Using Oracle RAC 12*c* Release 1 with RDSv3 over RoCE on IBM AIX

Dennis Massanari
IBM Systems Group
December 09, 2016

Table of contents

Abstract

Oracle Real Application Clusters (RAC) 12c Release 1 is certified to use the RDSv3 or UDP protocols with RoCE adapters on IBM AIX®. Use of RDSv3 over these adapters can allow you to achieve lower latency in Oracle RAC private communications without the need to add different communications adapters and switches to your network. This reduces the complexity of introducing RDSv3 to Oracle RAC deployments.

AIX 7.2 added support for RDSv3 over RoCE which has been certified for use with Oracle RAC 12c Release 1. Certification testing was done using the Mellanox 40GbE RoCE adapter MCX314A-BCBT which is supported for use with RDS on AIX 7.2 and with IBM POWER8™ processors, and is orderable from IBM as FC EC3A/EC4B, (LP)PCIE3 2 PORT 40GBE NIC ROCE. Certification testing also used the Mellanox MSX1710-BS2F2 Ethernet Switch which is orderable from IBM as FC 8831-NF2. In addition, the adapter can be used for TCP/UDP with AIX 6.1 and 7.1.

This white paper describes the steps necessary to install and leverage RDSv3 for Oracle RAC and AIX using RoCE for the Oracle RAC cluster interconnect.

Introduction

RDSv3 is a protocol that customers can use with Oracle RAC to optimize performance. Previously RDSv3 was only available to Oracle RAC customers using InfiniBand. The AIX 7.2 support for RDSv3 over RoCE and the certification of it for use with Oracle RAC 12*c* Release 1 provides a new option for customers with capable Ethernet infrastructure who desire the increased performance of the RDSv3 protocol. Using RDSv3 over RoCE the Oracle RAC database benefits from the low overhead, low latency, high bandwidth, and highly reliable characteristics of the RDSv3 protocol and the RoCE adapter.

Both UDP and RDSv3 protocols are supported over the cluster interconnect using RoCE adapters. The use of RDSv3 with Oracle RAC 12.1.0.2 on AIX requires the Oracle patch for bug number 12909465. Please contact Oracle support to obtain this patch. When using RDSv3 customers must utilize the Oracle RAC HAIP feature to create redundant interconnects.

The IBM FC EC3A/EC4B (Mellanox *MCX314A-BCBT*) adapter is optimized for enterprise data centers, high-performance computing, transactional databases, cloud computing, virtualization, storage, and other embedded environments. The adapter improves network performance by increasing available bandwidth to the processor and by providing enhanced performance to the applications by offloading to the adapter the movement of data between the nodes in the cluster.

Configuring RoCE Adapter in AIX

Configuration of the RoCE adapter is similar to the configuration of other Ethernet adapters, with the exception of some additional attributes related to RDSv3. The default settings are recommended, but please verify that the rdma attribute is set to “desired.”

The following example shows the AIX commands you can use to verify (lsattr) and change (chdev) the rdma attributes. Interface ent4 is used in this example.

```
# chdev -P -l ent4 -a rdma=desired
# lsattr -El ent4 -a rdma
rdma desired Request RDMA True
```

Configuring Network for RoCE

Configuration of the network for RDSv3 with RoCE requires the use of one of the following:

- Global Pause IEEE 802.3x port based Flow Control.
- Priority Flow Control (PFC) IEEE 802.1Qbb, priority based Flow Control.

Global Pause was used for this certification.

Instructions on how to configure a network switch are available in article “Network Considerations for Global Pause, PFC and QoS with Mellanox Switches and Adapters”
https://community.mellanox.com/docs/DOC-2022

AIX 6.1 TL9, AIX 7.1 TL3, AIX 7.2 and newer versions today support Global Pause. This is described below and will provide a reliable transport of the RDS traffic between the nodes. Global Pause should be enabled on all ports that are used for RDS Cluster communication.

To enable Global Pause on the RoCE adapter (ent4 in this example) set the flow_ctrl_rx and flow_ctrl_tx options to “yes” as shown below.

```
# chdev -P -l ent4 –a flow_ctrl_rx=yes

# chdev -P -l ent4 –a flow_ctrl_tx=yes

# lsattr -El ent4 | grep flow_ctrl

flow_ctrl       yes           Request flow control                    True

flow_ctrl_rx    yes           Receive pause frames                    True

flow_ctrl_tx    yes           Transmit pause frames                   True
```

Enable Global Pause in the switch as show in the following example for Eth1/2.

