.

Figures 15 and 16 show examples of the JSON payload in the request and response messages respectively for the 1KB message size.

```
{
  "type" : "object",
  "properties" : {
    "INPUT_MSG" : {
      "type" : "object",
      "properties" : {
        "IN_NUM_SEGS_IN" : {
          "maxLength" : 8,
          "type" : "string"
        },
        "IN_SIZE_PER_SEG_IN" : {
          "maxLength" : 5,
          "type" : "string"
        },
        "IN_NUM_SEGS_OUT" : {
          "maxLength" : 8,
          "type" : "string"
```

```
        },
        "IN_SIZE_PER_SEG_OUT" : {
          "maxLength" : 5,
          "type" : "string"
        },
        "IN_WAIT_TIME" : {
          "maxLength" : 8,
          "type" : "string"
        },
        "IN_DATA" : {
          "maxLength" : 962,
          "type" : "string"
        }
      }
    }
  }
}
```
Figure 15: API JSON Payload 1KB Request Message

After invoking the service, an HTTPS response is received from the IMS SP with a JSON payload containing the output transaction message. Figure 16 shows the response JSON payload after invoking a service for 1KB message size.

```
{
  "type" : "object",
  "properties" : {
    "OUTPUT_MSG_SEG" : {
      "type" : "object",
      "properties" : {
        "OUT_DATA_SEG" : {
          "maxLength" : 1020,
          "type" : "string"
        }
      }
    }
  }
}
```
Figure 16: API JSON payload 1KB Response Message

5.4.2 Tuning

To reduce the overhead in initializing TCP connections and the handshaking involved in SSL/TLS for establishing security associations, persistent connections were implemented to improve HTTPS performance. Persistent connections were enabled by setting http.maxConnections equal to the number of concurrent users in the client Java driver.

5.4.3 Execution

The REST driver was executed in z/OS Java batch jobs on a separate logical partition (LPAR) from the z/OS Connect EE IMS SP. The number of concurrent users varied from one to 90.

Each time a response was received from the z/OS Connect EE IMS SP, the driver compares the JSON payload with the expected JSON payload to ensure valid data was received.

The driver also has the capability of printing out statistics such as the current transaction rate and average client response time during runtime through a print statistics command. Figure 17 shows sample output of the print statistics command while running with 30 users.

```
Added 30 ims.perf.util.driver.ThreadPoolProperties threads to the
ZCEEAPITHREAD_POOL pool.
Added 30 ZceeApiThread threads.
Driver fully initialized.
All statistics were reset.
Printing statistics.
Pool Name: ZCEEAPITHREAD_POOL.
-->Class Name: ims.perf.util.driver.ThreadPoolProperties.
-->Active Threads: 30.
.
Statistics for: ZceeApiThread.
->Tx Rate: 20298.7118.
->Avg Resp Time (ms): 1.4760.
```

Figure 17: REST Driver Output from Print Statistics Command

5.5 Provider Performance Results

The z/OS Connect EE provider performance was evaluated by examining how the transaction rate, client response time and Service Time per transaction behaved under stress.

In this section, we investigate how scaling the number of concurrent users would affect the transaction rate, client response time and Service Time per transaction. Each user invokes a single API that randomly drives one of the 25 RESTful services. For each I/O message size (i.e. 1KB, 8KB and 30KB), the number of concurrent users were scaled from 1, 5, 10, 15, 20, 25, 30, 40, 50, 60, 70 and 80.

5.5.1 Transaction Rate

Figure 18 shows the effect on throughput, measured in Transactions Per Second (TPS), as the number of users increase:

- With 1KB I/O messages, throughput peaked at about 21,000 TPS before stabilizing.
- With 8KB I/O messages, throughput peaked at about 8,900 TPS before stabilizing.

- Lastly, with 30KB I/O messages, throughput peaked at about 3,500 TPS before stabilizing.

We found as the number of users increase, the transaction rate also increases until reaching its peak and stabilizing. As we increased the number of users with larger I/O messages, the transaction rate showed the same pattern, but peaked and stabilized with much less users at lower transaction rates.

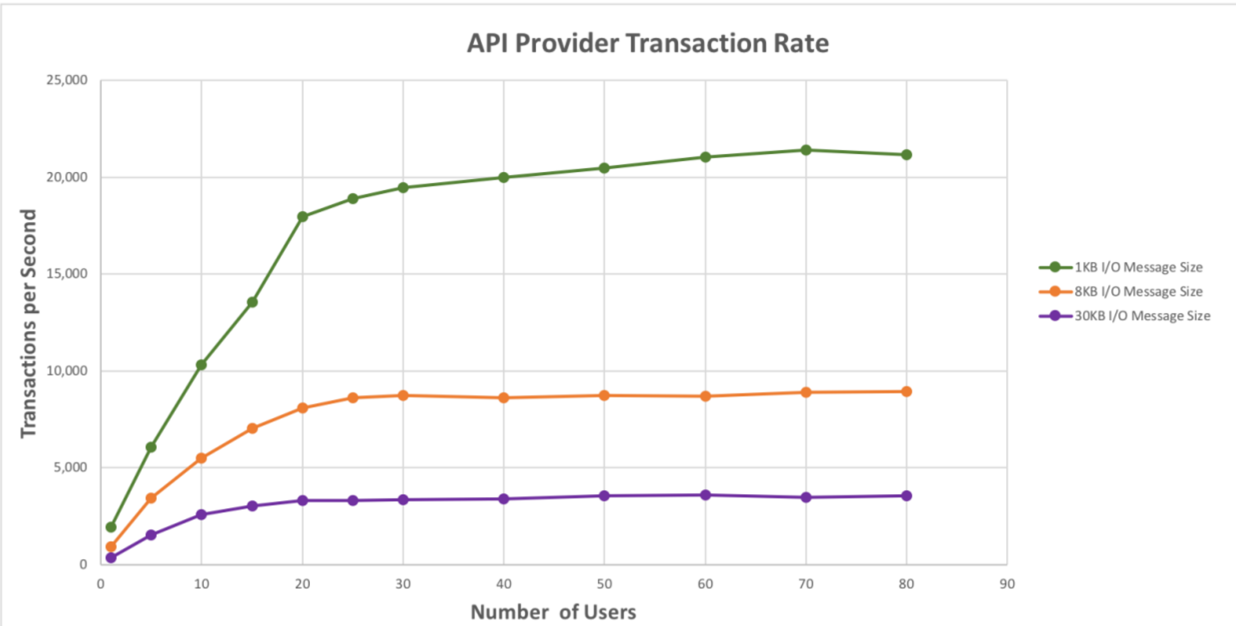

Figure 18: Scaling the Number of users – Transaction Rate

5.5.2 Client Response Time

Figure 19 shows the effect on client response times measured in milliseconds (ms), as the number of users increase:

- With 1KB I/O messages, the client response time gradually increased from less than one ms to three ms as the number of users increased.
- With 8KB I/O messages, the client response time increased at a greater rate from one ms to nine ms as the number of users increased.
- With 30KB I/O messages, the client response time increased at a greater rate from one to 22 ms as the number of users increased.

We found as the number of users increase, the client response time increases linearly. When increasing the number of users with larger I/O message sizes, the client response time continues to increase linearly but at a sharper incline.

Figure 19: Scaling the Number of users – Client Response Time

5.5.3 Service Time per Transaction

Figure 20 shows the effect on z/OS Connect EE Service Time per transaction, measured in microseconds (µs), as the number of users increase:

- With 1KB I/O messages, the z/OS Connect EE Service Time per Transaction ranged between 200-300 µs as the number of users increased.
- With 8KB I/O messages, the z/OS Connect EE Service Time per Transaction ranged between 600-800 µs as the number of users increased.
- With 30KB I/O messages, the z/OS Connect EE Service Time per Transaction ranged between 1900-2200 µs as the number of users increased.

