IBM z/OS Shared Memory Communications over RDMA: Performance Considerations

David Herr
Dan Patel

Table of Contents

Executive Summary

Mobile, cloud and social computing, smarter computing applications and analytics are driving the proliferation of data on the mainframe, creating a need for high performance communications to meet the demands of this increased workload. With the recent availability of RDMA over Converged Ethernet (RoCE), enterprise data centers now have an opportunity to realize the benefits of highly efficient RDMA-based networking solutions over existing Ethernet networking fabrics.

The IBM zEnterprize* EC12 (and BC12) and z/OS* V2R1 introduced optimized communications with an innovative solution: Shared Memory Communications – Remote Direct Memory Access (RDMA) or SMC-R. With SMC-R and the IBM System z RoCE Express feature, System z network capability takes a new leap, strengthening performance for sharing data and reducing data transmission network overhead.

This paper is provided for both IBM implementers and IBM customers who have an interest in the performance characteristics of the IBM z/OS SMC-R function. It is assumed that readers already have a basic background in TCP/IP protocols and the related z/OS implementation of those protocols.

The primary purpose of this paper is to describe the z/OS SMC-R protocol performance attributes and how these attributes can translate into better performance and reduced CPU consumption in customer environments.

Using the set of sample benchmarks (provided in this paper) of actual IBM middleware solutions, the reader will understand how this new technology can provide competitive advantages to their business critical workloads.

Summary of Topics

SMC-R is an open protocol defined in the informational RFC entitled *Shared Memory Communications over RDMA* (https://ietf.org/doc/draft-fox-tcpm-shared-memory-rdma/). This paper focuses exclusively on the IBM z/OS implementation of SMC-R.

The paper is organized into the following topics:

1. **Shared Memory Communications over RDMA Overview**
   is an introduction and overview of Shared Memory Communications over RDMA (SMC-R) concepts, architecture and z/OS implementation.
2. **Performance results**
   reviews performance comparisons between SMC-R and TCP/IP using a variety of workloads.

z/OS Shared Memory Communications over RDMA Overview

This section provides an overview of the Shared Memory Communications over RDMA support that is introduced in z/OS Communications Server V2R1.

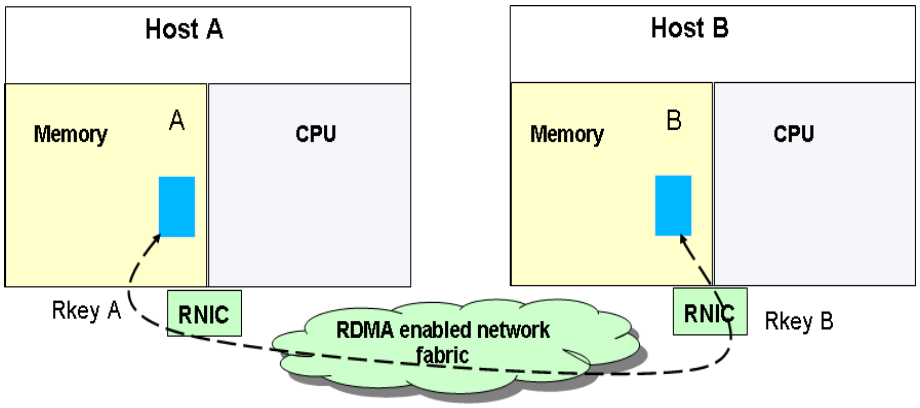

Figure 1 RDMA Overview

Remote Direct Memory Access (RDMA) is a communications technology that enables a host to make a subset of its memory, called Remote Memory Buffer (RMB), directly available to a remote host. By doing so, data can be transferred between hosts very efficiently and without any help from the CPU on the source or target host. Historically, RDMA has been confined to high-performance computing environments where the cost of maintaining RDMA-capable network fabrics such as InfiniBand was justified given the emphasis of performance over cost. However, RDMA is now available on standard Ethernet-based networks by using the industry (InfiniBand Trade Association) standard referred to as RDMA over Converged Ethernet (RoCE). With RoCE, the cost of adopting RDMA is lower because it can flow over the Ethernet fabrics that are already in place to carry IP network communications. Both standard TCP/IP and RDMA traffic can flow over the same physical LAN fabric at the same time, but RDMA network interface cards (RNICs, also referred to as RoCE host channel adapters (HCAs)), are required to do so. On System z, the 10Gb RoCE Express adapter serves as the RNIC.

z/OS Communications Server V2R1 introduces a new capability that combines the performance benefits of RDMA with the widely-used TCP/IP sockets programming interface. This function, called *Shared Memory Communication – RDMA (SMC-R)* allows your TCP sockets applications to benefit from direct, high-speed, low-latency, memory-to-memory (peer-to-peer) communications over RDMA transparently – no changes are required to application programs.

SMC-R provides an enterprise class of services for RDMA that are designed for enterprise-class data-center networks. Communicating peers (the z/OS TCP/IP stacks) dynamically learn about the shared memory capability by using traditional TCP/IP connection establishment flows. With this awareness, the TCP/IP stacks can switch from TCP network flows to more efficient direct memory access flows that use RDMA. The application programs are unaware of the switch to shared memory communications.

The remainder of this section will describe relevant characteristics of SMC-R communications in enough detail to provide a basis for the later performance discussion. For a more complete description of the z/OS SMC-R implementation, refer to the *z/OS Communications Server IP Configuration Guide Version 2 Release 1 (SC27-3650)*, Chapter 10.

SMC-R: A Hybrid Protocol

Shared Memory Communications over RDMA is a *hybrid protocol* that uses RDMA technology within an existing IP network topology. SMC-R connections are established and operate transparently to applications using their existing TCP socket connections. IP communications occur over OSA adapters as they have in the past, with the associated SMC-R connections being established over the RNICs. This requires that the RNICs be attached to the same network infrastructure as the OSAs.

SMC-R's reliance on existing IP network topology and TCP connection setup preserves critical TCP/IP operational and network management features, including compatibility with transport layer load balancers (e.g., Sysplex Distributor) and minimal or no topology changes to accommodate the use of RDMA.

SMC-R Eligibility

In order for two nodes to be eligible to communicate with SMC-R, several criteria must be met:

- Both must be enabled for SMC-R
- Both must have direct access to the same physical LAN fabric
- Both must have direct access to the same IP subnet and VLAN (if VLANs are defined)

The "direct access" requirements are based on the fact that the underlying RDMA connections are non-routable. This means that *SMC-R connections are not routable* as well. The direct access requirements ensure that a direct communication path exists at layer 2 between the SMC-R capable nodes, with no intervening IP router. The additional VLAN requirement further confines the traffic within the physical LAN fabric in cases where VLANs are in use.

The topology requirements are illustrated in Figure 2 showing how the SMC-R enabled TCP connections between HOST A and HOST B are allowed as a result of both hosts having OSA and RoCE adapters that are connected to the same physical LAN fabric, VLAN and IP subnet. Conversely, even though HOST C is attached to the same physical LAN fabric, it cannot establish any IP connections to HOST A or HOST B without an intervening IP router since it is connected to a different VLAN and IP subnet.

