Report on Summer Student Program JINR-2015 (students.jinr.ru) summer program:

Configuration of cluster environment from scratch with IPMI, ZFS and InfiniBand

By **Oleg Iakushkin** PhD student from Saint Petersburg State University (SPBU)

Under supervision of **Artem Petrosyan** from the Laboratory of Information Technologies of Joint Institute for Nuclear Research (LIT JINR)

2015

Table of Contents

About this document

This report intends to allow reproduction of performed work, analysis and expansion of provided results.

Task

Configuration of a SuperBlade system installed in a rack (an analog to Hybrilit[1] cluster) from scratch: from bios+raid setup to InfiniBand (IB) application selection and performance tuning and analysis. Use of Scientific Linux 6.6 OS is a must.

Setup

Given a SuperBlade server with two diskless systems (blade nodes) installed in a rack that looks like this:

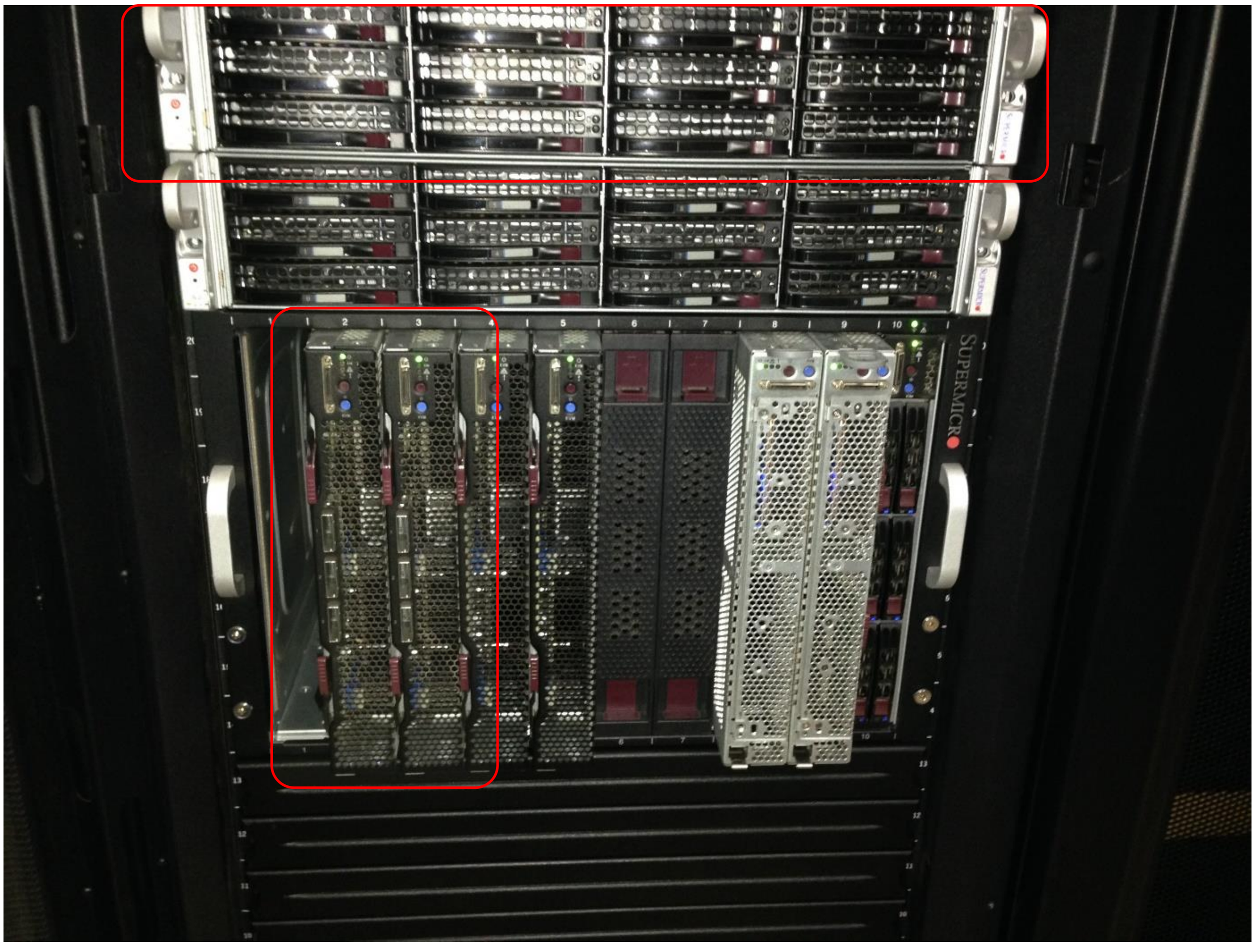


Figure 1: part of Hybrilit cluster, parts provided for experimentation selected in red. A server node and two blades.

Access

Server and blade were available via IPMI. After installation of Operating System (OS) access should be performed over SSH.

[1] http://hybrilit.jinr.ru/

LiveCD SMB/Windows Share hosting

IPMI allows remote .iso file mounting for OS installation onto Server. It supports Server Message Block (SMB) (aka Windows Share) protocol. To host SMB server with Scientific Linux 6.6 .iso LIT JINR cloud infrastructure[2] was used.

A *openvz_scientific_6-x86_64_krb_clst33* VM was allocated. Special dedicated user was created for files ownership. We installed and configured *samba* server to share user folder.

Samba configuration

A few lines were changed in samba configuration (/etc/samba/smb.conf) [global] group:

```
workgroup = WORKGROUP
wins support = yes
encrypt passwords = true
```

Then folder was shared:

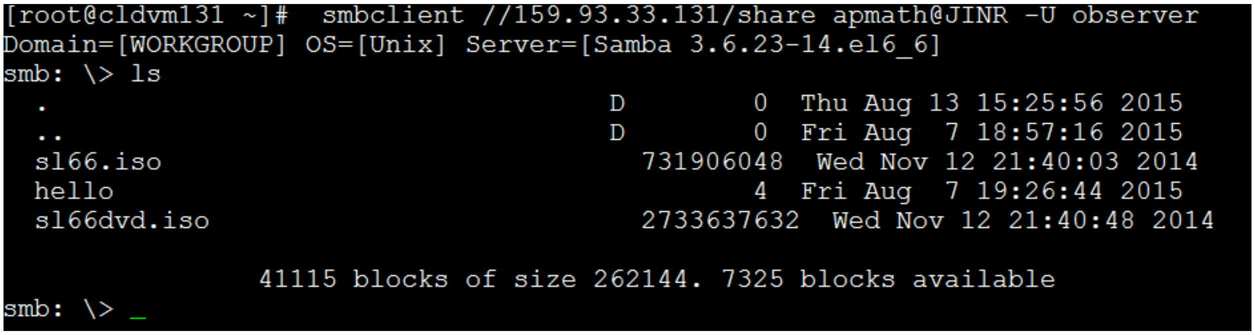
```
[share]
path = /home/observer/share
available = yes
read only = yes
browsable = yes
public = yes
guest ok = yes
create mask = 0755
```

Samba Server was restarted

```
sudo service smb restart
```

Local SMB testing

An sl66.iso LiveCD file was placed into /home/observer/share folder, rights to folder and its contents were passed to the user.



```
[root@cldvm131 ~]#  smbclient //159.93.33.131/share apmath@JINR -U observer
Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.6.23-14.el6_6]
smb: \> ls
  .                                   D        0  Thu Aug 13 15:25:56 2015
  ..                                  D        0  Fri Aug  7 18:57:16 2015
  sl66.iso                              731906048  Wed Nov 12 21:40:03 2014
  hello                                        4  Fri Aug  7 19:26:44 2015
  sl66dvd.iso                          2733637632  Wed Nov 12 21:40:48 2014

                41115 blocks of size 262144. 7325 blocks available
smb: \> _
```

Figure 2: SMB samba server was tested locally using smbclient.

VM configuration

Ports required to share data over SMB were opened.

```
-A INPUT -s 77.51.0.0/16 -p tcp -m state --state NEW -m tcp --dport 455 -j ACCEPT
-A INPUT -s 188.184.0.0/16 -p tcp -m state --state NEW -m tcp --dport 455 -j ACCEPT
-A INPUT -s 137.138.0.0/16 -p tcp -m state --state NEW -m tcp --dport 455 -j ACCEPT
-A INPUT -s 95.221.0.0/16 -p tcp -m state --state NEW -m tcp --dport 455 -j ACCEPT
```

[2] https://cloud.jinr.ru/

```
-A INPUT -s 91.203.80.0/22 -p tcp -m state --state NEW -m tcp --dport 455 -j ACCEPT
-A INPUT -s 62.84.96.0/19 -p tcp -m state --state NEW -m tcp --dport 455 -j ACCEPT
-A INPUT -s 159.93.0.0/16 -p tcp -m state --state NEW -m tcp --dport 455 -j ACCEPT

-A INPUT -s 77.51.0.0/16 -p tcp -m state --state NEW -m tcp --dport 139 -j ACCEPT
-A INPUT -s 188.184.0.0/16 -p tcp -m state --state NEW -m tcp --dport 139 -j ACCEPT
-A INPUT -s 137.138.0.0/16 -p tcp -m state --state NEW -m tcp --dport 139 -j ACCEPT
-A INPUT -s 95.221.0.0/16 -p tcp -m state --state NEW -m tcp --dport 139 -j ACCEPT
-A INPUT -s 91.203.80.0/22 -p tcp -m state --state NEW -m tcp --dport 139 -j ACCEPT
-A INPUT -s 62.84.96.0/19 -p tcp -m state --state NEW -m tcp --dport 139 -j ACCEPT
-A INPUT -s 159.93.0.0/16 -p tcp -m state --state NEW -m tcp --dport 139 -j ACCEPT
```