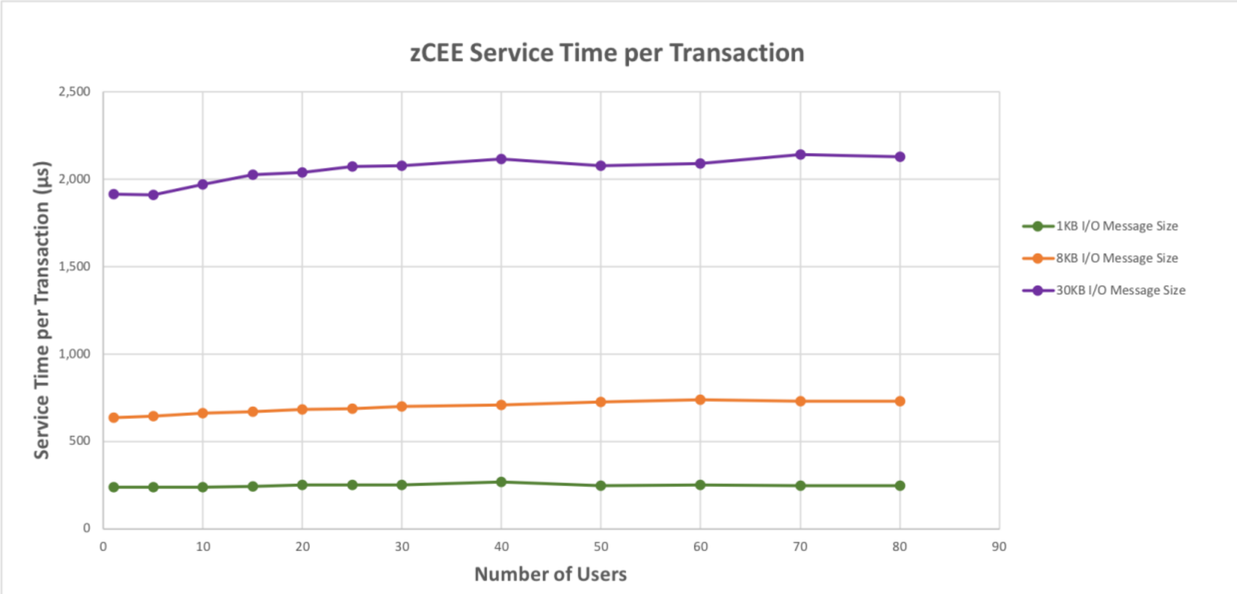

Figure 20: Scaling the Number of users – Service Time per Transaction

Figure 21 shows the z/OS Connect EE IMS SP Service Time per transaction for 30 users.

- With 1KB I/O messages, the total Service Time per transaction was 312 µs with z/OS Connect EE IMS SP contributing 252 µs of the total.
- With 8KB I/O messages, the total Service Time per transaction was 777 µs with z/OS Connect EE IMS SP contributing 698 µs of the total.
- With 30 KB I/O messages, the total Service Time per transaction was 2260 µs with z/OS Connect EE IMS SP contributing 2,078 µs of the total.

With any number of users from one to 80, we found Service Time per transaction stayed fairly consistent. Of the total Service Time per Transaction, the majority originated from the z/OS Connect EE IMS SP. The same pattern held true when increasing the I/O message size, but at a greater proportion.

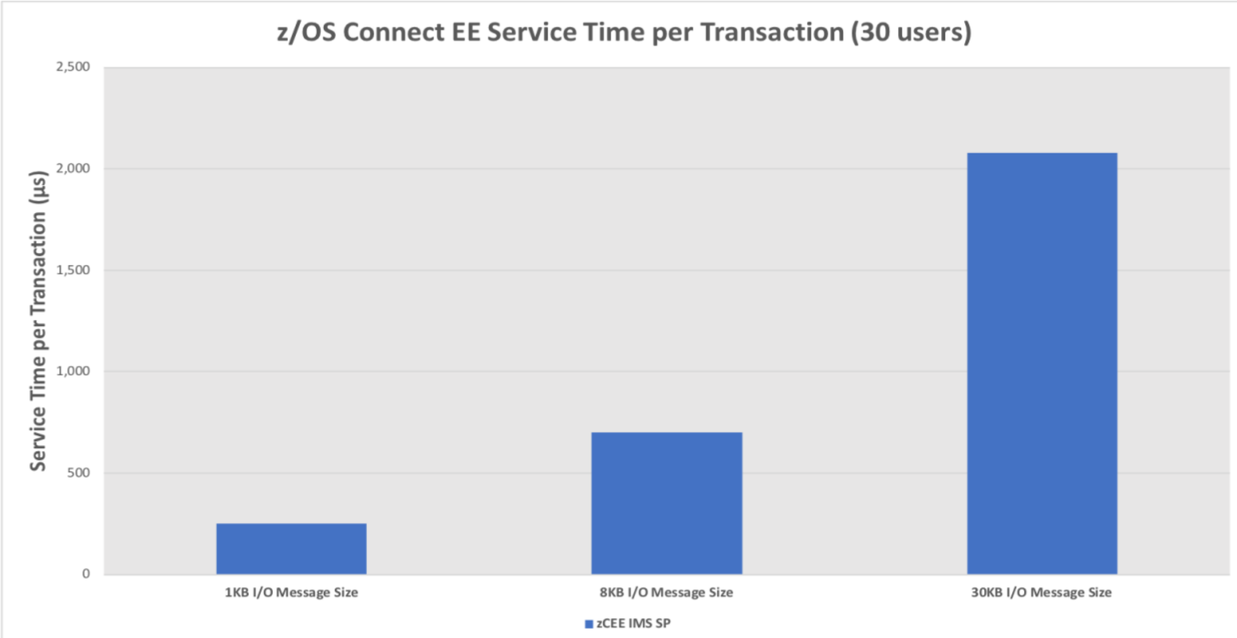

Figure 21: z/OS Connect EE Service Time per Transaction for 30 Users

Figure 22 shows the projected zIIP offload for z/OS Connect EE IMS SP if the same workload was executed with zIIPs enabled. The results show that approximately 99% of the processing is zIIP eligible as z/OS Connect EE is a Java based product.

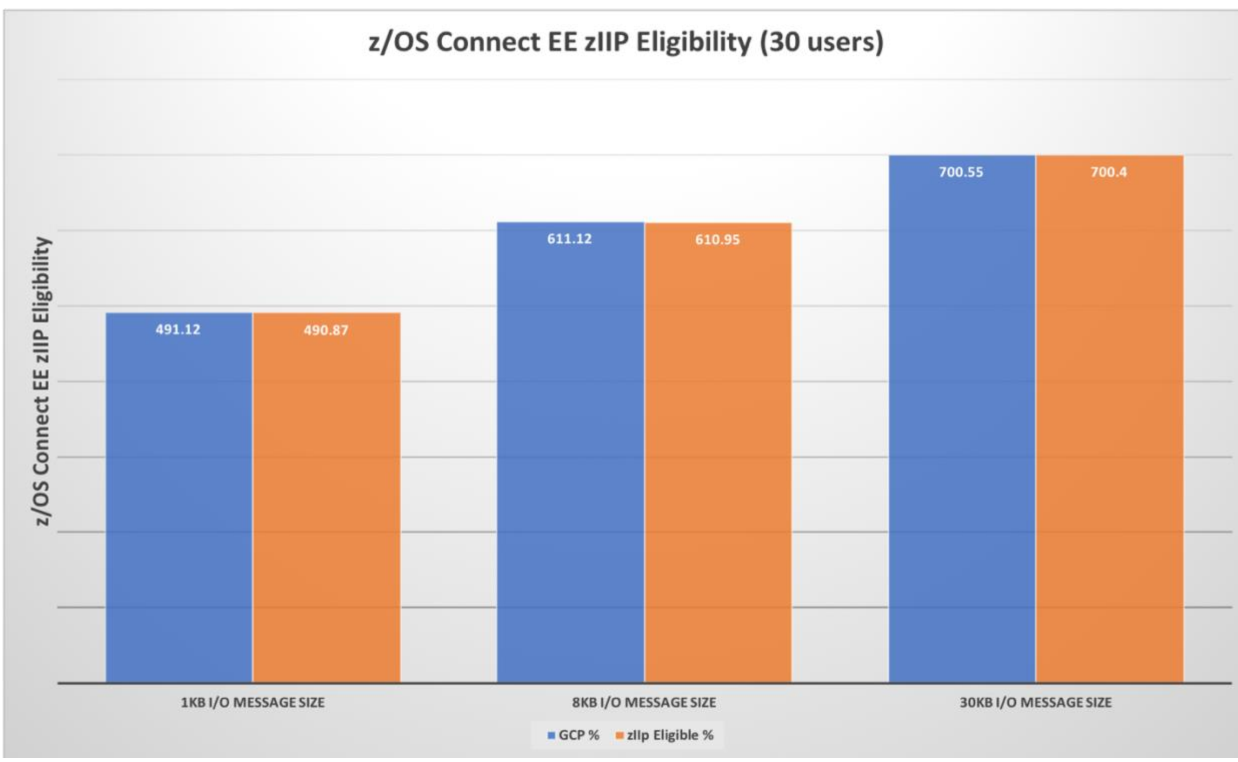

Figure 22: Projected z/OS Connect EE zIIP Eligibility for 30 Users

6 Requester Workload

With z/OS Connect EE, z/OS applications can consume RESTful APIs, thus utilizing the underlying services and data from external hosts (also called request endpoints). In this scenario, z/OS acts as a RESTful API consumer.

An intermediary named API requester is required to facilitate RESTful API calls by z/OS-based programs.

