z/OS Connect Enterprise Edition V2.0

Performance Summary for API Mapping

Version 1.1

26th August 2016

Alan Hollingshead
alan_hollingshead@uk.ibm.com

z/OS Connect EE
IBM UK Laboratories
Hursley Park
Winchester
Hampshire
SO21 2JN

Table of Contents

Illustration Index

1.1 Notices

This report is intended for Architects, Systems Programmers, Analysts and Programmers wanting to understand the performance characteristics of z/OS Connect EE V2.0. The information is not intended as the specification of any programming interfaces that are provided by z/OS Connect EE.

It is assumed that the reader is familiar with the concepts and operation of z/OS Connect EE V2.0.

References in this report to IBM products or programs do not imply that IBM intends to make these available in all countries in which IBM operates.

Information contained in this report has not been submitted to any formal IBM test and is distributed "asis". The use of this information and the implementation of any of the techniques is the responsibility of the customer. Much depends on the ability of the customer to evaluate this data and project the results to their operational environment.

The performance data contained in this report was measured in a controlled environment and results obtained in other environments may vary significantly.

1.2 Trademarks and service marks

All statements regarding IBM plans, directions, and intent are subject to change or withdrawal without notice.

1.3 Document History

1.0 First edition (8th January 2016)

1.1 Minor updates to descriptions of payload format (26th August 2016)

1.4 Terminology

Cost per transaction (ms)	-	CPU usage per transaction, in milliseconds
CP	-	Central Processor
CPU %	-	Percentage of CPU time used by transactions running on general purpose processors
EE	-	Enterprise Edition
GCP	-	General purpose Central Processor
PID	-	Product Identification Number
RMF	-	Resource Measurement Facility
SMF	-	System Management Facility
SSL	-	Secure Sockets Layer
Simulated Service Provider	-	A service provider written for the purpose of these performance tests used to simulate a z/OS subsystem
TPS	-	Number of Transactions Per Second
TT	-	Think Time in seconds. The time between individual requests.
Workload Driver	-	An application written for the purpose of these performance tests to simulate multiple simultaneous client requests
zIIP	-	IBM z Systems Integrated Information Processor

2. Overview

This report contains performance measurements for z/OS Connect EE V2.0, program number (PID) 5655-CEE.

IBM z/OS Connect EE V2.0 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, together with necessary detail to invoke underlying services in the z/OS subsystems (such as CICS or 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 JSON fields, and rearranging the order of JSON fields and data.

2.1 Focus of this report: API Mapping feature

This report focuses on the API mapping feature of z/OS Connect EE V2.0. The API mapping feature is not available in z/OS Connect V1.

The scenario will use a Simulated Service Provider written in Java to simulate a z/OS subsystem such as CICS or IMS.

The diagram in Illustration 1 shows the flow of requests and responses for the scenario:

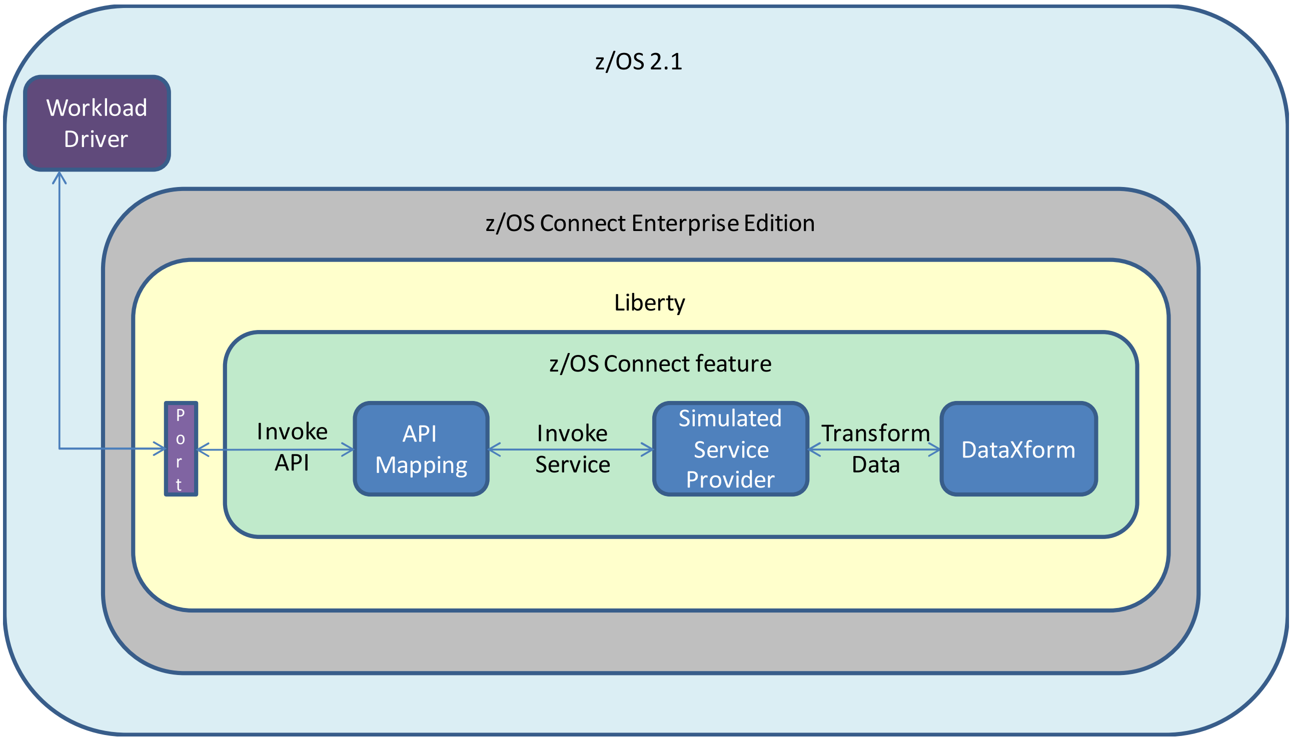


Illustration 1: Flow of requests and responses using API Field Mapping and DataXform

2.2 Flow of work

The flow of work is as follows:

1. The workload driver sends concurrent client HTTP JSON requests to z/OS Connect EE. See 2.3 Format of Requests on page 7 for details of the format of these requests.
2. The JSON request calls the API feature within z/OS Connect EE.
3. The API feature performs header and field mapping.
4. The API calls the service defined within the z/OS Connect feature and calls the Simulated Service Provider
5. The Simulated Service Provider receives the JSON payload using the Java getBytes method to convert the JSON to a byte array using the dataXform feature of z/OS Connect EE.
6. Depending on the content of the JSON request, the Simulated Service Provider sends a 1K or 4K payload in the response.

2.3 Format of Requests

The details of each request are described below.

2.3.1 URL

Each HTTP request specifies a URL with the following format:

- http://<host>:<port>/<basepath>/<relative path>

  For example:
  http://myhost:2222/performance.api/mapping/CABASIC-RS2/1

- The <basepath> in the example is “performance.api”

- The <relative path> in the example is “mapping/CABASIC-RS2/1”

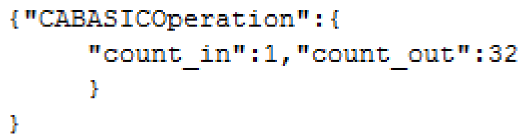
- The “1” at the end of the relative path is the path parameter “countIn” used by the API Mapping feature within the API Editor (see Illustration 7 on page 12).

2.3.2 Request Fields

Each JSON request consists of two fields and is approximately 50 bytes long.

- The first field is called “count_in” whose value is passed in as a “path parameter”.
- The second field is called “count_out”. The value in this second field is used to determine the size of the data to be returned by the Simulated Service Provider.

Illustration 2 shows an example of a JSON request.



```
{"CABASICOperation":{
     "count_in":1,"count_out":32
     }
}
```