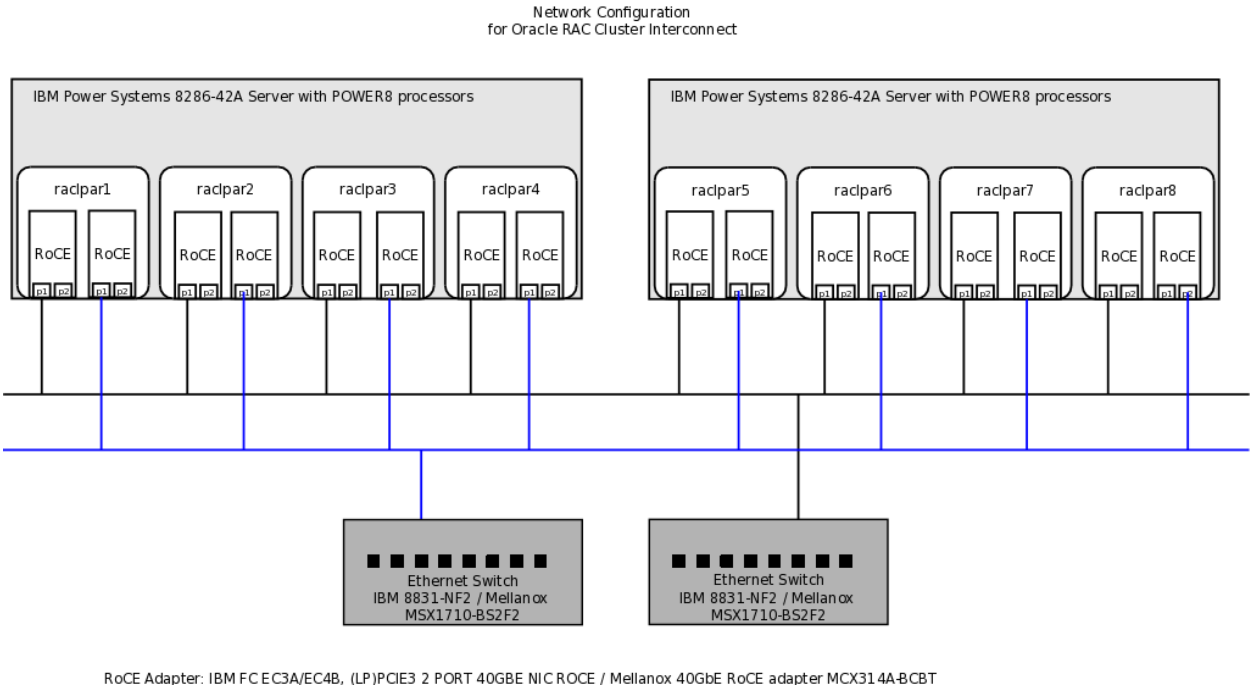
```
SX1710-Testlab-1 [standalone: master] > enable
SX1710-Testlab-1 [standalone: master] # configure terminal
SX1710-Testlab-1 [standalone: master] # interface ethernet 1/2 flowcontrol
receive on force
SX1710-Testlab-1 [standalone: master] # interface ethernet 1/2 flowcontrol send
on force
SX1710-Testlab-1 [standalone: master] # write memory
SX1710-Testlab-1 [standalone: master] # show interfaces ethernet 1/2

Eth1/2

 Admin state: Enabled
 Operational state: Down
 Description: N\A
 Mac address: f4:52:14:44:d6:ee
 MTU: 1500 bytes(Maximum packet size 1522 bytes)
 Flow-control: receive on send on
 Actual speed: 40 Gbps
```

```
Width reduction mode: Unknown
Switchport mode: access
MAC learning mode: Enabled
Last clearing of "show interface" counters : Never
60 seconds ingress rate: 0 bits/sec, 0 bytes/sec, 0 packets/sec
60 seconds egress rate: 0 bits/sec, 0 bytes/sec, 0 packets/sec
```

The following diagram shows the configuration of the Oracle RAC cluster interconnect used for the certification testing. The 40GbE cluster interconnect was configured with redundant adapters and redundant switches for high availability. If the switches used for the Oracle RAC cluster interconnect subnets are also used for other subnets then each subnet should be on its own VLAN.



Configuring RDSv3

In AIX 7.2 there are two RDS device drivers; the original RDSv2 device driver included for legacy applications only and the RDSv3 device driver which now includes code for RoCE as well as InfiniBand. The RDSv3 device driver is /usr/lib/drivers/ofed_rds and is included in the AIX fileset ofed.rds.rte. This section covers the use of the RDSv3 device driver for RoCE. The RDSv3 fileset ofed.rds.rte is on the AIX installation media.

