175 Cabot Street – Suite 210
Lowell, MA 01854 USA
www.cspi.com
NASDAQ: CSPI

DBLv5 Troubleshooting

Model:

N/A

Software:

DBLv5

Operating System:

Supports both the Linux and Windows Operating Systems.

Information:

The cause of defective network adapters can be easily determined. When this situation occurs, contact CSPI Technical Support to initiate a Return Merchandise Authorization (RMA) to obtain a replacement.

If the network appears to be functioning correctly but application transaction rates have not significantly improved after installing the ARC Series network adapter and the DBLv5 software, it is possible that the proprietary software features are not properly enabled. Only those hosts with valid DBLv5 licenses will demonstrate accelerated performance. Repeat the software licenses will demonstrate accelerated performance. Repeat the software license activation process on all suspect host machines to certify proper licensing requirements for all network adapters.

This chapter contains the following topics:

Hardware Installation and Performance Issues

Software Installation and System Configuration

13.1.- Hardware Installation and Performance Issues

The ARC Series network adapter is an x8 PCIe “Gen3” 10-Gigabit Ethernet network adapter. For optimal performance, properly seat the adapter in an x8 PCIe Gen3 expansion slot on the server.

The ARC Series adapter auto-negotiates operation in the widest available mode (x8,x4,x2,or x1) supported by the expansion slot into which it is installed, and at the highest data rate (8 or 5 or 2.5 gigatransfers per second GT/s). For optimal performance, verify that the adapter reports x8 “Gen3” (8 GT/s) PCIe link speed following seating into the PCIe slot on the server.

Two ways to determine if the network adapter is properly installed into a “Gen3” PCI Express slot:

Windows

Check the output of the **/opt/dbl/sbin/myri_info** directory file

Linux

Check the output of **lspci-vvv.**

Example:

Example header information from the output of **myri_info** for a "Gen3" 10-Gigabit Ethernet network adapter:

Usage:

```
#  ./myri_info  -b  0
Pci-dev at 1:00.0 vendor:product (rev) = 1c09:4258 (01)
    ▪ Behind bridge report root-port: 00:01.0 8086:0c01 (x8.3/x16.3)
Myri-10G-PCIE-8E –Link x8
        EEPROm String-spec:
                MAC=00:60:dd:43:cd:40
                SN=491940
                PC=10G-PCIE3-8E-2S
                PN=09-04669
                BOM=0.0.1.0.1
Firmware version 2.0.0 sha1: e28d461d878314f49560de21b345cb12
External Inputs:
PPS: Enabled, OK
10Mhz Clock: Disabled
```

pci_dev output results:

The ".3" notation refers to a PCIe 3.0 "Gen3" slot.

Behind the bridge root-port: 00:01.0 8086:0c01 (x8.3/x16.3) indicates that the adapter is running at x8 Gen3 speed (maximum capability)

The motherboard PCIe slot is x16-able.

The Myri-10G-PCIE-8E –Link x8 also indicates that the ARC Series network adapter is running optimally at x8 speed.

13.1.1. Operating systems using the lspci command

For operating system with the **lspci** command, look at the **lspci—vvv** output to examine the Link speed (**Lnk Sta**).

```
LnkSta: Speed 8Gt/s, Width x8, TrErr- Train-SlotClK+ DLActive- BWMgmt- ABWMgmt-
```

13.2.1.- Software installation and system configuration issues

Bug report scripts:

/opt/dbl/sbin/phx_bug-report is a diagnostic script included in the Linux DBLv5 software distribution. The script collects diagnostic information about a customer's system configuration, such as **uname** output, processor files such as **cpuinfo** and interrupts, **lspci**, kernel messages, **ethtool, myri_counterrs,** and so on. The script must be run as root.

Phx_bug_report.ps1 is a Windows powershell script in the Windows phx-tools distribution (Toolkit-Phoenix Group on the CSPi Customer Portal). The script extracts logging information from Windows Logs and prints them to **stdout**. The script must be run as an administrator.

Executing the script requires the following setting:

Set-executionpolicy remotesigned

The output determines if the DBLv5 license is valid, which driver is loaded, and records error statements. ***Note: if you cannot run Windows powershell, select **EventViewer -> Windows Logs** for error messages related to DBLv5 software. ***

Linux RPM-TGZ installation failure

If you encounter errors during the Linux RPM installation process, send the complete output from the RPM command, the kernel log output, and the **/tmp/myri_dbl.log.**

If you encounter errors during the Linux TGZ installation, send the complete output from the **sbin/rebuild.sh** command.

