Redbooks

ibm.com/redbooks

The Benefits of Co-location

Patty Driever

Jerry Stevens

z Systems

IBM

Point-of-View

Cost of data movements between IBM z Systems virtual servers takes a leap forward

Highlights

You can realize significant benefits, when co-locating the presentation, business logic, and data serving layers of a multitier workload onto a single physical server, using SMC-D within z/OS V2.2 and the new ISM devices on IBM z13 and IBM z13s platforms.

These are the significant benefits:

- Savings in the amount of processor resource used for network processing, freeing up that resource for other tasks
- Throughput improvements for both interactive and streaming workloads
- Reductions in latency of network communication between servers as data moves between them to complete an end-to-end transaction faster

The volume of data being generated and transmitted by cloud-based technologies (such as mobile, analytics and social computing applications) is continuously growing. This dynamic puts increasing pressure on businesses and their IT organizations. Businesses and IT organizations must provide fast access to data across the web, application, and database tiers that comprise most enterprise workloads.

For many businesses, a highly available environment and a responsive environment are directly correlated to increased revenue. This connection between performance and revenue drives a relentless focus on improving transaction response times. Often a key component of transaction response time is the network latency between the servers supporting the workload tiers and the latency between the client and servers.

The virtualization technology used with IBM® z Systems platforms allows you to efficiently, cost-effectively, and securely run multitier workloads on a single physical system. IBM z Systems™ have long been considered the gold standard for mission-critical online transaction processing (OLTP) workloads. The z Systems platforms host much of the operational data for many large banks, retailers, and other large enterprises around the world.

Using new developments in z Systems internal local area network (LAN) and device technology, along with a new sockets-based direct memory move protocol, transactional response times can be optimized for Transmission Control Protocol/Internet Protocol (TCP/IP) applications by reducing network latency and processor consumption. This new capability also eliminates the costs and complexity associated with procuring and managing physical network equipment.

This IBM Redbooks® Point-of-View publication discusses how the new Shared Memory Communications - Direct Memory Access (SMC-D) protocol, supported in the IBM z/OS® V2.2 operating system running over virtual Internal Shared Memory (ISM) devices now available on IBM z13™ platforms, can provide significant reductions in network latency between workload tiers of TCP/IP applications.

Optimized internal communications

The new SMC-D protocol running over ISM devices builds upon prior z Systems internal LAN technology, known as IBM z Systems™ HiperSockets™, and the prior “sockets-based RDMA” protocol, Shared Memory Communications over Remote Direct Memory Access (SMC-R). SMC-D provides a virtual optimized internal network communication path for TCP/IP applications. New ISM devices are represented as virtual Peripheral Component Interconnect Express (PCIe) functions, allowing for efficient data moves between logical partitions and virtual servers in a z Systems platform.

Understanding the technology components and their evolution from prior technologies (such as IBM z Systems HiperSockets and the SMC-R protocol) can facilitate a more complete understanding of the benefits realized with the current solution.

IBM z Systems HiperSockets

Co-location of multiple tiers of a workload onto a single z Systems platform allows for the exploitation of IBM z Systems internal communications. IBM z Systems HiperSockets is an internal LAN technology that provides low-latency connectivity between virtual machines within a physical z Systems central processing complex (CPC). HiperSockets is logically represented to the host as a standard network interface controller (NIC) with access to an Institute of Electrical and Electronics Engineers (IEEE) standard LAN. It is implemented fully within z Systems firmware, hence it requires no physical cabling or external network connection to purchase, maintain, or replace. The lack of external hardware components also provides for a very secure and low latency network connection, as data transfer occurs much like a cross-address-space memory move.

The performance benefits of HiperSockets when compared to using an external LAN connection over Open Systems Adapter-Express (OSA-Express) are achieved in large part because the physical layer (Ethernet) processing is avoided. Processing related to moving bytes across an adapter-to-memory bus, external port/wire, switch, and distance latencies are non-existent.

However, for the plethora of applications that use TCP/IP for network communications, the communications and packet-processing aspects of the software stack still remain. Data is formed into, then transmitted and received in fully compliant TCP/IP packets. Traditional IP routing is also used. HiperSockets does allow for a significantly larger frame size (up to 64 KB) to reduce the amount of packet processing required for large transmissions. The maximum transmission unit (MTU) size for HiperSockets is 56 KB compared to the maximum Ethernet jumbo frame size of 8192 bytes. All standards relevant to TCP/IP processing, such as checksum, segmentation, flow control, reordering, and retransmission are used when communicating over HiperSockets. This processing occurs at the appropriate operating system communication stack software layers. A significant amount of overhead is also incurred with the associated interrupt processing, memory management, context switching, and thread dispatching.