When using the RDSv3 device driver the socket buffer size (sb_max) should be increased to 4MB. Use the AIX no command to check and set the sb_max value, as shown in the following example.

```
# no -p -o sb_max=4194304

The “-p” option preserves the setting across system reboots.

# no -o sb_max
```

```
sb_max = 4194304
```

If the sb_max value is too small you will see a “not enough buffer space” error anytime you try and use the driver, as show in the following example.

```
# rdsctrl version

socket: There is not enough buffer space for the requested socket operation.
```

The RDSv3 device driver is not loaded by default. To load the RDSv3 device driver when using RoCE adapters you use 'rdsctrl load ofed'. To automatically load the device driver for the RoCE Ethernet adapter after every boot the following command should be added to /etc/inittab.

```
rdsRoCE:2:once:/usr/sbin/rdsctrl load ofed >/var/adm/ras/rdsRoCE_init.log 2>&1
```

Note in the above commands the “ofed” option is included on the rdsctrl command. When loading the RDSv3 device driver for the RoCE adapter both the “load” and “ofed” options of rdsctrl must be used. (Do not use the “ofed” option when loading the device driver for an InfiniBand adapter).

There are parameters for tuning the performance of the RDS device driver. For most systems and workloads, the default performance settings will work well. On very large systems with heavy interconnect workloads there may be some performance improvements from tuning these settings. The effect of changing these performance settings is highly workload dependent and some settings will impact the memory consumed by the RDS driver and therefore must only be changed while carefully monitoring the performance of your system to determine the optimal performance settings. If you make any changes to these settings, you should make the same changes on all the Oracle RAC nodes.

To change the performance settings the RDSv3 device driver must be loaded. The settings go into effect only for connections opened after the settings were changed, so close all the RDSv3 connections before making the changes. An easy way to do this without rebooting is to shut down all the Oracle RAC databases using RDSv3, exit any other processes that may be using RDSv3, then unload the device driver which results in all existing connections being closed. The device driver cannot be unloaded if there are any processes with RDSv3 connections opened. Once all connections are closed you can then reload the device driver and change the settings before restarting the Oracle RAC databases. The following examples show the commands that can be used to do this.

```
###--- check the settings
# rdsctrl get
rds_send_queue_max_pages              256           Pages
rds_recv_queue_max_pages              1024          Pages
rds_reconnect_min_delay               1             Clock ticks
rds_reconnect_max_delay               100           Clock ticks
rds_max_unacked_packets               16            Packets
rds_max_unacked_bytes                 16777216      Packets
rds_max_unsignaled_ib_packets         16            Packets
rds_max_unsignaled_ib_bytes           16777216      Bytes
rds_max_ib_conns_per_if               1024          Connections
rds_recv_queue_cq_batch               16            Packets
rds_recv_queue_cq_thread_th           64            Packets
```

```
rds_send_queue_cq_batch                 16              Packets
rds_send_queue_cq_signal                8               Packets
rds_send_queue_cq_thread_th             64              Packets
###--- check if there are any RDS connections
# rdsctrl info -n|grep -c -e "--C"
32
###--- shutdown Oracle databases
###--- unload RDS
# rdsctrl unload
Unloading kernel extension /usr/lib/drivers/ofed_rds ...
Cleanup of kernel extension /usr/lib/drivers/ofed_rds done
Unloaded kernel extension /usr/lib/drivers/ofed_rds (kmid -1607090176).
###--- confirm it is unloaded
# genkex|grep -i rds
#
###--- reload the RDS device driver
# rdsctrl load ofed
Loading kernel extension /usr/lib/drivers/ofed_rds ...
Kernel extension loaded successfully (kmid -1607483392).
Initializing kernel extension /usr/lib/drivers/ofed_rds ...
Initialization of kernel extension /usr/lib/drivers/ofed_rds done
RDS Configuration Environment: OFED
###--- Use 'rdsctrl version' to verify the RDSv3 device driver is loaded.
# rdsctrl version
RDS protocol version: 3.0
###--- Verify there are no RDS connections
# rdsctrl info -n|grep -c -e "--C"
0
###--- set desired values (just using sample values)
rdsctrl set rds_send_queue_max_pages=512
rdsctrl set rds_recv_queue_max_pages=2048
rdsctrl set rds_recv_queue_cq_batch=32
rdsctrl set rds_recv_queue_cq_thread_th=64
rdsctrl set rds_send_queue_cq_batch=32
rdsctrl set rds_send_queue_cq_thread_th=256
rdsctrl set rds_send_queue_cq_signal=24
###--- You can make the changes persistent if you want using the -p option
```

```
rdsctrl set -p rds_send_queue_max_pages=512

rdsctrl set -p rds_recv_queue_max_pages=2048

rdsctrl set -p rds_recv_queue_cq_batch=32

rdsctrl set -p rds_recv_queue_cq_thread_th=64

rdsctrl set -p rds_send_queue_cq_batch=32

rdsctrl set -p rds_send_queue_cq_thread_th=256

rdsctrl set -p rds_send_queue_cq_signal=24

###--- restart the Oracle RAC databases
```

A description of each or the RDSv3 performance settings along with the range of valid values is available from the 'rdsctrl help' command. The values shown in the examples above were found to work well for one large Oracle RAC cluster configuration and set of workloads with a heavy cluster interconnect load and resulted in a slight performance benefit. As mentioned before, the effect of changing these performance settings is highly workload dependent, some settings can adversely impact the performance and will impact the memory consumed by the RDS driver and therefore must only be changed while carefully monitoring the performance of your system to determine the optimal settings. If you make any changes to these settings you should make the same changes on all the Oracle RAC nodes.

