

Infiniband function test

Under Linux

Ruediger Teichert

14 Aug 2009

kontakt@teichert-online.de

Prerequisites

This test should work on any platform. The given board has 2 infiniband ports.

Besides the motherboard you will need:

- Infiniband cable
- Internet connection
- SLES 11 32 bit DVD

This test was done with (Suse Linux) **SLES 11 32bit**, it is compatible with openSUSE 11.1

Step 1: Install OS

- Do not plug the Infiniband cable during installation
 - It should not matter, but I avoided it not to get any Infiniband drivers onto the system, which may be hard to remove later
- Install the OS with all Development packages (C/C++ needed), but do **not** install any Infiniband support (you can select it, but do not do it!).
 - Note: installing the Infiniband support will install the openSM libraries whose removal is next to impossible later.
- Do **NOT** install Xen or VM or similar!
- Shutdown computer (type init 0) and plug infiniband cable into both ports.
- **NOTE:** set system time correctly or build will fail later!

Step 2: Get the OFED package

- For this test, an Alpha release of OFED 1.5 was used, filename is OFED-1.5-20090722-0600.tgz. You can retrieve OFED packages from <http://www.openfabrics.org/downloads/OFED/>
 - This package is newer than the packages provided with SLES 11

Step 3: Copy package

- Now copy the OFED-....tgz file onto your SUSE system. Work under root account.
- `#mkdir /root/install` will create an installation directory (never type the #).
- `#cp /media/usb/OFED*.tgz /root/install` will copy the file, assuming you are using a USB flash disk mounted as `/media/usb` (or use the XWindows GUI to copy the file via mouse).
 - Assuming there is only one file on your flash disk which starts with OFED and ends with .tgz

Step 4: uncompress file

- `#cd /root/install`
- `#gunzip OFED*.tgz`
- `#tar xf OFED*.tar`
- `#cd OFED*`
 - So we have created the OFED... installation folder inside `/root/install`
 - Again I assume there is only one file starting with OFED and ending with tgz or tar in `/root/install`

Step 5: start installation script

- `# ./install.pl` starts the installation script.
 - In case you cannot execute this file: use `#chmod +x install.pl` to make it executable.
 - » Again, never type any `#`
- In the menu, choose option 2 (install software).
- In the installation menu, choose option 3 (all packages).
- Problem: you will get a list of missing libraries. Keep this list, you will need it later.

Step 6: Install Smart Package Manager

- Open another terminal not to erase the list of missing packages (or use CTRL+ALT+F1 or CTRL+ALT+F2 etc. if you are in text mode to switch between screens) and install the Smart Package Manager:

```
#zypper ar -r
```

```
http://download.opensuse.org/repositories/smart/openSUSE\_11.1/smart.repo
```

- if in the future you are using SLES 11.1, you will need the openSUSE_11.2 repository as openSUSE version No. is always +1
- Install zypper via Yast in case zypper is not found.

Step 6 (cont.)

- `#zypper ref smart`
- `#zypper install smart`
 - Confirm with `t` (which means you trust the software) if asked to do so, confirm installations with `y` or `yes` or `YES` or whatever input is required (you will have a selection like `[YES/no]` sometimes)
 - Problem: If you are prompted that the “alias” would be already existing on the system (happens at least once in this or the next steps), just type a similar name at the prompt. Example: if the alias “`opensuse`” is existing, simply call it `opensuse2` or whatever you like.
- `#smart channel --add`
<http://linux01.gwdg.de/~pbleser/files/smart/opensuse-11.1.txt>
- **Confirm inclusion of all channels!**
 - Remember to change to `opensuse-11.2.txt` in case you are using SLES 11.1 in the future (or `...11.3.txt` for SLES 11.2 etc.)

Step 6 (cont.)

- `#smart mirror --add`
<http://linux01.gwdg.de/~pbleser/files/smart/mirrors-eu.txt>
- `#smart update`
 - This will take some time to read all the software package info from the server.
 - Afterwards, you could install the missing packages from Step 5 directly via several `#smart install <missingpackage>` commands (do NOT type the `< >`). It might show you several packages with similar names, choose the right one!
 - However, for convenience we will install the graphical user interface of Smart:
- `#smart install smart-gui`
 - Wait until the installation is finished.

Step 6 (cont.)

- Problem: sometimes signature keys can not be retrieved and installation stops. In that case type:
- `#smart config --set rpm-check-signature=false`

A quick-and-dirty solution, but okay for a test system. Only for a test system!

Step 7: install missing packages

- You do not need this step in case you opted for command line installation in step 6 of course
- Start the smart GUI (graphical interface) in one of the following ways:
 - Open a terminal and type `#smart --gui` (do not type `#`)
 - Or use XWindows to start it via mouse, With Gnome Desktop this is `<Computer>→Applications→More Applications→System-→Smart (Package Manager)`.
- Wait until Smart is updating its cache.
- Click on the magnifier/search symbol in the top tool bar (or click `Edit-→Find`).

Step 7 (cont.)

- A search box will open. Type the name of the first missing package (from step 5) completely and then press ENTER (or hit the magnifier/search symbol).
 - For me, the list of missing packages was:
 - » **zlib-devel**
 - » **pciutils-devel**
 - » **krb5-devel**
 - » **libevent-devel**
 - » **nfsidmap-devel**
 - » **libopenssl-devel**
 - » **cyrus-sasl-devel**
 - » **libltdl-devel**
 - » **tcl-devel**

Step 7 (cont.)