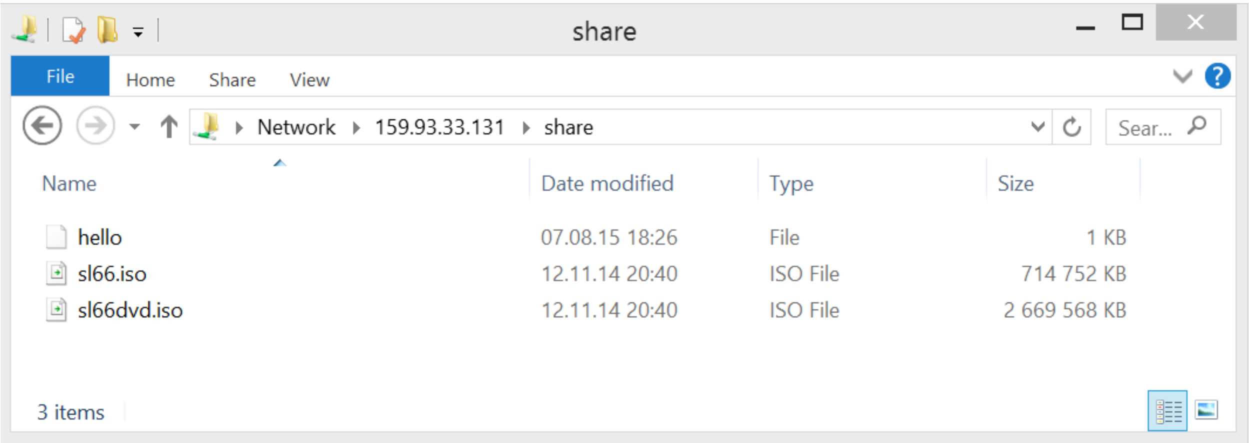
Lines were appended to `/etc/sysconfig/iptables`, and

```
service iptables restart
```

was performed.

Global SMB testing

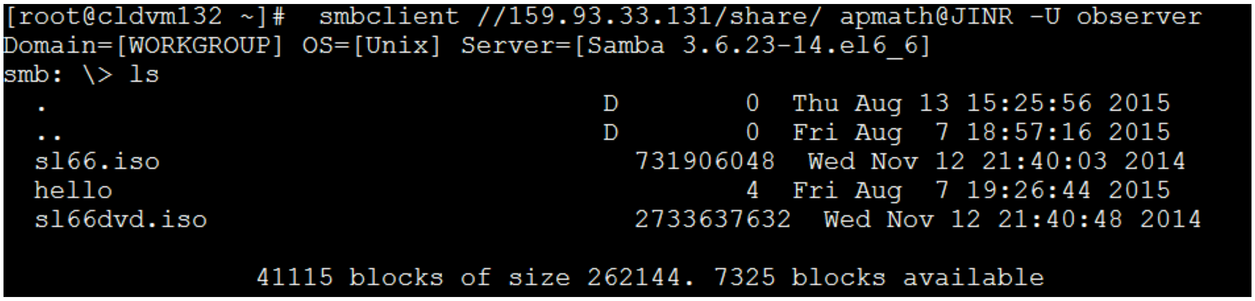
Samba was tested from remote Windows PC:



Name	Date modified	Type	Size
hello	07.08.15 18:26	File	1 KB
sl66.iso	12.11.14 20:40	ISO File	714 752 KB
sl66dvd.iso	12.11.14 20:40	ISO File	2 669 568 KB

Figure 3: Windows Explorer showing remote directory.

And from another cloud VM:



```
[root@cldvm132 ~]#  smbclient //159.93.33.131/share/ apmath@JINR -U observer
Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.6.23-14.el6_6]
smb: \> ls
  .                                   D        0  Thu Aug 13 15:25:56 2015
  ..                                  D        0  Fri Aug  7 18:57:16 2015
  sl66.iso                              731906048  Wed Nov 12 21:40:03 2014
  hello                                        4  Fri Aug  7 19:26:44 2015
  sl66dvd.iso                          2733637632  Wed Nov 12 21:40:48 2014

                41115 blocks of size 262144. 7325 blocks available
```

Figure 4: SMB client showing remote directory.

Now OS installation could be performed over IPMI.

Path to Mac -> Windows 8-> SMCIPMITool configuration

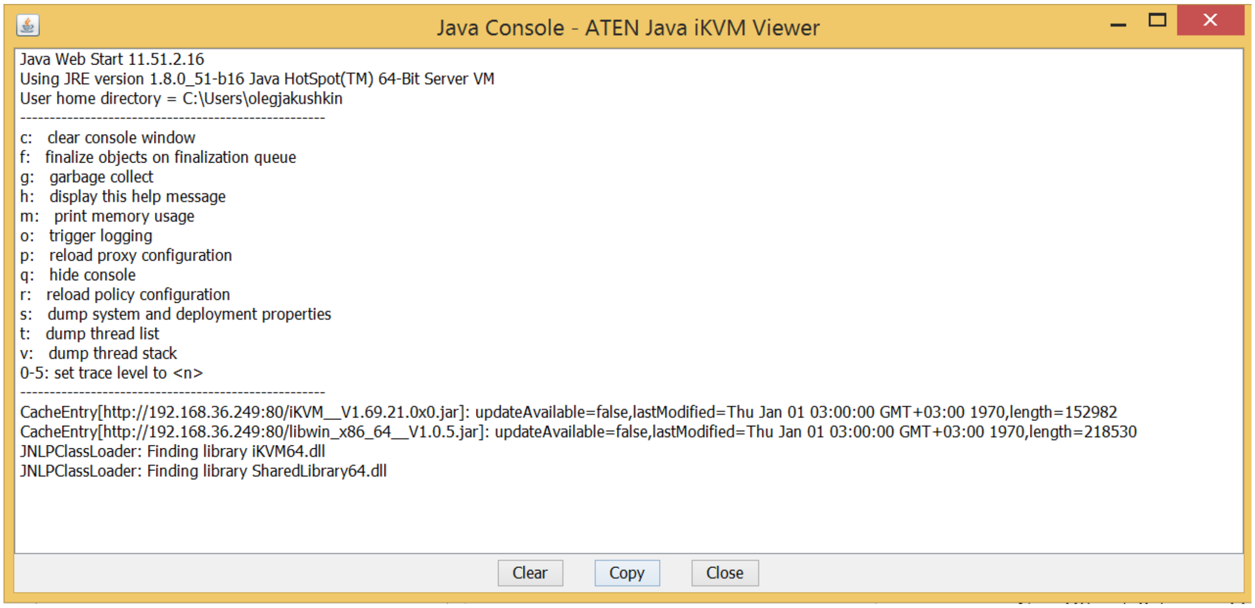
There are 3 ways to use IPMI: Web UI, GUI Client (IPMIView), Command Line Interface (SMCIPMITool).

To interact with server Supermicro provides IPMI iKVM Java viewer. It requires to use native libraries on that are provided for Windows and Linux.

So having a Mac Book workstation we had to install VM with Windows or Linux on top of it. Windows 8.1 was selected for ease of use.

IPMI Web Site Viewing

In hopes for remote work capabilities tunneling was studied. A cloud VM was used to tunnel connection. It allowed us to view website from any location. However we found out, underlying iKVM native libraries use UDP ports so one cannot use simple TCP tunnels.



```
Java Web Start 11.51.2.16
Using JRE version 1.8.0_51-b16 Java HotSpot(TM) 64-Bit Server VM
User home directory = C:\Users\olegjakushkin
----------------------------------------------------
c:   clear console window
f:   finalize objects on finalization queue
g:   garbage collect
h:   display this help message
m:   print memory usage
o:   trigger logging
p:   reload proxy configuration
q:   hide console
r:   reload policy configuration
s:   dump system and deployment properties
t:   dump thread list
v:   dump thread stack
0-5: set trace level to <n>
----------------------------------------------------
CacheEntry[http://192.168.36.249:80/iKVM__V1.69.21.0x0.jar]: updateAvailable=false,lastModified=Thu Jan 01 03:00:00 GMT+03:00 1970,length=152982
CacheEntry[http://192.168.36.249:80/libwin_x86_64__V1.0.5.jar]: updateAvailable=false,lastModified=Thu Jan 01 03:00:00 GMT+03:00 1970,length=218530
JNLPClassLoader: Finding library iKVM64.dll
JNLPClassLoader: Finding library SharedLibrary64.dll
```

Figure 5: iKVM connection error.

Web UI was effective to get a fast look at current overall server state. Its iKVM viewer is only updated with hos IPMI and has shown bugs like black not rendered parts of screen in latest Java 8 version.

Virtual media hosting was also unclear and could not mount our SMB stored iso file.

IPMIView

Same thing was with IPMIView iKVM interface – parts of it were rendered as black. Virtual Media facileties were acting unclearly.

SMCIPMITool

Shown to be best in iKVM rendering performance and provided clear manual on details such as virtual media use.

We started it with

```
# SMCIPMITool.exe 192.168.36.249 oleg password shell
```

And then it was used as interactive shell.

To mount and unmount an iso we used `vmwa` commands:

```
vmwa dev2iso \\159.93.33.131\share\sl66.iso
vmwa dev2stop
```

To start iKVM we used

```
kvmwa
```

command.

