IBM Z | **E**nterprise **N**etworking **S**olutions (**ENS**)

3.1: Z/OS COMMUNICATIONS SERVER PERFORMANCE SUMMARY REPORT

Created By: Kaji Rashad

Last Updated On: May 7th, 2024

Table of Contents

Trademarks, Notices, and Disclaimers

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries:

BigInsights
BlueMix
CICS*
COGNOS*
Db2*
DFSMSdfp
DFSMSdss
DFSMShsm
DFSORT
DS6000*
DS8000*
FICON*
GDPS*
HyperSwap
IBM*
IBM (logo)*
IMS
Language Environment*
MQSeries*
Parallel Sysplex*
PartnerWorld*
RACF*
Rational*
Redbooks*
REXX
SmartCloud*
System z10*
Tivoli*
UrbanCode
WebSphere*
z13
z14
z15
z16
zEnterprise*
z/OS*
zSecure
z Systems
z/VM*

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies:
Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce and is registered in the U.S. Patent and Trademark Office.
Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
OpenStack is a trademark of OpenStack LLC. The OpenStack trademark policy is available on the OpenStack website.
Red Hat is a trademark of Red Hat Inc, an IBM Company.
TEALEAF is a registered trademark of Tealeaf, an IBM Company.
Windows Server and the Windows logo are trademarks of the Microsoft group of countries.
Worklight is a trademark or registered trademark of Worklight, an IBM Company.
UNIX is a registered trademark of The Open Group in the United States and other countries.
* Other product and service names might be trademarks of IBM or other companies.

Notes:
Performance results are based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here. IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions. This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area. All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice and represent goals and objectives only. Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography. This information 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 www.ibm.com/systems/support/machine_warranties/machine_code/aut.html (“AUT”). No other workload processing is authorized for execution on an SE. IBM offers SE at a lower price than General Processors/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.

Acknowledgments

The z/OS Communications Server Performance Team would like to **thank** the following IBMers for their input on this report:

David Herr **|** Senior Software Engineer **|** z/OS Communications Server

Christopher Nyamful **|** Software Engineer **|** z/OS Communications Server

Michael Fitzpatrick **|** Senior Technical Staff Member **|** Architecture & Design for ENS

Tips for Reading This Document

1) Clicking on any row in the Table of Contents will take the reader to that specific section or subsection of the document[1]
2) All hyperlinks redirect to an external webpage or internal section/sub-section

[1] PDF application must support this feature

Preface

The performance measurements discussed in this document were collected using dedicated system environments. Results obtained in other configurations or operating system environments may vary significantly depending upon environments used. Therefore, no assurance can be given, and there is no guarantee that an individual user will achieve performance or throughput improvements equivalent to the results stated here.

The Central Processor Unit (CPU) numbers listed includes only z/OS host networking related CPU overhead (including dispatching costs) on **general Central Processors** (CPs) from the network device driver layer up through the application socket layer. The socket applications used in the micro-benchmarks for this publication have *no application logic*, so the CPU numbers represent the total application cost which in this case equates to the network related costs. With typical production workloads, network related cost of the overall application transaction cost will vary based on the type of workload. Network cost is typically a small percentage of transactional workloads but can have a larger cost for streaming workloads.

Note #1: In all benchmarks, the best practices recommended by z/OS Communications Server were utilized *when* applicable. The following best practices represent configuration options not enabled by default. Other options representing best practices that are enabled by default are not listed here (e.g., IPCONFIG CHECKSUMOFFLoad, READSTORAGE GLOBAL, GLOBALCONFIG ADJUSTDVipamss, etc.):

VTAM START Option:

- ✓ QDIOSTG = 126

TCP/IP Configuration Profile:

- ✓ INBPERF DYNAMIC
  - ○ WORKLOADQ (IWQ)

- Client & Server Side
  - ✓ IPCONFIG SEGMENTATIONOFFLoad (LSO)
  - ✓ IPCONFIG6 SEGMENTATIONOFFLoad (LSO)
  - ✓ IPCONFIG QDIOACCELerator
  - ✓ TCPCONFIG AUTODELAYAck
  - ✓ MSG_WAITALL[2]

Note #2: We used Jumbo Frames (e.g., HOST MTU 8192) in our test environment for **streaming workloads** (i.e., long lived flows). Our test environment is a controlled set-up for which we have control over the point-to-point links and routers. Readers need to investigate their own environment before using Jumbo Frames as *it potentially could not be a best practice for their environment*.

The reader should read [1] for a holistic z/OS Communications Server and OSA-Express best practices. For convenience, you can access it via this link: https://www.ibm.com/support/pages/node/6562173.

[2] A socket read flag utilized by the application to instruct the TCP layer to delay completion of a Socket Receive or Read call until the full length of the requested data is available in the TCP receive buffer [2]

Hardware Information

z16
Machine Type (Model): 3931 – A01

z15
Machine Type (Model): 8561 – T01

Cryptographic Coprocessor Level
Crypto Express-8S: 8.0.71
Crypto Express-7S: 7.3.26

Workload Naming Convention

Introduction

Readers can decipher the listed workloads in the following way:
[NameOfBenchmark][#OfClients](BytesSentByClient/BytesSentByServer)

For example, [RR][10](1B/100B) is interpreted as Request Response benchmark with 10 clients sending 1 byte and receiving 100 bytes from the server.

Generic Workloads

RRx(y/z): x number of clients doing **R**equest **R**esponse transactions where the client is opening a connection and performing a series of transactions sending y bytes and receiving a response of z bytes
CRRx(y/z): x number of clients doing **C**onnect **R**equest **R**esponse transactions where the client is performing a series of repeatable transactions consisting of opening a connection, sending y bytes, receiving a response of z bytes, and closing the connection
STRx(y/z): x number of clients doing **Str**eaming transactions where the client is opening a connection and performing a series of transactions sending y bytes and receiving a response of z bytes

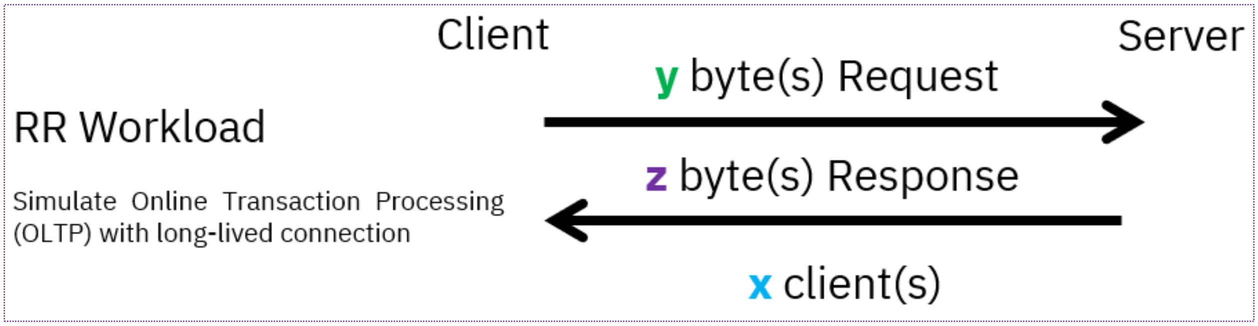
Client
Server
y byte(s) Request
RR Workload
z byte(s) Response
Simulate Online Transaction Processing (OLTP) with long-lived connection
x client(s)