Illustration 2: Example of JSON request requesting a 1K response

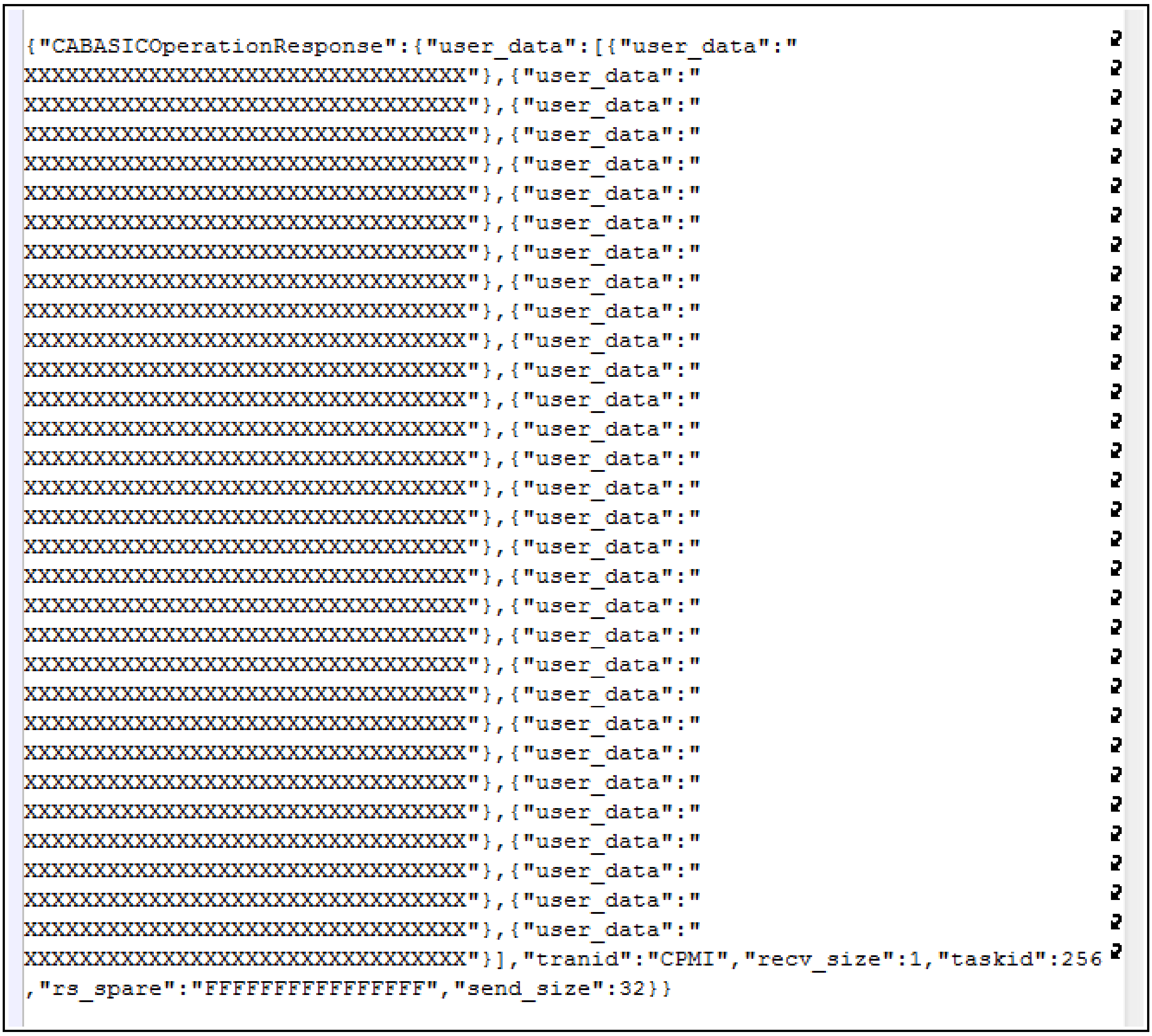
2.3.3 Response Fields

The size of the response to be sent by the Simulated Service Provider is determined by the "count_out" field in the request. In the example shown in Illustration 2, the "count_out" field is requesting 32 "blocks" of data. The Simulated Service Provider interprets this value by generating

1. An array called "user_data" with 31 elements each consisting of 32 alphabetic values (i.e. 31 x 32 bytes = 992 bytes);
2. Five fields of various values (e.g. tranid) totalling 32 bytes.

Therefore the total response size in this is example is 1K (i.e. 992 bytes + 32 bytes = 1024 bytes)

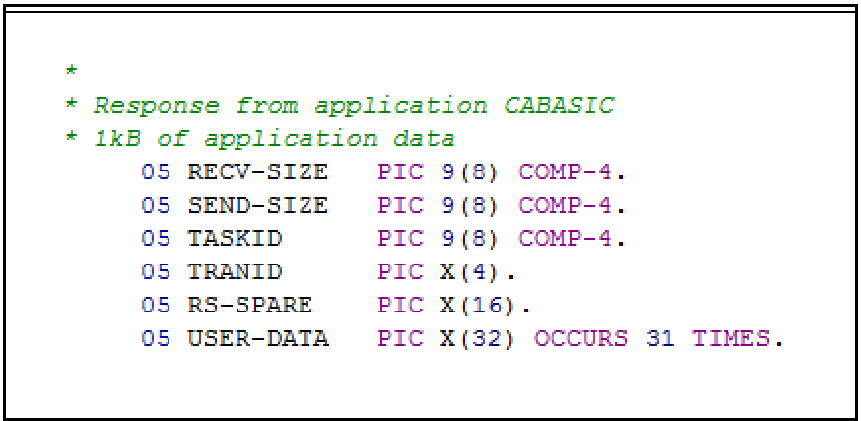
Illustration 3 shows the format of the 1K response (comprised of 31 array elements and five extra fields).



```
{"CABASICOperationResponse":{"user_data":[{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"},{"user_data":"
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"}],"tranid":"CPMI","recv_size":1,"taskid":256
,"rs_spare":"FFFFFFFFFFFFFFFF","send_size":32}}
```

Illustration 3: Example of JSON 1K response

The response data is based on the CICS COBOL copybook shown in Illustration 4:



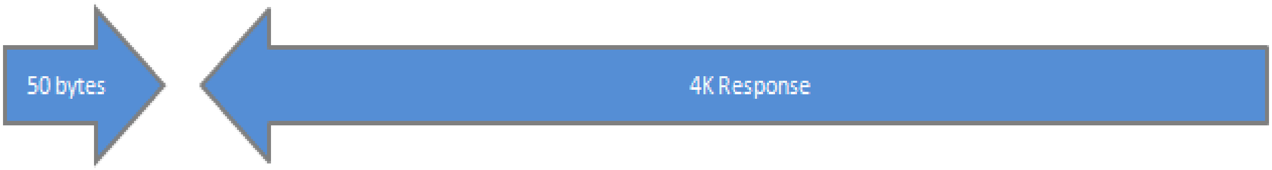
```
      *
      * Response from application CABASIC
      * 1kB of application data
           05 RECV-SIZE    PIC 9(8) COMP-4.
           05 SEND-SIZE    PIC 9(8) COMP-4.
           05 TASKID       PIC 9(8) COMP-4.
           05 TRANID       PIC X(4).
           05 RS-SPARE     PIC X(16).
           05 USER-DATA    PIC X(32) OCCURS 31 TIMES.
```

Illustration 4: COBOL copybook for 1K responses

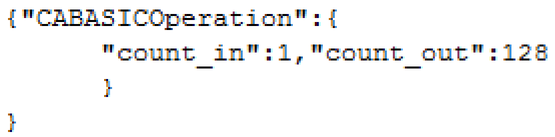
2.3.4 4K Responses

The illustrations in 2.3.2 Request Fields and 2.3.3 Response Fields are for 50 byte requests, requesting 1K responses.

For larger response sizes such as 4K, the data formats of the requests and responses follow a similar pattern to those for the 1K responses.



The JSON request is approximately 50 bytes in length, as shown in Illustration 5:



```
{"CABASICOperation":{
      "count_in":1,"count_out":128
      }
}
```

Illustration 5: Example of JSON request requesting a 4K response

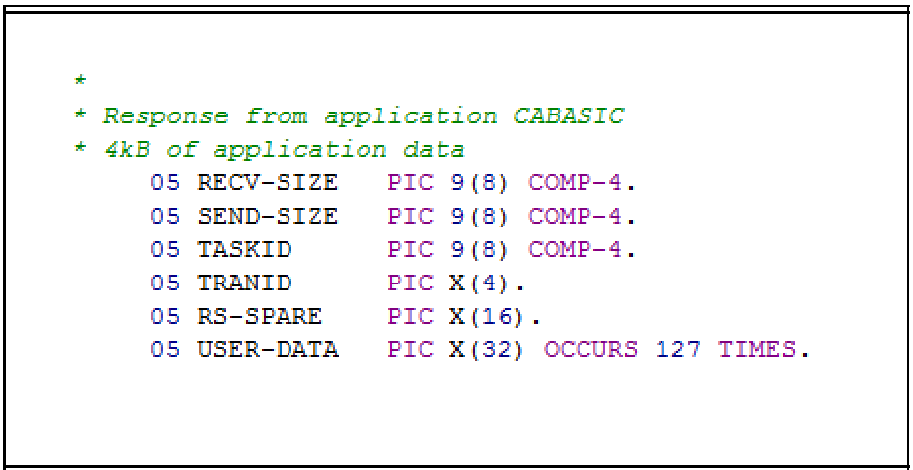
The second field, “count_out” requests a 4K response comprising of 128 “blocks” of data. The Simulated Service Provider interprets this value by generating

1. An array called “user_data” with 127 array elements consisting of 32 alphabetic values (i.e. 127 x 32 bytes = 4064 bytes)
2. Five fields of various values (e.g. tranid) totalling 32 bytes.

Therefore the total response size in this example is 4K (i.e. 4064 bytes + 32 bytes = 4096 bytes).

The 4K JSON response looks similar to the 1K JSON response shown in Illustration 3 on page 9 but with more “user_data” fields to make up the 4K of data.

The response data is based on the CICS COBOL copybook shown in Illustration 6: COBOL copybook for 4K responses.



```
*
* Response from application CABASIC
* 4kB of application data
     05 RECV-SIZE   PIC 9(8) COMP-4.
     05 SEND-SIZE   PIC 9(8) COMP-4.
     05 TASKID      PIC 9(8) COMP-4.
     05 TRANID      PIC X(4).
     05 RS-SPARE    PIC X(16).
     05 USER-DATA   PIC X(32) OCCURS 127 TIMES.
```

Illustration 6: COBOL copybook for 4K responses

2.4 API Mapping

The API Editor, new in z/OS Connect EE V2.0, is used to perform HTTP-to-JSON mapping.

For this performance report the simplest field mapping has been chosen using the path parameter “countIn” passed in the URL (see 2.3.1 URL on page 7). The API Editor maps this value to the JSON “count_in” field, and implicitly converts it from a string to an integer, as shown in Illustration 7. All other JSON fields from the HTTP request are automatically mapped without alteration.