As a rule of thumb if you change rds_recv_queue_max_pages or rds_send_queue_max_pages also change the other parameter proportionally to maintain the 4 to 1 ratio.

When the RDSv3 device driver is loaded, and later on-demand, memory is allocated and kernel threads are started to assist in RDSv3 processing. The following process names are for kernel threads used to assist in processing the RDSv3 messages.

```
krds_odma_pool_wq
krds_recv_comp
krds_recv_refill
krds_send
krds_send_comp
krdsd
```

Oracle patch installation for Oracle Grid Infrastructure

When using RDSv3 on AIX use of the Oracle RAC HAIP feature is required for high availability of the cluster interconnect. In addition to high availability HAIP provides load balancing across multiple cluster interconnect interfaces. AIX does not provide a network bonding solution for use with RDSv3.

Use of RDSv3 by Oracle RAC 12*c* Release 1 on AIX requires the Oracle patch for bug 12909465. This patch was originally introduced to provide support for HAIP with InfiniBand on AIX. This patch also contains a more general change in the behavior of HAIP which is necessary when using AIX for RDSv3 to work with HAIP over any type of interface and is therefore required when using RDSv3 over RoCE adapters. Please contact Oracle support to obtain this patch.

The ReadMe for the Oracle patch for bug 12909465 gives the steps to follow for installing the patch when using InfiniBand. These same steps can be used when installing on a system using RoCE, however when

using RoCE it is also possible to apply the patch on an Oracle RAC 12.1.0.2.5 system without applying the patch first to a software only install. The patch must be installed before enabling RDSv3 in an Oracle Database using Oracle RAC even when using RoCE.

Enabling RDSv3 in an Oracle Database with Oracle RAC

An Oracle Database with Oracle RAC will use UDP over the cluster interconnect by default. If you want the database to use RDSv3 then you must enable it using the command shown in the following example. If you do not have a shared Oracle home file system, then you need to execute this command on each node. All nodes must be configured to use the same protocol (UDP or RDSv3). After shutting down all instances running from the ORACLE_HOME you can enable RDSv3 for the Oracle databases running from that ORACLE_HOME by specifying the appropriate environment and then using the ipc_rds target of ins_rdbms.mk as shown in the example below.

```
…. set ORACLE_HOME and add ORACLE_HOME/bin to PATH
$ cd ${ORACLE_HOME}/rdbms/lib
$ make -f ins_rdbms.mk ipc_rds

…. if you want to go back to UDP use
$ make -f ins_rdbms.mk ipc_g

To verify that RDSv3 is being used look for the following message in the alert
log of the database or ASM instance:

cluster interconnect IPC version:Oracle RDS/IP (generic)
```

Because of the high bandwidth of the cluster interconnect it may be beneficial to increase the number of LMS processes in your database by changing the gcs_server_processes Oracle Database parameter. The optimal number of LMS processes is highly workload dependent and therefore should only be changed while carefully monitoring the performance of your system to determine the optimal performance. Please follow Oracle Database performance tuning best practices.

Software and Firmware Versions

The following software versions were used for the certification and these versions or newer are required for support of Oracle RAC 12c Release 1 on AIX using RDSv3 and RoCE.

AIX 7.2 TL0 SP1 (7100-00-01), plus the following ifixes:

IV79639s0a: IV79639

IV79848s1a: IV79848

IV80412m1a: IV80412+IV79441

APARs:

IV82220

IV81961

NOTE: all of the above ifixes and APARs are now available in SP02 (7200-00-02) which is highly recommended.

Oracle 12.1.0.2, plus the following

PSU3

Patch for bug 12909465

NOTE: The most recent available PSU is highly recommended.

The following system firmware versions were used and these versions or higher are highly recommended.

Power 8 8246-42A:

SV830_075

IBM FC EC3A/EC3B - Mellanox 40GbE RoCE adapter MCX314A-BCBT:

000200325100

The latest Power Systems™ adapter firmware for AIX is available from the IBM Support Fix Central website (https://www.ibm.com/support/fixcentral/). Select product “Power I/O Firmware” and then specify the Feature Code (FC).