Figure 1: Request response workload

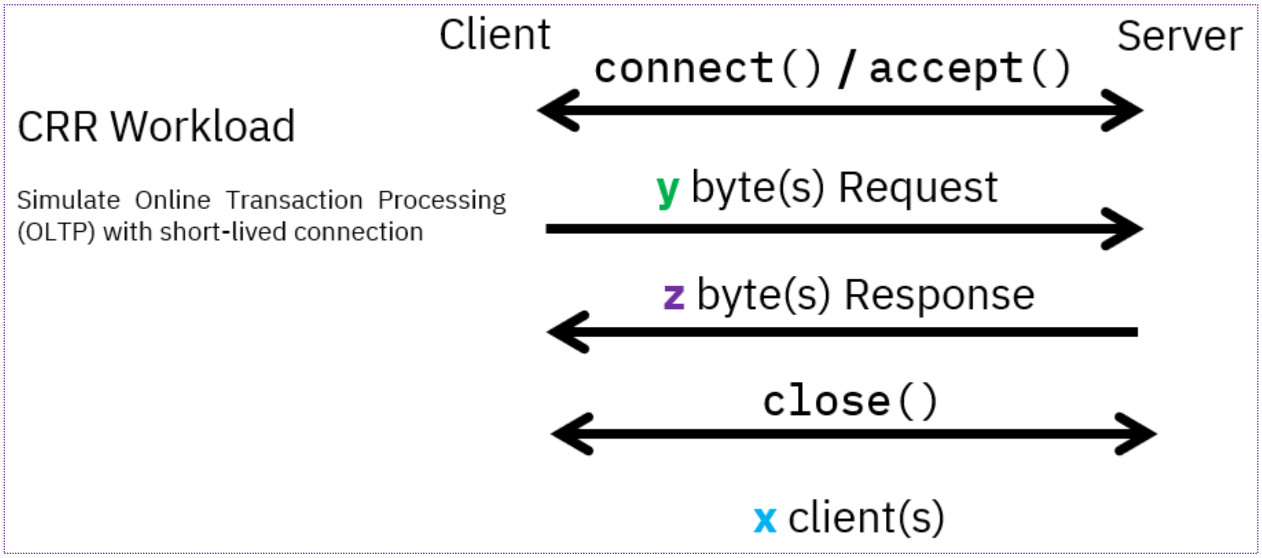

Figure 2: Connect request response workload

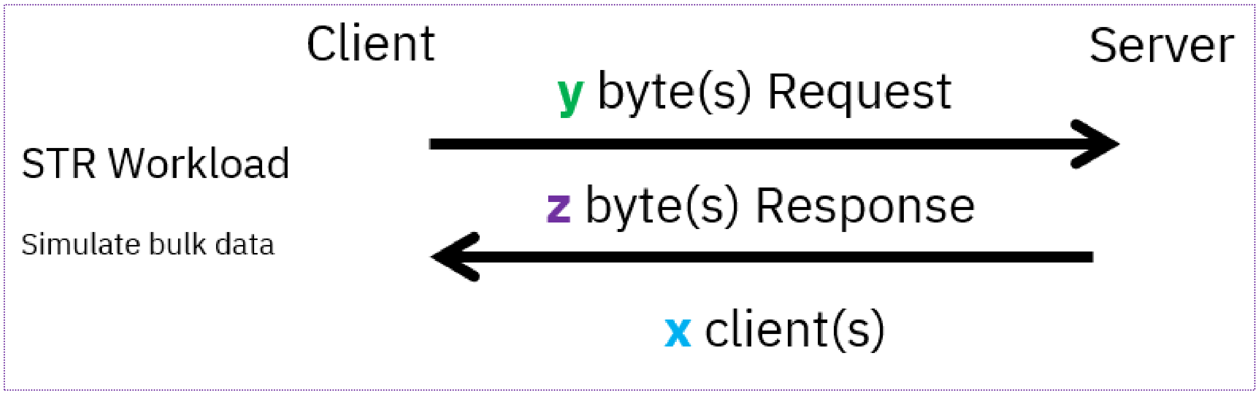

Figure 3: Streaming workload

Examples

RR40(100B/100B): In an instance of time, there are 40 clients browsing a webpage hosted on a server in which each HTTP GET request of 100 bytes contains a response of 100 bytes.

CRR9(200B/200B): In an instance of time, there are 9 clients sending a HTTP GET request containing 200 bytes and receives a response containing 200 bytes which allows them to log into their bank portal. The core difference between a RR and CRR workload is the duration of the connection. In CRR, the connection is closed after each transaction. A common use case for a bank portal is logging in to audit the balance before logging out.

STR3(1B/20MB): In an instance of time, there are 3 clients sending a 1 byte request and receiving a 20MB file in response.

CPU Cost/Tran & Transaction [Trans/sec]

On some graphs, the reader will observe a key legend of “CPU Cost/Tran” and “Transaction [Trans/sec]”. Our measurement uses the z/OS Resource Measurement Facility (RMF) to determine the average CPU utilization [12]. RMF results show the CPU utilization across all online CPs during a sampling interval which is taken into consideration when performing our calculations.

For example, if RMF result shows a LPAR utilization of 25% across 4 CPs then we translate this into 100% of 1 CP. If the sampling interval is 10 seconds and we are averaging 100% of 1 CP then the benchmark consumes 10 seconds of CPU during the sampling period. If there were 1 million transactions during the 10 second sampling interval, then there was a transaction rate of 1,000,000 trans / 10 seconds (or 100,000 trans/sec) and a CPU Cost per Transaction of (10 CPU Seconds) / (1,000,000 Transactions) or 10 [us]/tran.

Performance Best Practices: General

INBPERF DYNAMIC

Processing inbound traffic for the OSA-Express interface in Queued Direct Input Output (QDIO) mode dynamically exploits an OSA hardware function called Dynamic LAN Idle. The DYNAMIC setting reacts to changes in traffic patterns and dynamically sets the interrupt-timing values to optimize response times. Refer to this article for more information.

QDIO Inbound Workload Queueing (WORKLOADQ)

The core benefits of Inbound Workload Queueing (IWQ) are “finer tuning of read-side interrupt frequency to match the latency demands of the various workloads that are serviced” and “improved multiprocessor scalability as multiple OSA-Express input queues are efficiently serviced in parallel” [3]. Each queue is tailored for its specific need. For instance, the bulk queue is tailored for improved “in-order packet delivery on multiprocessor, which likely results in improvements to CPU consumption and throughput” [3]. QDIO IWQ provides benefit on both sides of the connection hence it is enabled on both sides in our test set-up when applicable. Note that WORKLOADQ requires the processing of inbound traffic for the QDIO interface to be set as DYNAMIC (e.g., INBPERF DYNAMIC WORKLOADQ). Refer to this article for more information.

IPCONFIG SEGMENTATIONOFFLoad (LSO)

Any large amount of data traveling over the network is broken down into smaller segments by the TCP/IP stack. This process can be CPU intensive. As an alternative, segmentation offload (i.e., Large Send Offload) is an OSA-Express feature. It reduces host CPU utilization and increases data transfer efficiency by offloading segmentation processing to OSA [4].

TCPCONFIG AUTODELAYAck

Reduction in network traffic and CPU utilization can be achieved by delaying the TCP acknowledgement (ACK) *depending* on the traffic pattern. AUTODELAYAck enables the TCP stack to “automatically enable or disable a delayed ACK in a TCP connection based on the characteristic of the traffic” [5].

`IPCONFIG QDIOACCELerator`

QDIO Accelerator specifies that inbound packets that are to be forwarded by a TCP/IP stack are eligible to be routed directly between any of the following combinations of interface types: a HiperSockets interface and an OSA-Express QDIO interface, two OSA-Express QDIO interfaces, and two HiperSockets interfaces. These packets arrive at the forwarding stack, but do not traverse all the TCP/IP layers for forwarding. Therefore, valuable TCP/IP resources (storage and CPU) are not expended for purposes of routing and forwarding packets. This option also applies to packets that would be forwarded by the Sysplex Distributor. Refer to this article more information.

`MSG_WAITALL`

MSG_WAITALL is beneficial in streaming workloads. The flag bit decreases the frequency of interrupts occurring for the application receiving data as less interrupts can result in improvements to CPU consumption and throughput. The receiving application is interrupted only when all requested data can be returned. To avoid blocking the application indefinitely, the flag bit should only be set in scenarios where the application expects to receive enough data to fill its buffer, or the connection will terminate.

Performance Best Practice: Security

HEAPPOOLS64

Application Transparent Transport Layer Security (AT-TLS) creates System SSL environments using the z/OS Language Environment (LE). These System SSL environments use the LE runtime default options or those specified in the CEEPRMxx PARMLIB member. The default LE runtime does not have HEAPPOOLS64 enabled. For large AT-TLS configurations, running without HEAPPOOLS64 enabled could result in additional contention for user heap storage across the different System SSL environments. This could lead to slow-downs or timeouts processing TLS handshakes. The LE heap contention issue is also a big factor when you use TLSv1.3 regardless of the workload. By enabling the HEAPPOOLS64[3] runtime option, this contention for user heap storage can be eliminated. The following measurements (e.g., Figure 4 & 5) were gathered under the same z/OS hardware and software environment used for the AT-TLS section of the *z/OS V2R5 Communications Server Performance Summary* report [6]. As evident by both figures, HEAPPOOLS64 has a positive impact on connections using AT-TLS. In our measurements, the positive impact was measured across all TLSv1.2 and TLSv1.3 ciphers. z/OS Communications Server APAR PH59425 will ensure HEAPPOOLS64 is always enabled.