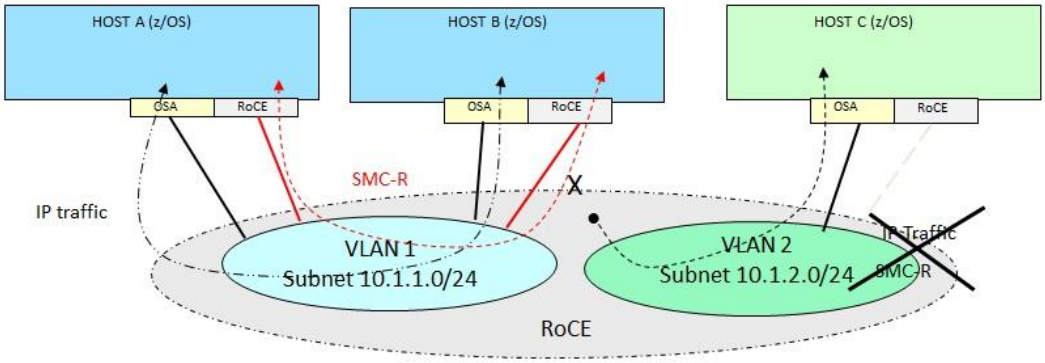

Figure 2 Network topology and SMC-R eligibility

SMC-R connection processing leverages existing IP topology (TCP/IP connection setup). Therefore, SMC-R connections are able to transparently "inherit" the same VLAN and IP subnet connection eligibility attributes of the associated TCP connection. When VLANs are in use, SMC-R connections then become VLAN qualified.

Since SMC-R's topology and eligibility requirements mimic those of IP, the level of trust that an enterprise has in its IP network infrastructure should mirror its trust in that infrastructure when SMC-R is deployed.

Enabling SMC-R and Connection Setup

SMC-R is enabled when you specify the SMCR parameter of the GLOBALCONFIG statement in the TCP/IP profile data set and include one or more Peripheral Component Interconnect Express (PCIe) function ID (PFID) values. Each PFID value represents an RNIC adapter (RoCE Express feature) that is configured by using the traditional hardware configuration definition (HCD) utility panels. TCP/IP activates the RNICs when the first SMC-R capable IP interface is started. By default, IPAQENET (IPv4) and IPAQENET6 (IPv6) interfaces with the OSD channel path ID type are enabled for SMC-R capability. For IPv4 QDIO interfaces defined via the DEVICE/LINK/HOME statements, you must first convert it to an IPAQENET INTERFACE statement. Each RNIC is associated with one or more OSD interfaces by a common PNETID (Physical Network ID) that is specified for each RNIC and OSD physical port using HCD.

All TCP connections that traverse SMC-R capable IP interfaces are eligible for SMC-R communications. The decision about whether an eligible connection will use SMC-R communications is made during traditional TCP connection establishment. The sequence of flows that determine whether or not to use

SMC-R on a given TCP connection is called *Rendezvous processing*, which is illustrated in Figure 3 below.

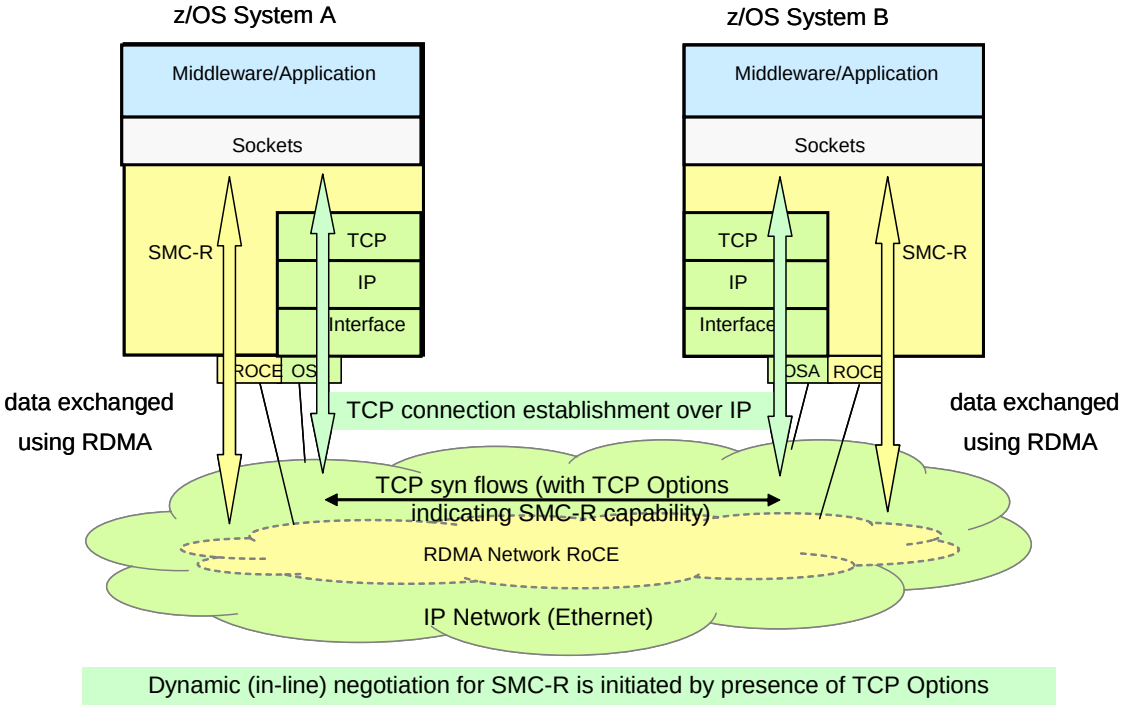

Figure 3: Rendezvous processing

The Rendezvous exchange of information occurs in three stages:

1. TCP connection establishment flows:
   TCP connections are still established using the standard three-way handshake mechanism. When SMC-R communications are enabled, the client adds TCP options settings in the SYN request to indicate that it supports SMC-R protocols. When SMC-R communications are enabled, the server also responds with TCP options settings for SMC-R in the SYN-ACK response. No additional exchange of information is required in this stage of the rendezvous processing.

2. In-band SMC-R Connection Layer Control (CLC) messages:
   After the TCP three-way handshake succeeds, the client and server negotiate the use of SMC-R for this TCP connection by using SMC-R CLC messages that flow as in-band data over the TCP connection. Conceptually, these flows are similar to the TLS/SSL handshake processing that occurs after the TCP connection is established, but they occur before any data is allowed to flow over the TCP connection (in-

cluding the TLS/SSL handshake). The CLC messages exchange the following information:

- Layer 2 addressing information (MACs and GIDs)
- RoCE credentials, consisting of
  - Remote memory buffer access information
  - Queue Pair and related information

3. SMC-R Link Layer Control (LLC) messages:
   Using the RoCE credentials exchanged in phase 2, an RDMA connection called an *SMC-R Link* is established between the two peers across *Reliable Connected Queue Pairs* (RC QPs). SMC-R LLC messages are then exchanged across the SMC-R Link to confirm that the RoCE information is correct and that the RC QPs that comprise the SMC-R link have connectivity. This stage is skipped if an existing SMC-R Link is used for this TCP connection.

Since the z/OS TCP/IP stack does not allow the client and server applications to exchange application data before or during rendezvous processing, the TCP connection can revert to IP protocols if there is a failure during the setup of the SMC-R communications. However, once the RoCE connection is confirmed by using the LLC messages, the TCP connection is committed to using SMC-R protocols and cannot fall back to using IP protocols if SMC-R communications encounter an error. The SMC-R protocol does provide the capability to setup redundant SMC-R links between two peers; in this case if a failure is encountered in the communication path for one SMC-R link, SMC-R will transparently move the connection to an alternate SMC-R link without any impact on the connection.