Shared Memory Communications over RDMA (SMC-R)

The IBM zEnterprise® EC12 system, IBM zEnterprise BC12 system, and IBM z/OS V2.1 introduced a new optimized communications protocol, SMC-R, which worked in conjunction with the IBM 10GbE RoCE Express feature. SMC-R maintains the TCP/IP socket application programming interface (API) for applications while significantly reducing the central processing unit (CPU) overhead of the networking software stack. Application data is transferred using direct memory-to-memory communication. To achieve this, SMC-R uses TCP to establish, manage, and terminate connections along with connection monitoring functions, such as traditional keep-alive processing. The TCP connection remains active while dynamically transitioning to exchange data out-of-band using Remote Direct Memory Access (RDMA), in a manner transparent to applications. It also does not require any IP topology changes or additional definitions, because the same IP interface can be used to support standard TCP/IP and SMC-R. This model preserves critical operational and network management features of TCP/IP, while providing significant latency, throughput improvements, and CPU savings. SMC-R was introduced to become an open, pervasive, industry protocol, and has been published as an informational Request for Comments (RFC)

(specifically RFC 7609) by the Internet Engineering Task Force. Although SMC-R and RoCE Express can also be used for logical partitioning (LPAR) communication (LPAR-to-LPAR), the solution was aimed primarily at optimizing communication between distinct physical servers. SMC-R provides “HiperSockets like” performance between servers.

Although HiperSockets has been widely accepted and deployed because of the latency and throughput benefits realized in comparison to communications over OSA, the chief inhibitor to further deployment and the greatest criticism has been the CPU overhead. Data movement, stack packet processing, and driver processing related to fully implementing the internal queued direct input/output (iQDIO) architecture are all incurred under the host’s processors. Technologies, such as RDMA, and protocols, such as SMC-R, have basically closed the gap, virtually eliminating the advantages in transaction and throughput rates once seen when using HiperSockets on the same CPC as compared to communicating off-platform between physical servers. The reduced CPU overhead of the networking software stack that is provided by the SMC-R protocol brings additional pressure on the CPU consumption rate of HiperSockets. The well-known problem related to the demise of Moore's Law also means that future improvements in internal network communications technology, such as HiperSockets must come from something other than the traditional incremental instruction machine-cycle-time-improvements in next processor generation technology.

Shared Memory Communications over Direct Memory Access (SMC-D)

If SMC-R, using RDMA, can vastly improve the transaction throughput and CPU consumption rates for unchanged TCP/IP applications for cross-CPC workloads, then it makes logical sense that a similar protocol based on Direct Memory Access (DMA) can provide comparable benefits for inter-processor (LPAR to LPAR) communications. The approach of using the SMC protocol (bypassing TCP/IP for data movement) without requiring the cost and complexity of additional hardware would solidify the advantages of co-locating multiple tiers of a workload onto a single z Systems platform. Therefore, the IBM z13 platform now brings to the forefront new ISM virtual PCIe (vPCIe) devices. ISM architecture enables optimized cross-LPAR TCP communications utilizing a new sockets-based DMA protocol named Shared Memory Communication - Direct Memory Access (SMC-D). SMC-D maintains the socket-API transparency aspect of SMC-R so that applications using TCP/IP communications can benefit immediately without requiring any application software or IP topology changes. SMC-D tightly couples the client and server sockets, eliminating the traditional stack processing. It provides for zero-copy outbound, moving data directly from user space to the target virtual server’s memory.

SMC-D completes the overall Shared Memory Communications solution, providing synergy with SMC-R. Both protocols use shared memory architecture, eliminating TCP/IP processing in the data path, yet preserving TCP/IP quality of service for connection management, load balancing, and security purposes. SMC-R and SMC-D can be exploited concurrently or independent of each other. The SMC connection protocol can dynamically determine the most optimal option.

Internal IBM benchmark testing was performed to demonstrate the throughput, latency (response time), and CPU consumption advantages that utilizing the SMC-D protocol across ISM devices has when compared to using the traditional TCP/IP protocol across HiperSockets.[1] Both interactive (request/response) workloads with small to medium message sizes were tested with varying numbers of concurrent TCP connections or threads active, and in each case a significant improvement was seen in the throughput and latency (response time) of the transaction. Substantial improvements were also seen in the amount of CPU consumed per transaction bidirectionally (on both the client and server sides). Summarizing these results at a very high level across the board, when using SMC-D, the throughput doubled while latency and CPU consumption were cut in half as compared to using traditional TCP/IP over HiperSockets. As the number of bytes transferred grew, even greater results were achieved.