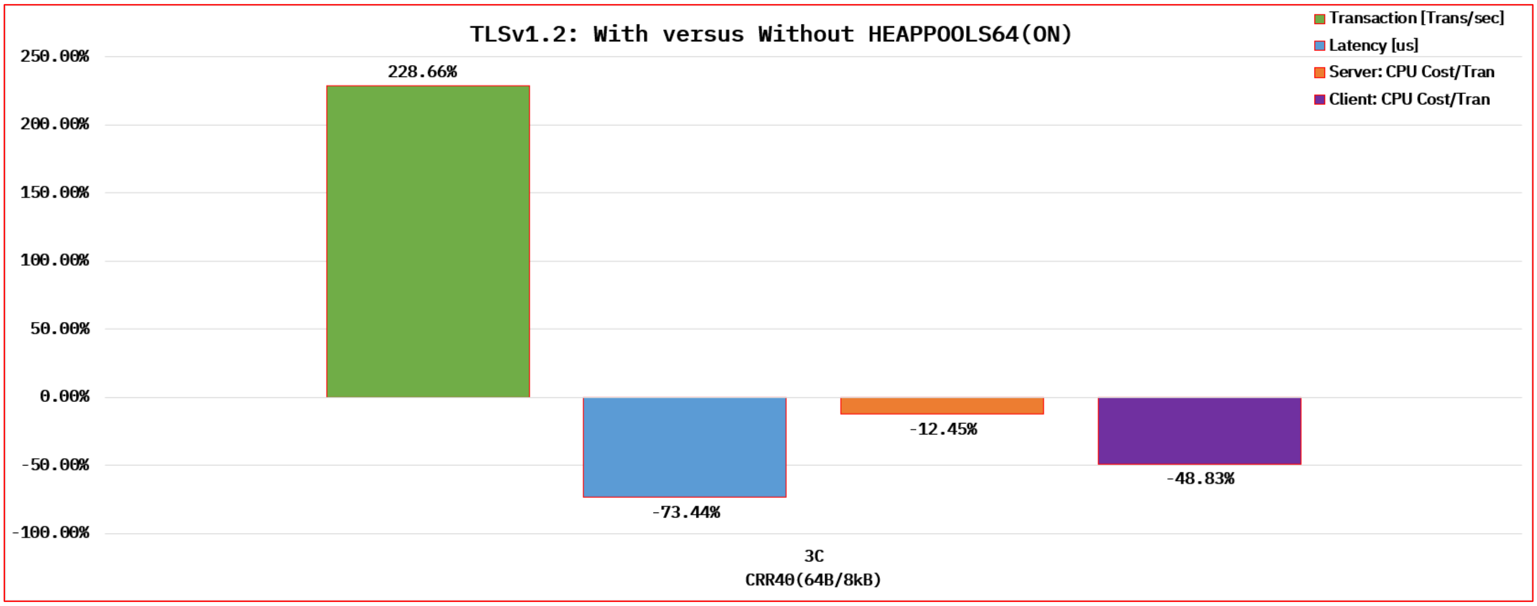

Figure 4

[3] Enabling HEAPPOOLS64 is referring to coding HEAPPOOLS64(ON) in the TCP/IP procedure where the default values were taken for each cell size and its associated pool count [16].

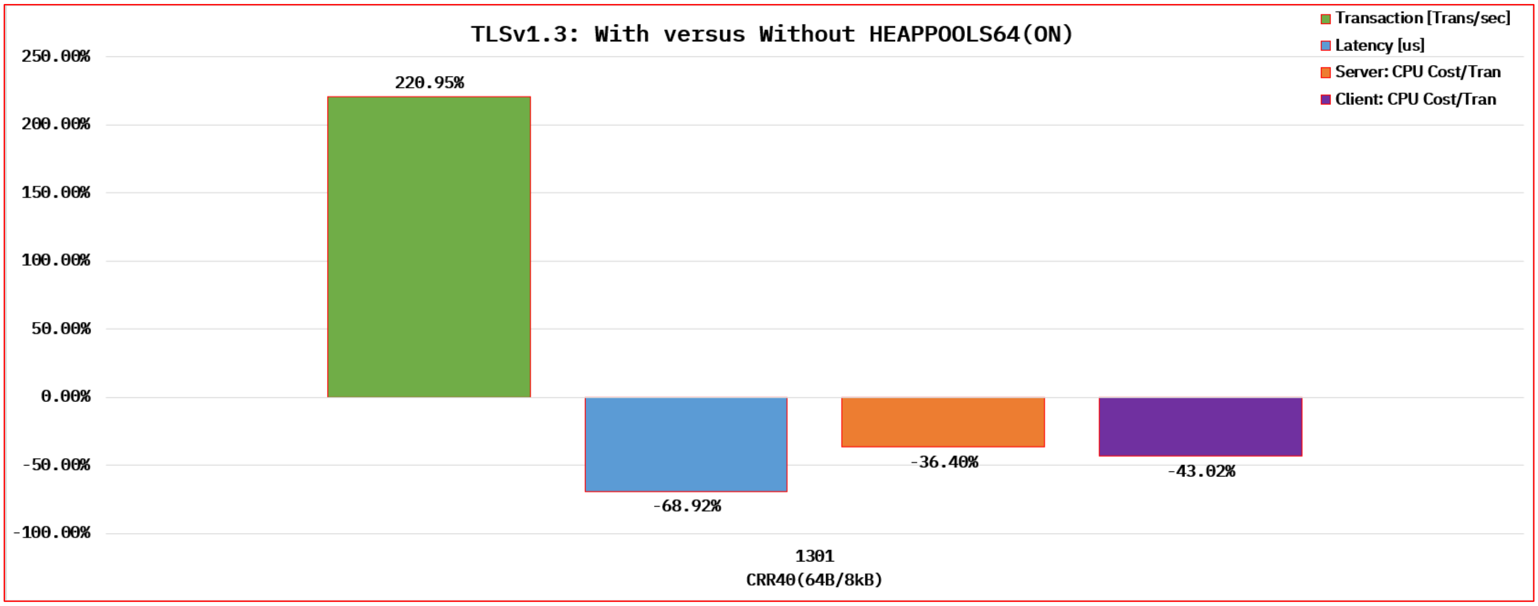

Figure 5

HEAPPOOLS64: TLSv1.2 versus TLSv1.3

To continue the above theme, we will now perform a comparison between TLSv1.2 against TLSv1.3. In this comparison, the main conclusion is as follows: *Enabling HEAPPOOLS64 is essential for migrating to TLSv1.3 which offers stronger security at a competitive cost*. z/OS Communications Server APAR PH59425 will ensure HEAPPOOLS64 is always enabled.

In the comparison, TLSv1.2 used the following cipher and signature pair: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (C030) and TLS_SIGALG_SHA256_WITH_RSASSA_PSS (0804).

In the comparison, TLSv1.3 used the following cipher and signature pair: TLS_AES_ 256_GCM_SHA384 (1302) and TLS_SIGALG_SHA256_WITH_RSASSA_PSS (0804).

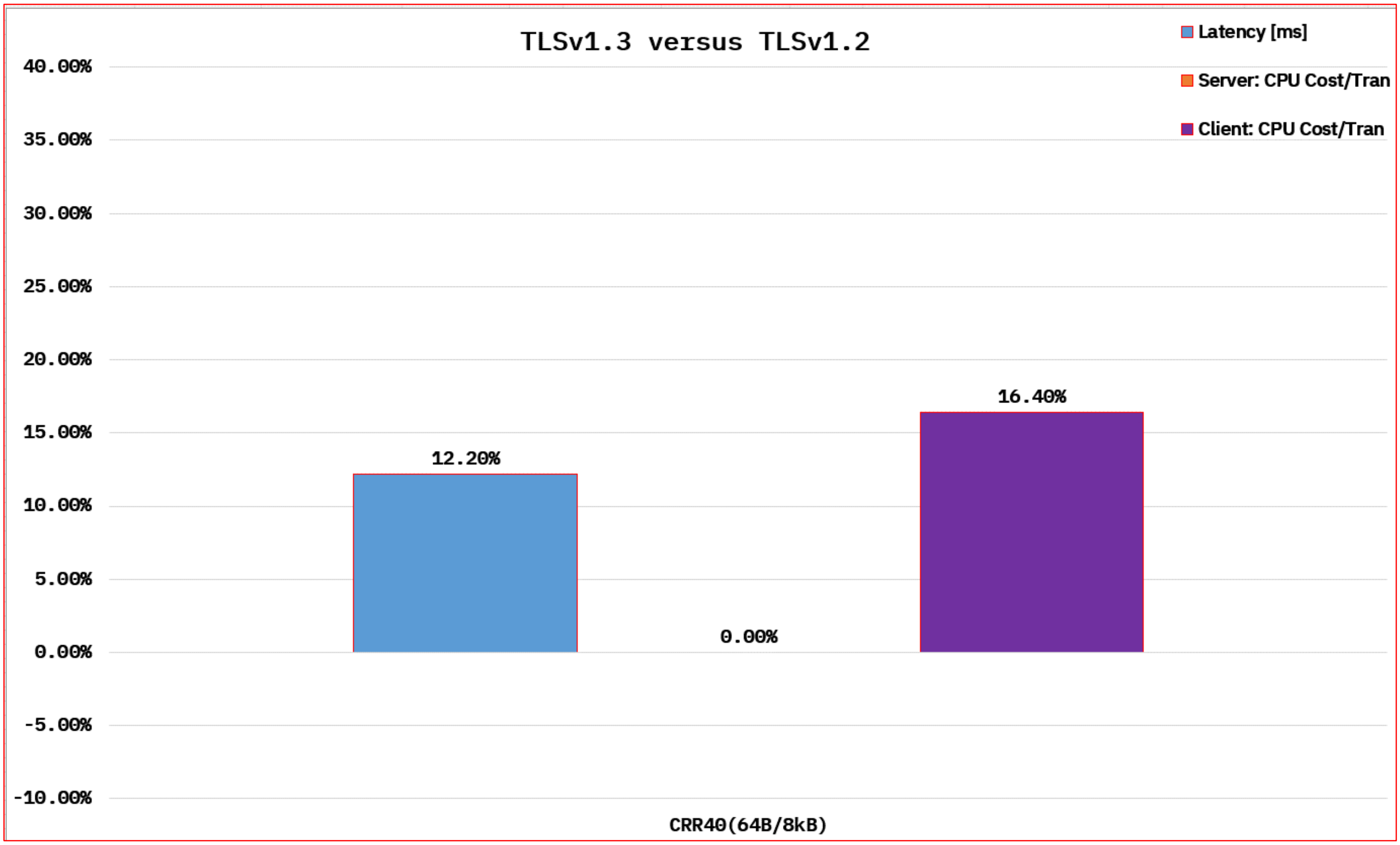

Figure 6

New TLS Performance Paper!

Recently, a new TLS performance paper was published by the z/OS Communications Server Performance Team. It focuses on AT-TLS performance using TLSv1.2 and TLSv1.3. The new paper, z/OS Communications Server TLS Performance Update, is accessible here.