Even though application data is sent out of band from the TCP connection with SMC-R communications, the TCP connection remains active in order to preserve the connection state for monitoring and management functions, load balancers, etc., and to support various stack functions, including connection termination processing.

Performance results

The results below compare workloads using the SMC-R protocol with the 10Gb RoCE Express feature to workloads using the TCP/IP protocol over OSA Express4S 10Gb and/or OSA Express5S 10Gb features (hereafter referred to as standard TCP/IP). The tests were conducted on IBM zEnterprise® zEC12 machines. The results in this paper were obtained using IPv4 workloads. Testing has shown comparable results with IPv6 workloads.

The results are divided into two main sections:

Part 1 – Sockets Micro benchmarks which were obtained using the Application Workload Modeler (AWM) tool in a laboratory environment

Part 2 - Macro benchmarks measuring z/OS middleware workloads exploiting TCP/IP sockets communications in a laboratory environment

The results obtained in other configurations or operating system environments may vary significantly depending upon environments used. Therefore, no guarantee can be given that other implementations will achieve performance gains equivalent to those described herein. Users of this document should verify the applicable data for their specific environment.

These results show that with the exception of very short-lived connections, SMC-R is recommended for all TCP/IP workloads. SMC-R provides throughput and response time improvements over TCP/IP and for workloads sending large messages or data (e.g., streaming connections) use of SMC-R also results in reduced CPU consumption.

For short-lived connections the additional connection setup processing for SMC-R (rendezvous), can outweigh any savings realized. It is recommended that for servers servicing short-lived connections users code NOSMCR on the server's PORT definition. For details, please refer to the z/OS Communications Server IP Configuration Reference Guide.

SMC-R and HiperSockets

Note that all benchmarks described in this document compare SMC-R versus standard TCP/IP over standard 10GbE using OSA Express. The question of how SMC-R compares to TCP/IP over HiperSockets also comes up frequently, despite the fact that these are inherently different technologies. HiperSockets focuses on optimized communications within a System z CPC, whereas SMC-R enables optimized communications across System z CPCs. While not discussed in detail in this document, internal IBM benchmarks have shown that SMC-R begins to approach the same levels of low network latency that Hiper-

Sockets offers. However, SMC-R is more CPU efficient, especially when larger payloads are involved in the traffic patterns. In contrast, HiperSockets does offer a network bandwidth advantage, since it is not constrained by the physical network bandwidth limitations that are present in physical networks such as 10GbE. Also note that there are several other key (non-performance-related) differences in the two technologies that also should be considered such as:

1. SMC-R is restricted to TCP workloads (HiperSockets supports all protocols)

2. SMC-R is a point-to-point solution (HiperSockets provides Broadcast and multi-cast support)

3. SMC-R requires RoCE hardware (HiperSockets does not require any additional hardware). In some environments this (internal communication) aspect is considered more secure.

4. SMC-R virtualization (sharing) capability lags HiperSockets virtualization capability.

SMC-R Remote Memory Buffer (RMB)

Before we get into the results, a brief overview of the SMC-R memory architecture and the Remote Memory Buffer (RMB) is in order. As mentioned in the overview section, SMC-R peers write directly into the remote partner's memory. Each TCP connection is allocated an element within the RMB. In some cases the element can be the entire RMB. The SMC-R protocol defines the RMBE sizes. The RMBE size used for a TCP connection is based on the TCP receive buffer size.

The various RMB and RMBE sizes supported by an Operating System and the TCP connection RMBE size selection criteria are based on each Operating System's SMC-R implementation. The OS RMBE supported sizes and size selection criteria (per TCP connection) are subject to change. The sizes currently supported by z/OS are 32KB, 64KB, 128KB, 256KB and 1MB. The z/OS default RMBE size is 64KB.

The RMB and RMBE size is transparent to socket applications. Socket applications can influence the RMBE size by issuing a SETSOCKOPT with the SO_RCVBUF option before establishing the connection. Additionally, some middleware applications may provide a configuration option to set the TCP receive buffer size. The RMBE size chosen will be the first RMBE size above the SO_RCVBUF value. For example, if the application issued SETSOCKOPT

with an SO_RCVBUF value of 256KB, the RMBE size used for that connection would be 1MB.

The RMBE size used can have an impact on SMC-R performance. A systems performance analyst should periodically monitor the connection's SMC-R statistical information along with the specific RMBE sizes uses for a given connection.

Micro Benchmark results

The micro benchmark performance data presented in this paper were collected using a dedicated system (dedicated CPs and dedicated OSA adapters) environment using the Application Workload Modeler (AWM). AWM uses a very lightweight socket application (no business logic) that stresses and measures the networking infrastructure. As a result, micro benchmarks are useful in showing the maximum possible benefits of the SMC-R protocol. We tested two types of AWM workloads. Request/Response (RR) patterns that are prevalent in Online Transaction Processing (OLTP) and streaming, or bulk data transfer, workloads (STR) that are prevalent in file transfer protocols.

Request/Response workloads with SMC-R performance summary:

For request/response workloads SMC-R provides substantial throughput and response time improvements. In the lab, we have measured as high as a 732% throughput improvement, with up to an 88% reduction in response time (Figure 4). SMC-R also provides increased CPU savings for these workloads as the message size increases (we measured up to 32KB). In the lab, we have measured CPU savings as high as 53% for a request/response workload sending and receiving 32KB of data (Figure 5).

Key to the micro benchmark results:

- For the blue bar (Raw Throughput) higher is better.
- For the green and yellow (CPU) and purple (Response time) bars lower is better.
- All results are SMC-R versus standard TCP/IP over OSA

Largest throughput/response time improvement – Figure 4

In this first AWM test, we ran a request/response workload (10 TCP connections) between two z/OS systems. Each side sends and receives 2KB of data on each transaction (small message). For this test TCP/IP used OSA Express4S 10Gb configured with an MTU of 1500 and NOSEGMENTATIONOFFLoad (no Large Send). For SMC-R we used the 10Gb RoCE Express feature with a 256KB RMBE and 1KB MTU.

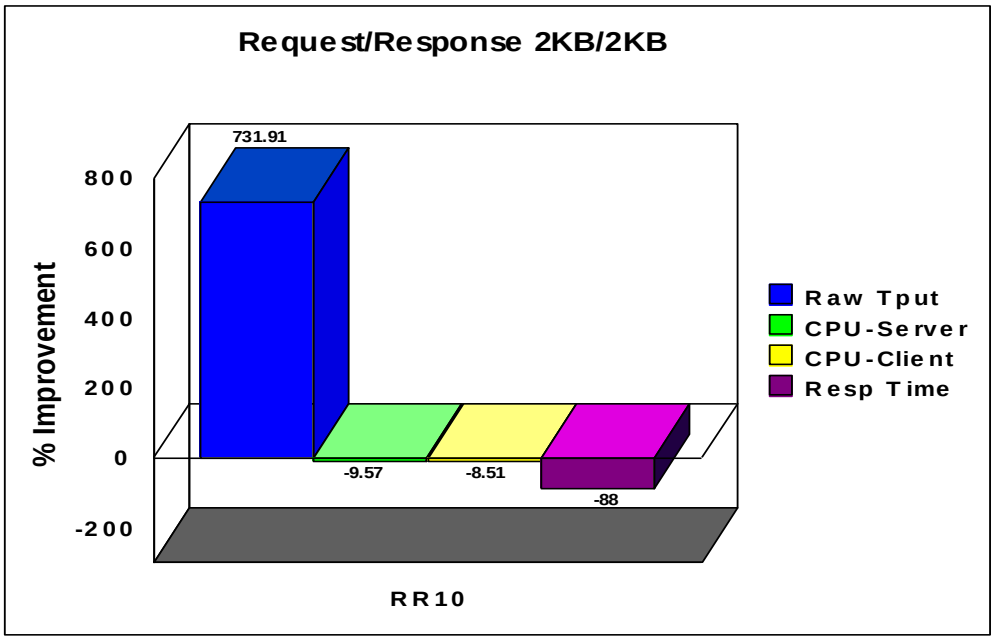

Figure 4: Request/Response AWM – RR10 2KB/2KB

In this test, we observed an increased throughput of almost 732% with a reduction in response time of 88%. Also, despite the smaller message size we still see approximately a 9% reduction in overall z/OS CPU on both the client and server with SMC-R when compared to standard TCP/IP.