6.1 Measurement Environment

The requester scenario evaluation was executed on z14 in a two LPAR configuration as shown in Figure 23:

- LPAR 1 hosts z/OS Connect EE V3.0.10, IMS 15 with 350 MPPs, IMS Connect 15, and web server with seven General Purpose Engines
- LPAR 2 hosts a Java-based workload driver with 10 General Purpose Engines

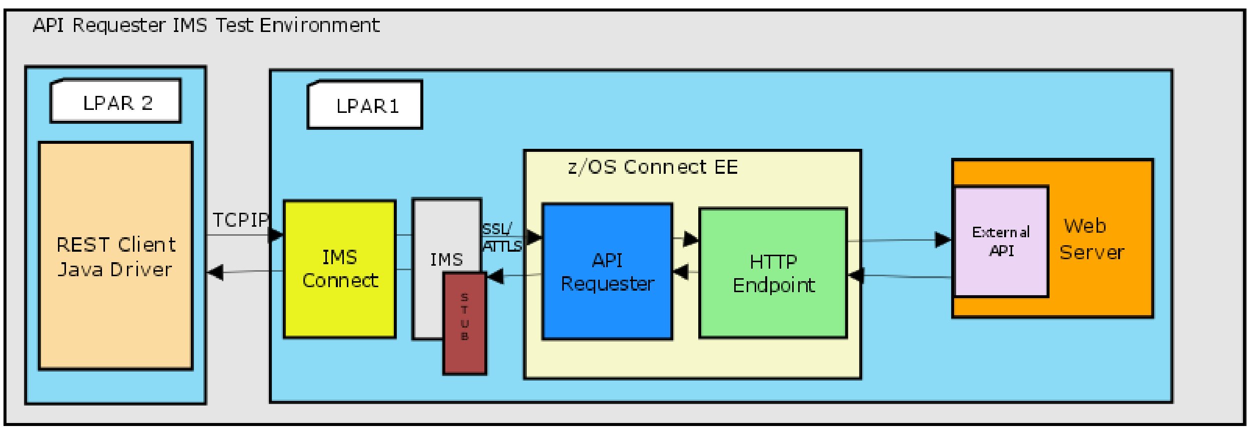

Figure 23: API Requester Environment Configuration

6.2 IMS

6.2.1 Transactions

The workload consists of 25 message-only transactions with transaction codes REQTX001 to REQTX025. All transactions run in MPP regions. Figure 24 shows the transaction attributes from issuing IMS command /DIS TRAN ALL below.

```
TRAN      CLS ENQCT   QCT   LCT  PLCT CP NP LP SEGSZ SEGNO PARLM    RC
REQTX001   1      0     0 65535 65535  1  1  1     0     0     0     0

REQTX002   2      0     0 65535 65535  1  1  1     0     0     0     0
REQTX003   3      0     0 65535 65535  1  1  1     0     0     0     0
REQTX004   4      0     0 65535 65535  1  1  1     0     0     0     0
REQTX005   5      0     0 65535 65535  1  1  1     0     0     0     0
REQTX006   6      0     0 65535 65535  1  1  1     0     0     0     0
REQTX007   7      0     0 65535 65535  1  1  1     0     0     0     0
REQTX008   8      0     0 65535 65535  1  1  1     0     0     0     0
REQTX009   9      0     0 65535 65535  1  1  1     0     0     0     0
REQTX010  10      0     0 65535 65535  1  1  1     0     0     0     0
REQTX011  11      0     0 65535 65535  1  1  1     0     0     0     0
REQTX012  12      0     0 65535 65535  1  1  1     0     0     0     0
REQTX013  13      0     0 65535 65535  1  1  1     0     0     0     0
REQTX014  14      0     0 65535 65535  1  1  1     0     0     0     0
REQTX015  15      0     0 65535 65535  1  1  1     0     0     0     0
REQTX016  16      0     0 65535 65535  1  1  1     0     0     0     0
REQTX017  17      0     0 65535 65535  1  1  1     0     0     0     0
REQTX018  18      0     0 65535 65535  1  1  1     0     0     0     0
REQTX019  19      0     0 65535 65535  1  1  1     0     0     0     0
REQTX020  20      0     0 65535 65535  1  1  1     0     0     0     0
REQTX021  21      0     0 65535 65535  1  1  1     0     0     0     0
REQTX022  22      0     0 65535 65535  1  1  1     0     0     0     0
REQTX023  23      0     0 65535 65535  1  1  1     0     0     0     0
REQTX024  24      0     0 65535 65535  1  1  1     0     0     0     0
REQTX025  25      0     0 65535 65535  1  1  1     0     0     0     0
```
Figure 24: IMS Requester Workload Transaction Attributes

6.2.2 Communication Stub Configuration

Before IMS application can make RESTful API calls through z/OS Connect EE, a communication stub in IMS needs to be configured. The communication stub is a load module for handling the HTTP connection and communication between IMS and z/OS Connect EE.

Figure 25 shows communication stub configuration in IMS dependent region.

```
//CEEOPTS  DD *
POSIX(ON),
ENVAR("BAQURI=1.2.3.4", "BAQPORT=3333",
"BAQUSERNAME=username", "BAQPASSWORD=password",
"BAQVERBOSE=OFF")
/*
```
Figure 25: IMS Communication Stub Configuration

The purpose of each variable in the ENVAR section is explained below.

BAQURI: Specifies the hostname of the z/OS Connect EE server to connect to

BAQPORT: Specifies the port number of the z/OS Connect EE server

BAQUSERNAME: Specifies the user name for connections

BAQPASSWORD: Specifies the password, in clear text, for the specified BAQUSERNAME to be authenticated with the z/OS Connect EE server

BAQVERBOSE: An optional value to enable verbose messages to assist in debugging of runtime and configuration issues. Valid values are ON or OFF

6.2.3 IMS Dependent Regions

The load module for the communication stub must be added to the STEPLIB concatenation for IMS dependent regions as shown below in Figure 26.

```
//STEPLIB  DD DISP=SHR,
//      DSN=IMSPERF.PGMLIB
//         DD DISP=SHR,
//      DSN=IMSPERF.RESLIB1
//         DD DISP=SHR,
//      DSN=IMSPERF.SCEERUN
//         DD DISP=SHR,
//      DSN=IMSPERF.SBAQLIB
```

Figure 26: IMS Dependent Region Configuration

6.2.4 COBOL Application

For an IMS application to call out to an API, it must include the generated request and response data structures and the API information file to communicate with z/OS Connect EE. The following steps were taken to configure the COBOL application for the requester scenario.

1. The COBOL application was modified to include the request, response and API information file as shown below.

```
* Copy request and response copybooks. These copybooks are
* generated by build toolkit and used to describe the structure
* of request and response.
* This procedure is general and necessary.
01 REQUEST.
COPY API00Q01.
01 RESPONSE.
COPY API00P01.
01 API-INFO-API1.
COPY API01I01.
```

2. Variables were declared for request and response messages as shown below.

```
* Request and Response segment, used to store request and
* response content.
01 BAQ-REQUEST-PTR            USAGE POINTER.
01 BAQ-REQUEST-LEN            PIC S9(9) COMP-5 SYNC.
01 BAQ-RESPONSE-PTR           USAGE POINTER.
01 BAQ-RESPONSE-LEN           PIC S9(9) COMP-5 SYNC.
```

3. The communication stub program name was specified as shown below.

```
77 COMM-STUB-PGM-NAME         PIC X(8) VALUE 'BAQCSTUB'.
```

4. The Request data was declared as shown below where <$PAYLOAD_SIZE$> is replaced with 1024, 8192 and 30720 according to the desired message size for the test:

```
01 REQ-DATA           PIC X(<$PAYLOAD_SIZE$>) VALUE ZEROS.
```

5. The data was prepared for the call as shown below.

```
* Use pointer and length to specify the location of
*  request and response segment.
* This procedure is general and necessary.
SET BAQ-REQUEST-PTR TO ADDRESS OF REQUEST.
MOVE LENGTH OF REQUEST TO BAQ-REQUEST-LEN.
SET BAQ-RESPONSE-PTR TO ADDRESS OF RESPONSE.
MOVE LENGTH OF RESPONSE TO BAQ-RESPONSE-LEN.
```

6. The communication stub was called as shown below.

```
CALL COMM-STUB-PGM-NAME USING
BY REFERENCE   API-INFO-API1
BY REFERENCE   BAQ-REQUEST-INFO
BY REFERENCE   BAQ-REQUEST-PTR
BY REFERENCE   BAQ-REQUEST-LEN
BY REFERENCE   BAQ-RESPONSE-INFO
BY REFERENCE   BAQ-RESPONSE-PTR
BY REFERENCE   BAQ-RESPONSE-LEN
```

7. The value of the response was retrieved when the API call was returned as shown below.

```
IF BAQ-SUCCESS
DISPLAY "The program call was successful."
DISPLAY "DATA: " outputData2
```

6.3 z/OS Connect Enterprise Edition

To establish communication and execute transactions on IMS, several resources need to be modified on the z/OS Connect EE install configuration. This section covers the types of resources that need to be modified.

6.3.1 Configure Request Endpoint

To route an API call request issued by the client application from z/OS Connect EE to the request endpoint, z/OS Connect EE needs to be configured to establish a connection with the request endpoint.