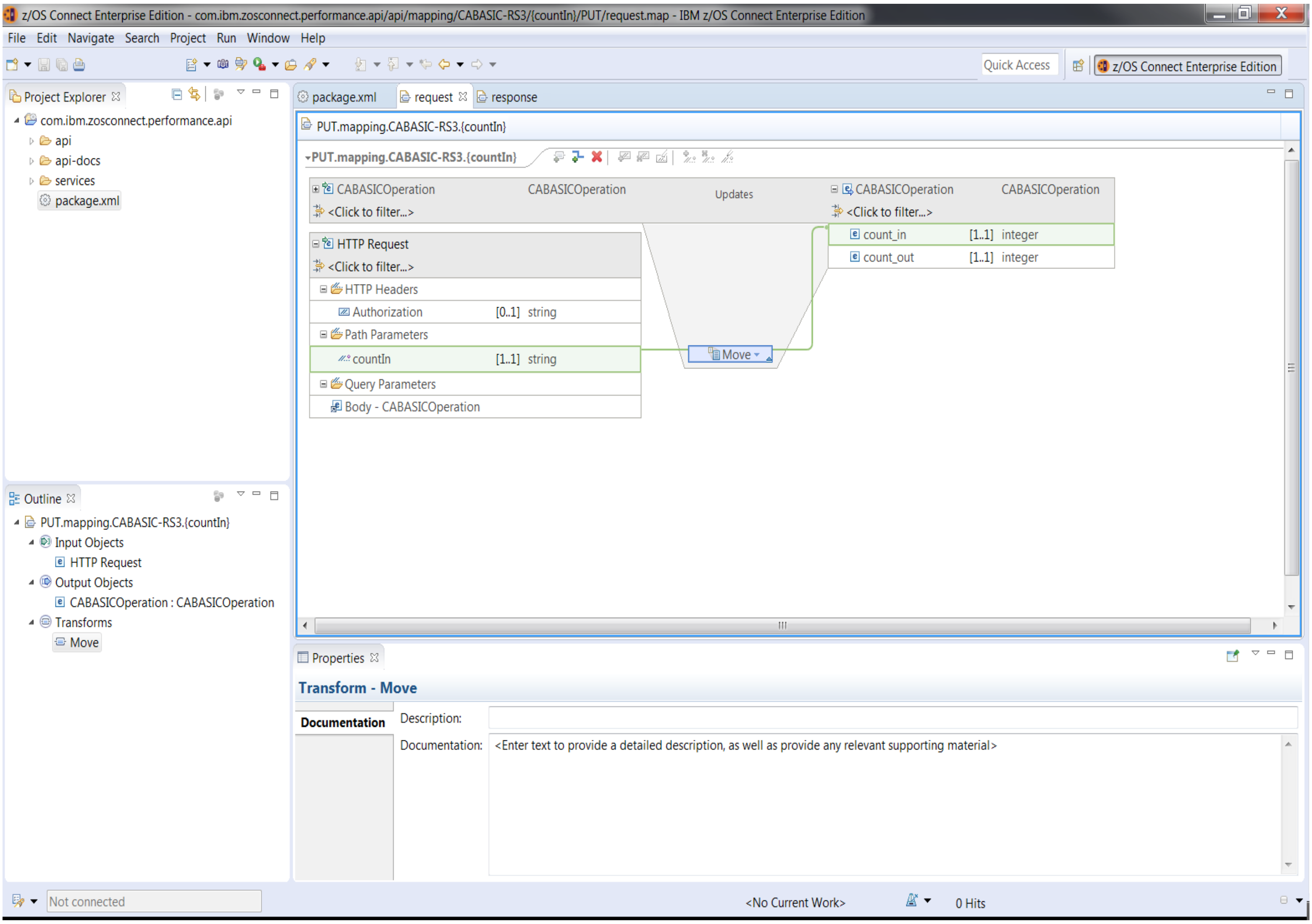

z/OS Connect Enterprise Edition - com.ibm.zosconnect.performance.api/api/mapping/CABASIC-RS3/{countIn}/PUT/request.map - IBM z/OS Connect Enterprise Edition
File Edit Navigate Search Project Run Window Help
Quick Access
z/OS Connect Enterprise Edition
Project Explorer
com.ibm.zosconnect.performance.api
api
api-docs
services
package.xml
package.xml
request
response
PUT.mapping.CABASIC-RS3.{countIn}
PUT.mapping.CABASIC-RS3.{countIn}
CABASICOperation
CABASICOperation
Updates
CABASICOperation
CABASICOperation
<Click to filter...>
<Click to filter...>
count_in [1..1] integer
count_out [1..1] integer
HTTP Request
<Click to filter...>
HTTP Headers
Authorization [0..1] string
Path Parameters
countIn [1..1] string
Move
Query Parameters
Body - CABASICOperation
Outline
PUT.mapping.CABASIC-RS3.{countIn}
Input Objects
HTTP Request
Output Objects
CABASICOperation : CABASICOperation
Transforms
Move
Properties
Transform - Move
Documentation
Description:
Documentation:
<Enter text to provide a detailed description, as well as provide any relevant supporting material>
Not connected
<No Current Work>
0 Hits

Illustration 7: API Editor mapping countIn to count_in

3. Test Environment

This chapter describes the environment the performance tests were run in. It includes details of the Workload Driver, and how z/OS Connect EE V2.0 was configured.

3.1 Hardware

- IBM System z13 2964-NE1 model 7A5
- 10GB of Central Storage (RAM)
- LPAR with 5 dedicated GCPs (no zIIPs)
- OSA-Express5S 10GB Ethernet

3.2 Software

- z/OS Connect Enterprise Edition (EE) V2.0
- z/OS V2.1
- IBM 64-bit SDK for z/OS Java Technology Edition, Version 8.0

3.3 Workload Driver

The workload driver, used to simulate multiple client requests, is a Java application that runs in its own JVM on the same LPAR as z/OS Connect EE. Requests issued by the workload driver specify “localhost” to minimise potential network latency.

Requests are issued by the workload driver at a regular pace, using a 100 ms “thinktime”. This is the time between individual requests being sent.

All scenarios issue HTTP PUT requests. There are no performance measurements for the HTTPS secure protocol as this is covered by Liberty performance reports, and not affected by z/OS Connect EE.

3.4 Asymmetric Payloads

Each payload was asymmetric, meaning that the size of data for the request is different to the size of data for the response. For example, a 50 byte payload request may generate a 4K response.

50 bytes
4K Response

3.5 z/OS Connect EE Configuration

z/OS Connect EE was configured with:

- IBM 64-bit SDK for z/OS Java Technology Edition, Version 8.0

- REGION size 0M
- MEMLIMIT 4G
- Maximum Java heap size (Xmx) 256M
- Other JVM system properties use default values (including GC mode: gencon)

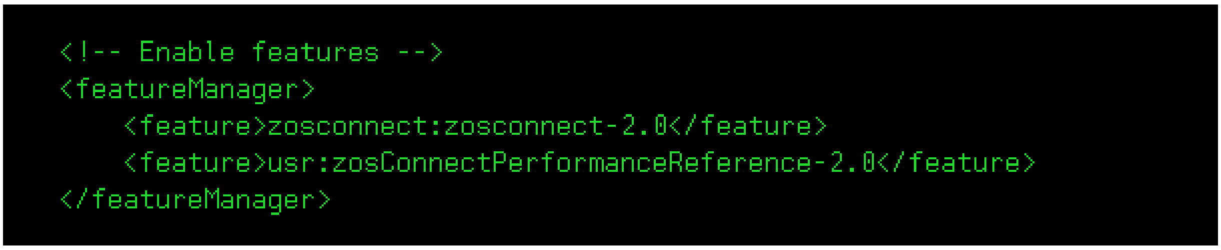
The server.xml file contains definitions for:

- zosconnect-2.0 feature
- performance reference user feature, created and used for this report
- services used
- data transformation

Note that the API used for these performance tests does not require any definitions in the server.xml file.

3.5.1 Feature Definitions

Illustration 8 shows the feature definitions specified in the server.xml configuration file.



```
<!-- Enable features -->
<featureManager>
    <feature>zosconnect:zosconnect-2.0</feature>
    <feature>usr:zosConnectPerformanceReference-2.0</feature>
</featureManager>
```

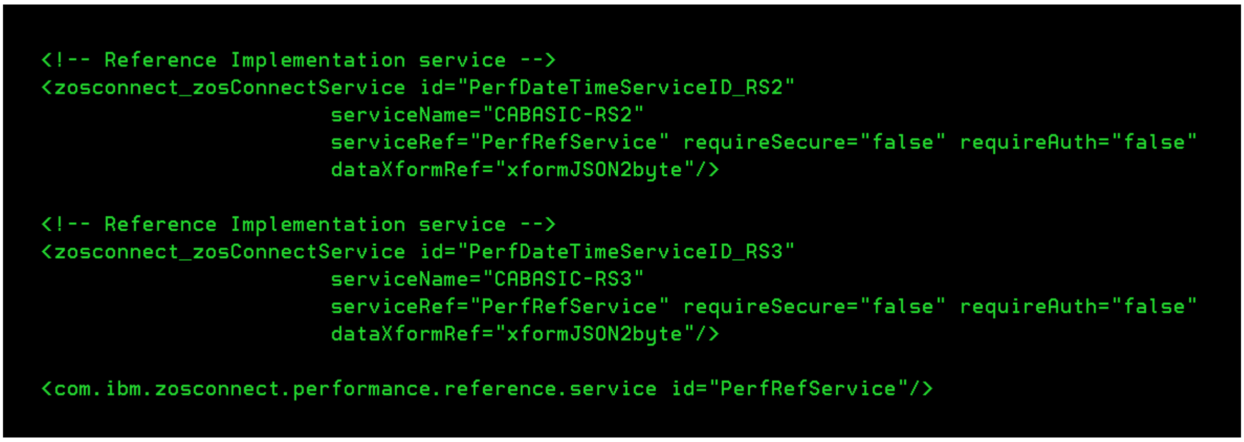
Illustration 8: Feature definitions in server.xml

- The first feature is for z/OS Connect EE V2.0.
- The second feature is for the Simulated Service Provider.

3.5.2 Service Definitions

Illustration 9 shows the service definitions specified in the server.xml configuration file.

- The serviceName CABASIC-RS2 is for the 1K response scenario
- The serviceName CABASIC-RS3 is for the 4K response scenario
- Both the services defined use the JSON to byte array data transformation feature supplied with z/OS Connect EE.