Largest CPU reduction – Figure 5

In this next AWM test, we ran the same request/response workload (10 TCP connections) but used a larger message size between the two z/OS systems. Each side sends and receives 32KB of data on each transaction. Figure 5 shows SMC-R compared to standard TCP/IP over an OSA Express4 configured with an MTU of 1500 and NOSEGMENTATIONOFFLoad (no Large Send). For SMC-R we used the 10Gb RoCE Express feature with a 256KB RMBE and 1KB MTU.

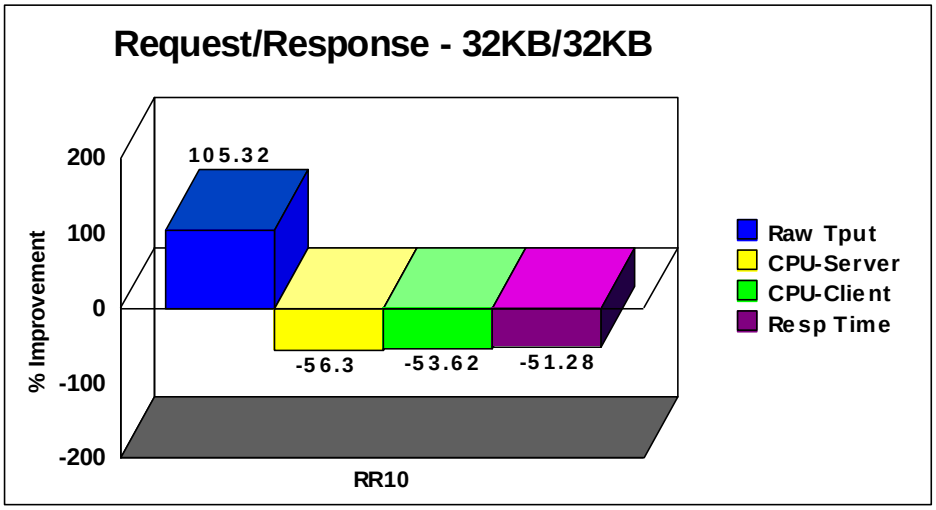

Figure 5: Request/Response AWM – RR10 32KB/32KB

In these results, we see an increased throughput of 105% with a 51% response time improvement. We are sending much larger messages and with SMC-R we begin to see saturation of the 10Gb network. This limits the throughput advantage SMC-R has over TCP/IP compared to what we saw in the smaller request/response test (Figure 4). In this test we are seeing almost 16Gb/sec of data between the client and server (each sending and receiving 32KB).

We also begin to see the real strength of SMC-R – moving large amounts of data. With the larger message sizes we see much higher overall z/OS CPU savings (reductions) - up to 56% with SMC-R when compared to standard TCP/IP.

Streaming workloads with SMC-R performance summary

Our streaming AWM workloads loop between the client sending 1 byte of data to the server and the server responding with 20MB. This tests the network performance with bulk data transfers.

In our labs, for streaming workloads, SMC-R provides substantial throughput, response time and CPU utilization improvements. In the lab, we have measured as much as a 54% throughput improvement, up to a 35% reduction in response time, and an overall z/OS CPU reduction of up to 67% on both the sender and receiver.

In our first test we compare SMC-R against standard TCP/IP using OSA Express4S 10Gb configured with a 1500 MTU and NOSEGMENTATIONOFFLoad (no Large Send in Figure 6). This is the most typical OSA configuration.

In this test, we ran an AWM streaming workload (1 TCP connection) between two z/OS systems. The client sends 1 byte of data to the server. The server responds by streaming 20 million bytes back to the client in a loop for 7 minutes. For SMC-R we used a 1KB MTU and a 1MB RMBE, the largest RMBE that can be used. A little further on we will show the advantage of using larger RMBEs for streaming workloads.

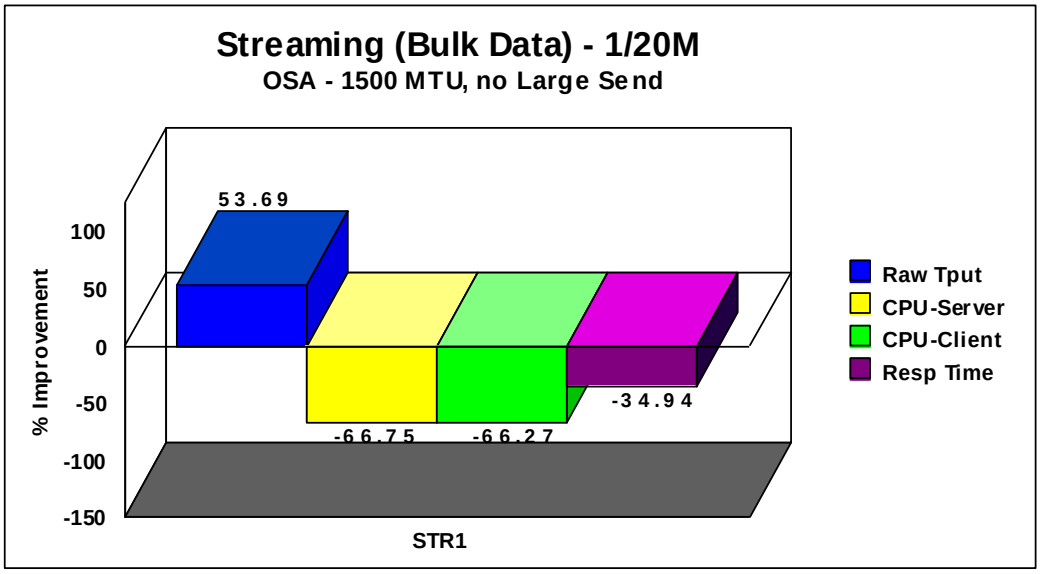

Figure 6: Streaming AWM – OSA small mtu, no Large Send

In this streaming test (Figure 6), we see an increased throughput of almost 54% with a 35% response time improvement with SMC-R versus standard TCP/IP. In fact, SMC-R is already saturating the 10Gb bandwidth network with

just this one streaming connection. We also see significant CPU improvement (reduction) moving large data over SMC-R versus standard TCP/IP. An overall z/OS CPU reduction of up to 66% on both the sender (server) and receiver (client) is observed.