In server.xml, the `zosconnect_endpointConnection` XML element was modified to set the value of the host and port attribute to the host and port of the request endpoint or web server as shown below in Figure 27.

```
<zosconnect_endpointConnection id="BluemixAPIConnect"
host="http://1.2.3.4" port="2222"/>
```

Figure 27: Server.xml Request Endpoint Connection

6.3.2 API Requester

To enable z/OS applications to call REST APIs, an intermediary called API requester was created and deployed on z/OS Connect EE. The following artifacts were needed for an API requester:

- API requester archive (.ara): A file needed by z/OS Connect EE to perform data mapping and transformation for REST API calls by IMS application.
- API information file: A file that contains information on the API that an IMS application calls.
- Data structures: Request and response data structures for each operation in an API.

The build toolkit was used to generate artifacts for the API requester. For each API, an .ara file and a summary report are generated. For each API operation, an API information file along with its request and response data structures are generated. The .ara file is then deployed to z/OS Connect EE.

Figures 28 and 29 show examples of the request and response data structures where <$PAYLOAD_SIZE$> is replaced with 1024, 8192 and 30720 according to the desired message size for the test.

```
* ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* This file contains the generated language structure(s) for
*  request JSON schema 'request_request.json'.
* This structure was generated using 'DFHJS2LS' at mapping level
*  '4.3'.
*
*
*      06 ReqBody.
*
*
* JSON schema keyword 'ReqBody->size' is optional. The number of
*  instances present is indicated in field 'Xsize-num'.
* There should be at least '0' instance(s).
* There should be at most '1' instance(s).
*        09 Xsize-num                     PIC S9(9) COMP-5 SYNC.
*
*
* Comments for field 'Xsize':
* This field represents the value of JSON schema keyword
*  'ReqBody->size'.
* JSON schema description: Size.
* JSON schema type: 'int'.
* JSON schema keyword 'maximum' value: '2147483647'.
* JSON schema keyword 'minimum' value: '-2147483648'.
*        09 Xsize                         PIC S9(9) COMP-5 SYNC.
*
*
* JSON schema keyword 'ReqBody->payload' is optional. The number
*  of instances present is indicated in field 'payload-num'.
* There should be at least '0' instance(s).
* There should be at most '1' instance(s).
*        09 payload-num                   PIC S9(9) COMP-5 SYNC.
*
*
*        09 payload.
*
* Comments for field 'payload2':
* This field represents the value of JSON schema keyword
*  'ReqBody->payload'.
* JSON schema description: Payload.
* JSON schema type: 'string'.
* JSON schema keyword 'minLength' value: '0'.
* JSON schema keyword 'maxLength' value: '<$PAYLOAD_SIZE$>'.
* This field contains a varying length array of characters or
*  binary data.
*          12 payload2-length              PIC S9999 COMP-5
*  SYNC.
*          12 payload2                     PIC X(<$PAYLOAD_SIZE$>).
*
*
* ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

06 ReqBody.
```

```
09 Xsize-num                     PIC S9(9) COMP-5 SYNC.

09 Xsize                         PIC S9(9) COMP-5 SYNC.

09 payload-num                   PIC S9(9) COMP-5 SYNC.

09 payload.
12 payload2-length               PIC S9999 COMP-5
SYNC.
12 payload2                      PIC X(<$PAYLOAD_SIZE$>).
```

Figure 28: Request Message Language Structure

```
* ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* This file contains the generated language structure(s) for
*  response JSON schema 'request_200_response.json'.
* This structure was generated using 'DFHJS2LS' at mapping level
*  '4.3'.
*
*
*      06 RespBody.
*
*
* JSON schema keyword 'RespBody->outputError' is optional. The
*  number of instances present is indicated in field
*  'outputError-num'.
* There should be at least '0' instance(s).
* There should be at most '1' instance(s).
*        09 outputError-num               PIC S9(9) COMP-5 SYNC.
*
*
*        09 outputError.
*
* Comments for field 'outputError2':
* This field represents the value of JSON schema keyword
*  'RespBody->outputError'.
* JSON schema description: Error message.
* JSON schema type: 'string'.
* This field contains a varying length array of characters or
*  binary data.
*          12 outputError2-length           PIC S9999 COMP-5
*  SYNC.
*          12 outputError2                  PIC X(255).
*
*
* JSON schema keyword 'RespBody->outputData' is optional. The
*  number of instances present is indicated in field
*  'outputData-num'.
* There should be at least '0' instance(s).
* There should be at most '1' instance(s).
*        09 outputData-num                PIC S9(9) COMP-5 SYNC.
*
*
*        09 outputData.
*
```

```
* Comments for field 'outputData2':
* This field represents the value of JSON schema keyword
*  'RespBody->outputData'.
* JSON schema description: Data.
* JSON schema type: 'string'.
* JSON schema keyword 'minLength' value: '0'.
* JSON schema keyword 'maxLength' value: '<$PAYLOAD_SIZE$>'.
* This field contains a varying length array of characters or
*  binary data.
*         12 outputData2-length            PIC S9999 COMP-5
*  SYNC.
*         12 outputData2                   PIC X(<$PAYLOAD_SIZE$>).
*
*
* ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

06 RespBody.

09 outputError-num              PIC S9(9) COMP-5 SYNC.

09 outputError.
12 outputError2-length          PIC S9999 COMP-5
SYNC.
12 outputError2                 PIC X(255).

09 outputData-num               PIC S9(9) COMP-5 SYNC.

09 outputData.
12 outputData2-length           PIC S9999 COMP-5
SYNC.
12 outputData2                  PIC X(<$PAYLOAD_SIZE$>).
```

Figure 29: Response Message Language Structure

Figure 30 shows an example of the API information file.

The API information file describes the API name, the API path, exposed API methods and other information that is required to communicate with the communication stub. To call the REST API, an IMS application must include the generated API information file along with the request and response message structures as described above.

```
03 BAQ-APINAME                PIC X(255)
VALUE 'Request-REST-Service_1.0.0'.
03 BAQ-APINAME-LEN            PIC S9(9) COMP-5 SYNC
VALUE 26.
03 BAQ-APIPATH                PIC X(255)
VALUE '%2Fims-request%2Fapi%2FRequestServices%2Frequest'.
03 BAQ-APIPATH-LEN            PIC S9(9) COMP-5 SYNC
VALUE 48.
03 BAQ-APIMETHOD              PIC X(255)
VALUE 'PUT'.
03 BAQ-APIMETHOD-LEN          PIC S9(9) COMP-5 SYNC
VALUE 3.
```

Figure 30: API information file

6.4 Web Server

The web server is a light weight web application server, also known as RESTful endpoint, that receives API requests and sends an echo response back to the IMS application via z/OS Connect EE.

6.5 Security

AT-TLS was enabled on the z/OS LPAR where z/OS Connect EE server was started by editing the TCPIP configuration member (located in SYS1.TCPPARMS) as shown below.

```
TCPCONFIG
  UNRESTRICTLOWPORTS
  TCPSENDBFRSIZE 262144
  TCPRCVBUFRSIZE 262144
  TTLS
```

The ICSF address space was first started before starting the PAGENT address space was started next to activate the AT-TLS policy.

6.6 REST Driver

A stand-alone Java application was used to drive the IMS application which in turn issues REST API requests to the web server through z/OS Connect EE using think time zero. Each request calls a REST API with 1KB, 8KB or 30KB input message sizes and receives a response.

6.6.1 Tuning

To take advantage of persistent connections, IMS dependent regions were changed to PWFI (Pseudo Wait-for-Input) = Y to minimize scheduling overhead.

6.6.2 Execution

The REST driver was executed in a z/OS Java batch job on a separate logical partition (LPAR) from the z/OS Connect EE. The number of concurrent users varied from one to 30.

Each time a response was received from an IMS application, the driver compares the JSON payload with the expected JSON payload to ensure the correct data was being received.

The driver also has the capability of printing out statistics such as the current transaction rate and average client response time during runtime through a print statistics command. Figure 31 shows sample output of the print statistics command while running with 15 users.

```
Printing statistics.
 Pool Name: ZCEEREQUESTERTHREAD.
 -->Class Name: ims.perf.util.driver.ThreadPoolProperties.
 -->Active Threads: 15.
 Statistics for: ZceeRequesterThread.
 ->Tx Rate: 5298.0456.
 ->Avg Resp Time (ms): 2.4719.
 Printed statistics in 0.017 seconds.
```

Figure 31: REST Driver Output from Print Statistics Command

6.7 Requester Performance Results

The z/OS Connect EE requester performance was evaluated by examining how the transaction rate, client response time and Service Time per transaction behaved under stress.

In this section, we investigate how scaling the number of concurrent users would affect the transaction rate, client response time and Service Time per transaction. When a transaction is invoked, the IMS application sends an API request to z/OS Connect EE. z/OS Connect EE then sends a request to the external API on web server and returns its response to IMS. For each I/O message size (i.e. 1KB, 8KB and 30KB), the number of concurrent users were scaled from 1, 2, 5, 7, 10, 13, 15, 20, 25 and 30.