SMCIPMITool errors

In case of *vmwa exception***:**

```
java.lang.NullPointerException
        at com.supermicro.ipmi.UDPSocket.sendPacket(UDPSocket.java:213)
        at com.supermicro.ipmi.RMCP.send(RMCP.java:155)
        at
com.supermicro.ipmi.IPMIMessagingCommand.GetSystemGUIDCommand(IPMIMessagingCommand.java:70)
        at com.supermicro.ipmi.IPMIMessagingCommand.getSystemGUID(IPMIMessagingCommand.java:891)
        at
com.supermicro.ipmi.IPMIMessagingCommand.getSystemGUIDByIP(IPMIMessagingCommand.java:918)
        at com.supermicro.ipmi.text.ShellCommand.getPrompt(ShellCommand.java:425)
        at com.supermicro.ipmi.text.ShellCommand.execute(ShellCommand.java:274)
        at com.supermicro.ipmi.text.SuperBladeTool.execute(SuperBladeTool.java:2186)
        at com.supermicro.ipmi.text.SuperBladeTool.main(SuperBladeTool.java:2127)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
        at java.lang.reflect.Method.invoke(Method.java:597)
        at com.zerog.lax.LAX.launch(DashoA10*..)
        at com.zerog.lax.LAX.main(DashoA10*..)
```

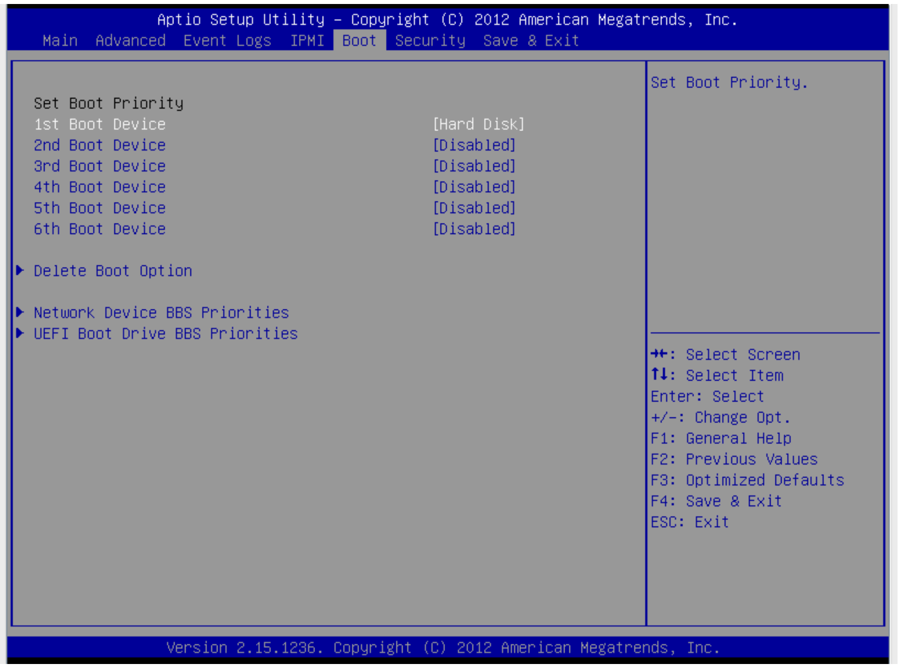
That kills *SMCIPMITool* process just call

```
ipmi reset
```

This will warmly restart IPMI and cure UDP socket error.

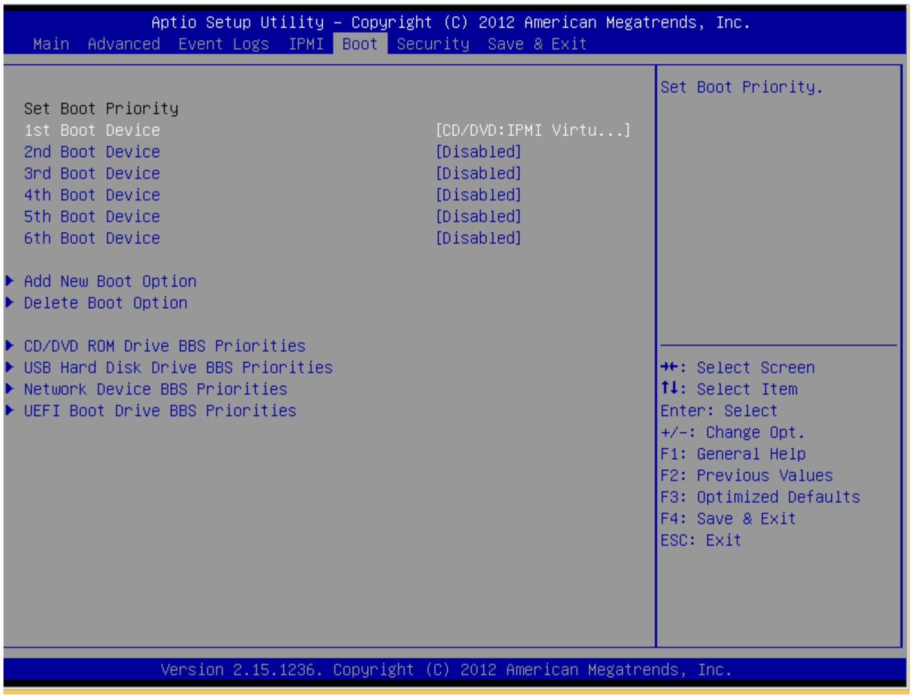
Server node

Server BIOS setup


Aptio Setup Utility – Copyright (C) 2012 American Megatrends, Inc.
Main Advanced Event Logs IPMI Boot Security Save & Exit
Set Boot Priority.
→←: Select Screen
↑↓: Select Item
Enter: Select
+/-: Change Opt.
F1: General Help
F2: Previous Values
F3: Optimized Defaults
F4: Save & Exit
ESC: Exit
Version 2.15.1236. Copyright (C) 2012 American Megatrends, Inc.

```
Set Boot Priority
1st Boot Device                              [Hard Disk]
2nd Boot Device                              [Disabled]
3rd Boot Device                              [Disabled]
4th Boot Device                              [Disabled]
5th Boot Device                              [Disabled]
6th Boot Device                              [Disabled]

▶ Delete Boot Option

▶ Network Device BBS Priorities
▶ UEFI Boot Drive BBS Priorities
```

Figure 6: BIOS before IPMI boot device selection


Aptio Setup Utility – Copyright (C) 2012 American Megatrends, Inc.
Main Advanced Event Logs IPMI Boot Security Save & Exit
Set Boot Priority.
→←: Select Screen
↑↓: Select Item
Enter: Select
+/-: Change Opt.
F1: General Help
F2: Previous Values
F3: Optimized Defaults
F4: Save & Exit
ESC: Exit
Version 2.15.1236. Copyright (C) 2012 American Megatrends, Inc.

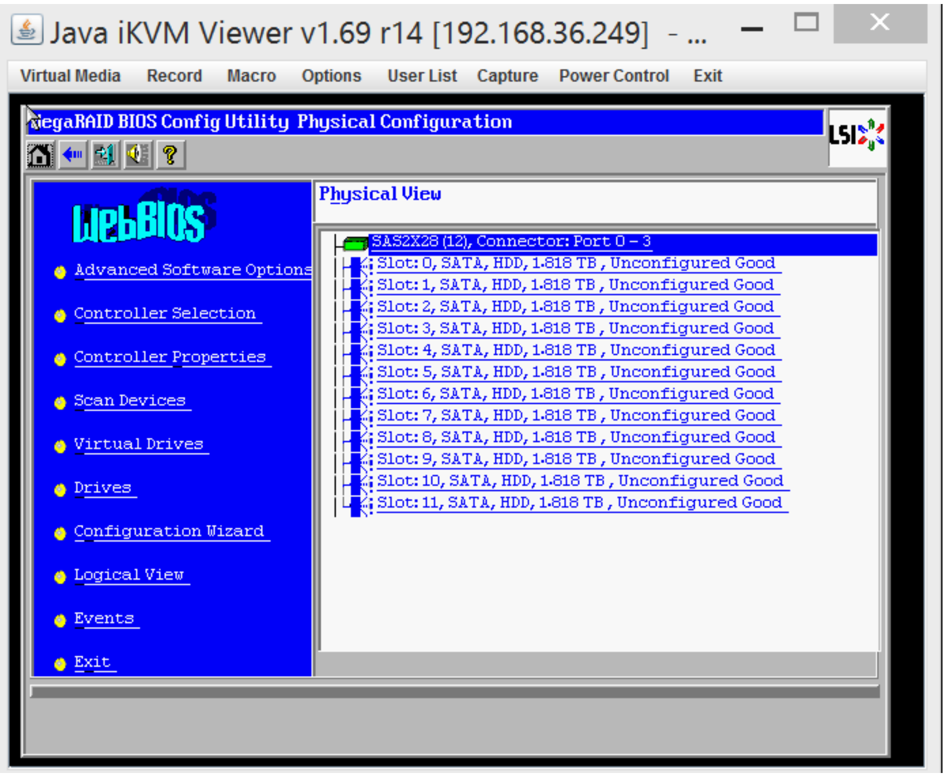
```
Set Boot Priority
1st Boot Device                              [CD/DVD:IPMI Virtu...]
2nd Boot Device                              [Disabled]
3rd Boot Device                              [Disabled]
4th Boot Device                              [Disabled]
5th Boot Device                              [Disabled]
6th Boot Device                              [Disabled]

▶ Add New Boot Option
▶ Delete Boot Option

▶ CD/DVD ROM Drive BBS Priorities
▶ USB Hard Disk Drive BBS Priorities
▶ Network Device BBS Priorities
▶ UEFI Boot Drive BBS Priorities
```

Figure 7: BIOS after IPMI boot device selection

Server Hardware RAID setup

We need to setup RAID, a ZFS one, yet hard drives shall be available for partitioning on the OS side. Thus we need to show hard drives to OS while hard drives shall not be inside RAID array.