[1] *SMC-D Overview and Performance V1*, ftp://public.dhe.ibm.com/software/os/systemz/pdf/SMC-D_Overview_and_Performance_V1.pdf

Figure 1 shows a performance comparison of workloads when using SMC-D verses using HiperSockets.

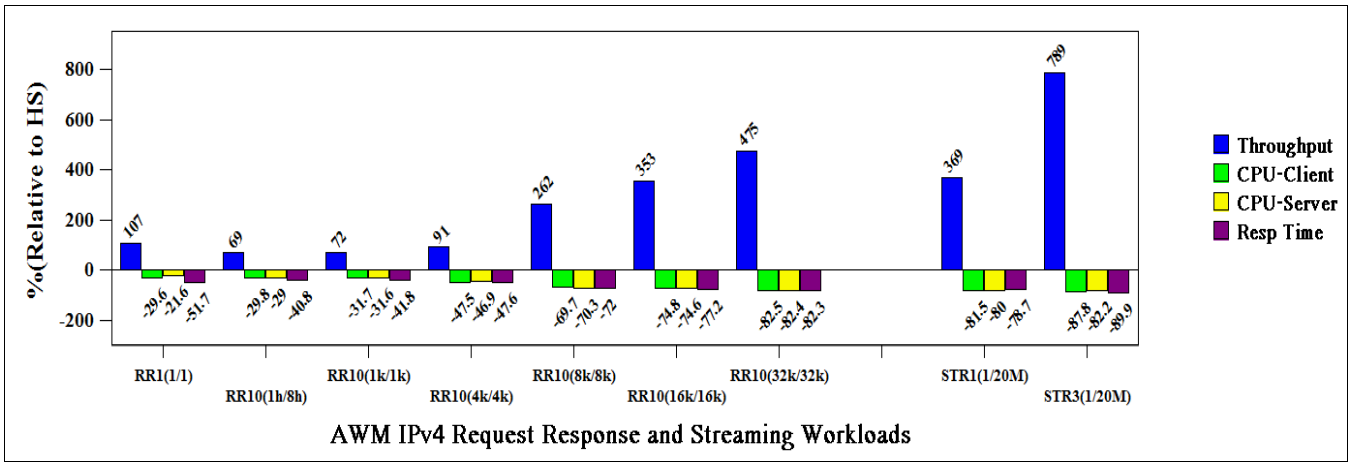

Figure 1 Performance comparison of workloads: SMC-D versus HiperSockets

The configuration of the environment used in the comparison (Figure 1) was a z13 platform with four CPs (all CPs in the same drawer) running z/OS V2R2.

Figure 1 summarizes the IBM internal (micro) benchmarks using standard tools and data points. Each transactional data point is denoted with `RRN(size)` where `RR` indicates a request/response pattern, `N` indicates the number of concurrent connections and threads and the payload size is indicated within the parenthesis. Notice that when moving left to right as the payload size grows, the benefits increase, demonstrating the scalability of the technology.

Although these benchmark results are simply measurements of the ultimate capability of the new technology, real application workloads might also realize these benefits in large measure. A prime and common example of a customer workload that exercises these interactive traffic patterns include workloads that are composed of an IBM WebSphere® Application Server interacting with an IBM DB2® database server on the back end.

In Figure 1 the last two data points (on the right) are related to streaming workloads. This type of benchmark testing was also performed on workloads with streaming data patterns (bulk data being transferred). In the streaming tests, up to 88% CPU savings was seen, and up to an almost 9 times increase in network throughput. Of particular note is that these significant CPU savings on streaming workloads were seen on both the client and sender sides; again, the benefits were bidirectional. The sender can move data more efficiently but the receiver can also receive the data now in a big “chunk,” getting interrupted less often because it is no longer receiving individual packets. These streaming workloads are reflective of the type of communication pattern deployed when transferring computer files between client and server hosts over a TCP/IP network using the standard network File Transfer Protocol (FTP). Internal benchmark testing using FTP between two z/OS instances running in two LPARs on the same CPC over ISM devices using the SMC-D communication protocol yielded a 60% CPU savings (network processing cost) on both the client and the server when compared to using HiperSockets.

As indicated previously, the SMC-R protocol over the IBM RoCE Express feature can be used for TCP/IP communications across hosts on the same server, although its primary intent was for cross-server communications. Use of the SMC-D protocol over ISM devices was designed for optimized on platform communications, and internal performance benchmarks completed demonstrate the scope of the advantages of deploying SMC-D for such an environment.