6.7.1 Transaction Rate

Figure 32 shows the effect on throughput, measured in Transactions Per Second (TPS), as the number of users increase:

- With 1KB I/O messages, throughput peaked at about 5,100 TPS before stabilizing.
- With 8KB I/O messages, throughput peaked at about 3,800 TPS before stabilizing.
- Lastly, with 30KB I/O messages, throughput peaked at about 2,100 TPS before stabilizing.

We found as the number of users increase, the transaction rate also increases until reaching its peak and stabilizing. As we increased the number of users with larger I/O messages, the transaction rate showed the same pattern, but peaked and stabilized with much less users at lower transaction rates.

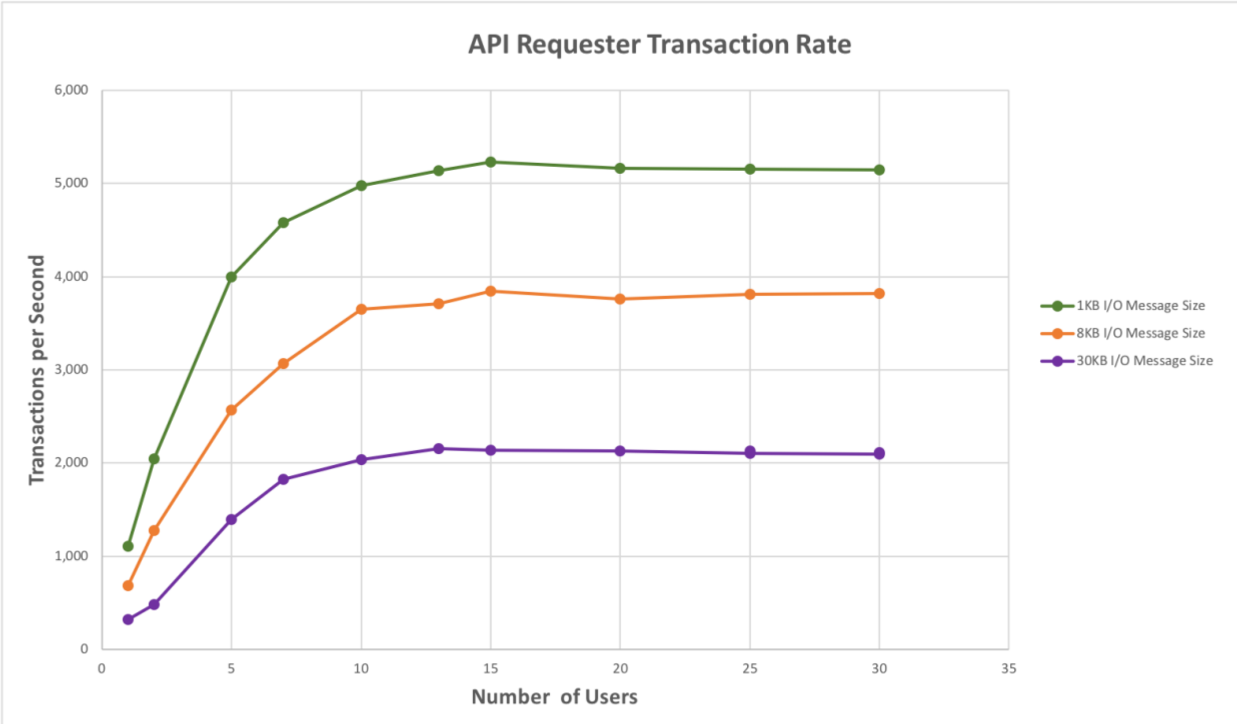

Figure 32: Scaling the Number of users Requester Scenario – Transaction Rate

6.7.2 Client Response Time

Figure 33 shows the effect on client response times measured in milliseconds (ms), as the number of users increase:

- With 1KB I/O messages, the client response time gradually increased from less than one ms to four ms as the number of users increased.
- With 8KB I/O messages, the client response time increased at a greater rate from one ms to five ms as the number of users increased.
- With 30KB I/O messages, the client response time increased at a greater rate from three ms to 14 ms as the number of users increased.

We found as the number of users increase, the client response time increases linearly. When increasing the number of users with larger I/O message sizes, the client response time continues to increase linearly but at a sharper incline.

Figure 33: Scaling the Number of users Requester Scenario – Client Response Time

6.7.3 Service Time per Transaction

Figure 34 shows the effect on z/OS Connect EE Service Time per transaction, measured microseconds (µs), as the number of users increase:

- With 1KB I/O messages, the z/OS Connect EE Service Time per Transaction ranged between 467-486 µs as the number of users increased.
- With 8KB I/O messages, the z/OS Connect EE Service Time per Transaction ranged between 807-966 µs as the number of users increased.
- With 30KB I/O messages, the z/OS Connect EE Service Time per Transaction ranged between 2,176-2,334 µs as the number of users increased.

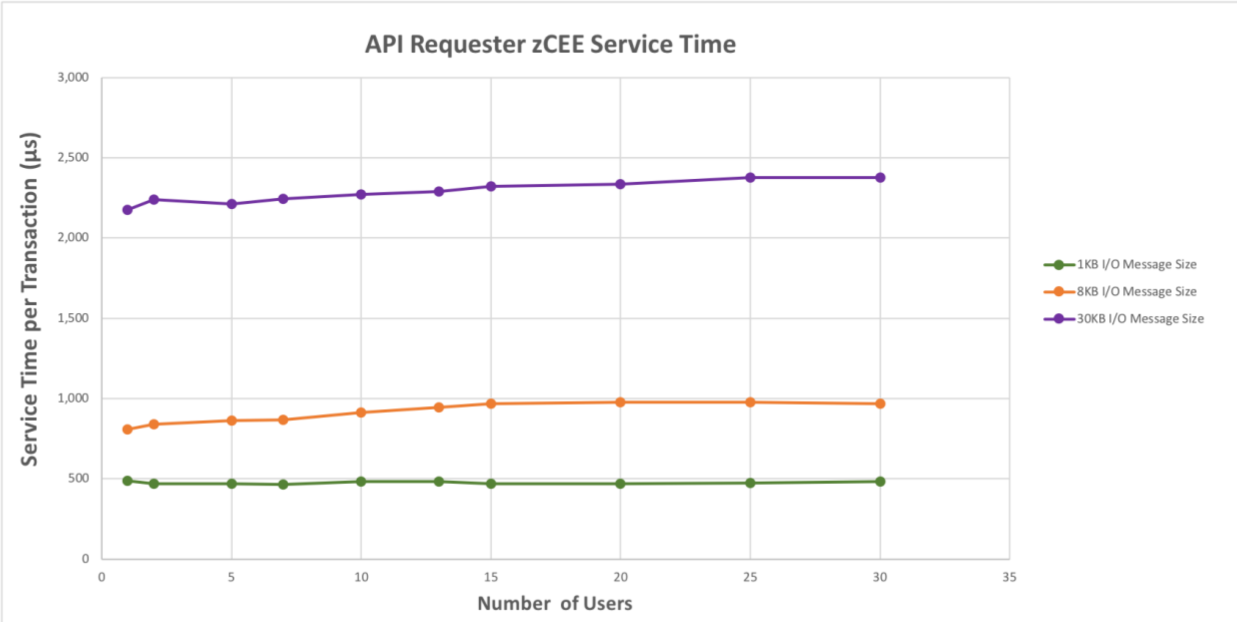

Figure 34: Scaling the Number of users Requester Scenario– Service Time per Transaction

Figure 35 shows the z/OS Connect EE Service Time per transaction for 10 users.

- With 1KB I/O messages, the total Service Time per transaction was 654 µs with z/OS Connect EE IMS SP contributing 481 µs of the total.
- With 8KB I/O messages, the total Service Time per transaction was 1,134 µs with z/OS Connect EE IMS SP contributing 913 µs of the total.
- With 30 KB I/O messages, the total Service Time per transaction was 2,649 µs with z/OS Connect EE IMS SP contributing 2,270 µs of the total.

With any number of users from 1 to 30, we found Service Time per transaction stays fairly consistent. Of the total Service Time, the majority originated from the z/OS Connect EE. The same pattern held true when increasing the I/O message size, but at a greater proportion.