Next we compared SMC-R against standard TCP/IP using OSA Express4 configured with jumbo frames (8992 MTU) and SEGMENTATIONOFFLoad (Large Send in Figure 7). SEGMENTATIONOFFLoad allows the TCP/IP stack to offload segmenting of the data to the OSA adapter. This saves CPU cycles on the sender's TCP/IP stack. For streaming (bulk data) workloads this is the most optimal configuration for OSA.

In this test, we ran an AWM streaming workload (1 TCP connection) between two z/OS systems. The client sends 1 byte of data to the server. The server responds by streaming 20 million bytes back to the client. This is done in a loop for 7 minutes. For SMC-R we are again using a 1MB RMBE and 1KB MTU.

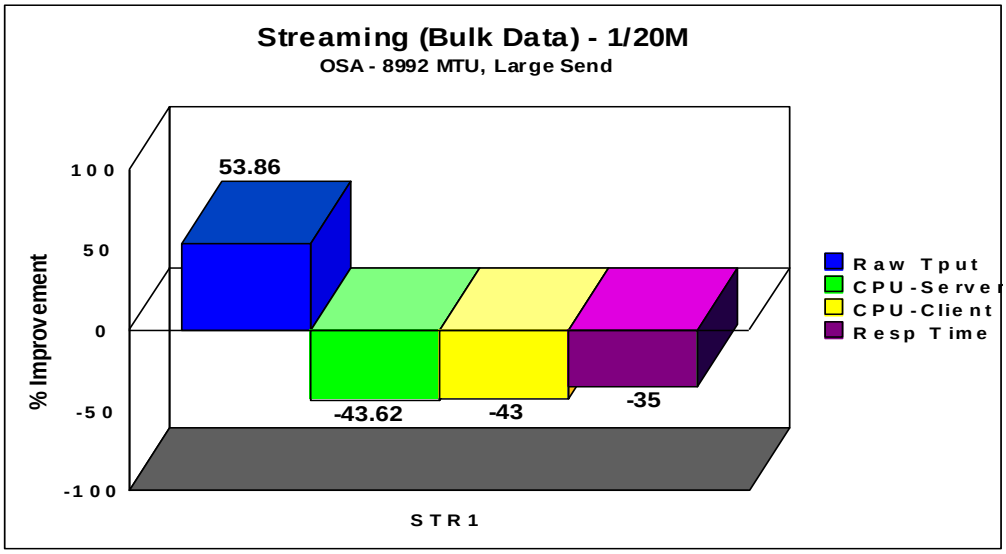

Figure 7: Streaming AWM – OSA jumbo frames, Large Send

In these results (Figure 7), we see about the same increased throughput and response time improvement with SMC-R that we see when compared against the typical OSA configuration (Figure 6). However, the advantage SMC-R had in overall z/OS CPU utilization when compared to the typical OSA (Figure 6) has narrowed some. Nevertheless, SMC-R still provides up to a 43% reduction in overall z/OS CPU utilization on both the sender and receiver.

One final note regarding streaming workloads with SMC-R: The SMC-R RMBE size that is used by the receiving side of a streaming connection can have an impact on the CPU utilization of the sending side. The larger the RMBE size used the more overall CPU savings the sending side will realize. In our lab, CPU cost per MB transferred on the sending stack is 18% lower when sending streaming data to a 1MB RMBE versus sending the data to a 256KB RMBE. As Figure 8 illustrates, a larger RMBE size means less wrapping of the RMBE and a lower frequency of space available notifications sent to the sender.

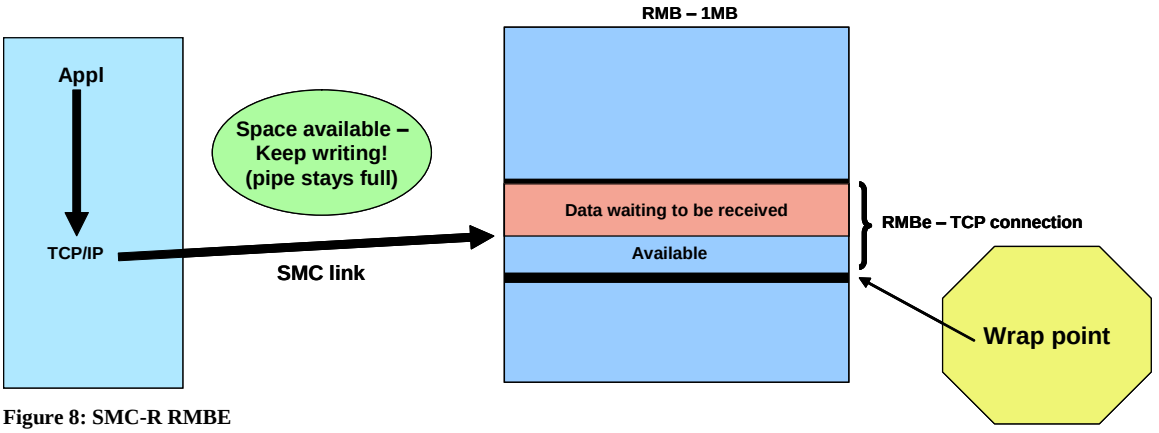

Figure 8: SMC-R RMBE

Sysplex Distributor Performance with SMC-R

Sysplex Distributor can be deployed with SMC-R with no additional configuration updates. If the client application resides on a z/OS host that is enabled for SMC-R and meets SMC-R criteria for connecting to the target z/OS system then sysplex distributed connection data can flow over SMC-R links between the client and target (server) systems.

This provides an additional performance advantage for SMC-R when compared to sysplex distributed workloads using standard TCP/IP. With standard TCP/IP, all inbound packets flow through the sysplex distributor stack before reaching the target (server) system. QDIO Accelerator helps mitigate the CPU cost to the sysplex distributor stack by pushing the routing tables to the DLC layer.

SMC-R takes this one step further and bypasses the sysplex distributor stack completely for all inbound data. Figure 9 illustrates the differences between standard TCP/IP (with QDIO Accelerator and without) and SMC-R. The dotted yellow line (line 1) highlights the flow of inbound data for TCP/IP distributed connections without QDIO Accelerator while the solid yellow line (line 2) shows the QDIO Accelerator path. Contrast that with the line 3 flow that shows SMC-R distributed connections where traffic between client and server can flow directly without traversing the Sysplex Distributor system.

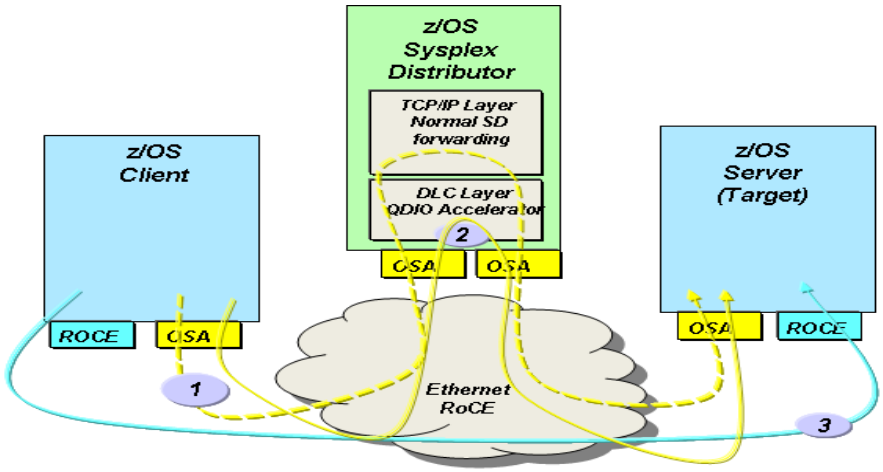

Figure 9: SMC-R with Sysplex Distributor

To demonstrate this advantage, we ran a request/response test (20 TCP connections) sending 100 bytes and receiving 800 (small messages). We compared an SMC-R sysplex distributed run against two standard TCP/IP runs. The first TCP/IP run has QDIO Accelerator enabled on the sysplex distributing stack. The second TCP/IP run does not have accelerator enabled. The results in Figure 10 show the significant advantages SMC-R provides. Almost all of the CPU on the distributing stack has been removed (as the inbound workload bypasses the distributing stack). The overall throughput (transactions per second) for SMC-R increases by up 295% when compared to standard TCP/IP without QDIO acceleration and by 248% when compared to standard TCP/IP with QDIO Acceleration.