IBM MTM 8831-NF2 - Mellanox MSX1710-BS2F2 Ethernet Switch:

X86_64 3.4.3002 2015-07-30 20:13:19 x86_64

Tips

Monitoring the status of the interface and HAIP failover

In AIX 6.1 TL9, AIX 7.1 TL3 and AIX 7.2 and newer versions, AIX provides the IFF_DEVHEALTH flag (defined in /usr/include/net/if.h) to reflect the status of the physical link associated with a network interface. With the Oracle patch for Bug 12909465 this is the bit Oracle uses to promptly detect a network failure and failover the HAIP address from the failed interface to a good interface. The status of this bit is not included in the ifconfig flags output. The following sample program can be used to monitor that bit. As noted, this bit is only defined in AIX 6.1 TL9, AIX 7.1 TL3 and AIX 7.2 and newer versions of AIX.

```
$ cat check_devhealth.c
#include <stdlib.h>
#include <sys/types.h>
#include <stdio.h>
#include <sys/socket.h>
#include <sys/ioctl.h>
#include <net/if.h>
#include <errno.h>
#include <net/if_netdev.h>
#include <time.h>
#include <sys/time.h>

int main(int argc, char *argv[]) {
   struct ifreq    ifreq[32];
   char           *ifname;
   int             fd[32];
   int             i;
   int             j;
   int             numi;
   int             starti;
   int             y;
```

```
   int             interval=0;
   int             iterations=1;
   int             flagh[32], flagr[32], flagu[32];
   int             pflagh[32], pflagr[32], pflagu[32];
   int             del=0;
   time_t          bin_time;
   struct tm *t;
   struct tm space;
   char time_string[64];
   char buf[256];

   for(i=0;i<32;i++) { flagh[i]=-1;pflagh[i]=-1;flagr[i]=-1;pflagr[i]=-1;flagu[i]=-
1;pflagu[i]=-1; }

   for (starti = 1; starti < argc; starti++) {
      if (strcmp(argv[starti], "-t") == 0) {
         starti+=1;
         interval=atoi(argv[starti]);
      }
      else if (strcmp(argv[starti], "-i") == 0) {
         starti+=1;
         iterations=atoi(argv[starti]);
      }
      else if (strcmp(argv[starti], "-d") == 0) {
         del=1;
      }
      else break;
   }

   if(starti >= argc) {
      printf("usage: %s [-t <interval>] [-i <iterations>] [-d] <interface
name>\nexamples:\n   %s -t 30 -i 1 ib0\n   %s en0\n   -d only shows
changes\n",argv[0],argv[0],argv[0]);
      return -1;
   }
   j=0;
   for(i=starti;i<argc;i++) {
      ifname = argv[i];
      printf("%14s ",ifname);
      fd[j] = socket(AF_INET, SOCK_DGRAM, 0);
      if(fd[j] == -1){
         fprintf(stderr, "Could not get socket.\n");
         return -1;
      }
      memset(&ifreq[j], 0, sizeof(ifreq[j]));
      strcpy(ifreq[j].ifr_name, ifname);
      j++;
   }
   numi=j;
   printf("\n");
   for(j=0;j<numi;j++) printf("-------------- ");
   printf("\n");
   for(j=0;j<numi;j++) printf("DEVH RUNN   UP ");
   printf("\n");
   for(j=0;j<numi;j++) printf("---- ---- ---- ");
   printf("\n");

   for(y=0;y<iterations;y++) {
      strcpy(buf,"");
      time(&bin_time);
      t = (struct tm *)localtime_r(&bin_time, &space);
```

```
sprintf(time_string,"%04d-%02d-%02d %02d:%02d:%02d",t->tm_year+1900,t->tm_mon+1,t->tm_
mday,t->tm_hour,t->tm_min,t->tm_sec);
      for(j=0;j<numi;j++) {
         if (ioctl(fd[j], SIOCGIFFLAGS, &ifreq[j]) < 0) {
             perror("SIOCGIFFLAGS failed");
             close(fd[j]);
             return -1;
         }
         if(ifreq[j].ifr_flags & IFF_DEVHEALTH) {flagh[j]=1;} else {flagh[j]=0;}
         if(ifreq[j].ifr_flags & IFF_RUNNING) {flagr[j]=1;} else {flagr[j]=0;}
         if(ifreq[j].ifr_flags & IFF_UP) {flagu[j]=1;} else {flagu[j]=0;}
         sprintf(buf,"%s%4d %4d %4d ",buf,flagh[j],flagr[j],flagu[j]);
      }
      if(del==0) {
         sprintf(buf,"%s %s",buf,time_string);
         printf("%s\n",buf);
      }
      else {
         for(j=0;j<numi;j++) {
            if(pflagh[j]!=flagh[j] || pflagr[j]!=flagr[j] || pflagu[j]!=flagu[j] ||
y==iterations-1) {
               sprintf(buf,"%s %s",buf,time_string);
               printf("%s\n",buf);
               break;
            }
         }
         for(j=0;j<numi;j++)
{pflagh[j]=flagh[j];pflagr[j]=flagr[j];pflagu[j]=flagu[j];}
      }
      fflush(stdout);
      if(y<iterations-1) sleep(interval);
   }
   return 0;
}
```