Performance Best Practice: Workload(s) Reading From Storage Devices

High Performance FICON for IBM z Systems (zHPF)

High Performance FICON for IBM z Systems (zHPF) obeys the Fibre Channel Protocol (FCP) which is a high-speed data transfer mechanism between many different objects such as workstations, mainframes, supercomputers, storage devices, and displays [9]. For our purpose, the objects of interest are mainframes and storage devices. *zHPF increases the number of I/O per second by reducing the FICON channel and Control Unit (CU) link protocol overhead [9]*. Another aid to zHPF performance is special hardware contained in the IBM ASIC[4] available on FICON Express 8S and later generation adapters that was designed specifically for zHPF [10]. The requirements to exploit zHPF involves using a DASD controller (i.e., hardware) supporting the protocol (e.g., IBM DS8000, EMC/Dell, Hitachi, etc.) running on the minimum operating system level [14].

Testing Environment: GET & PUT via FTP with zHPF

The following diagram showcases the test topology under which a single LPAR sent over multiple MVS datasets to another LPAR on the same CPC via the FTP protocol in a loop.

[4] An Application Specific Integrated Circuit (ASIC) is designed for a specific logic (e.g., ASIC designed for the sole purpose of processing packets) instead of being general purpose

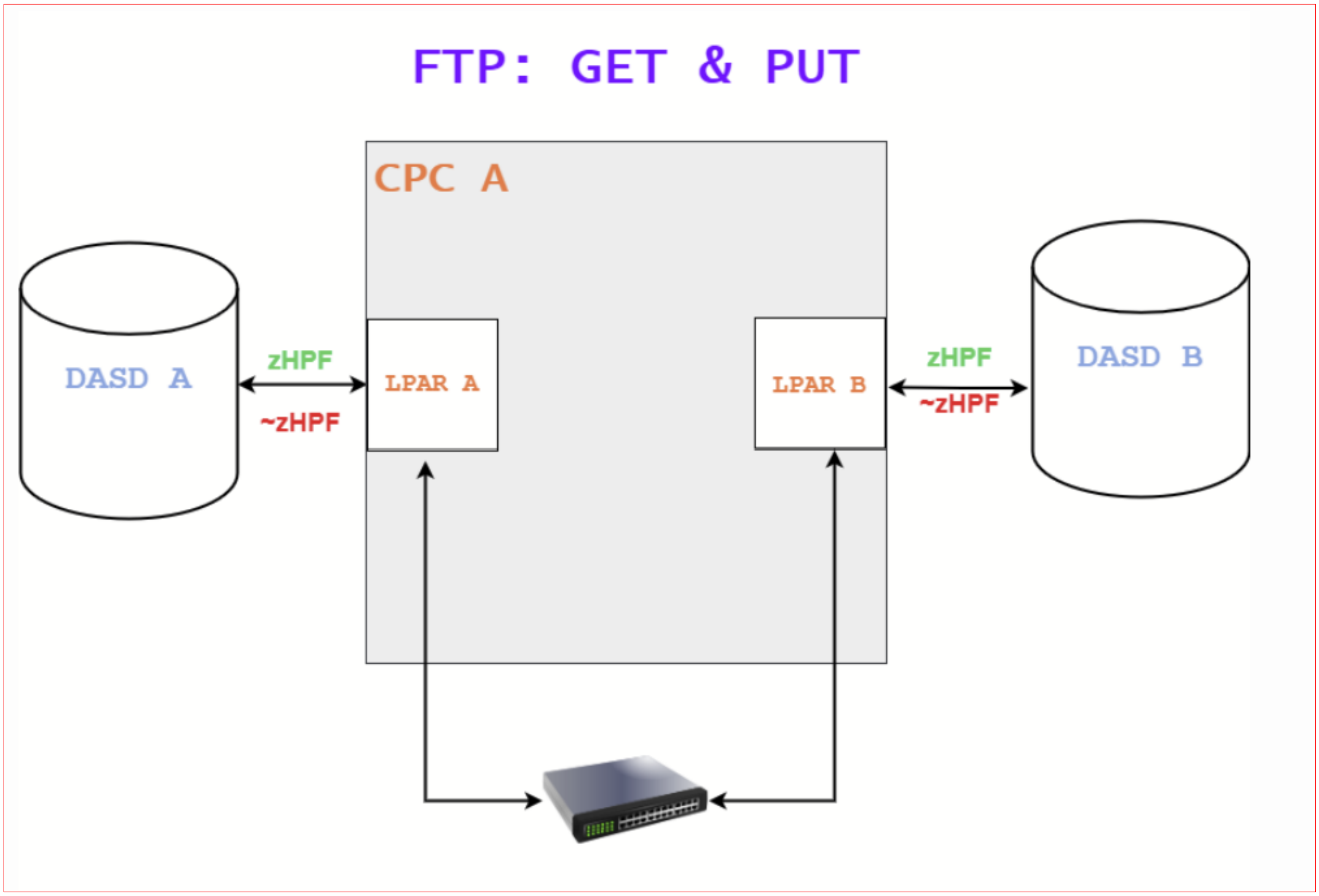

Figure 7: Test set-up for doing FTP GET & PUT while zHPF is enabled versus not enabled

z/OS Environment Configuration: GET & PUT via FTP with zHPF
Below is the environment configuration in which the data was collected:

- Central Processor Complex (CPC): z16
- Release: 3.1
- Number of CPUs: 2 (Dedicated) per LPAR
- Interface: OSA-Express 7S 10 GbE
- Workload:
  - GET
  - PUT
- File Transfer Type: ASCII

Observations

An increase in the number of I/O per second while doing the processing on a designated ASIC resulted in: higher throughput, lesser delay, lesser server CPU Cost, and lesser total duration when transferring a dataset via FTP as shown in the following figure.

Results

The above observations are represented in the following graph.

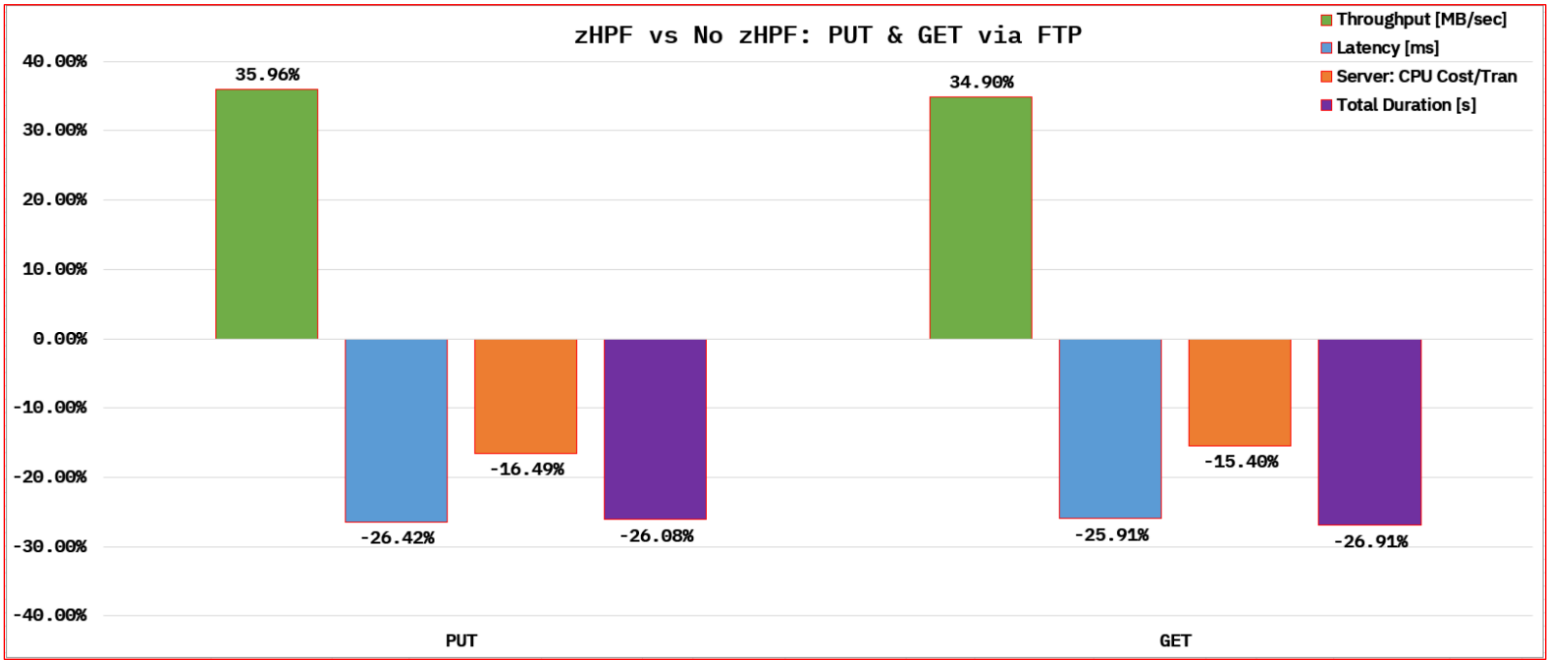

Figure 8: Using zHPF for reading & writing datasets for FTP transfers

3.1: New Functions

z/OS Container Platform!