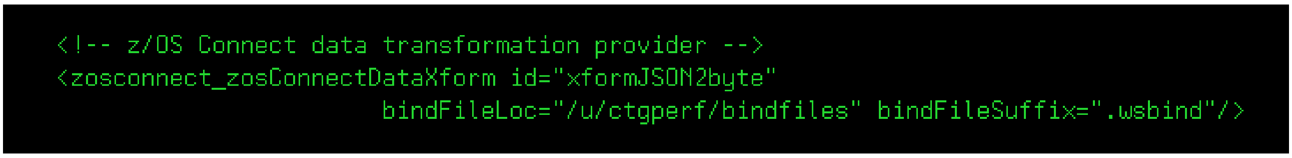
```
<!-- Reference Implementation service -->
<zosconnect_zosConnectService id="PerfDateTimeServiceID_RS2"
                      serviceName="CABASIC-RS2"
                      serviceRef="PerfRefService" requireSecure="false" requireAuth="false"
                      dataXformRef="xformJSON2byte"/>

<!-- Reference Implementation service -->
<zosconnect_zosConnectService id="PerfDateTimeServiceID_RS3"
                      serviceName="CABASIC-RS3"
                      serviceRef="PerfRefService" requireSecure="false" requireAuth="false"
                      dataXformRef="xformJSON2byte"/>

<com.ibm.zosconnect.performance.reference.service id="PerfRefService"/>
```

Illustration 9: Service definitions in server.xml

3.5.3 Data Transformation Definition

Illustration 10 shows the data transformation definition specified in the server.xml configuration file.

- The data transformer “zosConnectDataXform” uses the information in the bind files to transform data from/to different object types. For the performance tests in this report the data is transformed between JSON and a byte array. This byte array maps to a COBOL copybook structure, as shown in Illustration 4 on page 9 and Illustration 6 on page 11.
- The bind files were generated using the BAQLS2JS utility using COBOL copybooks to generate the artefacts.



```
<!-- z/OS Connect data transformation provider -->
<zosconnect_zosConnectDataXform id="xformJSON2byte"
                      bindFileLoc="/u/ctgperf/bindfiles" bindFileSuffix=".wsbind"/>
```

Illustration 10: Data transformation definition in server.xml

3.5.4 API

The API used for the performance runs is modelled using the API Editor and deployed using the z/OS Connect EE API Deployment Utility.

- The API package file, exported using the API Editor shown in Illustration 7 on page 12, has a suffix of .aar, for example, com.ibm.zosconnect.performance.api.aar. This contains the mappings for the scenarios used in this performance report. The .aar file is deployed to z/OS Connect EE using the API Deployment utility.
- The deployed API is automatically loaded upon invocation, and does not need to be defined in the server.xml file.

- There are no interceptors used in these performance tests.

3.5.5 Simulated Service Provider

The Simulated Service Provider, “PerfRefService”, shown in Illustration 9, is self-contained, and makes no network or cross-memory calls to a z/OS subsystem such as CICS or IMS. Instead, it simulates typical responses from a z/OS subsystem. As there are no network calls made by the Simulated Service Provider it allows for more accurate performance results for z/OS Connect EE V2.0 itself.

For this report the Simulated Service Provider receives the request and depending on the content of the request, generates a 1K or 4K response.

4. Performance Results for API Mapping

The following results were taken for the scenario described in 2.1 “Focus of this report: API Mapping feature“ on page 6:

- TPS (Transactions Per Second)
- Average Response time of requests
- CPU Cost Per Transaction
- CPU % Usage
- zIIP Eligibility

All requests are 50 bytes long and contain a simple object.
The structure of the response varies depending on the request.

- 1K responses consist of 31 array elements.
- 4K responses consist of 127 array elements.

4.1 Transactions Per Second for 1K Responses

Illustration 11 shows the Transactions Per Second (TPS) for 1K responses:

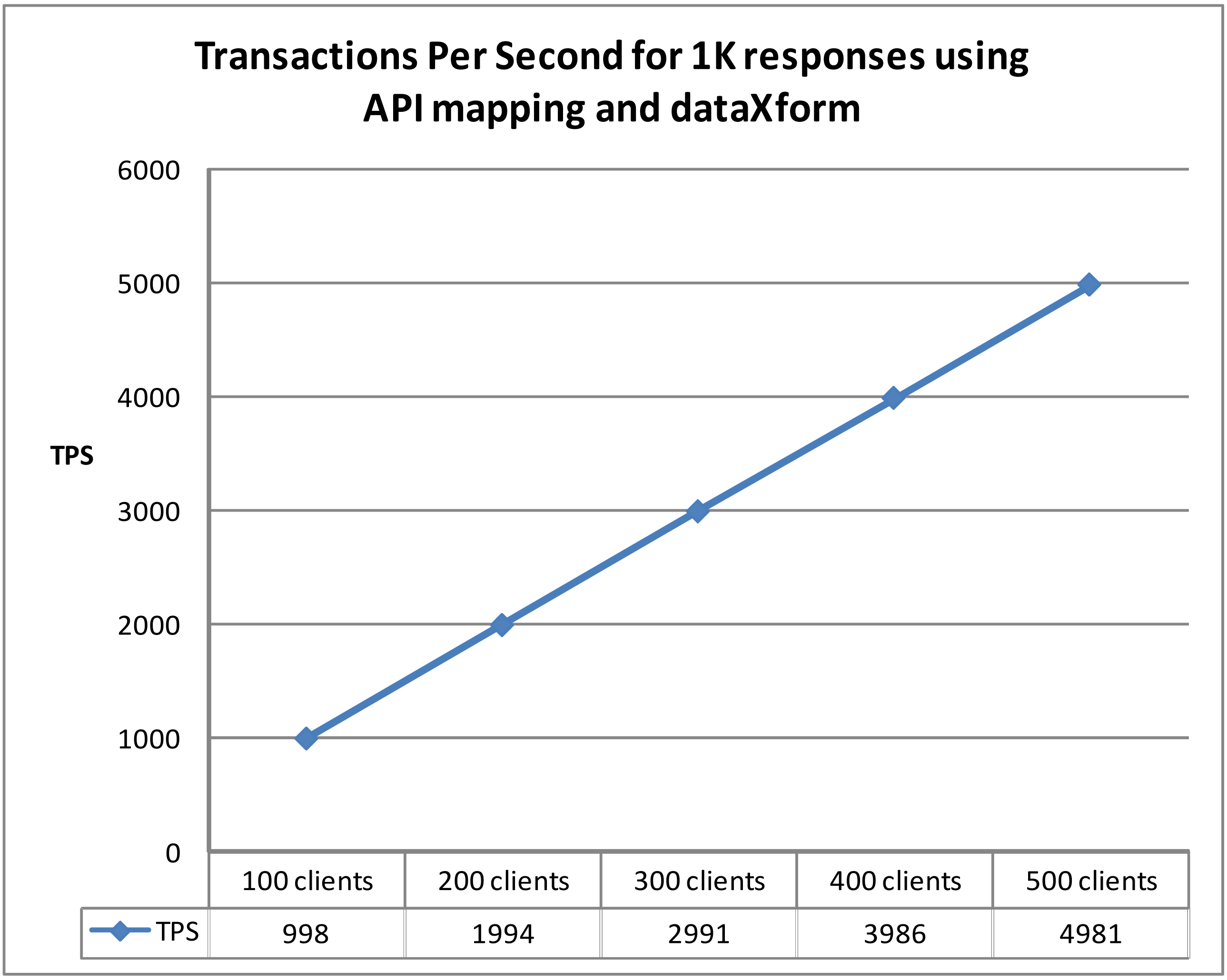


Illustration 11: TPS for 1K responses with increasing numbers of clients

4.1.1 Observations

- ✔ z/OS Connect EE demonstrated good scalability.
- ✔ Increasing the number of clients to run in parallel did not compromise the TPS.

4.2 Average Response time with 1K Responses

Illustration 12 shows the average response time (in milliseconds) for 1K responses.

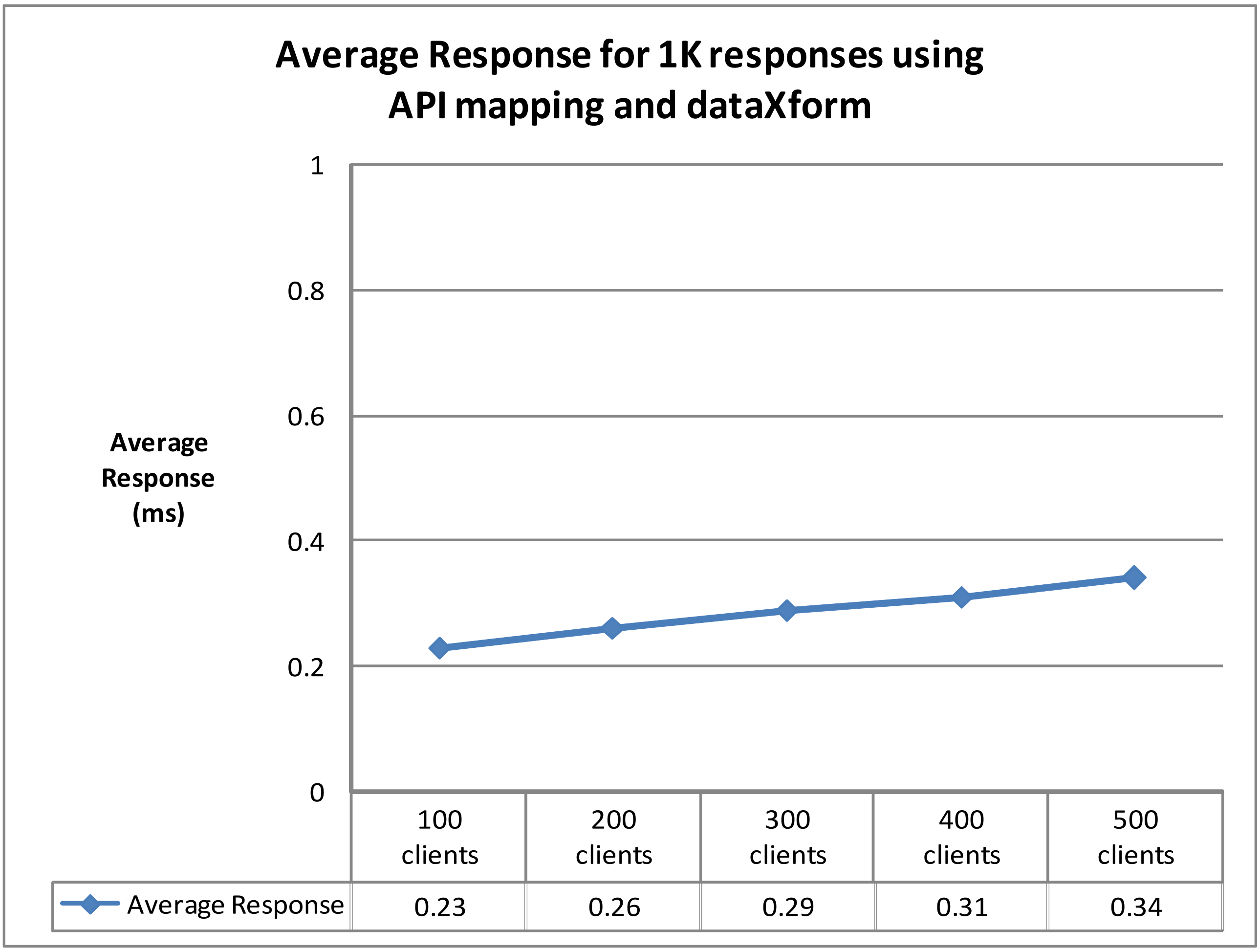


Illustration 12: Average response times for 1K payloads with increasing numbers of clients