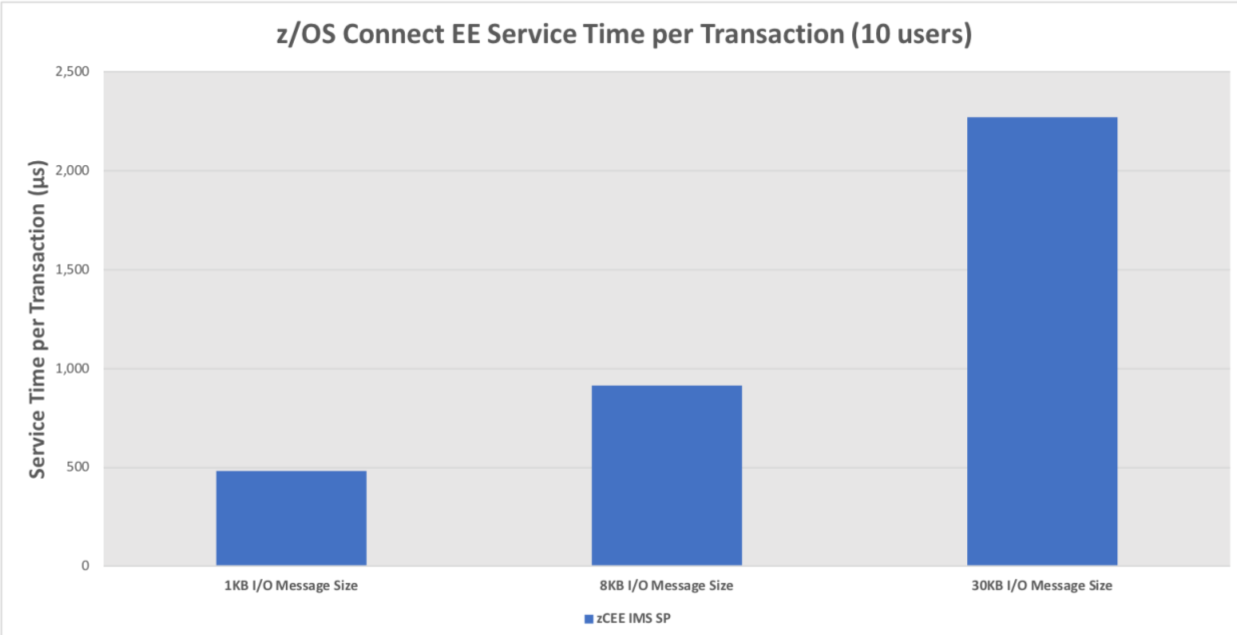

Figure 35: z/OS Connect EE Service Time per Transaction for 10 Users – Requester Scenario

Figure 36 shows the projected zIIP offload for z/OS Connect EE IMS SP if the same workload was executed with zIIPs enabled. The results show that approximately 99% of the processing is zIIP eligible as z/OS Connect EE is a Java based product.

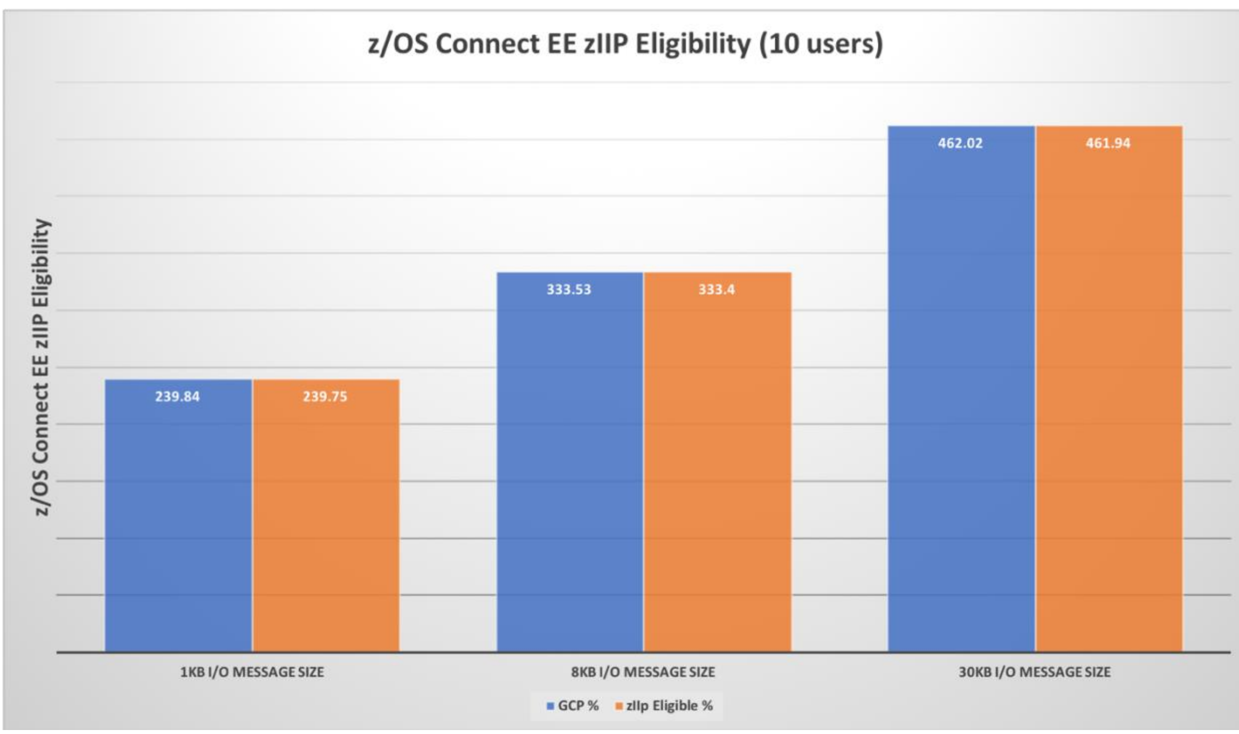

Figure 36: Projected z/OS Connect EE zIIP Eligibility for 10 Users– Requester Scenario

7 Conclusion

The z/OS Connect EE V3.0.10 delivers excellent performance and scalability with IMS V15 for both provider and requester scenarios. In addition, the z/OS Connect EE processing including data transformation from JSON to bytes and back is eligible for zIIP offload.

Overall, z/OS Connect EE and IMS 15 provide a fast, seamless and reliable solution for the API economy.

8 Resources

IBM z/OS Connect Enterprise Edition Announce and Information:

- **Introduction Page:** https://developer.ibm.com/mainframe/whats-new-zos-connect-ee/
- **Technical Specs:** https://www.ibm.com/support/knowledgecenter/SS4SVW_3.0.0/com.ibm.zosconnect.doc/backmatter/pdf.pdf

IMS 15 Information Center Link:
https://www.ibm.com/support/knowledgecenter/SSEPH2_15.1.0/com.ibm.ims15.doc/ims_product_landing_v15.html

IMS V15 Announcement Letter:
https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?infotype=an&subtype=ca&appname=gpateam&supplier=897&letternum=ENUS217-398

IMS Home Page: https://www.ibm.com/it-infrastructure/z/ims

9 Appendix A – Measurement Data Definitions

9.1 IMS Performance Analyzer for z/OS

Below are the IMSPA control statements that were used for processing IMS log data:

```
IMSPALOG      INPUTDD(LOGIN),
              OUTPUTDD(RPTOUT),
              PAGESIZE(60),
              PRINTAT(STOP)
IMSPALOG      ANALYSIS(
              TRANCODE,
              INCL(MSGSW),
              INCL(APPC)),
              INCL(BMP))
IMSPALOG      CPUR(
              DDNAME(CPURDD),
              R,T)
IMSPALOG      AVAIL(
              DDNAME(AVALDD),
              REGION,
              TRANCODE,
              PROGRAM)
IMSPALOG      SUMMARY(INCL(TRANCODE(*)),PRECISION(6),
              COMPLVL(3),
              FIELDS(TRANCODE,TRANCNT,INPUTQ,PROCESS,
              SYNCELAP,OUTPUTQ,DBIOCALL,DBIOTIME,
              LOCKTIME,CPUTIME,CPUZAAP))
IMSPALOG      IRUR(
              DDNAME(IRURDD),
              INTERVAL(0))
IMSPALOG      MSGQ(DDNAME(MQURDD),INTERVAL(100))
IMSPALOG      FPANALYSIS(TRANCODE)
IMSPALOG      FPIRUC(RESUSAGE,BUFFER)
IMSPALOG      FPRGNO
IMSPALOG      FPDBCALL
IMSPALOG      FPDBUPD
IMSPALOG      FPVSO
IMSPALOG      EXECUTE
```