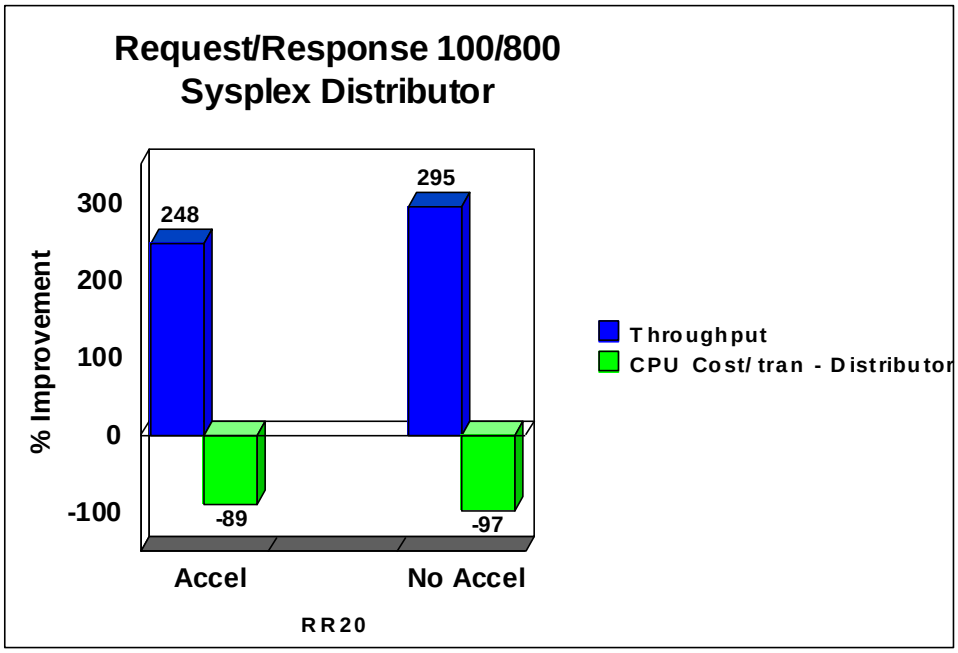

Figure 10: SMC-R with Sysplex Distributor performance

Macro Benchmark results

The macro benchmark results shown below were achieved by different labs across IBM using simulated application workloads. Like the micro benchmark tests, these results show SMC-R provides improvements in throughput and response time. CPU consumption was improved on some of the workloads depending on the communication pattern.

IBM WebSphere MQ for z/OS

This test used WebSphere MQ V7.1.0 between two zEC12 machines, each with 10 processors. On each peer (z/OS SYSA and z/OS SYSB in Figure 11) a queue manager was configured with 50 outbound sender channels and 50 inbound receiver channels with default options for the channel definitions.

A request/response workload was run where:

1. Six long running batch tasks per channel pair would connect to the requester queue manager and put 1 message out-of-syncpoint to a transmission queue. These tasks would then wait for a specific reply message on an indexed reply queue.

2. Two long running batch tasks would connect to the server queue manager and get and put the next available message in-syncpoint.

3. Once the requester batch tasks received their expected reply message, they would put another message.

4. This process is repeated until the test is ended.

The applications perform no additional business logic. The workload was increased by running applications from 1 to 50 channel pairs.

Each configuration was run with message sizes of 2KB, 32KB and 64KB where all messages were non-persistent.

Results show that for 64KB message sizes over one channel pair, WebSphere MQ can send up to three times (200% improvement) as many messages per second using SMC-R when compared to standard TCP/IP over OSA.

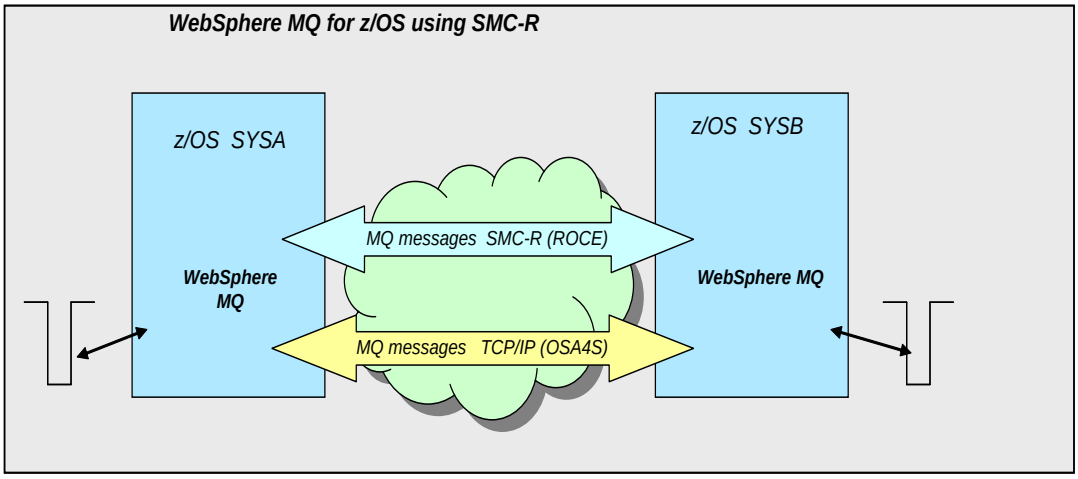

Figure 11: WebSphere MQ

The Websphere MQ results are based on internal IBM benchmarks using a modeled WebSphere MQ for z/OS workload driving non-persistent messages across z/OS systems in a request/response pattern. The benchmarks included various data sizes and number of channel pairs. The actual throughput and CPU savings users will experience may vary based on the user workload and configuration.

IBM WebSphere to DB2 communications

This test consisted of a WebSphere Application Server (WAS) application called Tradelite that simulates a stock trading application that uses z/OS DB2 as its database. The test used a workload client simulator (JIBE) running on a Linux x86 server communicating to the Tradelite application running on WAS using the Liberty profile on z/OS (z/OS SYSA in Figure 12). The JIBE client opened 40 concurrent HTTP TCP connections to WAS Liberty and these connections were always used standard TCP/IP. The WAS Liberty server was configured with 85 persistent connections to the z/OS DB2 on system SYSB. For each client request arriving over the 40 TCP/IP connections, the Tradelite application issued, on average, 3 JDBC/DRDA requests to the DB2 server (z/OS SYSB in Figure 12). The connections between WAS Liberty and DB2 are eligible for SMC-R. Tests were performed using both standard TCP/IP and SMC-R for these backend connections. The data exchanged between WAS Liberty and DB2 consist of small (approximately 100 bytes) data sizes.