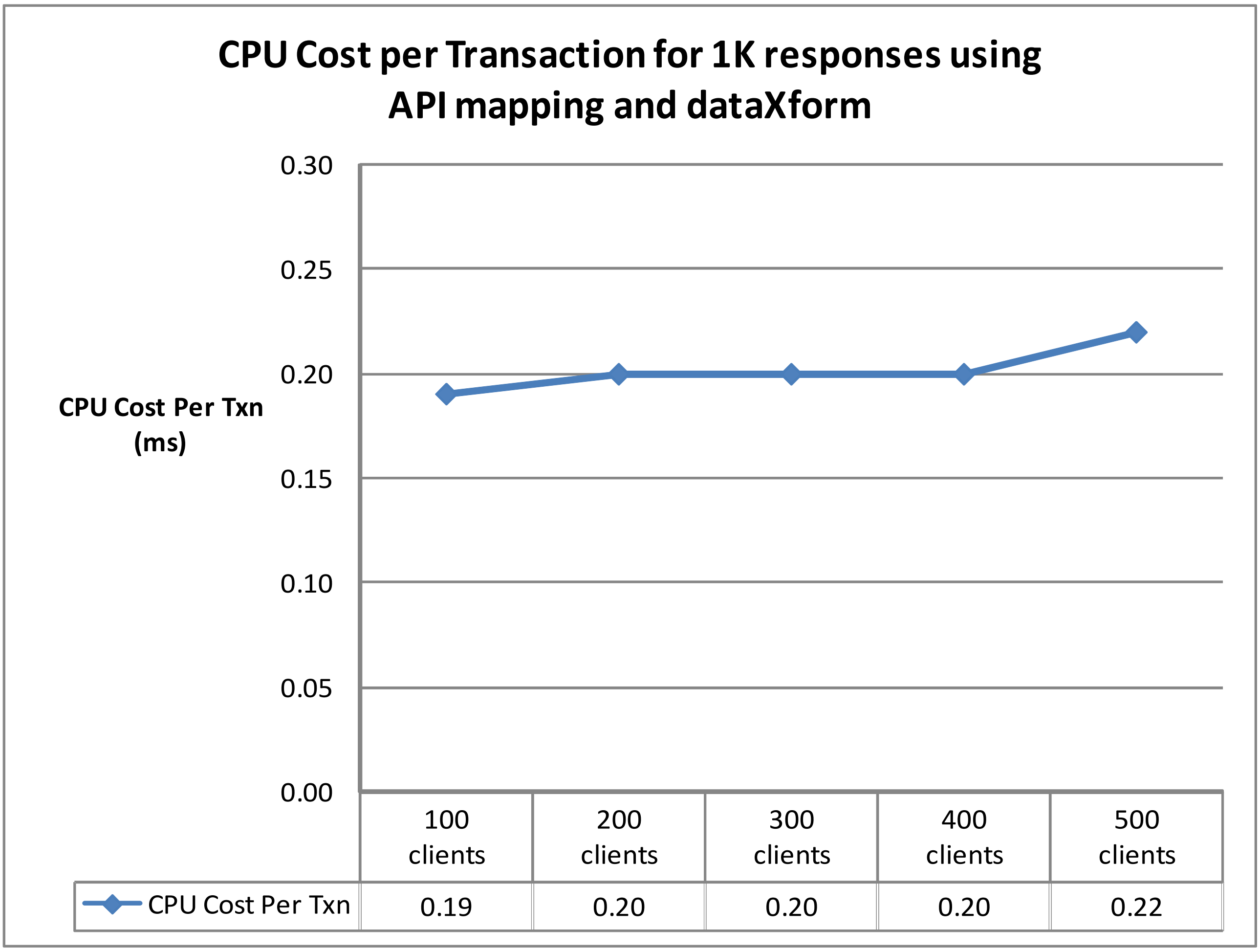
Ideally the average response time should be the same whether there is one user or 500 users. Realistically, as the number of clients increase, the average response time can be expected to also increase due to queues somewhere in the system. Typically a maximum throughput in the system will be reached somewhere in any configuration at which point requests will begin to queue, thus increasing the average response time.

4.2.1 Observations

- ✔ z/OS Connect EE demonstrated acceptable scalability for average response times.
- ✔ Increasing the number of clients to run in parallel did not create unacceptable response times.

4.3 *CPU Cost Per Transaction for 1K Responses*

Illustration 13 shows the CPU Cost Per Transaction for 1K responses:



	100 clients	200 clients	300 clients	400 clients	500 clients
CPU Cost Per Txn	0.19	0.20	0.20	0.20	0.22

Illustration 13: CPU Cost Per Transaction for increasing numbers of clients

4.3.1 Observations

- ✔ z/OS Connect EE demonstrated good scalability with minimal increase in the CPU Cost Per Transaction as the number of clients increased.
- ✔ Increasing the number of clients to run in parallel from 100 to 500 clients resulted in little or no change in CPU Cost Per Transaction for this scenario.

4.4 CPU % usage for 1K Responses

Illustration 14 shows the CPU % usage for 1K responses. The CPU % includes all the GCPs, allowing a theoretical maximum of 500%.