Most described way of doing such RAID controller configuration is using WebBios. There are even video tutorials for Supermicro motherboards[3].



Physical View

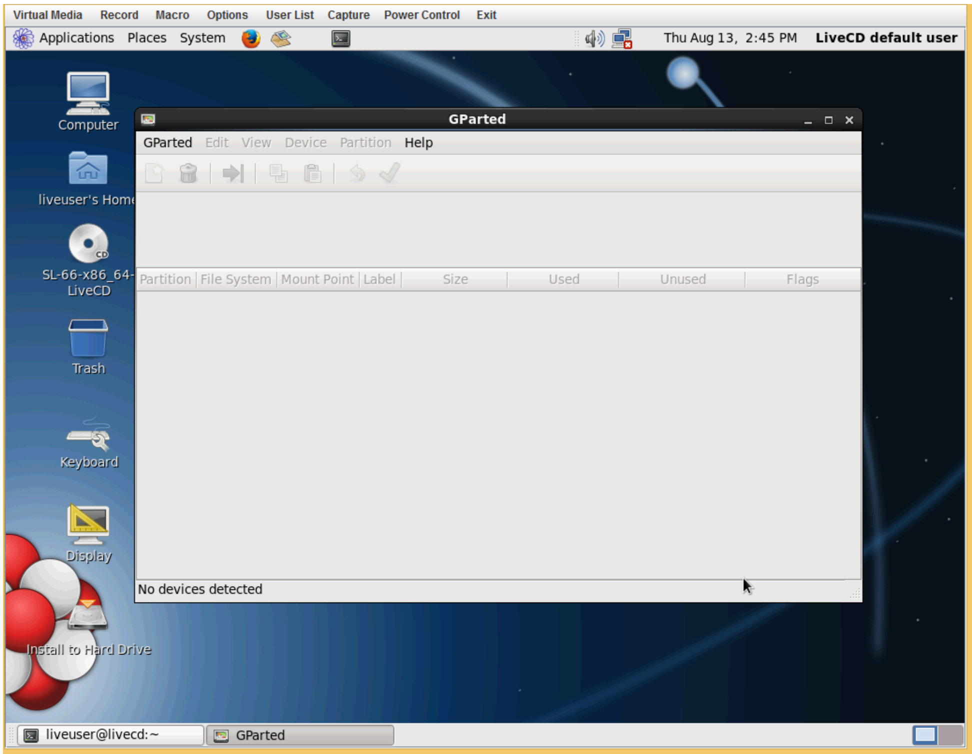
```
SAS2X28 (12), Connector: Port 0 - 3
Slot: 0, SATA, HDD, 1.818 TB , Unconfigured Good
Slot: 1, SATA, HDD, 1.818 TB , Unconfigured Good
Slot: 2, SATA, HDD, 1.818 TB , Unconfigured Good
Slot: 3, SATA, HDD, 1.818 TB , Unconfigured Good
Slot: 4, SATA, HDD, 1.818 TB , Unconfigured Good
Slot: 5, SATA, HDD, 1.818 TB , Unconfigured Good
Slot: 6, SATA, HDD, 1.818 TB , Unconfigured Good
Slot: 7, SATA, HDD, 1.818 TB , Unconfigured Good
Slot: 8, SATA, HDD, 1.818 TB , Unconfigured Good
Slot: 9, SATA, HDD, 1.818 TB , Unconfigured Good
Slot: 10, SATA, HDD, 1.818 TB , Unconfigured Good
Slot: 11, SATA, HDD, 1.818 TB , Unconfigured Good
```

Figure 8: WebBIOS RAID setup with mouse locked on top right corner.

WebBios only supports PS/2 mouse. IPMI provides only USB mouse emulation over iKVM. Thus mouse will always stay in the top left corner. So we can not setup RAID from WebBios over IPMI. Lets look at alternatives!

We could try to access it over LiveCD! After installing a LiveCD of Scientific Linux 6.6 we found out it has no internet connection… We tried to configure it but on Scientific Linux `gedit`, `vi` and others crush out of the box with I/O and segmentation errors respectively not being able to start network file creation. And GParted cannot see anything obviously.

[3] https://www.youtube.com/watch?v=woo_3PywYE0



No devices detected

Figure 9: LiveCD GParted can not see hard drives.

So our last resort will be a Preboot CLI " **MegaPCLI SAS RAID Management Tool**" over IPMI iKVM. Tool cannot scroll its output and nearly always output is larger than screen size thus iKVM video recorder is helping us to inspect what happened on command call. Also Preboot CLS is not a Linux thus traditional information pipelining and filtering will not work.

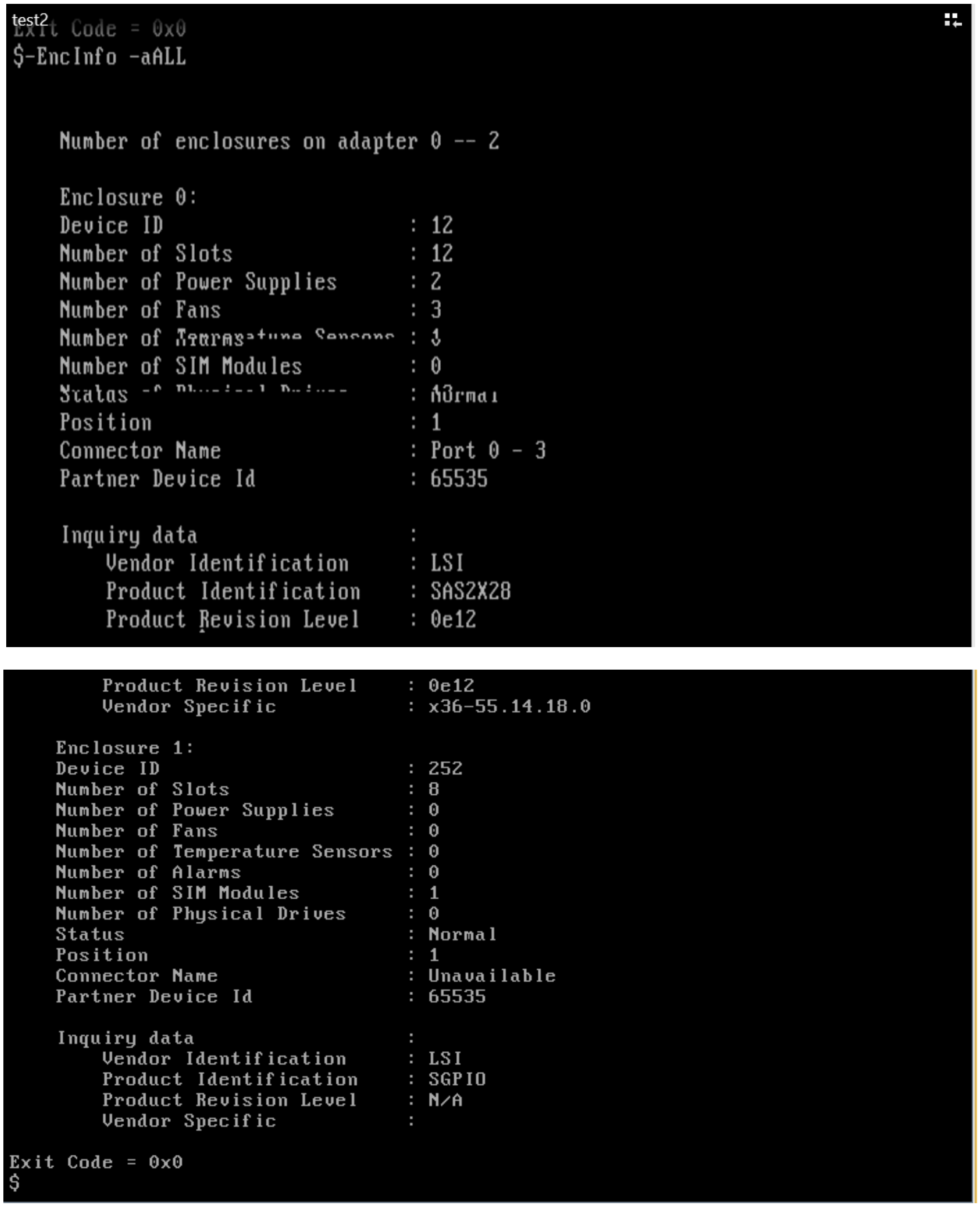
So first call would be `-h -aALL` that would print out a lot of information that is correlating with many other **MegaCLI** documents. For more introductory information on MegaCLI, MegaPCLI and its differences (Chapter 5) inspect Offical Manulal[4] and numerous blog posts like this one[5]. Note that sometimes we also used WebBIOS video tutorial for reference on how to read this help for example to correlate help line:

```
-CfgLDAdd -RX[E0:S0,E1:S1,...] [WT | WB] [NORA | RA] [Direct |
Cached] [CachedBadBBU|NoCachedBadBBU] [-szXXX [-szYYY ...]] [-strpszM]
[-Hsp[E0:S0,...]] [-AfterLdX] | -Force [FDE|CtrlBased] [-Cache] [-enblPI
-val] -aN
```