Results show a 40% improvement (reduction) in overall transaction response time is observed from the JIBE client's perspective when SMC-R is used for the communications between WAS Liberty and DB2 as compared to standard TCP/IP. These results are quite significant considering the fact only the backend communications (WAS to DB2) exploited SMC-R.

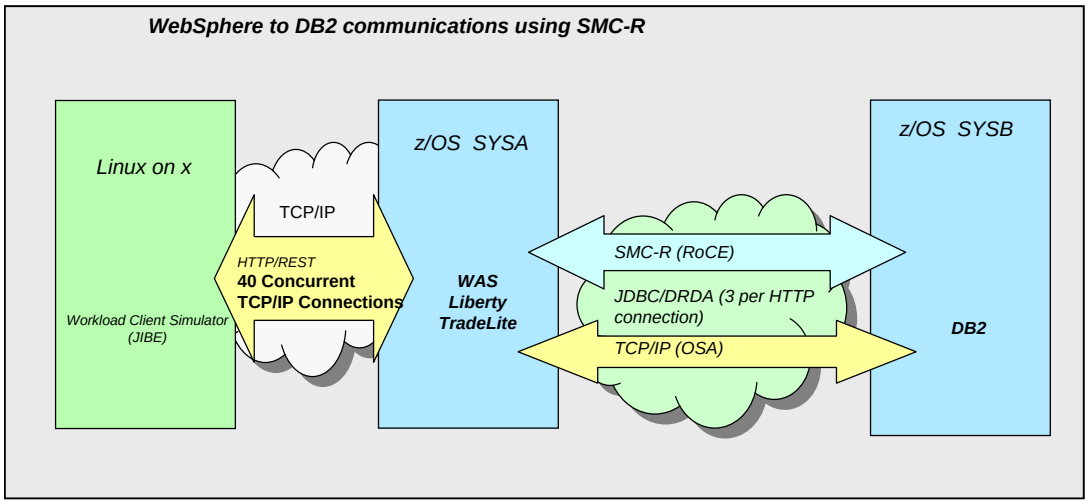

Figure 12: WebSphere Liberty to DB2

The Websphere-to-DB2 results are based on projections and measurements completed in a controlled environment. Results may vary by customer based on individual workload, configuration and software levels.

IBM CICS performance improvement

This test used TPNS (Teleprocessing Network Simulator) to drive connections to a front-end CICS (CICS A in Figure 13) on z/OS. For each of these connections the front-end CICS invokes a transaction that makes 5 Distributed Program Link (DPL) calls to a program that executes in the back-end CICS (CICS B in Figure 13) on another z/OS. These DPL calls are made over an IPIC (IP Interconnectivity) which are eligible for SMC-R. Messages of 32KB are passed as input on the DPL calls and the mirror transaction/program returns the same amount of data as output.

Results show up to a 48% improvement (reduction) in the CICS transaction response time, from the front-end CICS perspective, with SMC-R when compared to standard TCP/IP.

Additionally, a 10% reduction in overall z/OS CPU utilization is realized on the CICS systems. The CPU reduction is a result of the large data sizes exchanged between the two CICS systems.

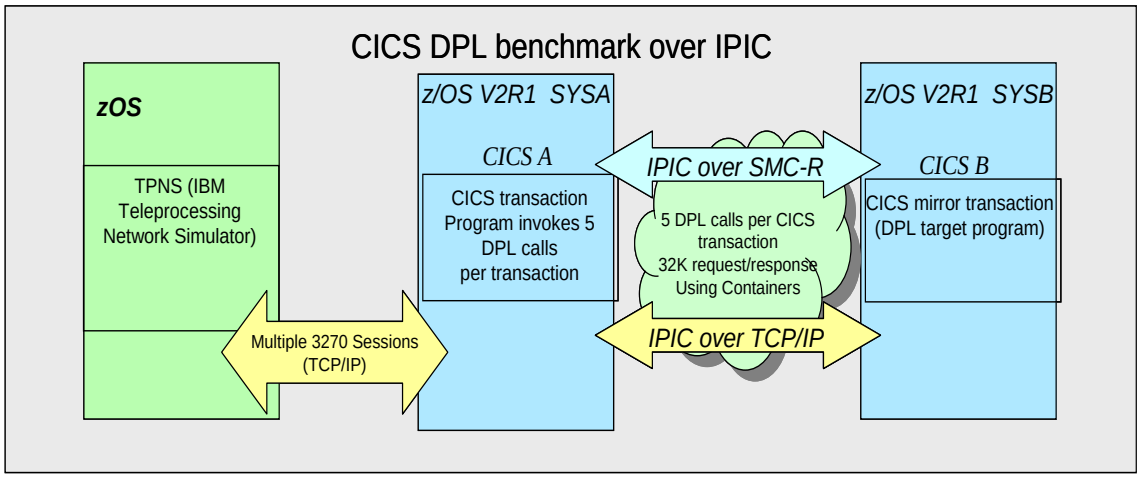

Figure 13: CICS

The CICS results are based on internal IBM benchmarks using a modeled CICS workload driving a CICS transaction that performs 5 DPL calls to a CICS region on a remote z/OS system, using 32K input/output containers. Response times and CPU savings measured on z/OS system initiating the DPL calls. The actual response time and CPU savings will vary based on users' workload and configuration.

DB2 server performance

This test used a DB2 DDF client (z/OS SYSA in Figure 14) to fetch large amounts of data (500,000 rows of 32KB each – about 15GB overall) from a DB2 server (z/OS SYSB in Figure 14). The test compared this data flowing over standard TCP/IP (yellow flow in Figure 14) against this data flowing over SMC-R (green/blue flow in Figure 14).

Results show up to a 37% improvement (reduction) in response time observed with SMC-R when compared to standard TCP/IP.

Additionally, a 45% reduction in overall z/OS CPU utilization is realized across both systems. The CPU reduction is a result of the large data transferred between the two systems.

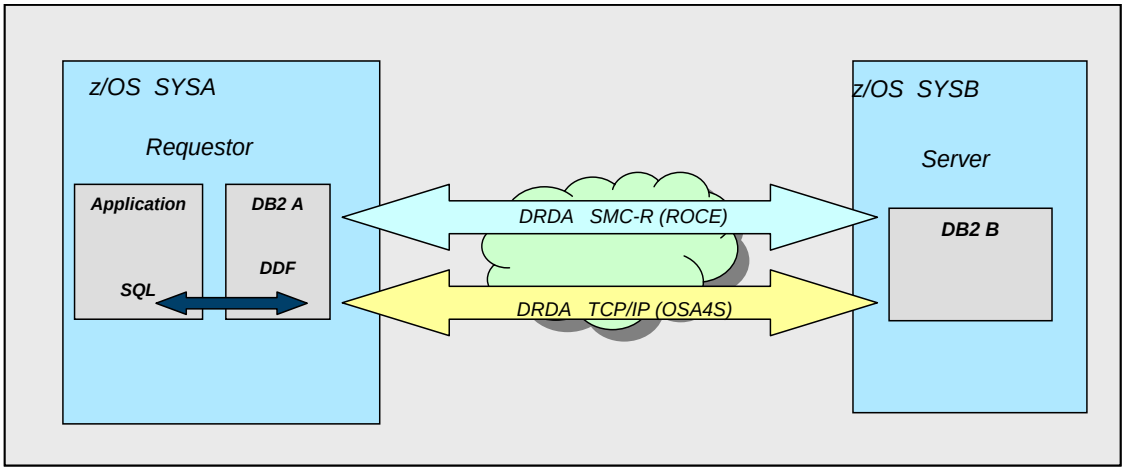

Figure 14: DB2 server

The DB2 server results are based on internal IBM benchmarks using a modeled DB2 workload executing a fetch of 500,000 rows (32KB per row) from a remote DB2 server. The actual response times and CPU savings any user will experience will vary.