Background: Generic

In general, containers became popular for their agility. It reduces application development and unit testing time. It eradicates the notion of "it worked on my local computer, but I am not sure why it is not working on your local computer". [15] describes containers concisely – "Containers are executable units of software in which application code is packaged along with its libraries and dependencies, in common ways so that the code can run anywhere – whether it be on desktop, traditional IT or the cloud."

Background: New Function Specific

z/OS Container Platform required many cross-team collaboration and contribution. One such team was "z/OS Communications Server" which contributed to the network connectivity of a container allowing an application running natively on your local computer communicate with an application running in a container on z/OS (e.g., client-server model). It is also possible to use z/OS Container Platform in a service-to-service environment.

Testing Environment: z/OS To z/OS vs z/OS Container Platform to z/OS Container Platform

The goal was to benchmark the following configuration than perform a comparison of the extracted metrics:

1. A client and a server running natively on z/OS (e.g., figure 9)
    a. Figure 9 shows client applications running natively on **z/OS UNIX** where it communicates with a remote server application also running natively on **z/OS UNIX**
2. A client and a server running natively in a z/OS Container Platform (e.g., figure 10)
    a. Figure 10 shows containerized client applications running on **z/OS** where it communicates with a remote containerized server application also running on **z/OS**

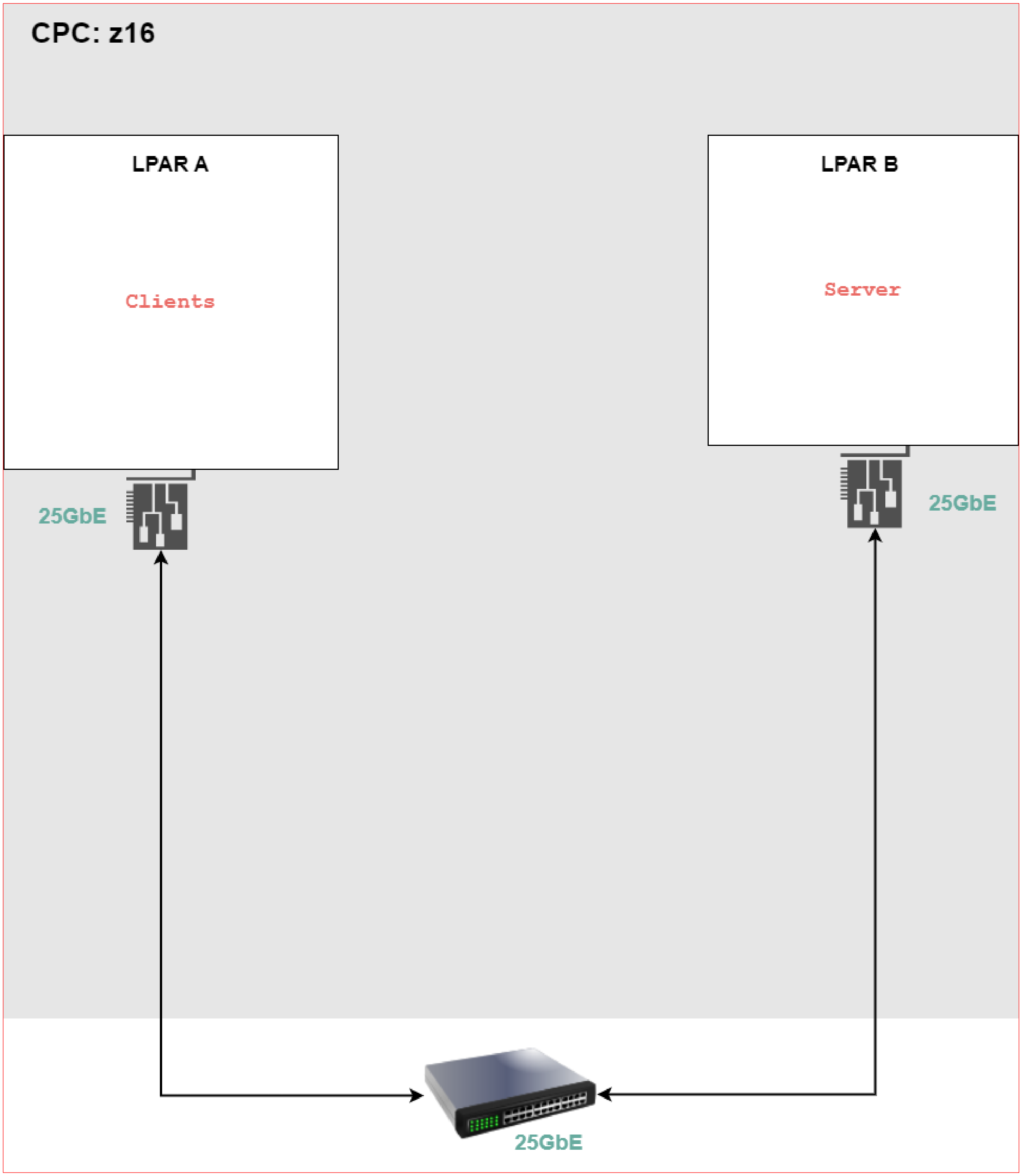

Figure 9: Test set-up where the clients and server run natively on z/OS UNIX

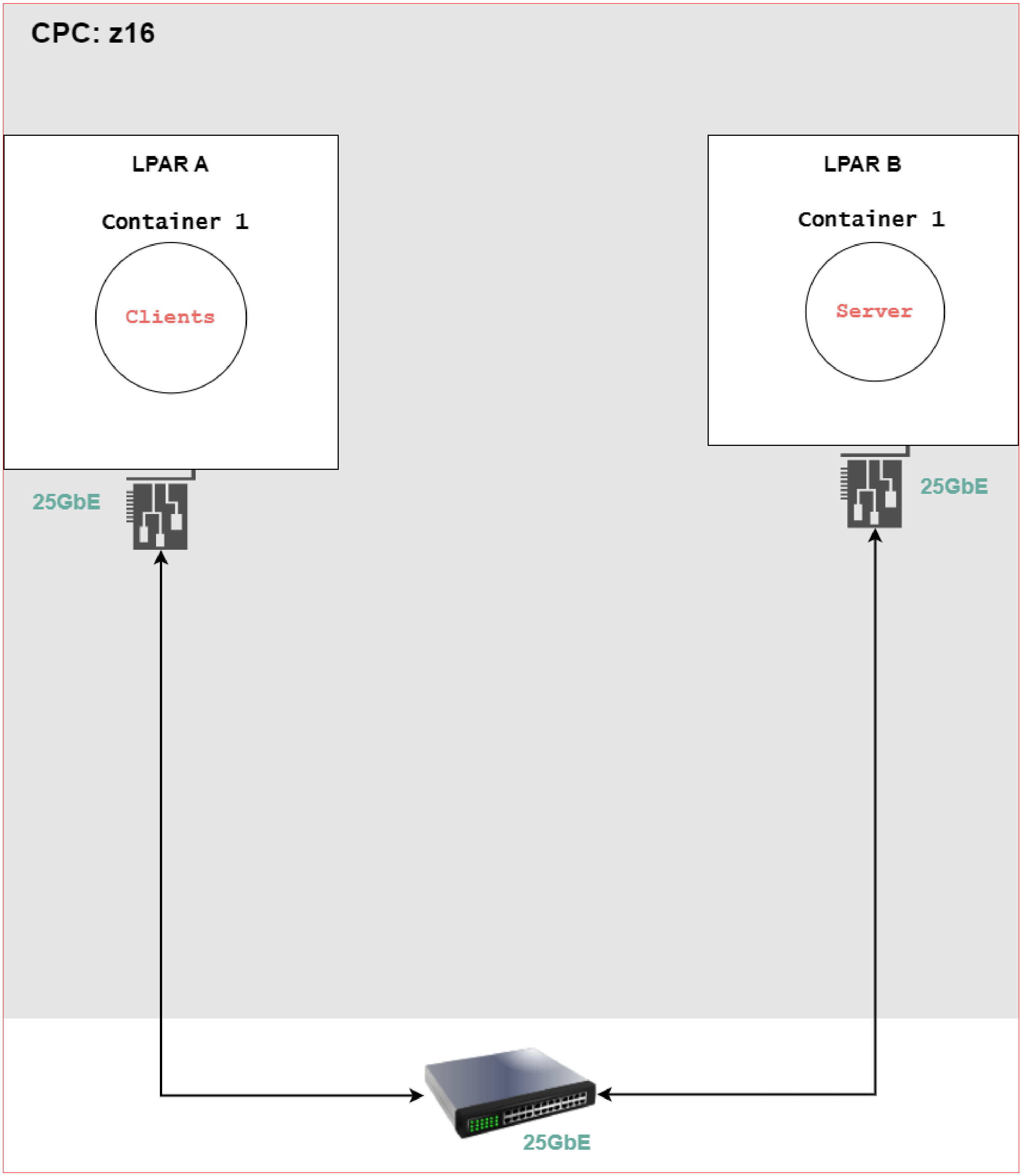

Figure 10: Test set-up where the client & server are containerized

z/OS Environment Configuration: z/OS Container Platform

Below is the environment configuration in which the data was collected:

- Central Processor Complex (CPC): z16
- Release: 3.1
- Number of CPUs: 2 (Dedicated) per LPAR
- Interface: OSA-Express 7S 25GbE
- Workloads:
    - RR10(1kB/1kB)
    - CRR10(64B/8kB)
    - STR3(20MB/1B)

Testing Outcome: Nil Difference!