- After typing zlib-devel and searching for it via the button, Smart will show a folder (package group):
 - » > Development/Languages/C and C++
- Click on the triangle symbol (left of “Development/...”) to expand the list. You will see a list of versions, probably only one version
 - » [*] zlib-devel.....1.2.3-104.137@i586
- The empty box (possibly with yellow star) means it is not installed!
- Click on the box symbol next to zlib... with a right mouse click and choose “Install” with a left mouse click.
- Now you will see a box with an arrow in it next to zlib...
- If you see more than one version choose the latest one.

Step 7 (cont.)

- Click on the gear-wheel symbol on the tool bar or click File->Execute Changes in the menu. Confirm with okay.
- Smart will now download the package and install it. Wait a while.
- Problem: If Smart stops downloading and opens an error message, just repeat the process.
- If a package is successfully installed you will see a black (or actually dark grey) square next to the package name.
- Afterwards proceed with the next missing package from the list. Remember to choose the latest version always!
- When finished, close Smart Package Manager.

Step 8: Repeat Step 5

- So now all missing devels are installed, so back to Step 5:
- `#cd /root/install/OFED*`
- `#./install.pl`
- Choose 2 “Install Software”
- Choose 3 “All packages”
- Now you should not see anymore error messages but instead a message:
UNINSTALLING THE PREVIOUS VERSION OF OFED
- Problem: If this fails with a notice (removal of old version failed), then you have old OFED/Infiniband libraries!

Step 8 (cont.)

- This is very hard to solve, as the openSM library rejects practically all un-installation attempts. It is usually faster to reinstall the test system if you get this problem!
- If everything works out fine, a long compilation process will start now with lots of screen output.
- Problem: compilation in the end might fail. Reading the mentioned log file complains about an “implicit definition of a function”. This is simply a kernel problem. Resolve as follows:
 - `#uname -a` will show the kernel version. Officially supported is 2.6.27-19.5-default as mentioned in the release notes of this OFED 1.5 Alpha build. However I had 2.6.27-19.5-pae kernel installed (supports address extension for 32 bit architectures).
 - It is possible to downgrade via YAST to 2.6.25-19.5-default, however that made problems and so we use SMART to change the kernel to 2.6.27-23.0.1

Step 9: change kernel (only if your compilation failed in step 8)

- To change to a newer default kernel, open Smart as before.
- Type kernel in the search window and scroll through the software list on the right hand side until you find:

[*] kernel-default 2.6.27.23-0.1.1

the box is empty or has a yellow star to show this software is not installed. If black box, it is installed!

- Right-click the box on the left (next to “kernel...”) and select “install”, confirm with OK and then choose File→Execute Changes and then confirm with OK. After installation, restart computer (type #init 6).

Step 9a: install kernel-source (only if your compilation failed in step 8)

- Open Smart again and search for “kernel-source” without the “.
- You will see
 - > Development/Sources
- Expand it and select
 - [*] kernel-source 2.6.27.23-0.1.1@i586
- Click on File→Execute Changes and confirm and quit.

Step 10: restart, rebuild (only if your compilation failed in step 8)

- As the compilation of “Step 8-last page” had stopped, we restart it again, this time with the right kernel.
- Choose the kernel 2.6.27-23.0.1.1 (default) in the SUSE boot list, in case there are several kernels!
- `#cd /root/install/OFED*`
- `# ./install.pl`
- Choose 2 and then 3 as before!
- It will this time succeed, hopefully...

Step 11: exit shell

... and if compilation was successful you are finally asked if you want to configure b0, b1, b2 and b3.

- Say No!
- Leave the installer with Q for Exit

Step 12: Install XRC support

- We still need to install XRC support, otherwise our test routines will crash and complain the kernel would not have XRC support. XRC is basically a storage communication protocol. We need to use Yast to take it from SLES 11 DVD:
- Open YAST Software management and type mlx in the search field. Among other software, you will see
`libmlx4-rdmav2`
- Install this library!

Step 13: Start test!

- Restart the computer, after restart you should see the LEDs lit on the Infiniband ports!
- Type `#ifconfig -a`, you should see `ib0` and `ib1`.
- Open a terminal and type:
- `#ifconfig ib0 10.0.1.1/24`
- `#ifconfig ib1 10.0.1.2/24`
- `#opensm`
- You will see some screen output. Open a 2nd terminal and type:
- `#ib_write_bw -a`

Step 13 (cont.)

- Open a 3rd terminal and type:

```
#ib_write_bw -a 10.0.1.1
```

- After successful data copy, go back to terminal 2 and repeat:

```
#ib_write_bw -a
```

- Then go to terminal 3 again and type:

```
#ib_write_bw -a 10.0.1.2
```

After data transport is finished we have successfully used Infinband!

Notes

- It should be necessary to change the config file ofed.conf a bit, but my test run without it.
- If you have trouble, locate ofed.conf (you can search by `#find / -name "ofed.conf"`) and change the line:
 - `nfsrdma=y`
 - to:
 - `nfsrdma=n`