With its meaning

Now lets get information on enclosures: `-EncInfo -aALL`

[4] http://www.cisco.com/c/dam/en/us/td/docs/unified_computing/ucs/3rd-party/lsi/mrsas/userguide/LSI_MR_SAS_SW_UG.pdf
[5] http://artipc10.vub.ac.be/wordpress/2011/09/12/megacli-useful-commands/



```
Exit Code = 0x0
$-EncInfo -aALL


    Number of enclosures on adapter 0 -- 2

    Enclosure 0:
    Device ID                     : 12
    Number of Slots               : 12
    Number of Power Supplies      : 2
    Number of Fans                : 3
    Number of Temperature Sensors : 1
    Number of SIM Modules         : 0
    Status of Physical Drives     : Normal
    Position                      : 1
    Connector Name                : Port 0 - 3
    Partner Device Id             : 65535

    Inquiry data                  :
        Vendor Identification     : LSI
        Product Identification    : SAS2X28
        Product Revision Level    : 0e12
```

```
        Product Revision Level    : 0e12
        Vendor Specific           : x36-55.14.18.0

    Enclosure 1:
    Device ID                     : 252
    Number of Slots               : 8
    Number of Power Supplies      : 0
    Number of Fans                : 0
    Number of Temperature Sensors : 0
    Number of Alarms              : 0
    Number of SIM Modules         : 1
    Number of Physical Drives     : 0
    Status                        : Normal
    Position                      : 1
    Connector Name                : Unavailable
    Partner Device Id             : 65535

    Inquiry data                  :
        Vendor Identification     : LSI
        Product Identification    : SGPIO
        Product Revision Level    : N/A
        Vendor Specific           :

Exit Code = 0x0
$
```

Figure 10: MegaPCLI help recovered from video.

We can now get info on individual discs using `PDInfo -PhysDrv [12:1] -aALL` where 12 means device id and 1 is a slot number (we have 12 slots, numbering starts from 0).

For ZFS we want to get 12 virtual drive groups with 1 hard drive per group.

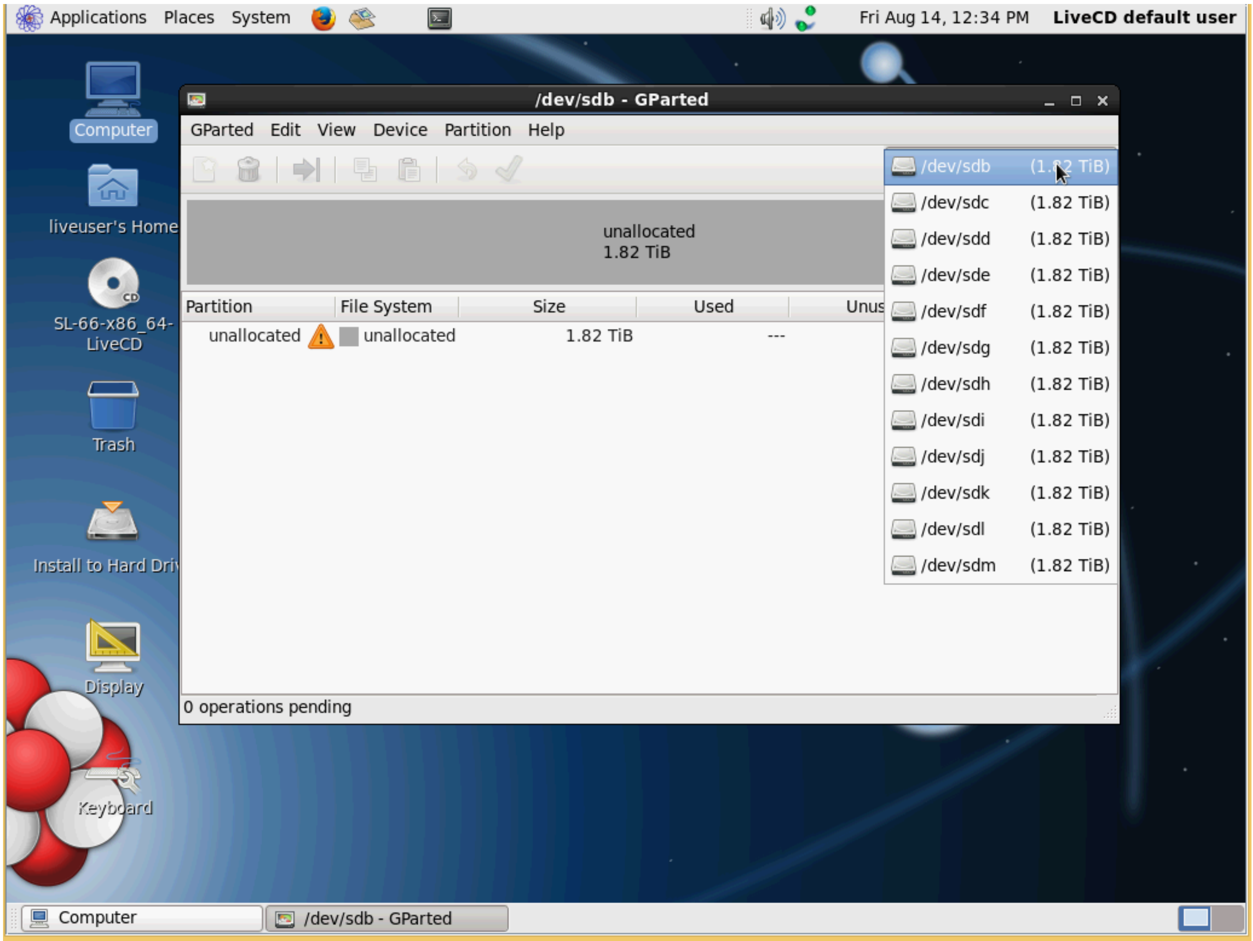
Following this manuals on this topic from University of California[6], Davis and University of Cambridge[7] we export all MegaRAID drives for Linux Software RAID

```
-CfgEachDskRaid0 WB adra cached -a0
```

[6] http://www.maths.cam.ac.uk/computing/docs/public/megacli_raid_lsi.html

[7] https://wiki.cse.ucdavis.edu/support:general:megacli

And get nice view in GParted when LiveCD is installed



Partition	File System	Size	Used	Unus
unallocated ⚠	▨ unallocated	1.82 TiB	---	

/dev/sdb	(1.82 TiB)
/dev/sdc	(1.82 TiB)
/dev/sdd	(1.82 TiB)
/dev/sde	(1.82 TiB)
/dev/sdf	(1.82 TiB)
/dev/sdg	(1.82 TiB)
/dev/sdh	(1.82 TiB)
/dev/sdi	(1.82 TiB)
/dev/sdj	(1.82 TiB)
/dev/sdk	(1.82 TiB)
/dev/sdl	(1.82 TiB)
/dev/sdm	(1.82 TiB)

0 operations pending

Figure 11: LiveCD GParted view after MegaPCLI call.

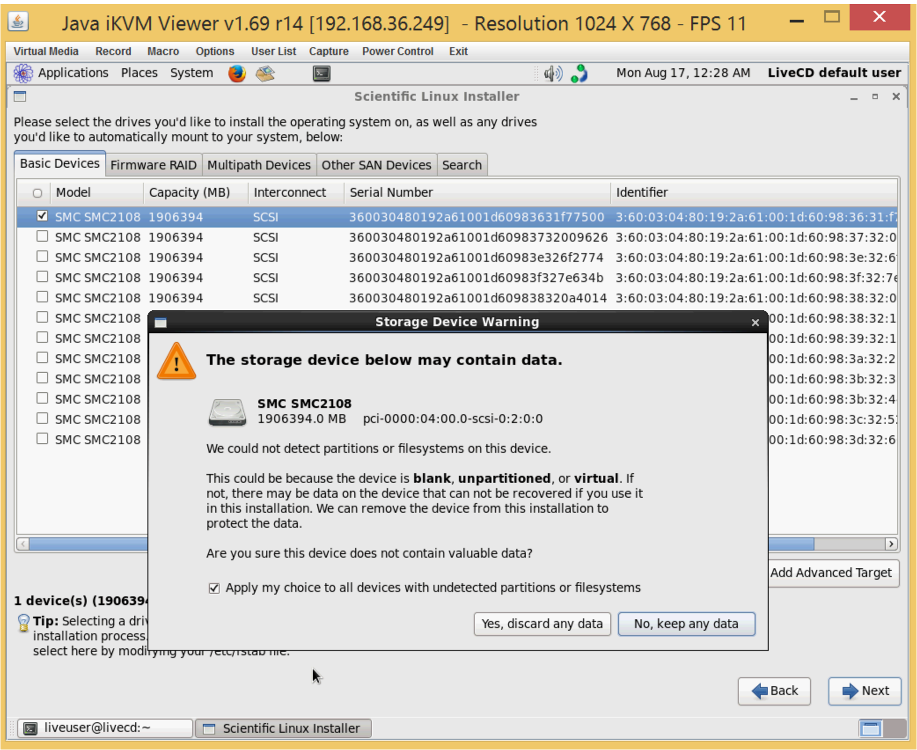
OS Installation

When we can see hard drives from LiveCD live gets much simpler! We will be mainly following this[8] blog post for OS installation and ZFS configuration.