The intent was to create a z/OS Container Platform for which the performance of containerized z/OS UNIX applications is equivalent to running them natively on z/OS. The expectation was met. In other words, *there is minuscule performance difference in the realm of network performance when an application runs directly on z/OS versus running containerized.*

z/OS UNIX SYSLOGD Support for Secure Logging Over TCP

Background: Generic

Syslog is a standard for message logging defined by RFC 5424 whereas syslog daemon (syslogd) is a server process running in the z/OS UNIX environment. Syslogd is an implementation of the message logging standard. The daemon gives flexibility to applications for local logging or sending logs to other daemons on the network [8].

Background: New Function Specific

Syslogd only supported UDP until the release of this new function. If customers wanted to protect syslogd traffic prior to this new function, then they would need to create IPsec VPNs. To eradicate this limitation, in z/OS 3.1, syslogd now has the capability to communicate with other syslogd daemons over TCP connections resulting in an increased reliability with easier traffic protection through AT-TLS.

Testing Environment: SYSLOGD via TCP vs UDP

The following diagram (e.g., figure 11) shows the test set-up. In a single CPC, three LPARs each have an application writing logs into syslogd who sends the logs to another LPAR on the same CPC over the following different protocols:

- UDP
- TCP
- TCP with AT-TLS

z/OS Environment Configuration: z/OS UNIX SYSLOGD Support for Secure Logging Over TCP

Below is the environment configuration in which the data was collected:

- Central Processor Complex (CPC): z16
- Release: 3.1
- Number of CPUs: 2 (Dedicated) per LPAR
- Interface: OSA-Express 7S 1GbE

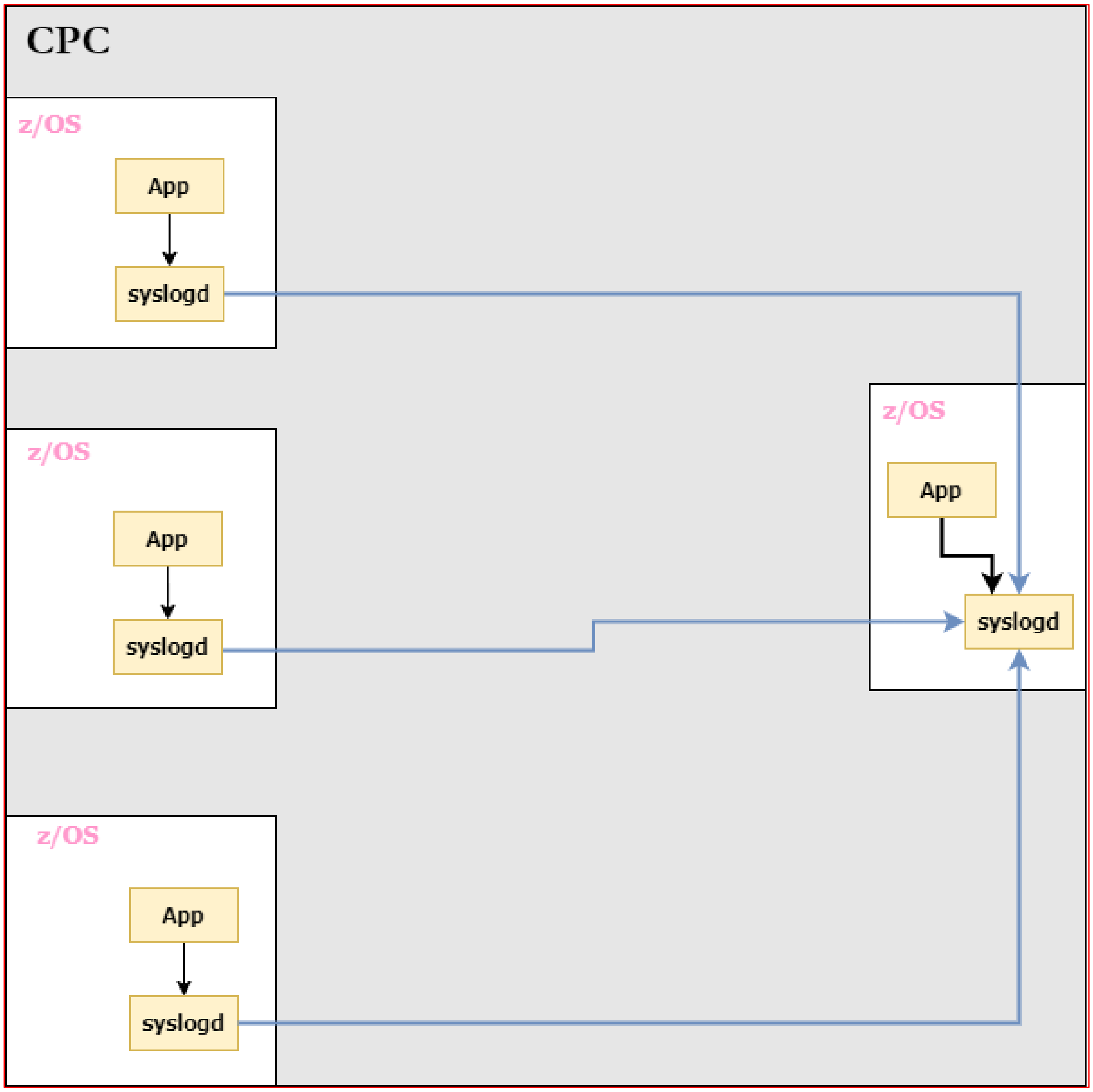

Figure 11: Multiple syslog daemons consolidating their logs

Testing Outcome: SYSLOGD via TCP vs UDP

The expectation is that if syslogd uses TCP as the transport layer instead of UDP then the performance difference of such choice should be nil or close to nil.

The expectation was met.

Testing Outcome: SYSLOGD via TCP when configuring AT-TLS

The same test set-up (e.g., figure 11) was used to measure the cost of using syslogd over TCP when configuring AT-TLS. The conclusion of the testing is as follows: *The server networking CPU cost per transaction increases by ~ 5% when the TCP connection is protected by AT-TLS.*

3.1 vs. V2R5: Release to Release Performance Comparison

3.1 vs. V2R5

Introduction

In this sub-section, the pure focus was on benchmarking the latest release, 3.1, against the previous release, V2R5.

z/OS Environment Configuration

Below is the environment configuration in which the data was collected:

- Central Processor Complex (CPC): z16
- Release: 3.1 & V2R5
- Number of CPUs: 2 (Dedicated) per LPAR
- Interface: OSA-Express 7S 10GbE, OSA-Express 7S 25GbE
- Workloads
    - RR60(4kB/4kB)
    - CRR40(64B/8kB)
    - STR6(1B/20MB)

Synopsis

Performance of 3.1, which consists of new functions and improved existing functions, is equivalent to V2R5.

SMC-Rv2: 3.1 vs. V2R5

Introduction

In this sub-section, the pure focus was on benchmarking SMC-Rv2 from a release-to-release perspective as this function is available on both releases. SMC-Rv2 is a protocol that implements the RoCEv2 standard (i.e., routable RoCE). The protocol will exploit your current routing methodology (e.g., static versus dynamic) and network topology (e.g., firewalls) [7]. The protocol also supports direct links.

z/OS Environment Configuration

Below is the environment configuration in which the data was collected:

- Central Processor Complex (CPC): z16
- Release: 3.1 & V2R5
- Number of CPUs: 2 (Dedicated) per LPAR
- Interface
  - TCP: OSA-Express 7S 25GbE
  - SMC-Rv2: RoCE Express3 25GbE
- Workloads
  - RR10(4kB/4kB), RR50(256kB/256kB)
  - STR1(1B/20MB); STR3(1B/20MB)
- Number of hops: 0

Synopsis

The goal of this measurement was to benchmark the performance of SMC-Rv2 in a directly attached set-up. The finding was as follows – *The protocol, SMC-Rv2, performance on 3.1 is equivalent to V2R5*.

Sysplex Distributor

Introduction

In this sub-section, the pure focus was on benchmarking Sysplex Distributor from a release-to-release perspective as this function is available on both releases.

z/OS Environment Configuration

Below is the environment configuration in which the data was collected:

- Central Processor Complex (CPC): z16
- Release: 3.1 & V2R5
- Number of CPUs: 2 (Dedicated) per LPAR
- Interface: OSA-Express 7S 25GbE
- Workloads
  - RR40(1kB/1kB)
  - STR1(20MB/1B)

Synopsis

Sysplex Distributor performance on 3.1 is equivalent to V2R5.

IPv6

Introduction

z/OS Communications Server supports IPv6 internet protocol addresses. In comparison to IPv4, IPv6 provides no practical limit on global addressability.

z/OS Environment Configuration

Below is the environment configuration in which the data was collected:

- Central Processor Complex (CPC): z15
- Release: 3.1 & V2R5
- Number of CPUs: 2 (Dedicated) per LPAR
- Interface: OSA-Express 7S 10GbE, OSA-Express 7S 25GbE
- Workloads
    - RR60(4kB/4kB)
    - CRR40(64B/8kB)
    - STR6(1B/20MB)

Synopsis

IPv6 performance on 3.1 is equivalent to V2R5.

Reminders!