As is shown in Figure 2, SMC-D over ISM devices can also provide significant improvements in all three key measurements of throughput, latency, and CPU cost over the same workloads using the SMC-R protocol over RoCE Express.

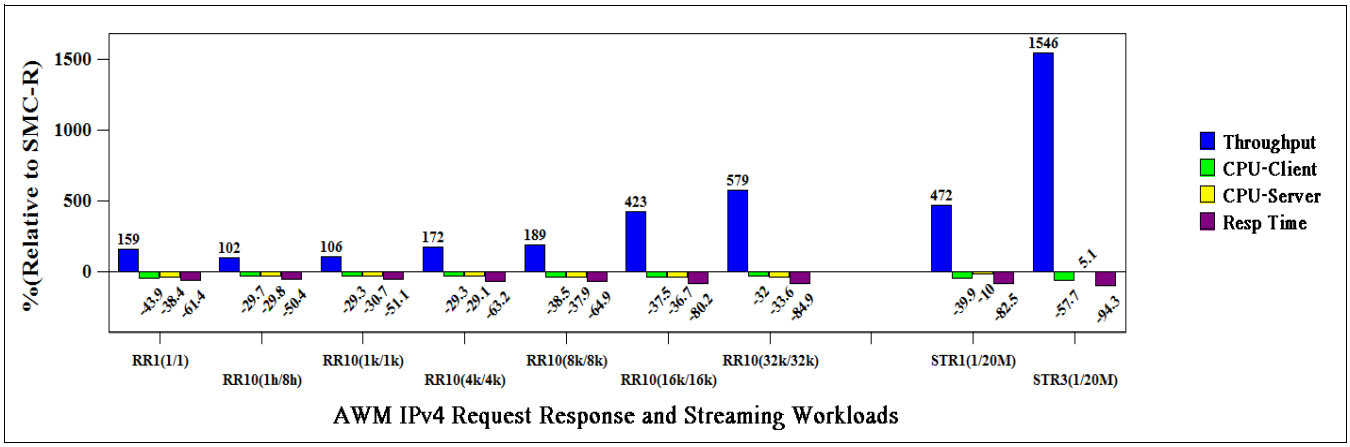

Figure 2 Performance comparison of workloads: SMC-D verses SMC-R

The configuration of the environment used in the comparison (Figure 2) was a z13 platform with four CPs (all CPs in the same drawer) running z/OS V2R2.

With the introduction of SMC-D and ISM architecture on the IBM z13 platforms, the case for co-location of workloads onto a single z Systems platform has another feature with the introduction of ISM devices and the SMC-D protocol. The speed, agility, and cost of moving data between virtual servers is vastly improved. Application workloads that currently exploit HiperSockets technology will find significant advantages in deploying the new ISM devices. Businesses that previously elected not to exploit HiperSockets technology, because of either CPU consumption concerns or the complexity associated with deploying and managing multi-homed IP network environments, should re-evaluate their situation. With the introduction of SMC-D and ISM on the IBM z13 platforms you have an opportunity to reduce your cost and increase throughput for your mission-critical workloads.

If you are not sure you have application workloads that might benefit from SMC-R or SMC-D, z/OS also provides a the IBM z/OS SMC-R Applicability Tool (SMC-AT). With SMC-AT, you can evaluate your existing TCP/IP workloads for SMC applicability. For more information about SMC-AT, see the *IBM z/OS SMC-R Applicability Tool* presentation:

`ftp://public.dhe.ibm.com/software/os/systemz/pdf/SMC_Applicability_Tool_Overview_3-03-15.pdf`

What's next: How IBM can help

To understand in more depth this or other networking solutions offered by IBM z Systems platforms, contact your local IBM representative. IBM has a broad range of expertise that can help you design the optimal networking solution for your existing and emerging workloads.

Resources for more information

For more information about the concepts highlighted in this document, see the following resources:

- *IBM z Systems Connectivity Handbook*, SG24-5444

  `http://www.redbooks.ibm.com/abstracts/sg245444.html`
- *IBM z13 and z13s System Technical Introduction*, SG24-8250

  `http://www.redbooks.ibm.com/abstracts/sg248250.html`
- Shared Memory Communications over RDMA Reference Information

  `http://www.ibm.com/software/network/commserver/SMCR/`

Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

Trademarks

REDP-5324-00

ISBN 0738455016

Printed in U.S.A.