Network Configuration for HANA Workloads on IBM POWER Servers

TARGET AUDIENCE:

Administrators for VIOS and Linux Operating Systems supported by SAP to run SAP HANA workloads.

OBJECTIVE:

Documentation of the basic setup and verification.

Valid for the Linux versions RHEL 7.3 and SLES 11 SP4 up to the latest versions of both released for HANA. This document covers only PowerVM based VIOS environments for 10Gb network connectivity with Largesend/Largereceive based on SEA. This document is valid for all HANA versions.

Special considerations for large OLTP loads starting with HANA2 SPS2 revision 1.

SCOPE:

Basic setup of VIOS/SLES/RHEL to utilize largesend/largereceive feature. Specific adapter, network or system tuning is beyond the scope of this document.

PREFACE:

Platform Large Send Offload (PLSO)

The Platform Large Send Offload (PLSO) becomes available on POWER-servers starting with the Server firmware 840.10. High level, the PLSO is a mechanism which watches the communication of a LPAR over the virtual Ethernet (ibmveth) and intervened under the following circumstances to optimize the throughput. One point of intervention happens if the Hypervisor detects that the communicating LPARs both are able to use largesend and largereceive but having an MTU of 1500 in the client LPAR. Then the Hypervisor tries to use jumboframes for most, but not all, parts of the communication. The throughput of such a communication may achieve around 80% of a...
communication where the LPAR would have used MTU=9000 natively. A second point where a PLSO intervention happens is when the communication partner resides in the same server. In this case the Ethernet-traffic bypasses most of the VIOs stack and Ethernet switch and is send directly to the communication partner. This bypassing can improve performance up to 200% of the throughput of a normal communication.

**Traditional Large Send (TLS)**

All LPARs on a server with firmware less than 840.10 and suitable VIOs level are running the Traditional Large Send (TLS) by default if their MTU is set to 9000. The larger ethernet packages are send without any intervention from the Hypervisor to the VIOs and from the VIOs to the Ethernet switch. To prevent being automatically switched to PLSO with a firmware update of the server it is strongly recommended to set the parameter “old_large_send” for ibmveth.

If you have a system configured successfully using TLS, especially a production system we do not recommend switching it to PLSO as this will not give additional performance.

For new installed systems we recommend using PLSO.

**PROCEDURES:**

------------------------------------------------------------------------------------------------------------------------

A. **Firmware and VIOS Requirements**

------------------------------------------------------------------------------------------------------------------------

1) **Verify POWER Systems Firmware level**

Minimum level: 840.10 or higher  
Recommended: 860.10 or higher

On SLES or RHEL execute:

```
# lscode

Version of System Firmware is FW860.10 (SC860_056) (t) FW860.10 (SC860_056)  
(p) FW860.10 (SC860_056) (b)  
Version of PFW is 11442016092881CF0681
```

If lscode is not installed on your system, it can be found in the package lsvpd “VPD/hardware inventory utilities for Linux”, or you can check the firmware version with the HMC.

2) **Verify VIOS level**

Minimum level: IOSLEVEL 2.2.5.0 or higher  
Recommended: IOSLEVEL 2.2.5.10 or higher
Log on to the console of the VIO Server over the hardware-management-console (HMC) as user padmin.

From the padmin shell execute:

```
$ ioslevel
2.2.5.10
$
```

3) **VIOS level is below 2.2.5.0 or your Firmware level is below 840.10**

If your VIOS level is below 2.2.5.0 or your Firmware level is below 840.10 you are not able to configure PLSO. It is highly recommended in this case to plan to update the infrastructure to the required leve.

As a temporary solution, associated with lower network performance and higher CPU load on the clients, you may want accomplish the traditional largesend. For TLS you will need a minimum ioslevel of 2.2.4.10 (which contains fix IV72825) or have the fix IV72825 installed separately.