***Note: To build a DBLv5 kernel module, configure the source kernel tree to match the running kernel.

Redhat example: You must install the **kernel-devel** package and the **kernel-headers** package from the RedHat distribution to build the DBLv5 kernel module.

Windows MSI installation failure

To confirm that the MSI software is installed correctly, open the Device Manager (Start->Run->devmgmt.msc). The text Myricom 10Gbps Ethernet NIC with DBL appears under Device Manager>Network adapters.

If you encounter an install error, the **Device Manager-> Network adapters** output reveals no DBLv5 devices or a yellow exclamation mark (!) next to the device. The presence of an exclamation mark indicates an error message.

Check the **Event Viewer** for error messages, such as:

1. "Insufficient number of MSI-X vectors…"
2. This message indicates that the system does not have enough resources to install the driver, mainly due to other NICs claiming additional resources.

To free up more resources, decrease the number of endpoints by creating a **MYRI_MAX_ENDPOINT** registry key and set it 4, as follows:

3. REG ADD HKLM\SYSTEM\CurrentControlSet\services\dbl/myri_max_endpoints /t REG_DWORD /d 4.
4. The keyword "dbl" indicates that you are using the DBLv5 driver.

If the Windows MSI installation fails when running the MSI installer:

Send CSPI Technical Support the log obtained by adding **/log <filename>** to the MSI install, as follows:

5. C:\> dbl-5.0.0_x64.msi /log dbl_install_log.txt

If the MSI installation fails with an ambiguous message stating:

"There is a problem with this Window Installer Package. A program run as part of setup did not finish as expected."

6. Verify that an earlier removal has unlocked all processes from DBLv5-related files. Run the following command from a command prompt to determine if DBLv5 is still in use:

C:\> tasklist /m dbl.dll

- The output may show processes still using DBLv5-related software

7. Reboot Windows- Alternatively, you can install Windows DBLv5 software using Windows msiexec as described in **Installing DBLv5 Software.**

Bonding for failover in TA mode:

DBLv5 provides limited bonding support (in DBLv5 TA mode- used only for failover). The failover mode in DBLv5 TA (**dblrun**) requires no additional bonding driver configuration. The bonding driver is configured to bond the Ethernet DBLv5 interfaces as usual for failover, and DBLv5 TA mode will use that configuration. Under Linux, this feature requires configuring and using the **BONDING.KO** module for a failover bond with the Ethernet devices exposed by DBLv5. Under Windows, this feature requires configuring the CSPi-supplied teaming driver for a failover team.
When TA mode is used, if it detects that the device you are using is part of a failover bond, TA mode executes a **dbl_open** on each underlying device and then the native bonding driver detects that a link is down and selects a new active slave. TA mode then detects that the active slave has changed and switched to using that slave for its operation.

TA Failure:

If the socket application fails to run in DBLv5 Transparent Acceleration (TA) mode, follow the troubleshooting steps described below to identify the source of TA failure.
Report the error to CSPI technical Support using the **–f** option in **dblrun** to generate the error log as follows:

$ dblrun –f logfile myapp.exe

Note: Some error logfiles are very large and cumbersome. Attempt to reduce the size of the logfile by reproducing the error with the smallest number of connections(feeds) and a minimum number of variables that still show the error. Send the smaller error logfile.
Two ways to verify that DBLv5 TA mode is being activated when running the application:

Option 1:

Set the environment variable DBL_IP_VERBOSE to 1 when executing the dblrun command as follows:
$ DBL_IP_VERBOSE=1 dblrun [options] executable arguments
A diagnostic output prints to the console.

Option 2:

Examine the output of **dblrun –v <appname>.**

13.2.2.- Troubleshooting Q&A

Does the application run using standard IP?

Verify that the sockets application runs using standard IP communication.
$ dblrun –d <application>
The **–d** option forces **dblrun** to bypass the transparent accelerations mode and run the application over standard IP.

Does the problem occur using UDP or TCP or both?

Determine if the problem occurs when using UDP or TCP by running the application in combination.
$ dblrun { -u, -t }
By specifying **–u** and **–t**, you deploy LSP in 'fallthrough' mode, using the base provider.