Let’s call Install to Hard drive icon on LiveCD desktop.

We will be installing on one hard drive

[8] https://rudd-o.com/linux-and-free-software/installing-fedora-on-top-of-zfs


Java iKVM Viewer v1.69 r14 [192.168.36.249] - Resolution 1024 X 768 - FPS 11
Virtual Media Record Macro Options User List Capture Power Control Exit
Applications Places System
Mon Aug 17, 12:28 AM LiveCD default user
Scientific Linux Installer
Basic Devices Firmware RAID Multipath Devices Other SAN Devices Search
Add Advanced Target
Yes, discard any data
No, keep any data
Back
Next
liveuser@livecd:~
Scientific Linux Installer

Please select the drives you'd like to install the operating system on, as well as any drives you'd like to automatically mount to your system, below:

○	Model	Capacity (MB)	Interconnect	Serial Number	Identifier
☑	SMC SMC2108	1906394	SCSI	360030480192a61001d60983631f77500	3:60:03:04:80:19:2a:61:00:1d:60:98:36:31:f
□	SMC SMC2108	1906394	SCSI	360030480192a61001d60983732009626	3:60:03:04:80:19:2a:61:00:1d:60:98:37:32:0
□	SMC SMC2108	1906394	SCSI	360030480192a61001d60983e326f2774	3:60:03:04:80:19:2a:61:00:1d:60:98:3e:32:6
□	SMC SMC2108	1906394	SCSI	360030480192a61001d60983f327e634b	3:60:03:04:80:19:2a:61:00:1d:60:98:3f:32:7
□	SMC SMC2108	1906394	SCSI	360030480192a61001d609838320a4014	3:60:03:04:80:19:2a:61:00:1d:60:98:38:32:0
□	SMC SMC2108				00:1d:60:98:38:32:1
□	SMC SMC2108				00:1d:60:98:39:32:1
□	SMC SMC2108				00:1d:60:98:3a:32:2
□	SMC SMC2108				00:1d:60:98:3b:32:3
□	SMC SMC2108				00:1d:60:98:3b:32:4
□	SMC SMC2108				00:1d:60:98:3c:32:5
□	SMC SMC2108				00:1d:60:98:3d:32:6

1 device(s) (190639

Tip: Selecting a dri installation process. select here by modifying your /etc/fstab file.

Storage Device Warning

The storage device below may contain data.

SMC SMC2108
1906394.0 MB pci-0000:04:00.0-scsi-0:2:0:0

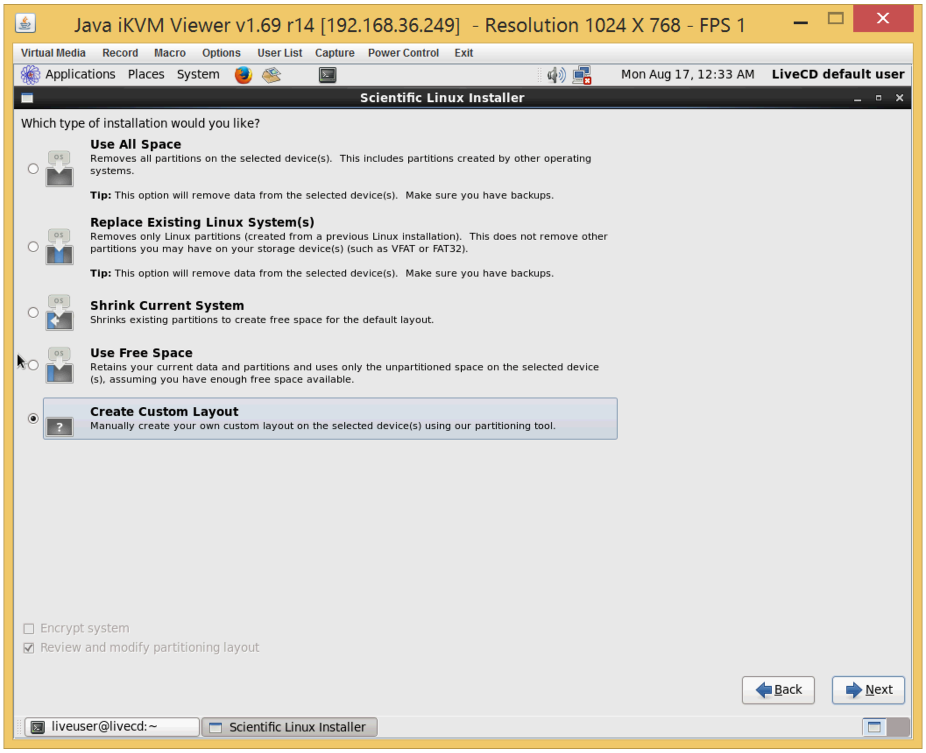
We could not detect partitions or filesystems on this device.

This could be because the device is **blank**, **unpartitioned**, or **virtual**. If not, there may be data on the device that can not be recovered if you use it in this installation. We can remove the device from this installation to protect the data.

Are you sure this device does not contain valuable data?

☑ Apply my choice to all devices with undetected partitions or filesystems

Figure 12: LiveCD Installation step.


Java iKVM Viewer v1.69 r14 [192.168.36.249] - Resolution 1024 X 768 - FPS 1
Virtual Media Record Macro Options User List Capture Power Control Exit
Applications Places System
Mon Aug 17, 12:33 AM LiveCD default user
Scientific Linux Installer
Back
Next
liveuser@livecd:~
Scientific Linux Installer

Which type of installation would you like?

○ **Use All Space**
Removes all partitions on the selected device(s). This includes partitions created by other operating systems.

Tip: This option will remove data from the selected device(s). Make sure you have backups.

○ **Replace Existing Linux System(s)**
Removes only Linux partitions (created from a previous Linux installation). This does not remove other partitions you may have on your storage device(s) (such as VFAT or FAT32).

Tip: This option will remove data from the selected device(s). Make sure you have backups.

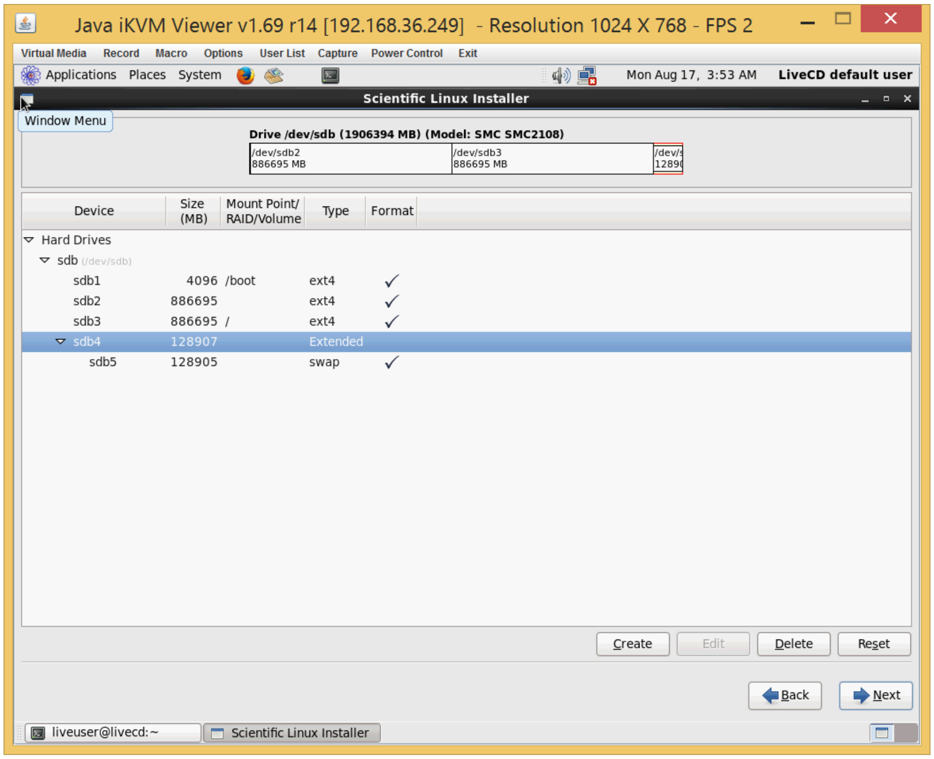
○ **Shrink Current System**
Shrinks existing partitions to create free space for the default layout.

○ **Use Free Space**
Retains your current data and partitions and uses only the unpartitioned space on the selected device (s), assuming you have enough free space available.

● **Create Custom Layout**
Manually create your own custom layout on the selected device(s) using our partitioning tool.

□ Encrypt system
☑ Review and modify partitioning layout

Figure 13: LiveCD Installation step.