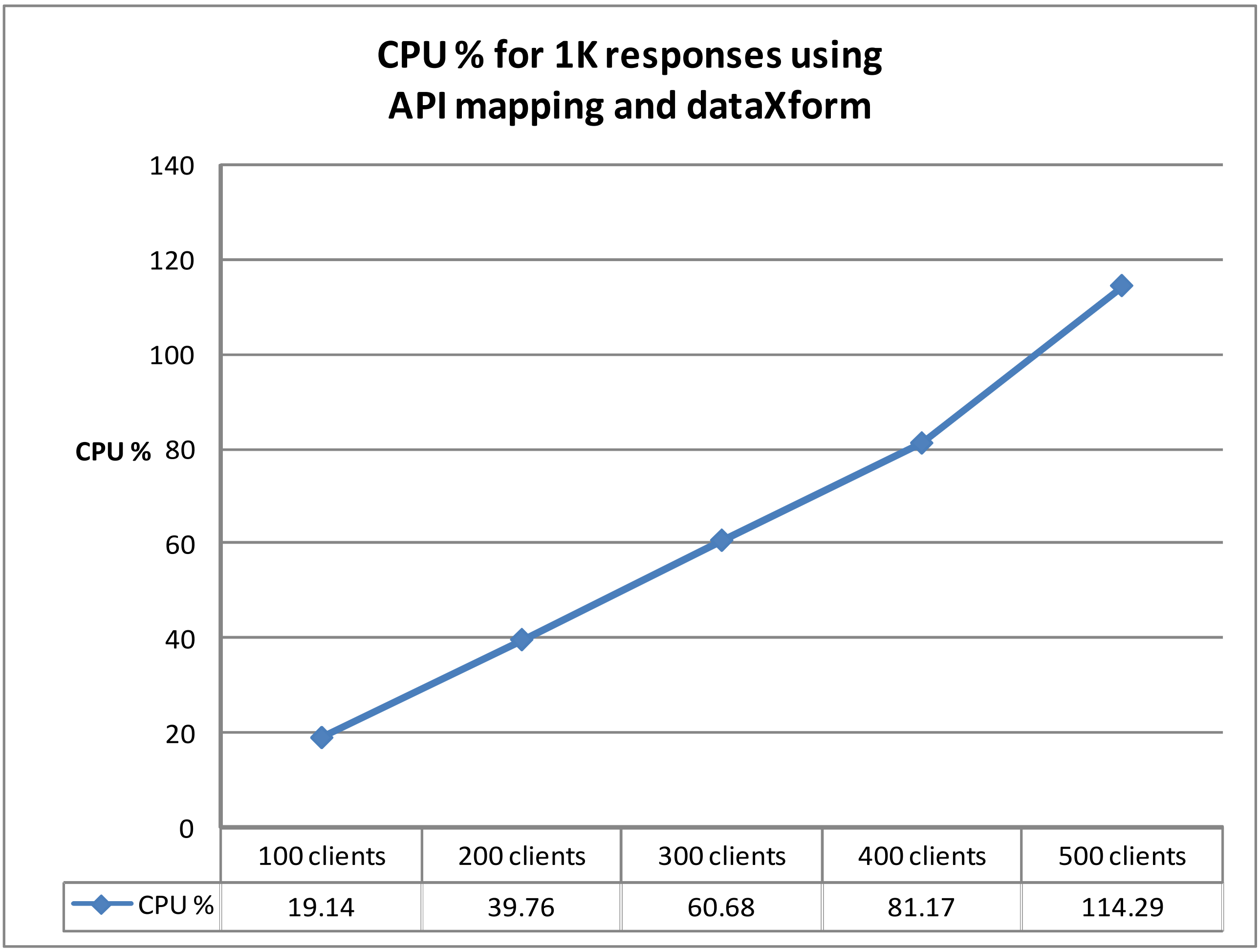


Illustration 14: CPU % usage for increasing numbers of clients

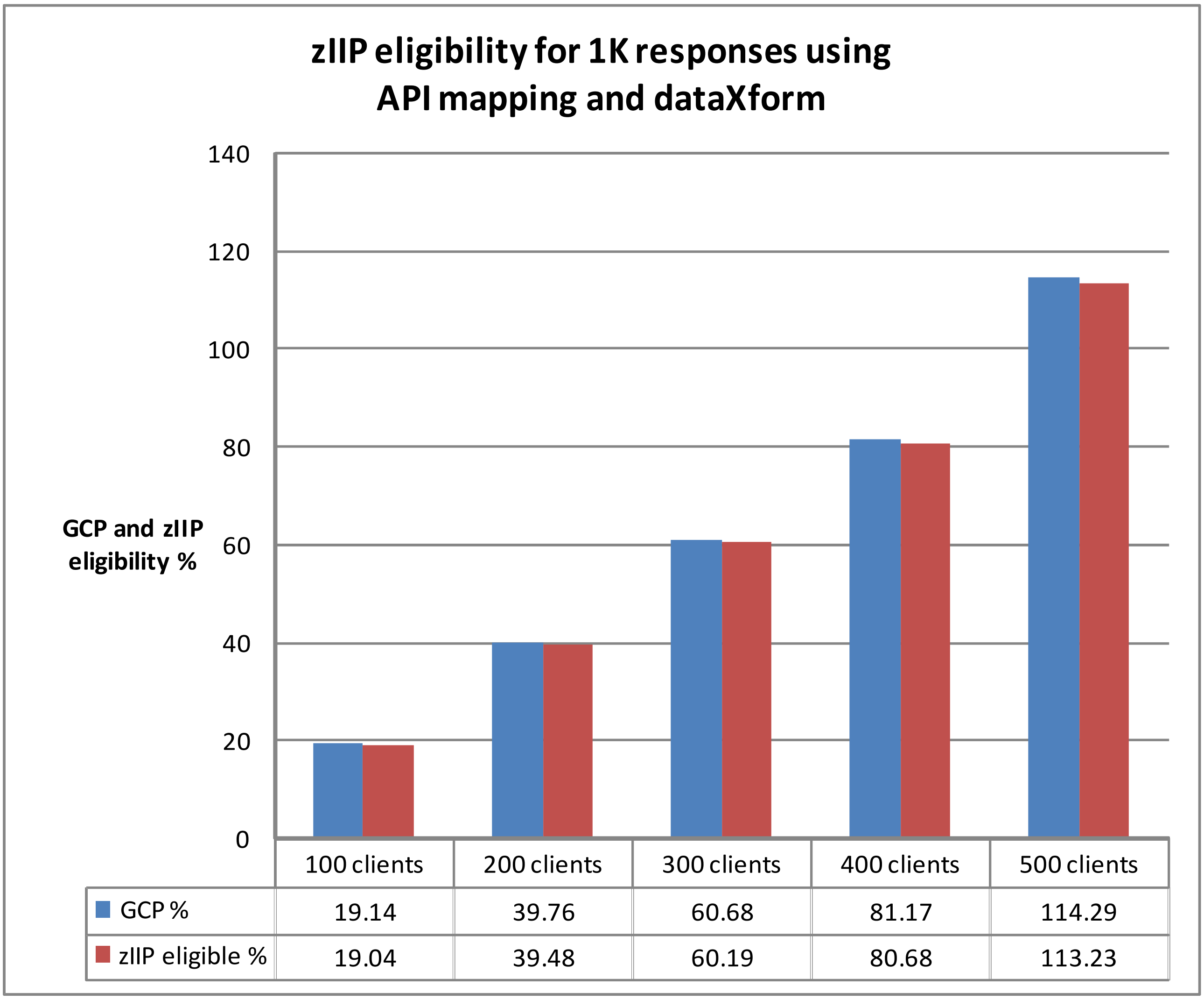
4.4.1 Observations

- ✔ z/OS Connect EE demonstrated good scalability.
- ✔ The CPU % usage increased linearly as the number of clients running in parallel were increased.

4.5 zIIP Eligibility for 1K Responses

Illustration 15 shows the GCP % usage and zIIP eligibility for this scenario.

The environment is configured with five GCPs .



	100 clients	200 clients	300 clients	400 clients	500 clients
🟦 GCP %	19.14	39.76	60.68	81.17	114.29
🟥 zIIP eligible %	19.04	39.48	60.19	80.68	113.23

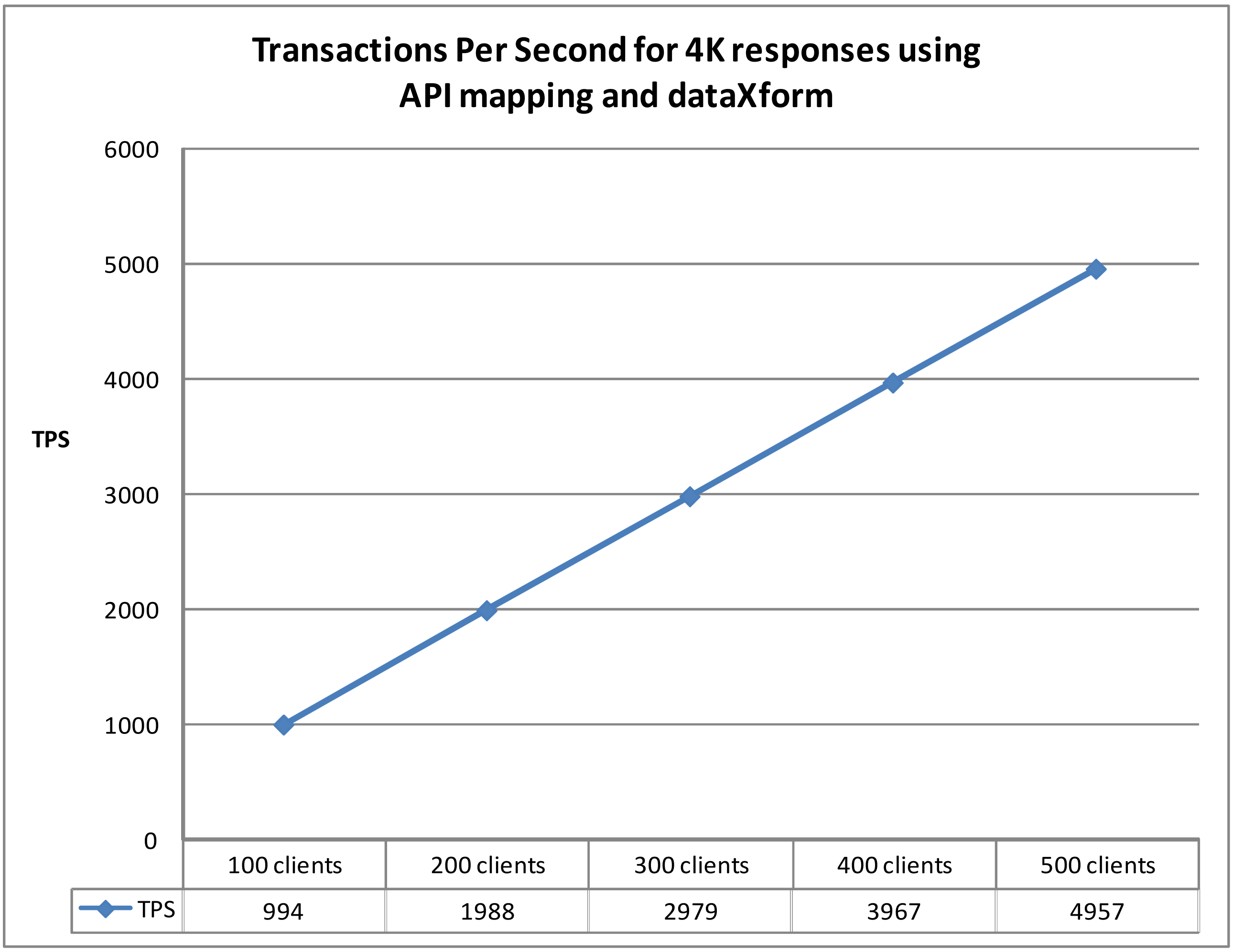
Illustration 15: GCP CPU % and zIIP eligibility for increasing numbers of clients

4.5.1 Observations

- ✔ z/OS Connect EE is a Java-based product and so over 99% of the product is eligible to be offloaded to zIIP.
- ✔ The potential usage of a zIIP scaled well with the increasing numbers of clients.