SMC via Sysplex Distributor

Introduction

As a reminder, we wanted to resurface [13] which informs readers on the usage of SMC with other z/OS Communication Server functions. In this specific instance, the other function is "Sysplex Distributor". The article conveys how bypassing the distributing node results in performance improvements. Such a statement is correct which we will showcase in this section. The following diagram gives a visual representation of how the bypass is performed.

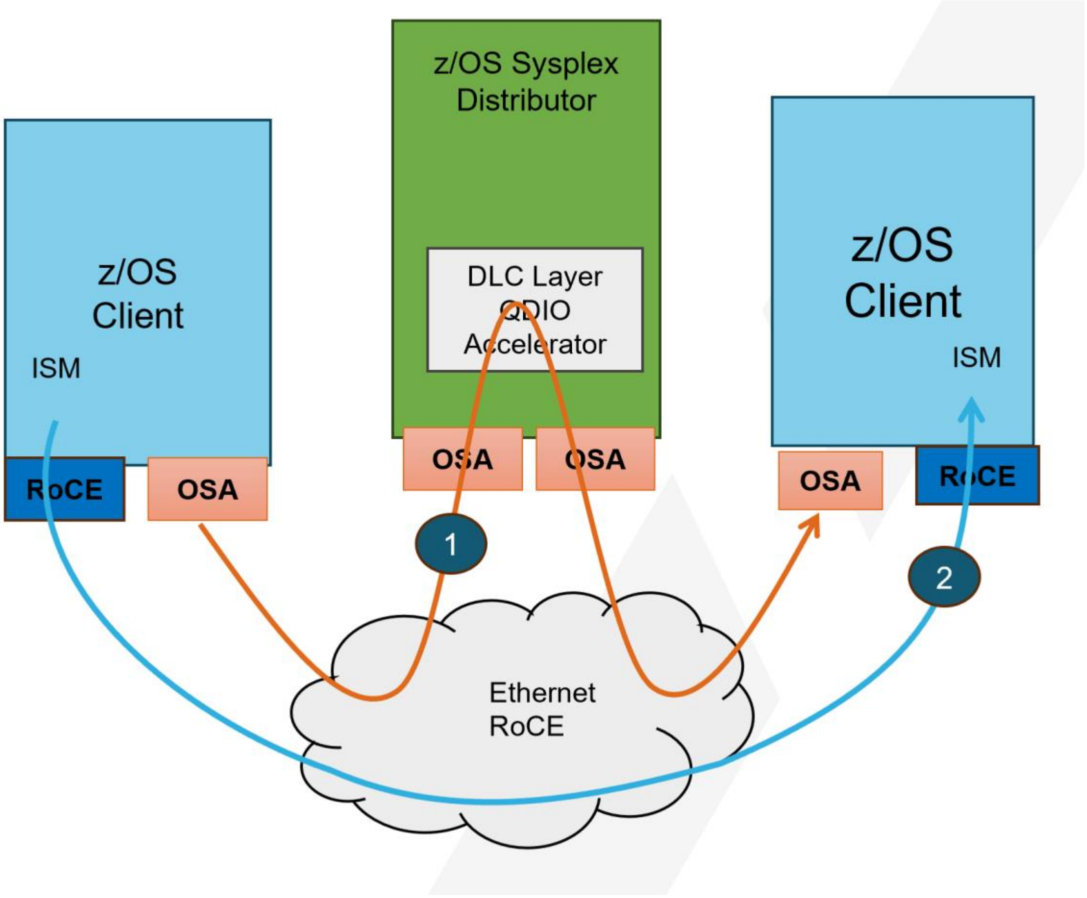

Figure 12

Line one represents the connection establishment path. It uses OSA to understand whether the client and server are SMC eligible. If so then it performs the SMC connection negotiation. After successful negotiation, it uses line two for data movement. The Sysplex Distributor (SD) node is bypassed during data movement which equates to lesser CPU usage on the SD host.

z/OS Environment Configuration

Below is the environment configuration in which the data was collected:

- Central Processor Complex (CPC): z16
- Release: 3.1
- Number of CPUs: 2 (Dedicated) per LPAR
- Interfaces
  - OSA-Express 7S 25GbE
  - RoCE Express2 25GbE
- Workloads
  - RR20(100B/800B)
  - RR10(1kB/1kB)
  - STR1(20MB/1B)

Synopsis

As evident by the following figures, an enormous CPU reduction is achievable by using SMC with Sysplex Distributor as stated by [13]. The following figures exemplifies the benefit of SMC via Sysplex Distributor instead of using TCP/IP. The higher CPU savings bar represents the comparison of SMC against TCP/IP not using QDIOACCELerator whereas the lower CPU savings bar represent the enablement of QDIOACCELerator.

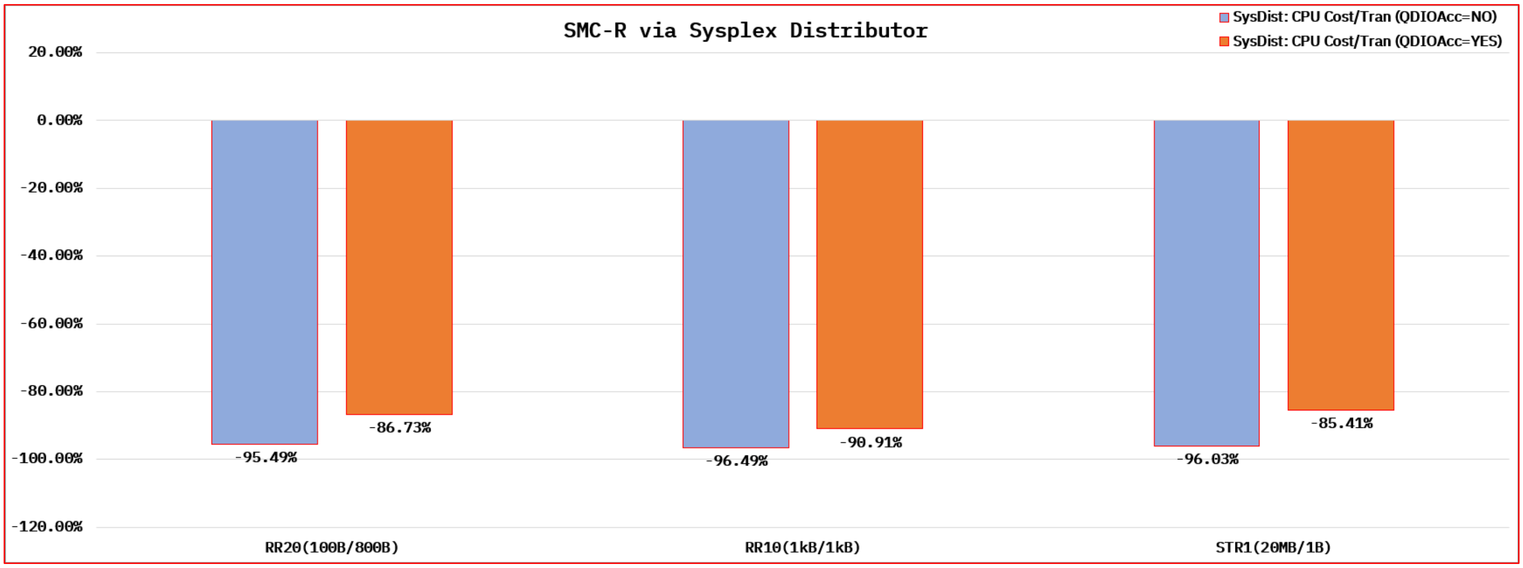

Figure 13: Comparing SMC-R via Sysplex Distributor against TCP/IP via Sysplex Distributor with and without QDIOACCELerator

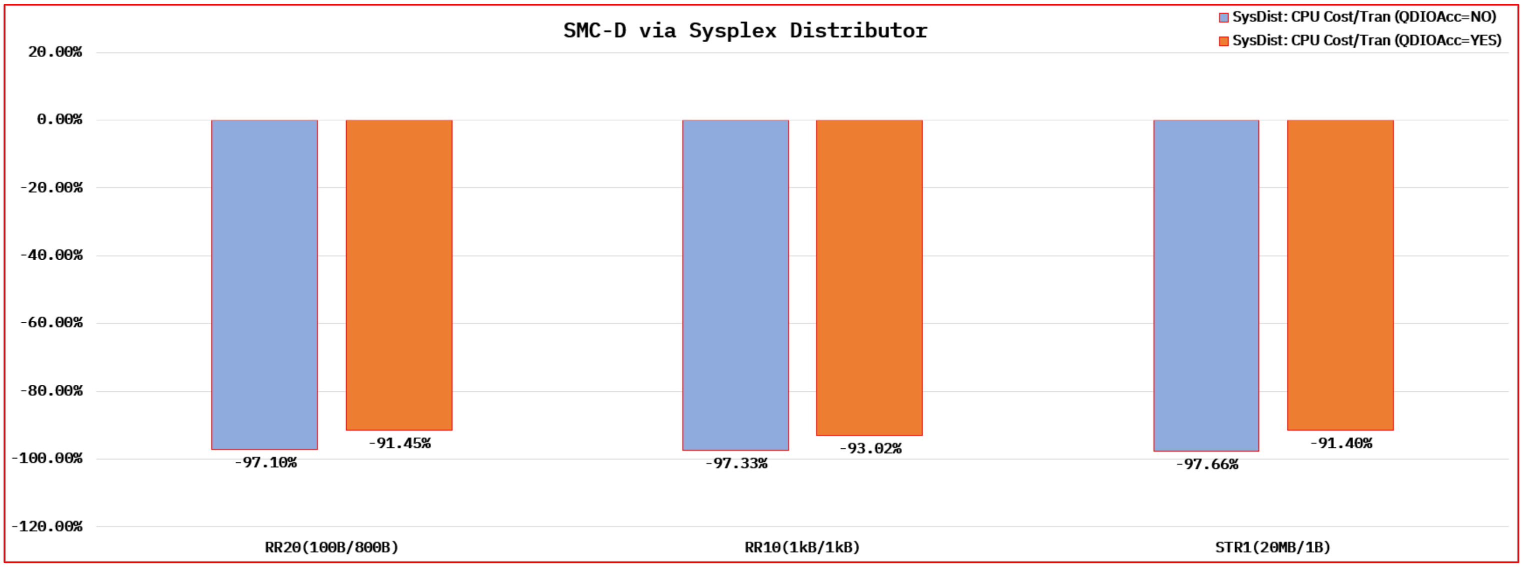

Figure 14: Comparing SMC-D via Sysplex Distributor against TCP/IP via Sysplex Distributor with and without QDIOACCELerator