9.2 RMF ERB Members

Below are the contents of the ERBRMFxx PARMLIB member used for RMF:

```
/*****************************************************************/
/* PART 1: MEASUREMENTS                                          */
/*****************************************************************/
  CACHE                    /* CACHE STATISTICS                   */
  CHAN                     /* CHANNEL STATISTICS                 */
  CPU                      /* CPU STATISTICS                     */
  CRYPTO                   /* CRYPTO HARDWARE                    */
  DEVICE(NODASD)           /* DIRECT ACCESS DEVICES MEASURED     */
  DEVICE(NMBR(5000:54FF))
  DEVICE(NOTAPE)           /* TAPE DEVICES MEASURED              */
  DEVICE(NOCHRDR)          /* CHARACTER READER DEVICES MEASURED  */
  DEVICE(NOUNITR)          /* UNIT RECORD DEVICES MEASURED       */
  DEVICE(NOCOMM)           /* COMMUNINCATION DEVICES MEASURED    */
  DEVICE(NOGRAPH)          /* GRAPHICS DEVICES MEASURED          */
  DEVICE(NONMBR)           /* NO SELECTION BY DEVICE NUMBERS     */
  DEVICE(NOSG)             /* NO SELECTION BY STORAGE GROUPS     */
NOENQ                      /* ENQUEUES MEASURED                  */
NOESS                      /* ENTERPRISE DISK SYSTEMS MEASURED   */
NOFCD                      /* FICON DIRECTOR MEASURED            */
  IOQ(NODASD)              /* DASD I/O QUEUEING MEASURED         */
  IOQ(NMBR(5000:54FF))
  IOQ(NOTAPE)              /* TAPE I/O QUEUEING MEASURED         */
  IOQ(NOCHRDR)             /* CHARACTER READER I/O QUEUING       */
  IOQ(NOUNITR)             /* UNIT RECORD DEVICE I/O QUEUING     */
  IOQ(NOCOMM)              /* COMMUNICATION I/O QUEUING          */
  IOQ(NOGRAPH)             /* GRAPHICS DEVICE I/O QUEUING        */
  IOQ(NONMBR)              /* NO SELECTIVITY BY LCU NUMBERS      */
  PAGESP                   /* PAGE/SWAP DATASET STATISTICS       */
  PAGING                   /* PAGING DATA                        */
NOTRACE                    /* TRACE                              */
  VSTOR(IFBCTR01,IFBDLR01,IFBDBR01,HWS1, MP010001)
  WKLD                     /* WORKLOAD MANAGER DATA              */
/*****************************************************************/
/* PART 2: TIMING                                                */
/*****************************************************************/
  CYCLE(1000)              /* SAMPLE EVERY 1000 MILLISECONDS     */
  INTERVAL(2M)             /* REPORT INTERVAL                    */
NOSTOP                     /* REMAIN ACTIVE UNTIL OPERATOR STOP  */
  SYNC(00)                 /* INTERVAL SYNC                      */
/*****************************************************************/
/* PART 3: REPORTING / RECORDING OF DATA                         */
/*****************************************************************/
NOOPTIONS                  /* OPERATOR MAY NOT CHANGE OPTIONS    */
  RECORD                   /* WRITE SMF RECORDS EVERY INTERVAL   */
  REPORT(REALTIME)         /* WRITE REPORTS EACH INTERVAL        */
  SYSOUT(E)                /* INTERVAL REPORTS TO CLASS E        */
/*****************************************************************/
/* PART 4: USER EXITS                                            */
/*****************************************************************/
  EXITS                    /* TAKE USER EXITS                    */
```

10 Appendix B – System Parameters and Specifications

10.1 IMS Startup (DFSPBxxx)

Below are the contents of the IMS startup member:

```
AOIP=2047M,         AOIS=N,           ARC=0,            ARMRST=Y,         AUTO=N,
CIOP=2047M,         CMDMCS=Y,         CPLOG=16M,        CRC=@,
DBRCNM=IFBDBR01,    DESC=7,           EPCB=600,         FIX=01,           FMTO=D,
IMSID=FR01,         ISIS=N,           MAXPST=500,       MCS=14,
PRDR=IMSPRDR,       PSBW=4000K,       PST=150,          PSWDC=R,
RCLASS=IMS,         RES=Y,            RRS=N,            RSRMBR=00,        SPM=01,
SRCH=0,             SUF=M,            SVC2=203,         TRACK=NO,         TSR=L,
VSPEC=01,           WADS=S,           WKAP=800K,        YEAR4=N,
ALOT=1440,          AOI1=N,           APPC=N,           APPCSE=N,
ASOT=1440,          BSIZ=0,           CCTCVCAN=N,       CMDP=2047M,
CSAPSB=3000K,       DBBF=24000,       DBFP=0,           DBFX=4800,
DBWP=1200K,         DC=000,           DFSDF=001,        DLINM=IFBDLR01,
DLIPSB=6000K,       DLQT=60,          DMB=3000K,        DMHVF=0,          DSCT=M,
DYNP=2047M,         EMHB=30M,         EMHL=256,         ETO=Y,            EXVR=Y,
FBP=1200K,          FESTIM=1,         FP=Y,             FPOPN=N,
FPRLM=N,            FPWP=2047M,       FRE=800,          GRNAME=OTMAGRP,
HIOP=2047M,         HSBMBR=00,        IOVFI=7200,       IRLM=Y,
IRLMNM=IR01,        LGMSGSZ=0,        LGNR=10,          LHTS=256,
LSO=S,              LTERM=Y,          LUMC=2047M,       LUMP=2047M,
MSDB=.,             NHTS=256,         NLXB=0,           ODBASE=N,
OTHR=150,           OTMA=Y,           OTMAASY=N,        OTMAMD=N,
OTMANM=IFBCTR01,    OTMASE=N,         OTMASP=N,
PIINCR=64K,         PIMAX=1M,         PSB=3000K,        QBUF=4000,
QBUFHITH=1,         QBUFLWTH=1,       QBUFMAX=0,        QBUFSZ=0,
QTL=50,             QTU=75,           RCF=N,            RCFTCB=1,
RECA=50,            RECASZ=2048,      RVFY=N,           SAV=999,
SGN=N,              SHMSGSZ=0,        SOD=0,            SVSODR=NONE,
TCORACF=N,          TRN=N,            VAUT=1,           UHTS=256,
FESEXIT=Y,          MFSEXITF=1,       MFSEXITS=126,     USERMSGS=Y
```

10.2 IMS Storage Pool (DFSSPMxx)

Below are the contents of the IMS storage pool member:

```
FPL=CESS,(128,32,16,Y),(256,32,16,Y),(512,32,16,Y)
FPL=CESS,(1024,32,16,Y),(2048,32,16,Y),(4096,16,8,Y)
FPL=CESS,(8192,8,4,Y),(16384,4,2,Y)
FPL=FPWP,(256,64,32,Y),(512,64,32,Y),(1024,32,16,Y)
FPL=FPWP,(2048,32,16,Y),(4096,16,8,Y),(8192,16,8,Y)
FPL=FPWP,(16384,4,2,Y)
FPL=HIOP,(216,50,10,Y),(384,800,50,Y),(552,500,100,Y)
FPL=HIOP,(968,500,100,Y),(1280,30,10,Y),(1640,200,50,Y)
```

```
FPL=HIOP,(3072,400,100,Y),(4096,400,100,Y)
FPL=CIOP,(192,128,16,Y),(448,32,16,Y),(1024,32,16,Y)
FPL=CIOP,(2048,32,16,N),(8192,16,8,N)
```

10.3 IMS VSPEC (DFSVSMxx)

Below are the contents of the IMS VSPEC member:

```
VSRBF=2048,2000
VSRBF=4096,4000
VSRBF=8192,2000
VSRBF=12288,600
VSRBF=20480,500
IOBF=(2048,20000,Y,Y,OA2K,)
IOBF=(2048,20000,Y,Y,OB2K,)
IOBF=(2048,20000,Y,Y,OC2K,)
IOBF=(2048,20000,Y,Y,OD2K,)
IOBF=(4096,4000,Y,Y,O04K,)
IOBF=(8192,1000,Y,Y,O08K,)
IOBF=(12288,600,Y,Y,O12K,)
```

10.4 IMS Connect (HWSCFGxx)

Below are the contents of the IMS Connect configuration member:

```
HWS(ID=HWS1,RACF=N,RRS=N)
TCPIP=(HOSTNAME=TCPIP, ECB=Y, EXIT=(HWSSMPL1, HWSSOAP1),
  IPV6=N,MAXSOC=24022,NODELAY=Y, PORTID=(4441))
DATASTORE(ID=FR01,GROUP=OTMAGRP,MEMBER=HWS1,TMEMBER=IFBCTR01)
IMSPLEX(MEMBER=HWS1,TMEMBER=PLEX1)
```