How many UDP sockets are you trying to accelerate?

In DBLv5, the DBL stack for UDP acceleration supports up to 16 shared DBL endpoints per network adapter. If you have more sockets under DBLv5 TA enter the following:
$ dblrun –b 1 myapp.exe

The **dblrun –b 1** option multiplexes the UDP sockets using the general TA BSD stack. The same option applies if several applications are accelerated together with few available endpoints.

How are the ARC series network adapters connected?

DBLv5 TA mode (**dblrun**) supports loopback communication on the loopback address (IP 127.0.0.1). Running loopback with DBLv5 TA mode (**dblrun**) demotes a socket, passing traffic over internal OS dependent paths- faster than sending packets through DBLv5.

Do you have a network configuration with IP addresses on the same subnet?

Adding two IP addresses to the same subnet prints the following error message:
"Two IPs on the same subnet detected. TA mode is prevented for this configuration (see FAQ). Exiting."
This problem is more than likely a duplicate route in the system's routing table. This situation can arise a couple of ways:
You have manually added a duplicate route to the routing table.
There are two different interfaces on the same subnet.
DBLv5 does not support a network configuration with IP addresses on the same subnet. Having two interfaces on the same subnet causes confusion when you're trying to decide which interface to use for the normal Linux TCP stack.
Examine your system routing table carefully and check for any duplicate entries where the destination/netmask are the same. Check that you are not putting two or more interfaces on the same subnet.

LD_PRELOAD error message?

Have you ever seen the following **LD_PRELOAD** error message when you were using Linux DBLv5 TA mode?

> **Error: ld.so: object '/opt/dbl/lib/libdbl_preload.so' from LD_PRELOAD cannot be preloaded: ignored.**

If so, the error message is indicating that your target application is 32-bit. Linux DBLv5 only supports 64-bit applications.

Winsock error message?

You may have seen this Winsock error message if you've been using Windows DBLv5 TA mode.

> 20.09.2012 13:44:35.960 | 1948 | ERROR | 10021001 | dispatcher::issue false read result: [error 10045: "the attempted operation is not supported for the type of object referenced."]

The Winsock error messages indicates that you are using LSP11 for Winsock 1.1 applications, but your application is making overlapped calls for LSP22, which requires registration. Register the library **ipdbl_lsp22.dll** for Winsock 2.2.

Are you running a Windows .NET application built with AnyCPU?

If you are using Windows DBLv5 TA mode with a **.NET** application, you may have seen the following error message.
(dblrun info) Unable to accelerate the [64-bit] application. Install the DBLv5 TA LSP for the corresponding chains
Although you can build .NET applications for x86, x64, and AnyCPU, you cannot determine the bitness when compiling for AnyCPU. **Dblrun.exe** posts a warning to that effect. In this case the **LSP dll**, which matches the native OS bitness, needs to be registered.
.NET apps specifically targeting x64 or x86 do not see the **dblrun.exe** warning. If you specify **dblrun –v you.NetApp** and see a)transp info) in the command prompt, it confirms the appropriate DBLv5 TA chains have been intercepted.

13.2.3.-Software Counters

The **myri_counters** tool provides low-level DBLv5 hardware and software counters for traffic that pass through the network adapter in DBLv5 mode.

Linux:

```
$ /opt/dbl/bin/myri_counters
```

Windows:

```
$ [INSTALL_DIR]\bin\myri_counters.exe
```

By default, the **myri_counters** output only displays on port/board 0. Dual-port adapters appear to **myri_counters** as different port/boards. If you have a dual-port network adapter installed in the host, you must specify the command-line argument **–p <port_num>** to obtain the counters output for each port. The environment variable **port_num** has an integer value from 0 to n-1, where "n" represents the number of network adapters installed in the host and running the DBLv5 driver, as follows:

```
$ /opt/dbl/bin/myri_counters –p 0
$ /opt/dbl/bin/myri_counters –p 1
```

If a host has two dual port adapters, you would assign –p0 and –p1 to the ports of one network adapter and –p2 and –p3 to the ports of the second adapter. To clear and reset the counters requires root privileges. To clear the counters on a specific port of a network adapter, enter the following:

```
# /opt/dbl/bin/myri_counters –p <port_num> -c
```

13.2.4.- Environmental Variables

DBLv5 software offers a variety of environment variables for debugging, timestamping, and for customizing the software configuration to meet application requirements.