Output showing the status (IFF_DEVHEALTH, IFF_UP, NDD_RUNNING) of en4 and en7 across the cluster at a point in time:

```
oracle@rac221 $ dsh /userdata/dennis/bin/check_devhealth en4 en7
rac221.ppc:            en4            en7
rac221.ppc: -------------- --------------
rac221.ppc: DEVH RUNN   UP DEVH RUNN   UP
rac221.ppc: ---- ---- ---- ---- ---- ----
rac221.ppc:    1    1    1    1    1    1  2016-07-08 08:42:04
rac222.ppc:            en4            en7
rac222.ppc: -------------- --------------
rac222.ppc: DEVH RUNN   UP DEVH RUNN   UP
rac222.ppc: ---- ---- ---- ---- ---- ----
rac222.ppc:    1    1    1    1    1    1  2016-07-08 08:42:04
rac224.ppc:            en4            en7
rac224.ppc: -------------- --------------
rac224.ppc: DEVH RUNN   UP DEVH RUNN   UP
rac224.ppc: ---- ---- ---- ---- ---- ----
rac224.ppc:    1    1    1    1    1    1  2016-07-08 08:42:04
rac223.ppc:            en4            en7
rac223.ppc: -------------- --------------
rac223.ppc: DEVH RUNN   UP DEVH RUNN   UP
rac223.ppc: ---- ---- ---- ---- ---- ----
rac223.ppc:    1    1    1    1    1    1  2016-07-08 08:42:04
rac226.ppc:            en4            en7
rac226.ppc: -------------- --------------
```

```
rac226.ppc: DEVH RUNN   UP DEVH RUNN   UP
rac226.ppc: ---- ---- ---- ---- ---- ----
rac226.ppc:    1    1    1    1    1    1  2016-07-08 08:42:04
rac227.ppc:            en4            en7
rac227.ppc: -------------- --------------
rac227.ppc: DEVH RUNN   UP DEVH RUNN   UP
rac227.ppc: ---- ---- ---- ---- ---- ----
rac227.ppc:    1    1    1    1    1    1  2016-07-08 08:42:04
rac225.ppc:            en4            en7
rac225.ppc: -------------- --------------
rac225.ppc: DEVH RUNN   UP DEVH RUNN   UP
rac225.ppc: ---- ---- ---- ---- ---- ----
rac225.ppc:    1    1    1    1    1    1  2016-07-08 08:42:04
rac228.ppc:            en4            en7
rac228.ppc: -------------- --------------
rac228.ppc: DEVH RUNN   UP DEVH RUNN   UP
rac228.ppc: ---- ---- ---- ---- ---- ----
rac228.ppc:    1    1    1    1    1    1  2016-07-08 08:42:04
```

Output showing the status en1, en4 and en7 on one node during an interface failure test. This example is using the “-d” option to show only the changes in status:

```
oracle@rac221 $ check_devhealth -t 1 -i 1800 -d en1 en4 en7
 en1  en4  en7
---- ---- ----
   1    1    1  2016-03-18 21:32:29
   1    0    1  2016-03-18 21:39:28
   1    0    0  2016-03-18 21:39:30
   1    1    0  2016-03-18 21:44:43
   1    1    1  2016-03-18 21:44:45
   1    1    1  2016-03-18 22:02:29
```

Monitoring the RoCE adapter

The entstat.mlxent command provides additional data on the Mellanox RoCE adapter not provided by the generic AIX entstat command. The enstat.mlxent command provides detailed device statistics including packet and byte counts and errors on each queue.

To check Flow Control and Global Pause, in addition to using lsattr to confirm flow_ctrl_tx and flow_ctrl_rx are set to yes as shown above, you can use the entstat.mlxent command to monitor if Global Pause is being engaged. If the network becomes congested, you will see nonzero counts in the PAUSE counters under the Hardware Stats section as shown in the following example:

```
# entstat.mlxent ent4|egrep "PAUSE|VLAN Priority|Size/Type|Hardware Stats:"
Hardware Stats:
  Frame Size/Type     Transmit Statistics   Receive Statistics
  VLAN Priority: 0
    PAUSE frames                      68              12578
    PAUSE duration                298206                578
    PAUSE Transitions               6289                 34
  VLAN Priority: 1
    PAUSE frames                      68              12578
    PAUSE duration                298206                578
    PAUSE Transitions               6289                 34
:
```

```
:
```