10.5 z/OS Connect Server Configuration

Below are the contents of the z/OS Connect EE service configuration (Server.xml) for provider scenario:

```
<server description="IMS Mobile Server for Managing and Accessing IMS
zConnect Services">
<include
location="/ZCEE310/var/zosconnect/servers/imsmobile/resources/imsmobile-
config/interactions/ims-interactions.xml" optional="true"/>
<include
location="/ZCEE310/var/zosconnect/servers/imsmobile/resources/imsmobile-
config/connections/ims-connections.xml" optional="true"/>
<include
location="/ZCEE310/var/zosconnect/servers/imsmobile/resources/imsmobile-
config/services/ims-services.xml" optional="true"/>
```

```
<include location="/ZCEE310/var/zosconnect/servers/imsmobile/ims-admin-
services.xml" optional="true"/>

<variable name="imsmobile.install.dir" value="/ZCEE310/imsmobile/wlp-ext"/>

<zosconnect_services
location="/ZCEE310/var/zosconnect/servers/imsmobile/resources/zosconnect/serv
ices" updateTrigger="disabled">
</zosconnect_services>

<zosconnect_zosConnectAPIs
location="/ZCEE310/var/zosconnect/servers/imsmobile/resources/zosconnect/apis
" updateTrigger="disabled">
  <zosConnectAPI name="vartranapi"/>
</zosconnect_zosConnectAPIs>

      <!-- Enable features -->
      <featureManager>
            <feature>imsmobile:imsmobile-2.0</feature>
            <feature>zosconnect:zosconnect-2.0</feature>
      </featureManager>

      <httpEndpoint host="*" httpPort="10443" httpsPort="7089"
id="defaultHttpEndpoint"/>

      <!-- Tuning -->
      <httpOptions maxKeepAliveRequests="-1"/>

      <applicationMonitor updateTrigger="mbean"/>

    <imsmobile_imsServiceManager imsRegistryHome="./registry"
imsTechnicalGroup="SYS1" imsTechnicalID="IMSGUEST"/>

      <!-- Tracing definition -->
<logging maxFileSize="20" maxFiles="10" traceFileName="imsmobile.log"
traceFormat="BASIC" traceSpecification="com.ibm.j2ca.RAIMSTM=OFF"/>

      <!-- Basic authentication registry definition -->
    <basicRegistry id="basic1" realm="zosConnect">
      <user name="omvsadm" password="{xor}PjMzbiw7KjE="/>
           <user name="imsmobile" password="{xor}NjIsMm89bjM6"/>
      </basicRegistry>

      <!-- Keystore definition ECC -->
      <keyStore id="defaultKeyStore" password="GiveMeLiberty"/>

      <!-- SSL definition -->
      <ssl enabledCiphers="TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA"
id="defaultSSLConfig" keyStoreRef="defaultKeyStore" sslProtocol="TLSv1"/>

      <!-- Allow fail over to basic-auth. Right now we do CLIENT-CERT -->
      <webAppSecurity allowFailOverToBasicAuth="true"/>

      <!-- Security Role definition for authentication -->
```

```
	<authorization-roles id="zos.connect.access.roles">
		<security-role name="zosConnectAccess">
			<user name="omvsadm"/>
			<user name="imsmobile"/>
		</security-role>
	</authorization-roles>

</server>
```

Below are the contents of the z/OS Connect EE service configuration (Server.xml) for requester scenario:

```
<server description="IMS Mobile Server for Managing and Accessing IMS
zConnect Services">
<include
location="/zcee/var/zosconnect/servers/imsmobile/resources/imsmobile-
config/interactions/ims-interactions.xml" optional="true"/>
<include
location="/zcee/var/zosconnect/servers/imsmobile/resources/imsmobile-
config/connections/ims-connections.xml" optional="true"/>
<include
location="/zcee/var/zosconnect/servers/imsmobile/resources/imsmobile-
config/services/ims-services.xml" optional="true"/>
<include location="/zcee/var/zosconnect/servers/imsmobile/ims-admin-
services.xml" optional="true"/>

	<variable name="imsmobile.install.dir" value="/zcee/imsmobile/wlp-
ext"/>

	<!-- Enable features -->
	<featureManager>
		<feature>imsmobile:imsmobile-2.0</feature>
		<feature>zosconnect:apiRequester-1.0</feature>
	</featureManager>

	<zosconnect_endpointConnection id="BluemixAPIConnect"
host="http://1.2.3.4" port="2222"/>

	<zosconnect_apiRequesters>
		<apiRequester name="reqpgm"/>
	</zosconnect_apiRequesters>

	<zosconnect_zosConnectManager requireSecure="false"
requireAuth="false"/>

	<httpEndpoint host="*" httpPort="10443" httpsPort="7089"
id="defaultHttpEndpoint"/>

	<!-- Tuning -->
	<httpOptions maxKeepAliveRequests="-1"/>

	<applicationMonitor updateTrigger="mbean"/>
```

```
    <imsmobile_imsServiceManager imsRegistryHome="./registry"
imsTechnicalGroup="SYS1" imsTechnicalID="IMSGUEST"/>

      <!-- Basic authentication registry definition -->
    <basicRegistry id="basic1" realm="zosConnect">
      <user name="omvsadm" password="{xor}PjMzbiw7KjE="/>
          <user name="imsmobile" password="{xor}NjIsMm89bjM6"/>
      </basicRegistry>

      <!-- Keystore definition (client certificate) -->
      <keyStore id="defaultKeyStore" password="imsmobile" location="P12DER"
type="PKCS12"/>

      <!-- Define a trust store (certificate authority) -->
      <keyStore id="defaultTrustStore" password="password”
location="key.jks"/>

      <!-- Define the SSL configuration -->
      <ssl id="defaultSSLConfig" keyStoreRef="defaultKeyStore"
trustStoreRef="defaultTrustStore" sslProtocol="SSL_TLSv2"/>

      <!-- Allow fail over to basic-auth. Right now we do CLIENT-CERT -->
      <webAppSecurity allowFailOverToBasicAuth="true"/>

      <!-- Security Role definition for authentication -->
      <authorization-roles id="zos.connect.access.roles">
            <security-role name="zosConnectAccess">
                  <user name="omvsadm"/>
                  <user name="imsmobile"/>
            </security-role>
      </authorization-roles>

</server>
```

Below are the contents of the Swagger 2.0 document for requester scenario:

```
{
   "swagger":"2.0",
   "info":{
      "version":"1.0.0",
      "title":"Request REST Service"
   },
   "host":"1.2.3.4:4444",
   "basePath":"/ims-request/api",
   "tags":[
      {
         "name":"RequestServices"
      }
   ],
   "schemes":[
      "http"
   ],
   "paths":{
```

```
    "/RequestServices/request":{
      "put":{
        "tags":[
          "RequestServices"
        ],
        "summary":"Request data",
        "description":"This service returns the requested data.",
        "operationId":"request",
        "consumes":[
          "application/json"
        ],
        "produces":[
          "application/json"
        ],
        "parameters":[
          {
            "in":"body",
            "name":"body",
            "required":false,
            "schema":{
              "$ref":"#/definitions/Input"
            }
          }
        ],
        "responses":{
          "200":{
            "description":"successful operation",
            "schema":{
              "$ref":"#/definitions/Output"
            }
          }
        }
      }
    }
  },
  "definitions":{
    "Output":{
      "type":"object",
      "properties":{
        "outputError":{
          "type":"string",
          "description":"Error message"
        },
        "outputData":{
          "type":"string",
          "description":"Data",
          "maxLength":1024
        }
      },
      "description":"Output"
    },
    "Input":{
      "type":"object",
      "properties":{
```

```
            "size":{
               "type":"integer",
               "format": "int32",
               "description":"Size",
               "maxLength":1024
            },
            "payload":{
               "type":"string",
               "description":"Paylod",
               "maxLength":1024
            }
         },
         "description":"Input"
      }
   }
}
```

10.6 z/OS Connect Proc

Below are the contents of z/OS Connect EE proc:

```
// SET ZCONHOME='/ZCEE310'
// SET PARMS='imsmobile'
//*
//ZCON     EXEC PGM=BPXBATSL,REGION=0M,MEMLIMIT=NOLIMIT,
//         PARM='PGM &ZCONHOME./bin/zosconnect start &PARMS.'
//STDOUT   DD   SYSOUT=*
//STDERR   DD   SYSOUT=*
//STDIN    DD   DUMMY
//STDENV   DD   *
_BPX_SHAREAS=YES
JAVA_HOME=/java/v800/bit64/J8.0_64
WLP_USER_DIR=/ZCEE310/var/zosconnect
JVM_OPTIONS=-Xms4096m -Xmx4096m
```

10.7 IRLM Configuration

Below are the contents of the IRLM procedure:

```
//IRLM1    JOB MSGLEVEL=1,MSGCLASS=E,CLASS=A,
//   LINES=999999,TIME=1440,REGION=0M,
//   MEMLIMIT=NOLIMIT
//*
//IRLM1    PROC IRLMNM=IR01,IRLMID=1,SCOPE=LOCAL,
//   DEADLOK='500,1',MAXCSA=64,
//   PC=YES,MAXUSRS=8,GROUP=IRLMGRP1,
//   LOCKTAB=IRLMLOCKTBL1
//IRLM     EXEC PGM=DXRRLM00,
//             PARM=(&IRLMNM,&IRLMID,&SCOPE,&DEADLOK,&MAXCSA,&PC,
//             &MAXUSRS,&GROUP,&LOCKTAB)
```

```
//STEPLIB  DD DISP=SHR,
//      DSN=IMSPERF.IRLM23.RESLIB
//PROCLIB  DD DISP=SHR,
//      DSN=IMSPERF.JS.AUTOSRVR.PROCLIB
//SYSABEND DD SYSOUT=*
//*
//         PEND
//IRLM1    EXEC IRLM1
```

10.8 WLM Service Class Definitions

The following table lists details of the main WLM service class definitions:

Service Class Description	Importance	Execution Velocity
IRLM	1	90
IMS Control Region	1	75
IMS DBRC Region	1	80
IMS DL/I Region	1	70
IMS Connect	1	90
IMS MPP regions	2	90
z/OS Connect EE Server	1	90

11 Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBMproducts. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:
© (your company name) (year).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. _enter the year or years_.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

The terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside of your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.