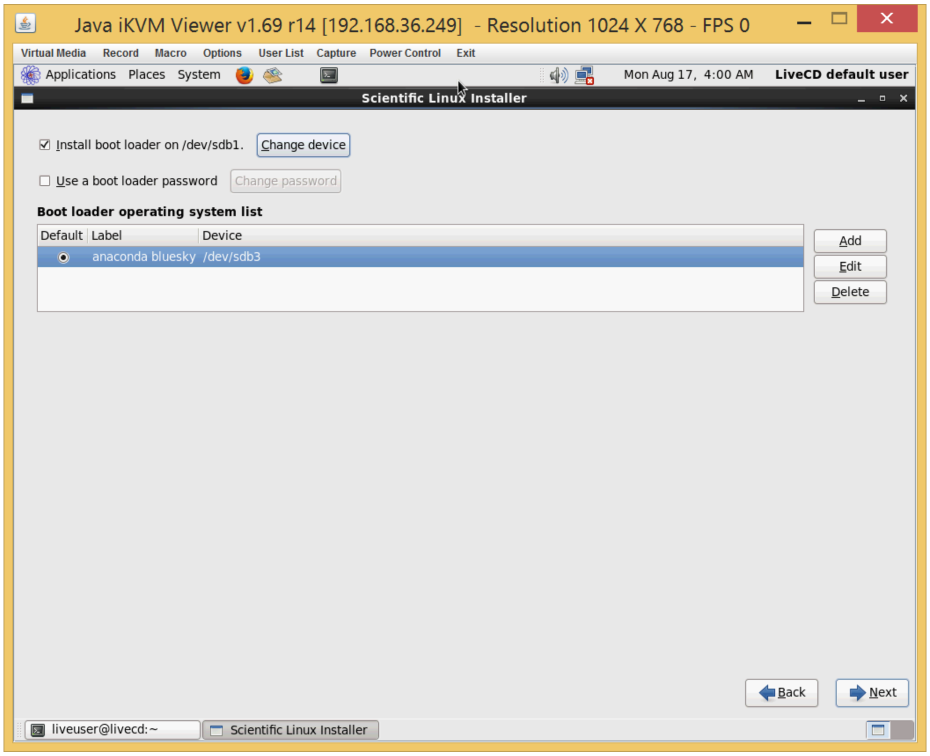


Drive /dev/sdb (1906394 MB) (Model: SMC SMC2108)

/dev/sdb2 886695 MB | /dev/sdb3 886695 MB | /dev/s 1289

Device	Size (MB)	Mount Point/ RAID/Volume	Type	Format
Hard Drives				
sdb (/dev/sdb)				
sdb1	4096	/boot	ext4	✓
sdb2	886695		ext4	✓
sdb3	886695	/	ext4	✓
sdb4	128907		Extended	
sdb5	128905		swap	✓

Figure 14: LiveCD Installation step.



☑ Install boot loader on /dev/sdb1.

☐ Use a boot loader password

Boot loader operating system list

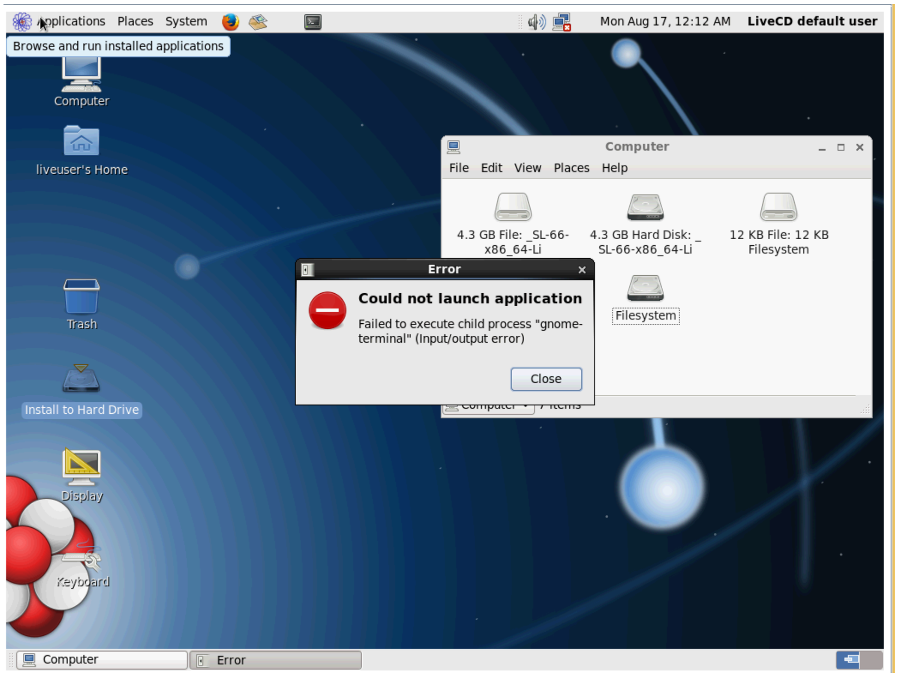
Default	Label	Device
●	anaconda bluesky	/dev/sdb3

Figure 15: LiveCD Installation step.

After LiveCD finishes installation, we stop virtual media device and perform power reset.

LiveCD Errors

Mainly errors occur when IPMI virtual media that mounts LiveCD Drive over SMB lost connection. Errors may include failed application launch, system initialization fails.



Could not launch application

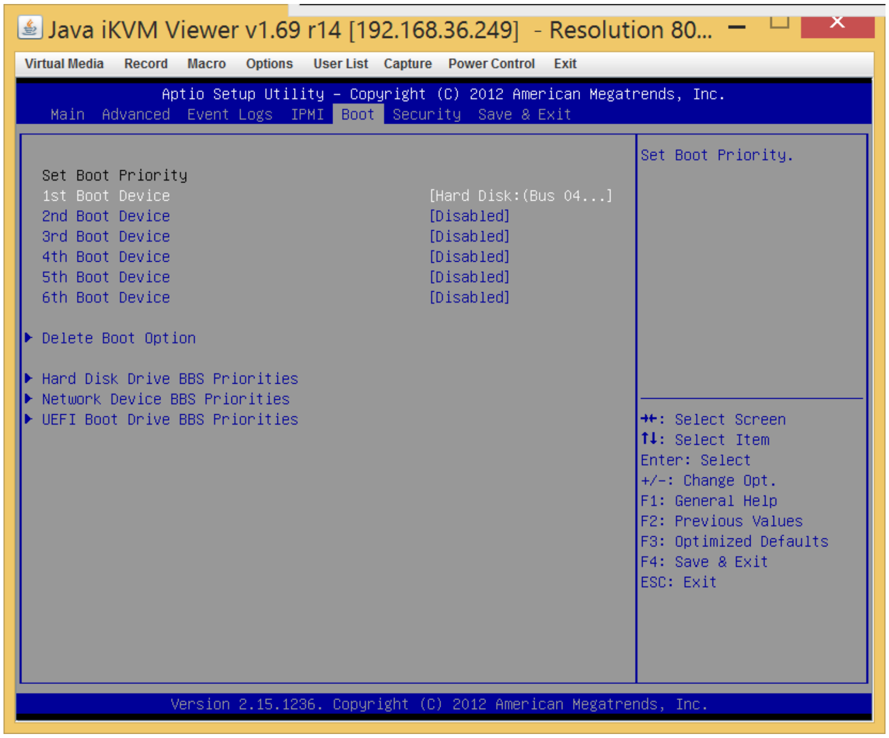
Failed to execute child process "gnome-terminal" (Input/output error)

Figure 16:Errors due to IPMI Virtual Media connection loss

In such cases virtual media device stop+start+Pover Reset helps

```
vmwa dev2stop
vmwa dev2iso \\159.93.33.131\share\sl66.iso
Power Reset button on iKVM top menu bar
```

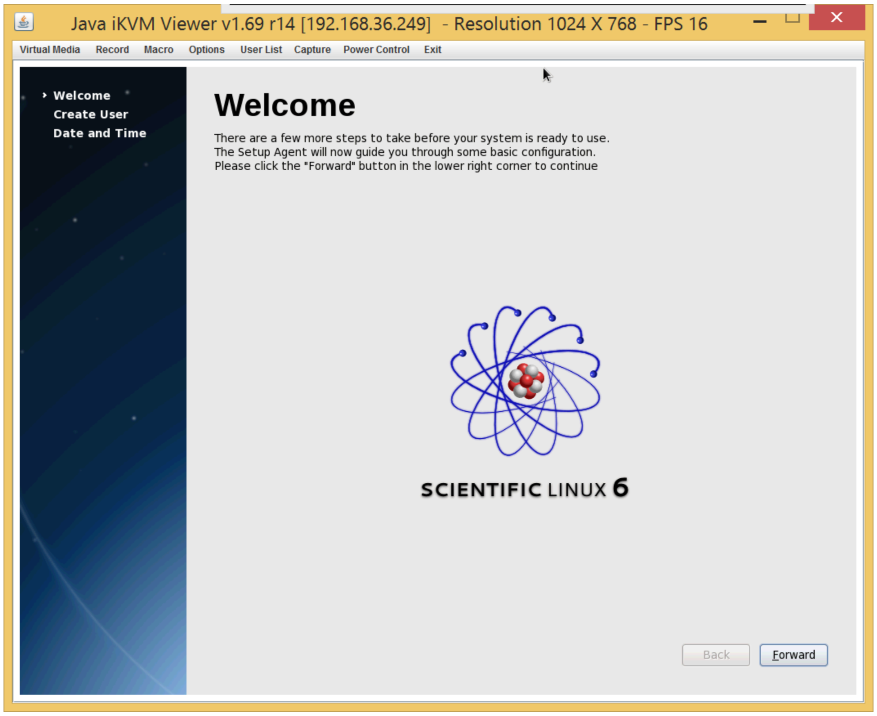
Then we open BIOS and select another boot device



```
Aptio Setup Utility – Copyright (C) 2012 American Megatrends, Inc.
  Main  Advanced  Event Logs  IPMI  Boot  Security  Save & Exit

  Set Boot Priority                                        Set Boot Priority.
  1st Boot Device                    [Hard Disk:(Bus 04...]
  2nd Boot Device                    [Disabled]
  3rd Boot Device                    [Disabled]
  4th Boot Device                    [Disabled]
  5th Boot Device                    [Disabled]
  6th Boot Device                    [Disabled]

▶ Delete Boot Option

▶ Hard Disk Drive BBS Priorities
▶ Network Device BBS Priorities                            →←: Select Screen
▶ UEFI Boot Drive BBS Priorities                           ↑↓: Select Item
                                                           Enter: Select
                                                           +/-: Change Opt.
                                                           F1: General Help
                                                           F2: Previous Values
                                                           F3: Optimized Defaults
                                                           F4: Save & Exit
                                                           ESC: Exit

          Version 2.15.1236. Copyright (C) 2012 American Megatrends, Inc.
```

Figure 17:New boot device is selected.