DBL_IP_VERBOSE

Setting **DBL_IP_VERBOSE** to **1** prints additional diagnostic output to the console.

```
$ DBL_IP_VERBOSE=1 ./test/dbl_pingpong –s -1 10.0.0.1
```

OR `$ DBL_IP_VERBOSE=1 dblrun [options] executable arguments`

DBL_CONFIG

The environment variable **DBL_CONFIG**, steers various MTCP variables in DBLv5 TCP API. These variables include, but are not limited to, congestion modules and delayed ACK.

DBL_ST_MODE

For single-threaded applications, set the environment variable DBL_ST_MODE to 1 for a faster TCP stack for DBLv5 TCP extensions.

13.2.5.- Functional failures

Some socket-based applications may experience a functional failure because DBLv5 behavior differs from the regular OS stack.

Linux assigns a substantial portion of the transparent sockets service to BSD-style Linux sockets emulation. In the event a functional failure occurs, enter the following command to save tracing output:

```
$ dblrun –f output <application_name>
```

Send the output file to CSPI Technical Support

13.2.6.- Performance Issues

Refer to the Tuning chapter to familiarize yourself with tuning DBLv5. If these suggestions do not address the issue, follow these steps

8. Verify that the Transparent Socket Acceleration (DBLv5 TA mode) is effectively loaded.

Check for the following information after running DBLv5:

<process_name> [<pid>] DBL 5.0.0. <build_id> Copyright 2016 CSPI

9. Verify that the network adapter is installed in a PCIe expansion slot that can sustain 10-Gigabits per second. Refer to the Testing chapter for configuration suggestions.
10. Monitor **myri_counters** for accelerated traffic. Check that the network is not experiencing "receive data buffering" overflow conditions.
11. Check the network adapter runtime counter values as described below:

Examine **Packets Received (all ports)** and **Packets RX (this port)** counter values to verify that traffic is passing directly to a user space stack.

Examine **Packets Drop Filter, Packets Drop Filter (HW), Packets RX Filter, Packets Drop Incoming,** and **PCIE FIFO** counter values for traffic that the network adapter and stack cannot sustain. The counter values are zero in normal operation.

By default, the **myri_counters** output only displays on port/board 0. Dual-port adapters appear to **myri-counters** as different ports/board. If you have a dual-port network adapter installed in the host, you must specify the command line argument **–p <port_num>** to obtain the counters output for each port. The environment variable **PORT_NUM** has an integer value from **0** to **n-1**, where **n** represents the number of network adapters installed in the host and running the DBLv5 drive.

13.2.7.- Network Adapter Timestamps

DBLv5 supports socket timestamps on ARC Series network adapters.

The timestamp is made available through the socket interface. The socket interface, running the **SO_TIMESTAMP[NS]** socket option, is a high precision clock synchronized at startup to system host time (returned by the **gettimeofday()** function call). The timestamp automatically attaches to the packet when it arrives at the network adapter.

Linux users can access the clock as a POSIX clock or through the PTP protocol.

Caution: Be careful running NTP services (that rely on network time protocols) on the POSIX clock! NTP services cause variations in system host times that may exceed inter-packet arrival times.***

Note: While you can inject timestamps into Windows packets, there is no interface to adjust the clock after it is initially set by loading the driver.***

13.2.8.- Synchronization

There are different levels of timestamp synchronization:

Network Adapter-to-host synchronization (local sync)

Host-to-host synchronization (global sync)

Network Adapter-to-host synchronization (local sync)

Use the **phc2sys** tool in Linux to run Network adapter-to-host synchronization. Set the ARC Series network adapter to "master" clock, for higher resolution and accuracy (recommended).

Host-to-host synchronization (global sync)

Most systems use **NTPD** to run Host-to-host synchronization (global sync). More precise options to **NTPD** exist; however, it only raises the accuracy level from milliseconds to tens of milliseconds. Since both protocols are not typically run on dedicated networks and because there is typically a lot of host overhead in processing the protocol, the time can only be so accurate.

The ARC Series network adapters connect externally to a PPS or a 10MHz input to improve adapter clock accuracy. When loading the driver, set the module variable inputs as follows:

"myri_clk_enable_pps=1"

"myri_clk_enable_10mhz=1"

Revision	**Date**	**Change**
1	6/30/2016	Initial Draft
2	8/18/2016	Feedback Edits