DB2 IBM Relational Warehouse Workload (IRWW)

IRWW is a modeled OLTP workload that consists of seven transactions. Each transaction consists of one to many SQL statements, each performing a distinct business function in a predefined mix.

This test used 15 concurrent threads (z/OS SYSA in Figure 15), each connected to a DB2 server (z/OS SYSB in Figure 15) using JDBC/DRDA.

Results show up to a 39% improvement (reduction) in response time with up to a 66% increase in throughput observed with SMC-R when compared to standard TCP/IP.

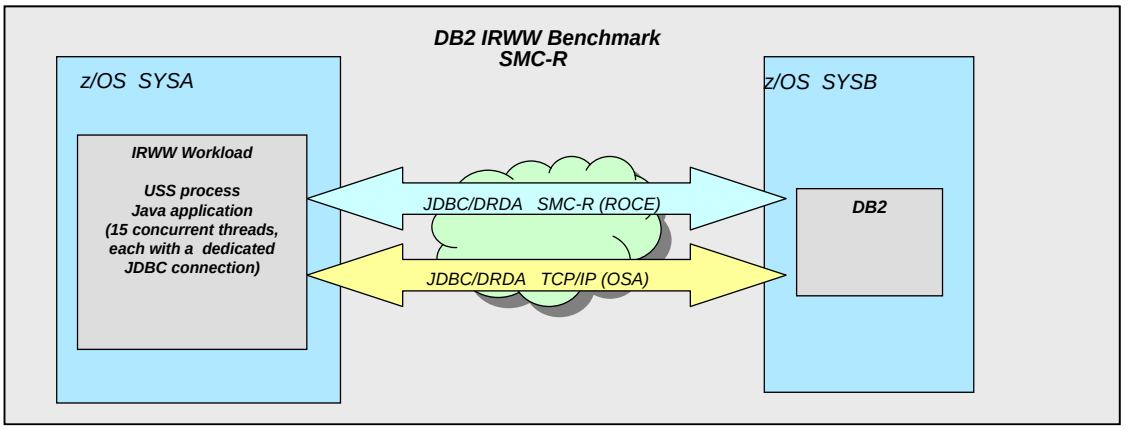

Figure 15: DB2 IRWW

The DB2 IBM Relational Warehouse Workload (IRWW) results are based on internal IBM benchmarks using a modeled DB2 IRWW workload deployed on a z/OS system accessing a remote z/OS DB2 server using JDBC/DRDA. The actual response times any user will experience will vary.

z/OS FTP performance with SMC-R

This test compares a single FTP connection using SMC-R versus standard TCP/IP. We measured both PUTs and GETs with large binary file (1200 MB) transfers. The PUT sends a 1200 MB file from the z/OS FTP client to the z/OS FTP server. A GET reverses the direction. This is performed in a loop using the same file. This test is unique in that the file I/O latency (reading and writing the files to DASD) is so much higher than network latency that throughput (MB/sec) between SMC-R and standard TCP/IP is about the same. In this case, SMC-R's primary benefit is CPU improvement (reduction).

Figure 16 shows SMC-R CPU consumption versus standard TCP/IP over an OSA Express4S 10Gb configured with an MTU of 1500 and NOSEGMENTATIONOFFLoad (no Large Send). For SMC-R we used the 10Gb RoCE Express feature with a 256KB RMBE and 1KB MTU.

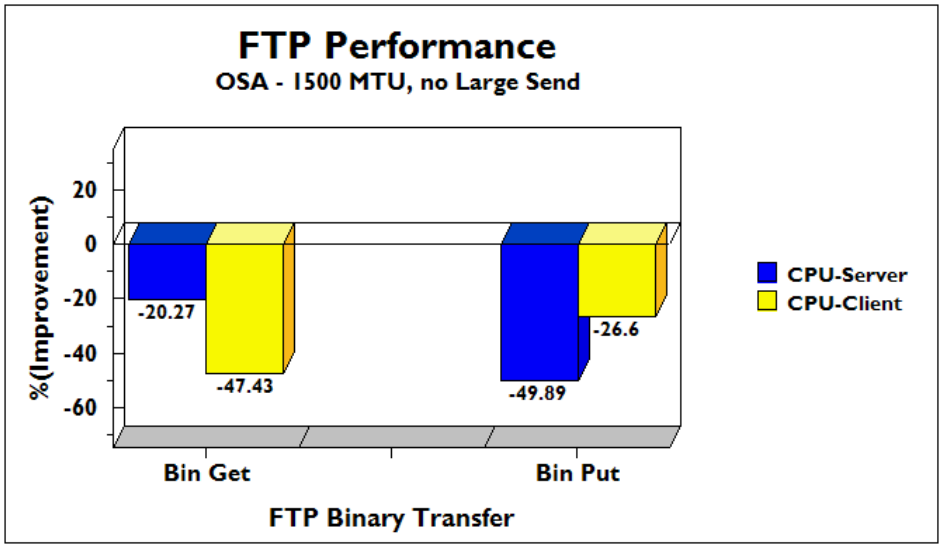

Figure 16: z/OS FTP

The results (Figure 16) show that, for FTP Puts and Gets, SMC-R realizes a CPU improvement (reduction) of up to almost 50% on the receiving side of the FTP transfer and up to 26% on the sending side versus standard TCP/IP.

The FTP performance measurements were collected using a dedicated system environment. The 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 to achieve

equivalent performance or throughput improvements. Users of this document should verify the applicable data for specific environment.

Conclusion

In this document the sample micro benchmarks demonstrate the potential savings and the macro benchmarks demonstrate the actual (sample) savings offered by the IBM System z/OS SMC-R solution combined with the RoCE Express feature. While each customer's actual savings will vary (based on environment and workload related variables) the overall efficiency offered by RDMA technology is compelling.

When this level of savings can be provided without changing existing application software, data center network (Ethernet) infrastructure, and IP topology, and while preserving the key networking quality of services demanded by enterprise networks, then IBM z/OS V2R1 customers will find that the SMC-R solution provides a highly competitive solution while also delivering on "time-to-value".

Acknowledgments and Contributions

This paper was a collaborative effort. Many thanks to the following individuals for their contributions to this paper.

- Gus Kassimis
- Jerry Stevens
- Mike Fox
- Chris Meyer
- John Burgess
- Peter Bunk
- Tony Sharkey
- Todd Munk
- Nguyen Dao

About the Authors:

Dave Herr is a Senior Software Developer in IBM Software Group's z/OS Communications Server team, focusing on performance. He has 26 years of experience as a developer and performance analyst. Dave can be reached at dherr@us.ibm.com.

Dan Patel is a Senior Software Engineer for z/OS Communications Server at IBM Research Triangle Park working on the Performance team. Dan has been with IBM for 29 years leading the performance benchmark and measurement team. Dan can be reached at danpatel@us.ibm.com

IBM Systems and Technology Group
Route 100
Somers, New York 10589
U.S.A.
Produced in the United States of America,
09/2013

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

Performance is in Internal Throughput Rate (ITR) ratio 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.

ZSW03258-USEN-00