If the result of the ioslevel is lower than 2.2.4.10, verify if IV72825 is installed:

```
$ print "instfix -ik IV72825" | oem_setup_env
  All filesets for IV72825 were found.
$`

*If your VIOS level is below 2.2.3.52 or equal 2.2.3.52 and the fix is not installed stop your investigations here. Do not change any settings and contact the responsible admin to upgrade the VIOS level first.*

If you met the prerequisites for TLS continue the instructions as for PLSO and be sure to execute step 5 outlining the additional steps only mandatory for TLS based configurations.

---------------------------------------------
B. VIOs largesend status
---------------------------------------------

1) **Check The "largesend" Status Of Your VIOs**

Check if jumbo_frames and largesend is already enabled:
In our example the value of 1500 for MTU (Maximum transfer unit) for the device en6 shows, that jumbo_frames are not enabled. In case it would be enabled you should see a value of 9000.

If you will only see the "lo0" lines in the output of the "lstcpip" command, you do not have any IP Address set on your SEA. In this case you can check the largesend option setting with the following two commands (#2 and #3)

2) Find out your SEA:

$ lsdev -type adapter | grep -i Shared

ent6 Available Shared Ethernet Adapter

$ 

3) Check largesend, large_receive and jumbo_frames:

$ lsdev -dev ent6 -attr largesend,large_receive,jumbo_frames

value
1
no
no

$ 

This output would indicate, that "largesend" is set but not "large_receive" and "jumbo_frames".

C. VIOS Setup

If largesend is not enabled, or you want to verify your configuration follow these instructions:

1) Verify the physical adapter

Verify the physical adapter used by SEA supports the largereceive capability. 
\textit{(10Gb adapters have this capability. However, some 1Gb adapters do not.)}
2) Change the Setup of the VIOs
   i. List Adapters/Devices of your VIO
      First find out which Ethernet devices are located on your VIO as padmin:

      $ ladev | grep ent

      ent0  Available  PCIe2 4-Port Adapter (10GbE SFP+)  (e41xxa123xxx123x)
      ent1  Available  PCIe2 4-Port Adapter (10GbE SFP+)  (e41xxa123xxx123x)
      ent2  Available  PCIe2 4-Port Adapter (1GbE RJ45)   (e41xxa123xxx223x)
      ent3  Available  PCIe2 4-Port Adapter (1GbE RJ45)   (e41xxa123xxx223x)
      ent4  Available  Virtual I/O Ethernet Adapter (1-lan)
      ent5  Available  Virtual I/O Ethernet Adapter (1-lan)
      ent6  Available  Shared Ethernet Adapter

      $ ladev -dev ent6 -attr | grep adapter

      adapter_reset no    Reset real adapter on HA takeover      True
      ctl_chan            ent5 Control Channel adapter for SEA failover True
      pvid_adapter        ent4 Default virt adapter to use for non-VLAN-tag(...) True
      real_adapter        ent0 Physical adapter associated with the SEA True
      virt_adapters       ent4 List of virt adap associated with the SEA (...) True

      $ rmdev -dev en6 -ucfg

      en6 Defined

      $
vi. Unconfigure the Physical Adapter:

```bash
$ rmdev -dev ent6 -ucfg
ent6 Defined
$
```

vii. Change the Devices:
- Change the SEA:

```bash
$ chdev -dev ent6 -attr jumbo_frames=yes largesend=1 large_receive=yes
ent6 changed
$
```

- Change the Physical Adapter:

```bash
$ chdev -dev ent0 -attr large_receive=yes large_send=yes jumbo_size=9014 jumbo_frames=yes
ent0 changed
$
```

viii. Optional:
As further optimization you may want to resize buffers of the trunk adapter VEA (the "virt_adapter" from "lsdev -dev ent6 -attr", ent4 in our example) to have a better fitting with jumbo_frames.

- Unconfigure and Change the VEA:

```bash
$ rmdev -dev ent4 -ucfg
ent4 Defined
$

$ chdev -dev ent4 -attr min_buf_tiny=2048 max_buf_tiny=4096 min_buf_small=2048 max_buf_small=4096 min_buf_medium=512 max_buf_medium=1024 min_buf_large=96 max_buf_large=256 min_buf_huge=96 max_buf_huge=128
ent4 changed
$
```

3) Rescan the configuration to make to devices available again without reboot and restore the routes for the Ethernet interface:

IBM Systems Solution for SAP HANA on POWER and IBM System Storage © Copyright IBM Corporation, 2017 P. 6
4) Verify Your Changes On The VIO

i. Check success of your changes with:

```
$ lsdev -dev en6 -attr jumbo_frames, large_receive, large_send

Value
yes
yes
yes
$
```

ii. Check the Physical Adapter for its values:

```
$ lsdev -dev en6 -attr jumbo_frames, large_receive, large_send

Value
yes
yes
$
```

iii. Check the SEA for its values:

```
$ lsdev -dev en6 -attr jumbo_frames, large_receive, large_send

Value
yes
yes
1
$
```

5) Redo the changes on the second VIO

D. SLES and RHEL Requirements

IBM Systems Solution for SAP HANA on POWER and IBM System Storage © Copyright IBM Corporation, 2017
1) SLES 11 SP4 or RHEL 7.3 up to latest versions released for HANA (both with latest maintenance/security updates).

i. First check the version of your virtual Ethernet driver:

```
# uname -roi
3.10.0-514.6.2.el7.ppc64le ppc64le GNU/Linux
#
```

```
# modinfo ibmveth
filename: /lib/modules/3.0.101-71 ppc64/kernel/drivers/net/ibmveth.ko
version: 1.05 <--- You will need 1.05 or higher
license: GPL
description: IBM Power Virtual Ethernet Driver
......
......
#
```

ii. Enable TSO and MTU 9000 (Jumbo frames) for the target ibmveth adapter(s)

```
# ethtool -K eth0 tso on <--- ## The device name may be different on your system
#
# ip link set dev eth0 mtu 9000 <--- ## Devicename may differ on your system
```

iii. Make both changes persistent across reboots:

Change to network configuration directory and edit target adapter config file: ifcfg-eth# (on most systems: ifcfg-eth0)

- For SLES:

- For RHEL:

IBM Systems Solution for SAP HANA on POWER and IBM System Storage © Copyright IBM Corporation, 2017
NOTE: Using MTU 9000 is unavoidable to get above 9Gb/s throughput as verified by the SAP HWCCT (Hardware Configuration Check Tool) and HCoT. The infrastructure prerequisite for using this setting is that all components from the sender to the receiver can handle MTU 9000 settings. However, if the infrastructure does not allow for this the business SLAs can likely be achieved with a lower bandwidth. In this case apply all settings described in this document except for the MTU setting.

**MTU 1500 hosts or networks may become unreachable after setting the MTU to 9000 if the components between hosts can not cope with the MTU size!**

2) Edit /etc/sysctl.conf to set/add larger tcp send/recv spaces in the Kernel

```bash
# cd /etc/sysconfig/network
# vi <config file>

BROADCAST=''
ETHTOOL_OPTIONS=''
BOOTPROTO='static'
IPADDR='xx.xx.xx.xx/xx'
MTU=''
NAME='Virtual Ethernet card 0'
NETWORK=''
REMOTE_IPADDR=''
STARTMODE='auto'
USERCONTROL='no'
ETHTOOL_OPTIONS='--K iface tso on ' ~ iface internally replaced with devicename
MTU='9000'

- For RHEL:

# cd /etc/sysconfig/network-scripts
# vi <config file>

BROADCAST=''
ETHTOOL_OPTIONS=''
BOOTPROTO='static'
IPADDR='xx.xx.xx.xx/xx'
MTU=''
NAME='Virtual Ethernet card 0'
NETWORK=''
REMOTE_IPADDR=''
STARTMODE='auto'
USERCONTROL='no'
ETHTOOL_OPTS='--K eth0 tso on ' ~ devicename may differ in your config file
MTU='9000'
```

IBM Systems Solution for SAP HANA on POWER and IBM System Storage © Copyright IBM Corporation, 2017 P. 9
3) **Activate Your Changes for the Kernel**

Push out changes:

```
# sysctl -p
```

4) **Activate Your Changes for the Ethernet driver**

In case the OS is SLES 12, SLES 15 or RHEL use the following command to restart the service:

```
# service network restart
```

In case the OS is SLES 11 use the following command:

```
# /etc/init.d/network restart
```

5) **Do additional Changes only for TLS**

Skip this section if you want to use PLSO.

To set the “old_large_send” parameter permanent execute:

```
#echo "options ibmveth old_large_send=1" > /etc/modprobe.d/50-ibmveth.conf
```

and reboot your LPAR.

To check the “old_large_send” parameter:

```
# cat /sys/module/ibmveth/parameters/old_large_send
Y (Y: enabled N: disabled)
```

and

IBM Systems Solution for SAP HANA on POWER and IBM System Storage © Copyright IBM Corporation, 2017 P. 10
6) Interrupt Tuning for large OLTP workloads (optional)

Apply these recommendations only if you can observe the following symptoms described below on your system:

Large OLTP systems can be associated with a high amount of small network packets between the application servers and the database server. The PowerVM Virtual Ethernet adapter (ibmveth) provides only one interrupt queue which means at any given moment only one CPU can service the interrupts. In periods of high traffic, this could drive up utilization on that CPU and bottleneck the traffic flowing into the server. To overcome this, consider the use of ethernet adapters which contain multiple interrupt queues. The IRQ-Balancer of the Operating system will assign the different interrupt queues to different CPUs and avoid this bottleneck.

E. Verification and Test

NOTE: Besides using the SAP HWCCCT (Hardware Configuration Check Tool) or SAP HCoT it I highly recommended to run following verification steps.

1) Test Jumbo Packet settings

For the test you need an external system which is connected over 10 Gb, has largesend/largereceive enabled and jumbo frames set. We name it EXT_UNIX here.

i. Check if jumbo size packages are configured:
ii. Simple test to check if jumbo sized packages are transmitted:

```bash
# ip addr show eth0
## Devicename may be different on your system
```

```bash
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc pfifo_fast state UNKNOWN qlen 1000
    link/ether fa:ab:d0:xx:xx:xx brd ff:ff:ff:ff:ff:ff
    inet 10.xxx.xxx.xx/23 brd 10.xxx.xxx.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::f8ab:xxxx:xxxx:xxxx/64 scope link
       valid_lft forever preferred_lft forever
#
```

```bash
ii. Start iperf SERVER on the SLES or RHEL LPAR using the default port 5001
```

```bash
# ping -c 10 -s 8900 <ipaddr_of_EXT_UNX_system>
PING <ipaddr_of_EXT_UNX_system> (12.xxx.xxx.xx) 8900(8928) bytes of data.
8908 bytes from 12.xxx.xxx.xx: icmp_seq=1 ttl=64 time=0.235 ms
8908 bytes from 12.xxx.xxx.xx: icmp_seq=2 ttl=64 time=0.213 ms
.......]
.......]
.......]
------ 12.21.xxx.xx ping statistics ----
10 packets transmitted, 10 received, 0% packet loss, time 8997ms
rtt min/avg/max/mdev = 0.213/0.241/0.256/0.019 ms
#
```

2) Large_receive test EXT_UNX -> VIOS -> SLES or RHEL

For further tests a streaming tcp load is needed to test largesend/largereceive such as ftp or iperf. The examples below use iperf which is available most Unix and Linux systems.

NOTE: iperf -c (client mode) transmits packets.
      iperf -s (server mode) receives packets.

The following test sequence checks if packets received by the VIOS are aggregated before transfer to SLES LPAR.

i. Start iperf SERVER on the SLES or RHEL Lpar using the default port 5001

```bash
# iperf -s
```

ii. Reset SEA adapter statistics on VIOS
iii. On the EXT_UNIX system, start an iperf CLIENT with four parallel connections

```bash
# iperf -w 400m -c <ip_addr_SLES_RHEL_lpar> -P4
.....
.....
.....
```

iv. Check VIOS SEA for aggregated receive packets

```bash
$ entstat -all ent6 | grep -i "Aggregat"
```

```plaintext
Receive TCP Segment Aggregation: Enabled
Receive TCP Segment Aggregation Large Packets Created: 283270
Receive TCP Packets Aggregated into Large Packets: 583692
Receive TCP Payload Bytes Aggregated into Large Packets: 490290060
Receive TCP Segment Aggregation Average Packets Aggregated: 2
Receive TCP Segment Aggregation Maximum Packets Aggregated: 16
```

3) Large_send test SLES or RHEL -> to AIX/EXT_UNIX

This checks that largesend packets from SLES or RHEL are being resegmented for transmission by VIOS SEA/Real adapter.

i. Reset SEA adapter statistics on VIOS:

```bash
$ entstat -reset ent6
.....
.....
.....
```

ii. Start iperf server on EXT_UNIX system on default port 5001

```bash
# iperf -s
```
iii. On SLES or RHEL system start iperf client with four parallel connections:

```
# iperf -w 400m -c <ipaddr_of_EXT_UNX_system> -P4
.....
.....
.....
.....
#`
```

iv. Check VIOS SEA for TCP segmentation offloaded packets:

```
$ entstat -all ent6 | grep -i "Segmentation Offload"  ## Devicename may differ
Transmit TCP Segmentation Offload: Enabled
Transmit TCP Segmentation Offload Packets Transmitted: 202656  <---
Transmit TCP Segmentation Offload Maximum Packet Size: 62702  <---
$
```

v. Check SLES or RHEL lpar tx_large_packets count:

```
# ethtool -S eth0 | grep -i "large"  ## Devicename may differ
tx_large_packets: 316534  <---
rx_large_packets: 0  <---  Note: This counter is NOT used in TLS
fw_enabled_large_send: 0
#`
```
Disclaimer and Special Notices

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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems.

Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

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 the names and addresses used by an actual business enterprise 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.

ANY INFORMATION HEREIN IS PROVIDED “AS IS” WITHOUT WARRANTY OR INDEMNIFICATION OF ANY KIND BY IBM AND DO NOT ANY EXPRESS OR IMPLIED, WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS OR USAGE FOR PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.