RDSCTRL command

The rdsctrl command is used to load, configure RDSv3 and monitor RDSv3 connections and statistics.

```
# rdsctrl -h
Usage:  rdsctrl set [-p] <RDS-tunable-name>=<value>
        rdsctrl get [<RDS-tunable-name>]
        rdsctrl default [-p] [<RDS-tunable-name>]
        rdsctrl load [ofed]
        rdsctrl unload
        rdsctrl ras [-p] <minimal | normal | detail | maximal>
        rdsctrl ras extract
        rdsctrl info {<flags>}
        rdsctrl ping [<IPv4-address>]
        rdsctrl conn restart <source-IPv4-address> <destination-IPv4-address>
        rdsctrl conn kill <source-IPv4-address> <destination-IPv4-address>
        rdsctrl trace start <trace-file-path> <maximum-data-(in-bytes)-per-RDS-
fragment>
        rdsctrl trace stop
        rdsctrl trace report <trace-file-path>
        rdsctrl version
        rdsctrl help [<RDS-tunable-name>]
```

The ‘rdsctrl info –n’ command shows the state of each Reliable Connection (RC) established on the system. Following is an example

```
#  rdsctrl info -n

RDS Connections:
      LocalAddr      RemoteAddr           NextTX           NextRX Flg
 169.254.35.163 169.254.179.224            38732         22661699 --C
169.254.159.169 169.254.191.254           143471         40943791 --C
 169.254.35.163 169.254.211.211            32348         63017466 --C
 169.254.35.163  169.254.211.14            33972         26928246 --C
 169.254.35.163 169.254.225.212            57473         15563428 --C
:
```

In the above example the local and remote addresses are the HAIP addresses used by the Oracle Database. The “--C” indicated the RC has been established. A value of “-c-“indicates the connection is in the process of being established. If any other state persists in the Flg column it would indicate there is a problem with the connection.

The 'rdsctrl -c' output across multiple samples can be used to determine the amount of data being sent and received using the RDSv3 protocol. The following sample script shows the average values over a specified interval.

```
INTERVAL=30
ITERATIONS=999999999
if [ "$1" != "" ]
then
   INTERVAL=$1
   shift
fi
if [ "$1" != "" ]
then
   ITERATIONS=$1
fi
while [ $((ITERATIONS+1)) -gt 0 ]
```

```
do
   rdsctrl info -c | \
   awk -v DATE="`date \"+%Y/%m/%0d_%0H:%0M:%0S\"`" '/ send_queued /{S+=$2}; \
        / recv_delivered /{R+=$2}; \
        / copy_to_user /{CT+=$2}; \
        / copy_from_user /{CF+=$2}; \
        / send_rdma_bytes /{SRB+=$2}; \
        / send_rdma /{SR+=$2}; \
        / recv_rdma_bytes /{RRB+=$2}; \
        END{printf("%s %d %d %d %d %d %d\n",DATE,S,R,CT,CF,SRB,SR,RRB)}'
   sleep ${INTERVAL}
   ITERATIONS=$((ITERATIONS-1))
done | awk -v I=${INTERVAL} 'function phead() { \
      print "                    send         receive      copy to      copy
from    send RDMA    send RDMA    receive RDMA"; \
      print "Date_Time           queued (pps) del (pps)    user (MBps)  user
(MBps)  bytes (MBps) (pps)        bytes (MBps)"; \
      print "------------------- ------------ ------------ ------------ -------
----- ------------ ------------ ------------"; \
   } \
   BEGIN{phead();F=1;PR=0;PS=0;PCT=0;PCF=0;PSRB=0;PSR=0;PRRB=0;X=I*1000000}; \
   {if(F==1){F=0} else{printf("%s %12d %12d %12.2f %12.2f %12.2f %12d
%12.2f\n",$1,($2-PS)/I,($3-PR)/I,($4-PCT)/X,($5-PCF)/X,($6-PSRB)/X,($7-
PSR)/I,($8-PRRB)/X)}; \
   PS=$2;PR=$3;PCT=$4;PCF=$5;PSRB=$6;PSR=$7;PRRB=$8}'

Sample Output:
                    send         receive      copy to      copy from    send
RDMA    send RDMA    receive RDMA
Date_Time           queued (pps) del (pps)    user (MBps)  user (MBps)  bytes
(MBps) (pps)        bytes (MBps)
------------------- ------------ ------------ ------------ ------------ -------
----- ------------ ------------
:
2016/03/27_00:15:02       170939       275280      1243.61      1363.54
0.00            0         0.00
2016/03/27_00:15:32       174578       286684      1289.02      1392.94
0.00            0         0.00
2016/03/27_00:16:02       174633       286973      1288.62      1395.81
0.00            0         0.00
2016/03/27_00:16:32       176023       296863      1334.29      1402.21
0.00            0         0.00
2016/03/27_00:17:02       172483       281766      1263.18      1375.97
0.00            0         0.00
2016/03/27_00:17:32       172780       290111      1299.49      1377.92
0.00            0         0.00
2016/03/27_00:18:02       172195       283960      1281.14      1374.12
0.00            0         0.00
2016/03/27_00:18:32       173934       290723      1308.45      1389.06
0.00            0         0.00
2016/03/27_00:19:02       170293       282704      1275.49      1357.99
0.00            0         0.00
:
```