4.6 Transactions Per Second for 4K Responses

Illustration 16 shows the Transactions Per Second (TPS) for 4K responses:



	100 clients	200 clients	300 clients	400 clients	500 clients
TPS	994	1988	2979	3967	4957

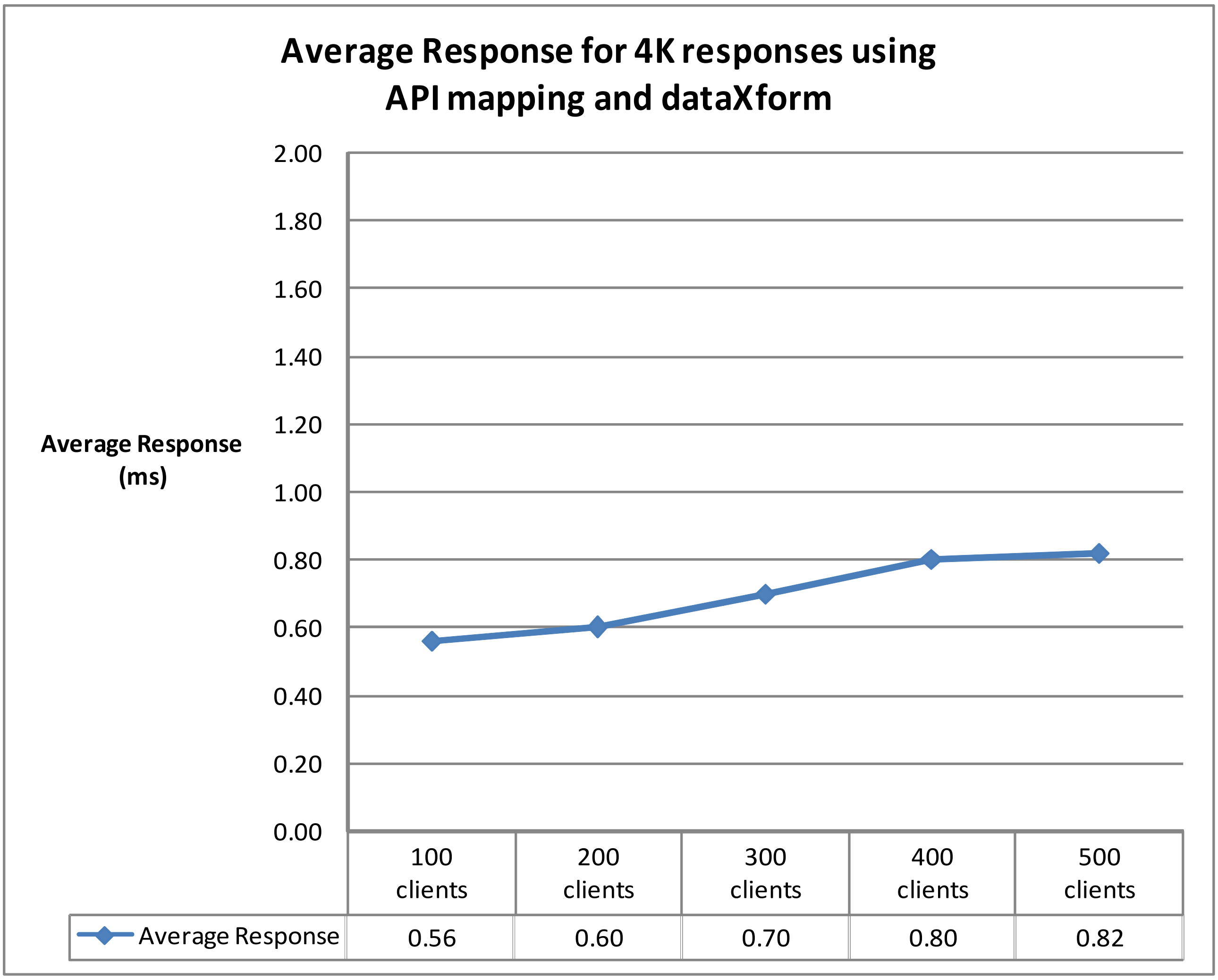
Illustration 16: TPS for 4K responses with increasing numbers of clients

4.6.1 Observations

- ✔ z/OS Connect EE demonstrated good scalability.
- ✔ Increasing the number of clients to run in parallel did not compromise the TPS.

4.7 Average Response time with 4K Responses

Illustration 17 shows the average response time (in milliseconds) for 4K responses.



	100 clients	200 clients	300 clients	400 clients	500 clients
Average Response	0.56	0.60	0.70	0.80	0.82

Illustration 17: Average response times for 4K payloads with increasing numbers of clients

Ideally the average response time should be the same whether there is one user or 500 users. Realistically, as the number of clients increase, the average response time can be expected to also increase due to queues somewhere in the system. Typically a maximum throughput in the system will be reached somewhere in any configuration at which point requests will begin to queue, thus increasing the average response time.

4.7.1 Observations

- ✔ z/OS Connect EE demonstrated acceptable scalability for average response times.

4.8 CPU Cost Per Transaction for 4K Responses

Illustration 18 shows the CPU Cost Per Transaction for 4K responses:

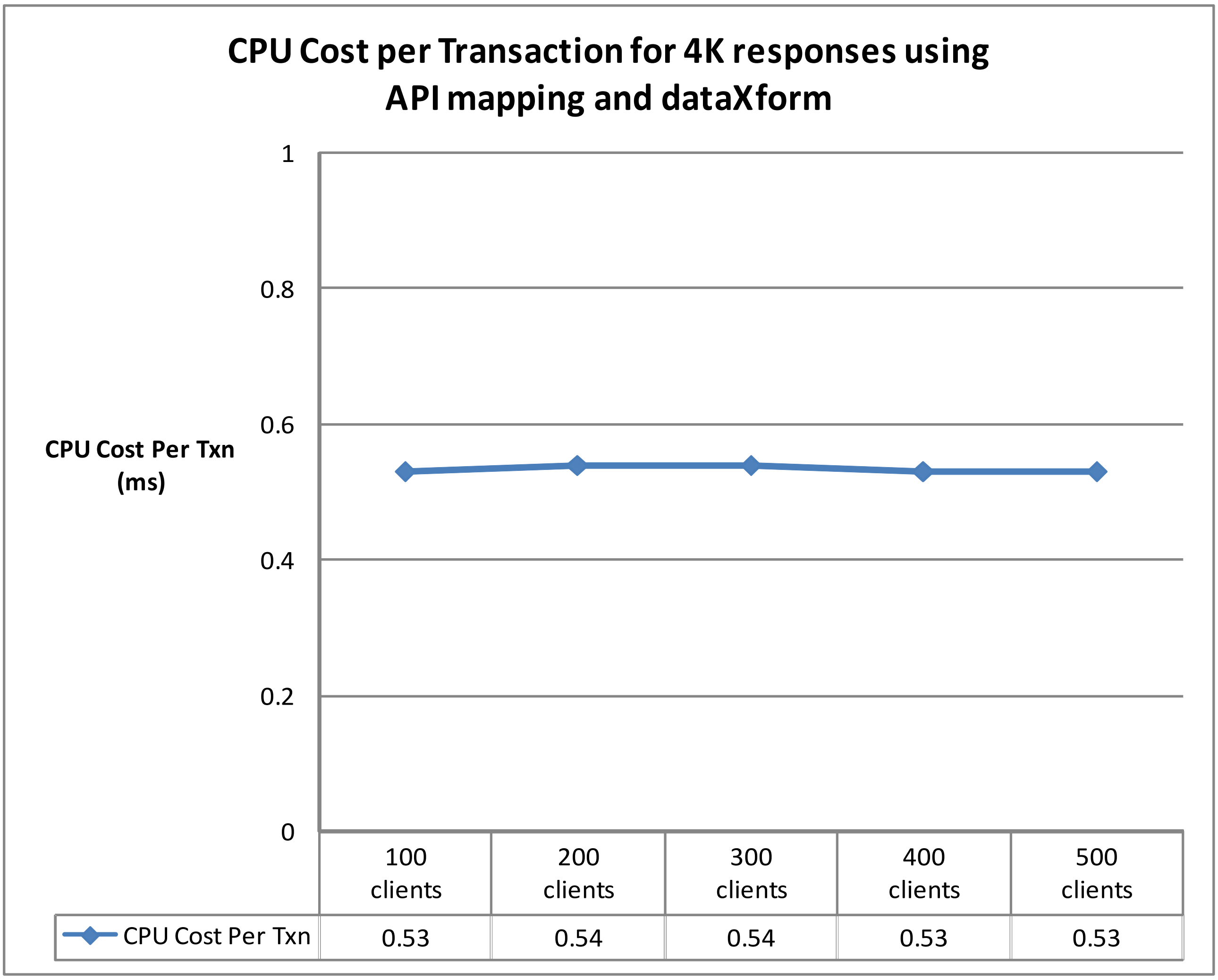


Illustration 18: CPU Cost Per Transaction for increasing numbers of clients

4.8.1 Observations

- ✔ z/OS Connect EE demonstrated good scalability with minimal increase in the CPU Cost Per Transaction as the number of clients increased.
- ✔ Increasing the number of clients to run in parallel from 100 to 500 clients resulted in little or no change in the CPU Cost Per Transaction for this scenario.

4.9 CPU % usage for 4K Responses

Illustration 19 shows the CPU % usage for 4K responses. The CPU % includes all the GCPs, allowing a theoretical maximum of 500%.