Miscellaneous Testing

SMC-Rv2 at Distance

Introduction

In this sub-section, the pure focus is on explaining recent testing of the SMC-Rv2 implementation of the RoCEv2 standard over distance such as 10 [km].

In the previous report, a general background was given for the topic then a comparison was completed of the SMC-Rv2 protocol over a single versus multiple subnets [6]. In the latter case, the number of hops in between the different subnets was one.

A benchmark was performed spanning multiple subnets over a greater distance. Such benchmark is a repeat of the SMC-R over distance measurements taken in 2014 with the only difference being the workload traversing multiple subnets [11]. The study in 2014 compared RoCE against TCP over a distance.

z/OS Environment Configuration

Below is the environment configuration in which the data was collected:

- Central Processor Complex (CPC): z15
- Release: V2R5
- Number of CPUs: 4 (Dedicated) per LPAR
- Interface
  - OSA-Express 7S 10GbE, OSA-Express 7S 25GbE
  - RoCE Express2 10GbE, RoCE Express2 25GbE
- Workloads
  - RR10(32kB/32kB)
  - STR1(1B/20MB)
  - STR3(1B/20MB)
- Number of hops: 2

Synopsis

As evident by the following figure, SMC-Rv2 over distance such as 10 [km] performs much better than TCP in terms of transactions, delay, and CPU cost. The RR workloads were measured over 10GbE NICs & RNICs whereas the long-lived flows (i.e., streaming) were measured over 25GbE NICs & RNICs. A more detailed study will be published soon on this specific topic.

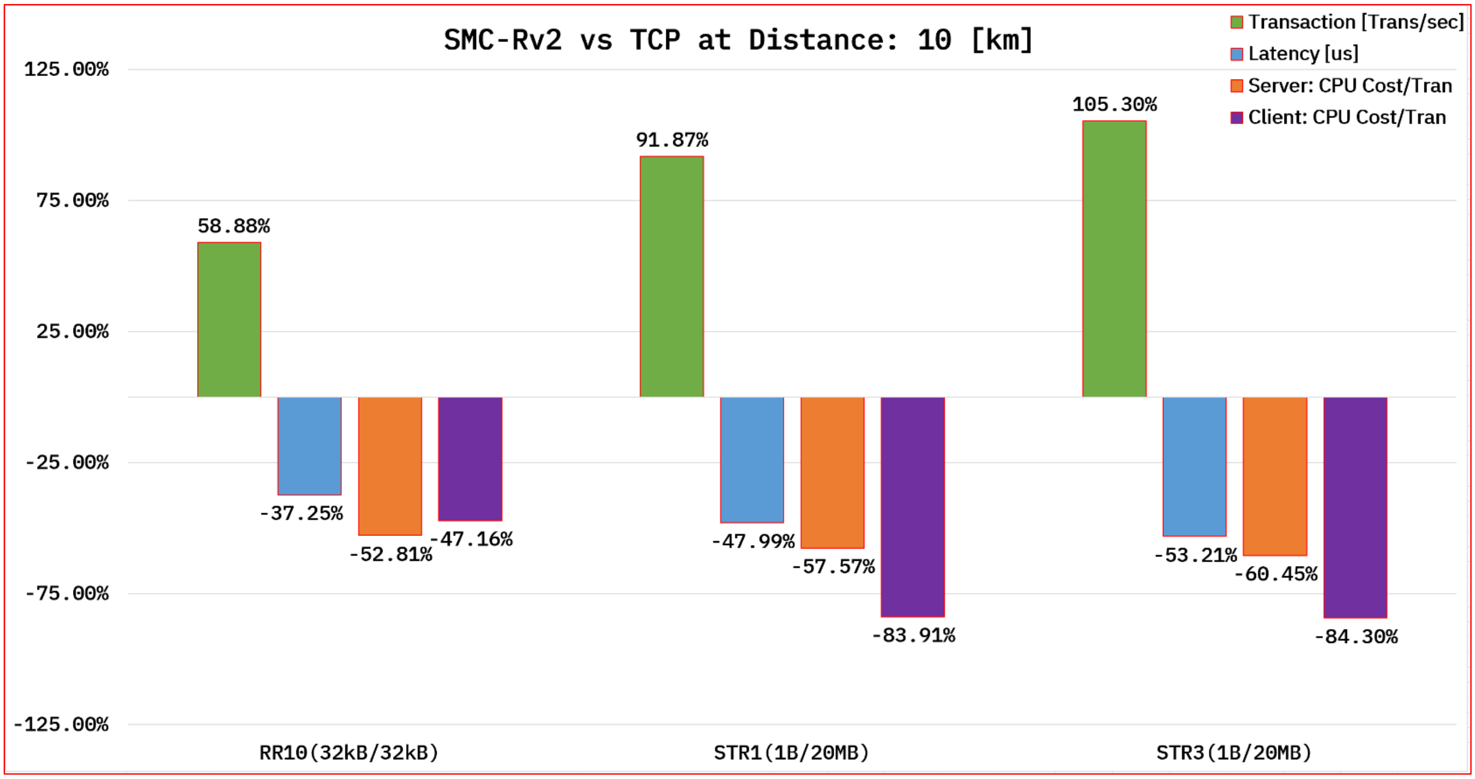

Figure 15: Comparison of SMC-Rv2 against TCP at a 10 [km] distance

References

z/OS Communications Server Performance Index

The following index contains all z/OS Communications Server Performance related publications. The posted materials are updated as necessary.

URL: https://www.ibm.com/support/pages/node/317829

Additional References

[1] J. Stevens, "IBM z/OS Communications Server and OSA-Express Best Practices," IBM, Raleigh, NC, Technical Report. Version 1.2, 08 August 2023.

[2] C. Rufus, *IBM z/OS V2R1 Communications Server TCP/IP Implementation Volume 3: High Availability, Scalability, and Performance,* 1st ed. USA: IBM, 2013, pp. 292 - 293

[3] "*QDIO inbound workload queueing",* IBM. Accessed On: Mar. 31, 2020. [Online]. Available: Here

[4] D. Herr, *Getting the most out of your OSA (Open Systems Adapter) with z/OS Comm Server*. RTP: IBM, 2013, [Online]. Available: share.confex.com/share/121/webprogram/Handout/Session13222/SHARE%20OSA_Boston.pdf.

[5] B. White, O. Ferreira, T. Missawa, T. Sudewo, *IBM z/OS V2R2 Communications Server TCP/IP Implementation: Volume 3 High Availability, Scalability, and Performance.* USA: IBM, 2016, pp: 305.

[6] K. Rashad, "V2R5: z/OS Communications Server Performance Summary Report," IBM, Raleigh, NC, Technical Report. 11 May 2022.

[7] "SMC-Rv2", IBM. Accessed On: December 15, 2023. [Online]. Available: Here

[8] "More flexible! z/OS UNIX syslogd support for secure logging over TCP", IBM. Accessed on: December 31, 2023. [Online]. Available: Here

[9] Steve Guendert, "Understanding High Performance FICON (zHPF) Part 1: Protocol specifics," Brocade, 2013.

[10] Lou Ricci, "High Performance Ficon Demystified," IBM Corp., Anaheim, CA, USA, 2011.

[11] G. Kassimis, "Shared Memory Communications over RDMA (SMC-R) Performance update: SMC-R over distance," IBM, Raleigh, USA, Tech. Report. Sept. 2014.

[12] "Abstract for Resource Measurement Facility User's Guide", IBM. Accessed on: February 8, 2024. [Online]. Available: Here

[13] "Sysplex distributor", IBM. Accessed on: February 20, 2024. [Online]. Available: Here

[14] H. O'Neal, *More Data, Less Chatter: Improving Performance on z/OS with IBM zHPF*. RTP: IBM, 2015, [Online]. Available: Here.

[15] "What are containers", IBM. Accessed on: February 8, 2024. [Online]. Available: Here

[16] "HEAPPOOLS64 (C/C++ and AMODE 64 only)", IBM. Accessed on: April 23, 2024. [Online]. Available: Here