Summary

The certification of Oracle RAC 12*c* Release 1 to use the RDSv3 or UDP protocols with RoCE adapters on AIX allows you to achieve lower latency in RAC private communications without the need to add different communications adapters and switches to your network. This reduces the complexity of introducing RDSv3 to Oracle RAC deployments.

RDSv3 is a protocol that customers can use with Oracle RAC to optimize performance. Previously RDSv3 was only available to Oracle RAC customers using InfiniBand. The AIX 7.2 support for RDSv3 over RoCE and the certification of it for use with Oracle RAC 12*c* Release 1 provides a new option for customers with capable Ethernet infrastructure who desire the increased performance of the RDSv3 protocol. Using RDSv3 over RoCE the Oracle Database with Oracle RAC benefits from the low overhead, low latency, high bandwidth, and highly reliable characteristics of the RDSv3 protocol and the RoCE adapter.

The IBM FC EC3A/EC4B (Mellanox MCX314A-BCBT) adapter is optimized for enterprise data centers, high-performance computing, transactional databases, cloud computing, virtualization, storage, and other embedded environments. The adapter improves network performance by increasing available bandwidth to the processor and by providing enhanced performance to the applications by offloading to the adapter the movement of data between the nodes in the cluster.

Resources

These Web sites provide useful references to supplement the information contained in this document:

- IBM Power Servers
  http://www-03.ibm.com/systems/power/hardware/systemp.html

- IBM Publications Center
  https://www-05.ibm.com/e-business/linkweb/publications/servlet/pbi.wss?CTY=US

- IBM Redbooks
  www.redbooks.ibm.com/

- Mellanox Ethernet Adapters
  http://www.mellanox.com/page/products_dyn?product_family=127&mtag=connectx_3_en

- Mellanox Ethernet Switches
  http://www.mellanox.com/page/products_dyn?product_family=252&mtag=sx1000

About the author

Dennis Massanari is a Software Engineer with IBM Systems Group, Oracle ISV Enablement. He focuses on IBM Power™ Systems running Oracle Database. He is based in Redwood Shores, CA, USA.

Appendix 1: Flow Control Standards

What is 802.3x Flow Control (Global Pause)? The Ethernet standard (802.3) was designed as unreliable. There was no guarantee for packets to reach the required destinations as it was designed to be done in upper layer protocols (e.g. TCP).

Later, the IEEE 802.3x (Annex 31B of 802.3) flow control standard was defined for applications that cannot build reliability on the upper layer's protocols. It enables receiving buffer feedback (e.g. overflow) from a receiver to its sender.

The pause action (XOFF) is a control frame sent by the receiver to alert the sender that the receiver buffer is stressed and will potentially be overflowed. The sender responds by stopping transmission of any new packets until the receiver is ready to accept them again. The pause frame contains a timeout value. The sender will wait during this timeout or until XON control message is received.

IEEE 802.3x suffers from a basic disadvantage: after a link is paused, a sender cannot generate any more packets. As a result, when using flow control on a port (global pause), the Ethernet link cannot carry multiple traffic flows that require different QoS behavior, as when enabling flow control on a port, it pauses all types of traffic on that port including the ones that require high QoS. Moreover, if the link is between two switches in the network, the pause action may block servers flows that do not need to be paused.

What is 802.1Qbb Priority Flow Control (PFC)? IEEE 802.1Qbb PFC extends the basic IEEE 802.3x to multiple classes (8 classes). It enables applications that require flow control to coexist on the same link with applications that can manage without flow control. PFC defines each one of the eight different types of flows that can be subject to flow control. In case of L2 network, PFC uses the priority bits within the VLAN tag (IEEE 802.1p) to differentiate up to eight types of flows that can be subject to flow control (each one independently).

Note: PFC and Global pause flow control cannot be running together on the same interface, either one of them can be enabled.

Trademarks and special notices

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

IBM, the IBM logo, AIX, Power, Power Systems, and POWER8 are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

The information provided in this document is distributed “AS IS” without any warranty, either express or implied.

The information in this document may include technical inaccuracies or typographical errors.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance 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 or performance improvements equivalent to the ratios stated here.

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.