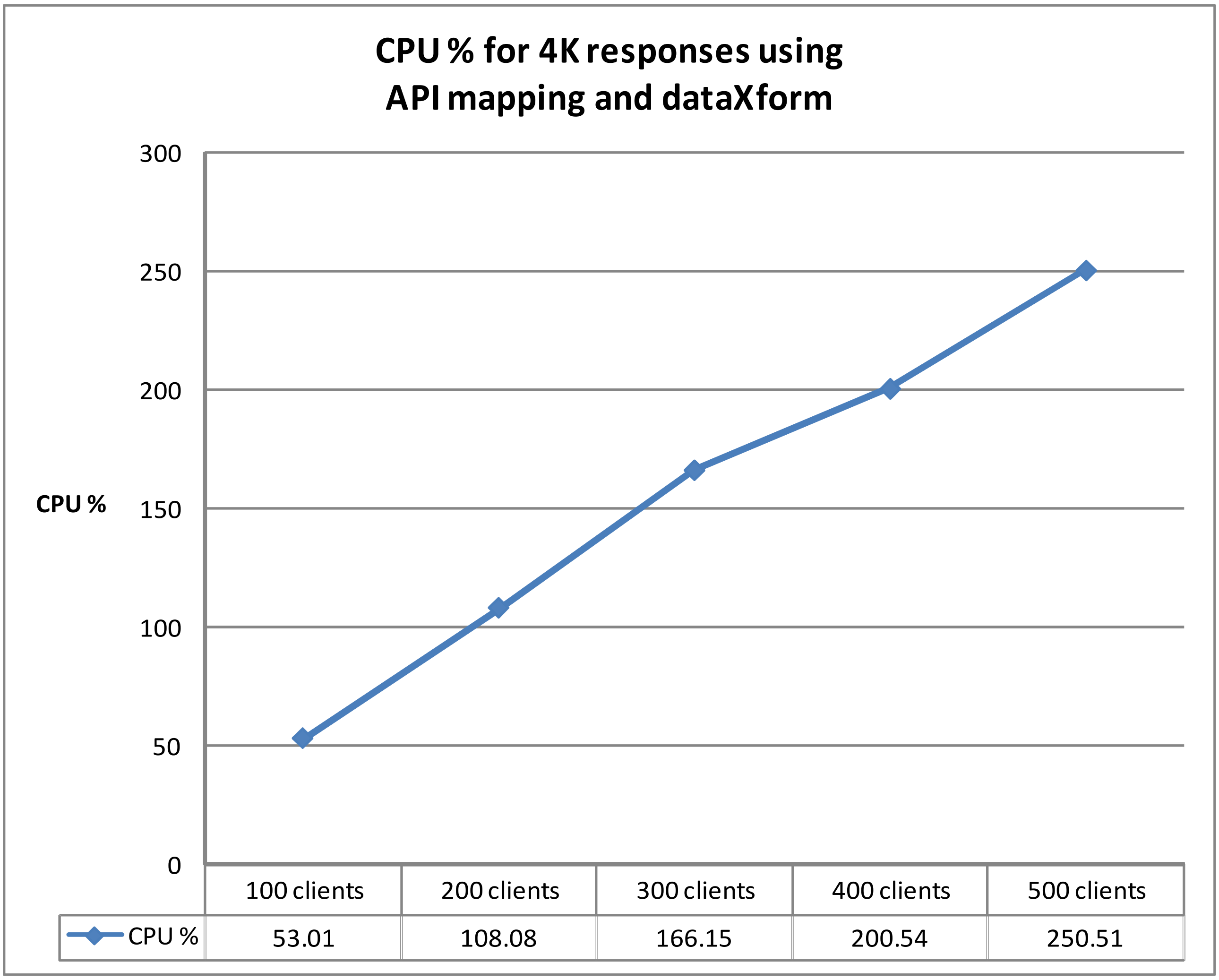


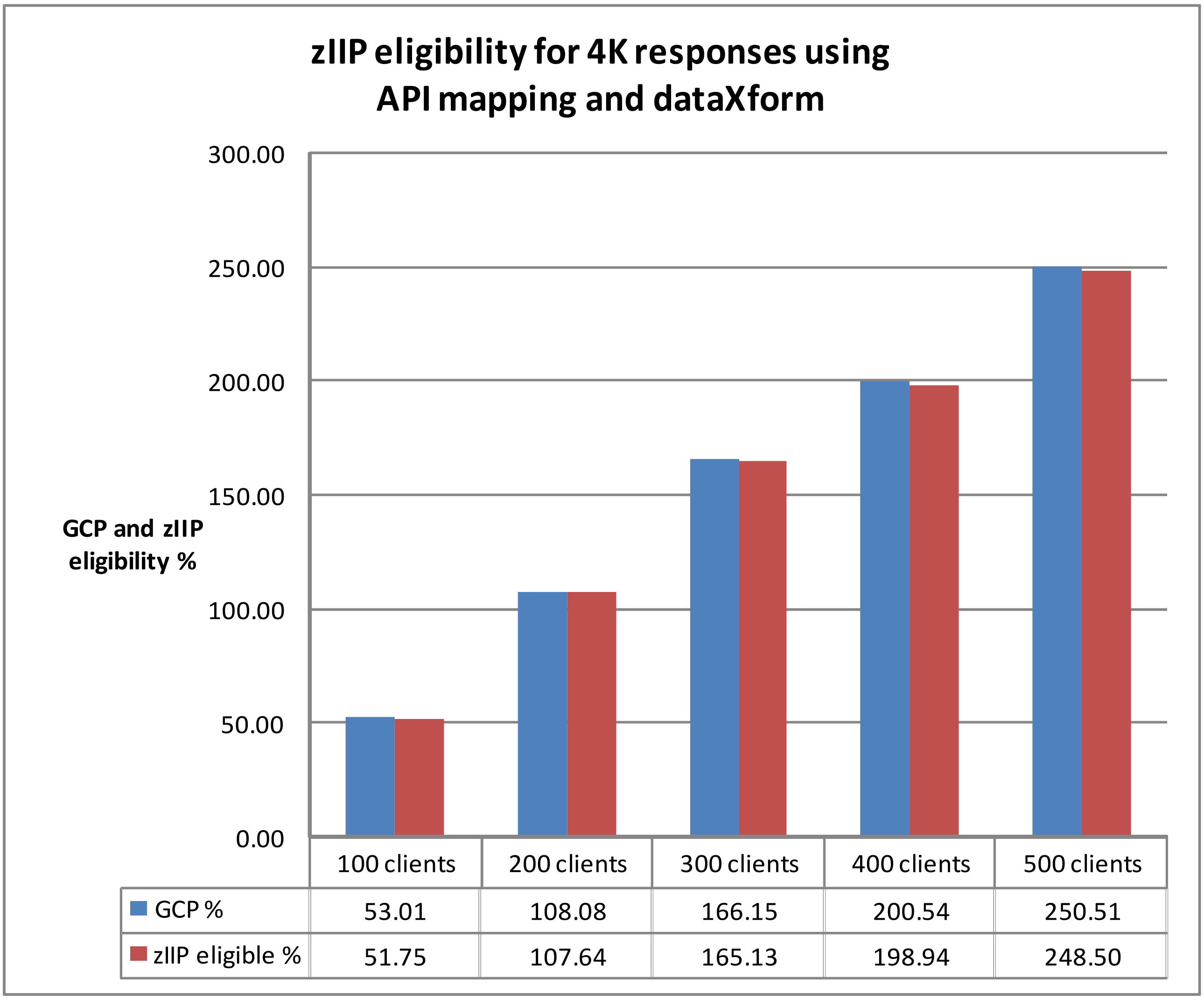
Illustration 19: CPU % usage for increasing numbers of clients

4.9.1 Observations

- ✔ z/OS Connect EE demonstrated good scalability.
- ✔ The CPU % usage increased linearly as the number of clients running in parallel were increased.

4.10 zIIP Eligibility for 4K Responses

Illustration 20 shows the GCP % usage and zIIP eligibility used for this scenario. The environment is configured with five GCPs.



	100 clients	200 clients	300 clients	400 clients	500 clients
🟦 GCP %	53.01	108.08	166.15	200.54	250.51
🟥 zIIP eligible %	51.75	107.64	165.13	198.94	248.50

Illustration 20: GCP CPU % and zIIP eligibility for increasing numbers of clients

4.10.1 Observations

- ✔ z/OS Connect EE is a Java-based product and so over 99% of the product is eligible to be offloaded to zIIP.
- ✔ The potential usage of the zIIP scaled well with the increasing numbers of clients.

5. Large request sizes, small response sizes

In addition to the performance results shown in this report, a number of performance tests were run whereby the request was much larger than the response. For example, a 4K request triggered a 50 byte response, as shown by the red arrow below.

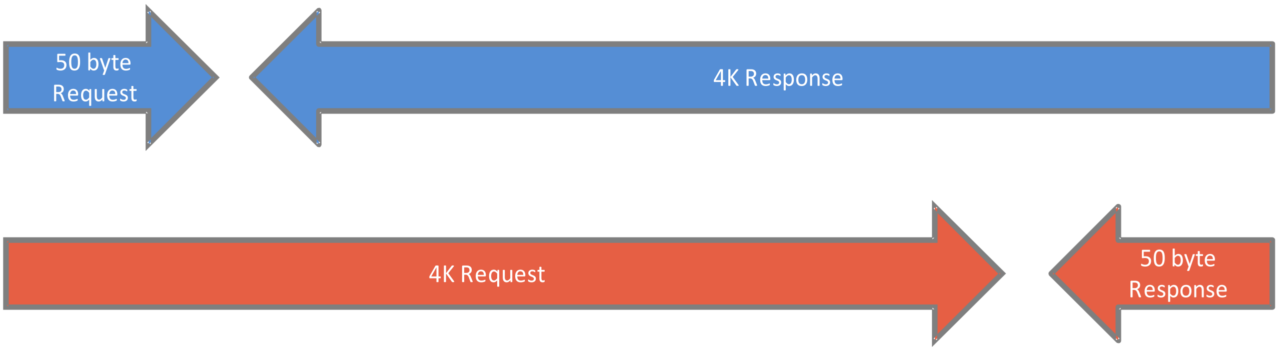


Illustration 21: Do large requests perform as well as large responses?

5.1 Observations

Whether the scenario used a large request, or a large response, the results for z/OS Connect EE V2.0 showed little to no difference in TPS, CPU cost per transaction, or CPU % usage.

Therefore these results have not been shown in this report, and readers can use the values given for larger responses as a guide.

6. Conclusions

Customers should consider the following:

1. A single z/OS Connect EE V2.0 server managing payloads containing large numbers of array elements that used the API mapping feature and transformed data between JSON and byte array data structures, demonstrated good scalability for

   - ✔ Transactions Per Second
   - ✔ Average Response time of requests
   - ✔ CPU Cost Per Transaction
   - ✔ CPU % Usage
   - ✔ potential zIIP offload.

2. The performance of large requests with small responses, showed no difference when compared to small requests with large responses.

3. z/OS Connect EE V2.0 is almost entirely written in Java and so will benefit from offloading work to one or more zIIPs.

Note:

1. The payload sizes for the responses in this report are up to 4K.
2. Analysis of other payload sizes, or different hardware, have not been completed at this time, so there is no guarantee that equivalent observations will be seen in other configurations.
3. Due to the effects on system performance of machine hardware, levels of software configuration and payload, equivalent observations might not be seen on other systems.