We do not see out drive partitions here – only hard drive system abstraction, select it, save and reboot.



Welcome

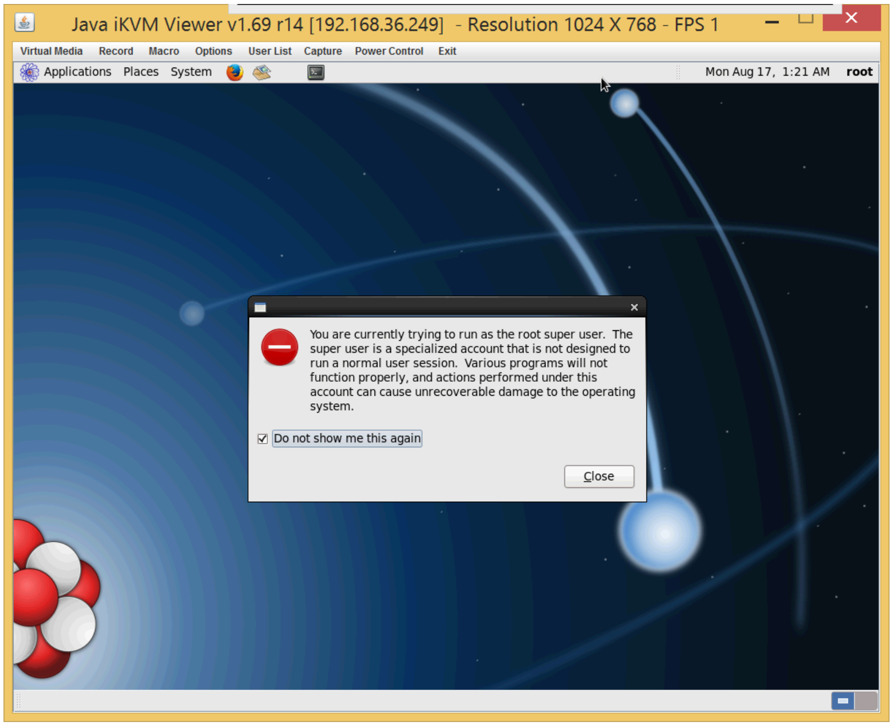
There are a few more steps to take before your system is ready to use.
The Setup Agent will now guide you through some basic configuration.
Please click the "Forward" button in the lower right corner to continue

Figure 18: Scientific Linux welcome screen.

System will load.

Set up the network and SSH

Lets login as superuser



You are currently trying to run as the root super user. The super user is a specialized account that is not designed to run a normal user session. Various programs will not function properly, and actions performed under this account can cause unrecoverable damage to the operating system.

☑ Do not show me this again

Close

Figure 19: We logged into root on installed system.

We create `ifcfg-eth0`

```
gedit /etc/sysconfig/network-scripts/ifcfg-eth0
```

and fill it with public network settings

```
DEVICE="eth0"
BOOTPROTO="static"
BROADCAST="159.93.36.255"
DNS1="159.93.14.7"
DNS2="159.93.17.7"
GATEWAY="159.93.36.1"
IPADDR="159.93.36.249"
NETMASK="255.255.255.0"
NM_CONTROLLED="yes"
ONBOOT="yes"
TYPE="Ethernet"
```

And same for local connection `ifcfg-eth1`

```
DEVICE="eth1"
BOOTPROTO="static"
IPADDR="10.1.36.1"
NETMASK="255.255.255.0"
NM_CONTROLLED="yes"
ONBOOT="yes"
TYPE="Ethernet"
```

Having networks working we shall be able to browse internet and install applications. Lets install `lshw` and add `HWADDR` to our configuration:

```
Yum -y install lshw
lshw -class network
```

Find serial numbers from networks with logical names that correlate to eth0 and eth1. Add `HWADDR` to network configurations like

```
HWADDR="0c:c4:7a:31:1b:20"
```

To both *`ifcfg-eth0`* and *`ifcfg-eth1`* respectively.

We shall configure `/etc/hosts.allow`

```
ALL: 10.1.36.0/24
ALL: 159.93.0.0/16
```

Now lets set up SSH server.

```
yum -y install openssh-server openssh-clients
chkconfig sshd on
service sshd start
```

And open tcp port 22. Edit `/etc/sysconfig/iptables` adding

```
-A INPUT -s 159.93.0.0/16 -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -s 10.1.36.0/24 -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
```

And calling

```
service iptables restart
```

Now we can connect over SSH and be free from iKVM as default window to our server.

ZFS Configuration

For ZFS installation we will mainly follow this[9] instruction. We will be using SSH to perform all next operations.

Prepare Operating system

Lets update kernel and reboot

```
yum update kernel
yum –y install nano
yum install htop flex bison
```

Disable SELinux: edit `/etc/selinux/config`

```
SELINUX=disabled
```

And call

```
setenforce 0
```

Lets prepare tools

```
yum install -y git patch kernel-devel gcc zlib-devel libuuid-devel libtool automake autoconf
```

[9] https://rudd-o.com/linux-and-free-software/installing-fedora-on-top-of-zfs

and prepare development tools (following instructions for by Linux@CERN[10])

```
wget -O /etc/yum.repos.d/slc6-devtoolset.repo http://linuxsoft.cern.ch/cern/devtoolset/slc6-devtoolset.repo
yum install devtoolset-2
scl enable devtoolset-2 bash
```

Install OFED IB

InfiniBand Installation: get and unpack

```
wget https://openfabrics.org/downloads/OFED/ofed-3.18/OFED-3.18-rc3.tgz
```

Install packages

```
yum install libnl-devel libudev-devel libnl-devel
yum install tcl tk tcl-devel glib2-devel
```

Build and install Open Fabrics Enterprise Distribution (OFED) libraries

```
./install.pl --all --without-libfabric --without-libfabric-devel --without-libfabric-debuginfo --without-fabtests-debuginfo  --without-fabtests --without-libiwpm
```

At the end installation will output something like

```
Device (15b3:1003):
        81:00.0 Network controller: Mellanox Technologies MT27500 Family [ConnectX-3]
        Link Width: 8x
        PCI Link Speed: Unknown

Installation finished successfully.
```

Test

Install ZFS

```
sudo yum localinstall --nogpgcheck https://download.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
sudo yum localinstall --nogpgcheck http://archive.zfsonlinux.org/epel/zfs-release.el6.noarch.rpm
sudo yum install -y kernel-devel zfs
sudo yum install spl
chkconfig zfs on
zpool status
```

We can now create ZFS Partitions.

```
zpool create -f zfs-data /dev/sdb
chmod 755 /zfs-data
zfs list
```

Yet our OS is installed on ext4! We would like to move it onto ZFS. Just moving files will not be enough – we need to make sure our boot loader supports our file system. GRUB 0.97 () is our default bootloader. It does not support ZFS, not developed any more and is considered Legacy. We want to update GRUB to version 2.

Download and build Grub 2.

Thus we shall download it from here[11] and compile using this instructions[12].

[10] http://linux.web.cern.ch/linux/devtoolset/#dts30
[11] http://www.gnu.org/software/grub/grub-download.html
[12] http://www.linuxfromscratch.org/lfs/view/development/chapter06/grub.html

```
cd /usr/src/spl-0.6.4.2/
./configure --prefix=/usr/src/build/
make -j24
make install
cd ../zfs-0.6.4.2/
./autogen.sh
./configure --with-spl=/usr/src/spl-0.6.4.2/ --prefix=/usr/src/build/
make -j24
make install
cd ../
git clone git://git.savannah.gnu.org/grub.git
cd grub/
./autogen.sh
./configure --prefix=/usr                      --sbindir=/sbin         --sysconfdir=/etc
--disable-grub-emu-usb             --disable-efiemu   --disable-werror --enable-libzfs  --with-
platform=efi LDFLAGS=-L/usr/src/build/lib/ CPPFLAGS=-I/usr/src/build/include/
make -j24
make install
```

Now we can check if zfs is supported

```
[root@localhost grub]# grub-probe  /zfs-data
zfs
```

Configure Grub 2

We will be mostly following GRUB2 Migration instructions[13] (note that `/boot` is installed on `/dev/sda1`)

```
[root@localhost src]# grub-install --grub-setup=/bin/true /dev/sda
Installing for i386-pc platform.
Installation finished. No error reported.
[root@localhost src]# grub-mkconfig -o /boot/grub/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-2.6.32-573.3.1.el6.x86_64
Found initrd image: /boot/initramfs-2.6.32-573.3.1.el6.x86_64.img
Found linux image: /boot/vmlinuz-2.6.32-504.el6.x86_64
Found initrd image: /boot/initramfs-2.6.32-504.el6.x86_64.img
done
```

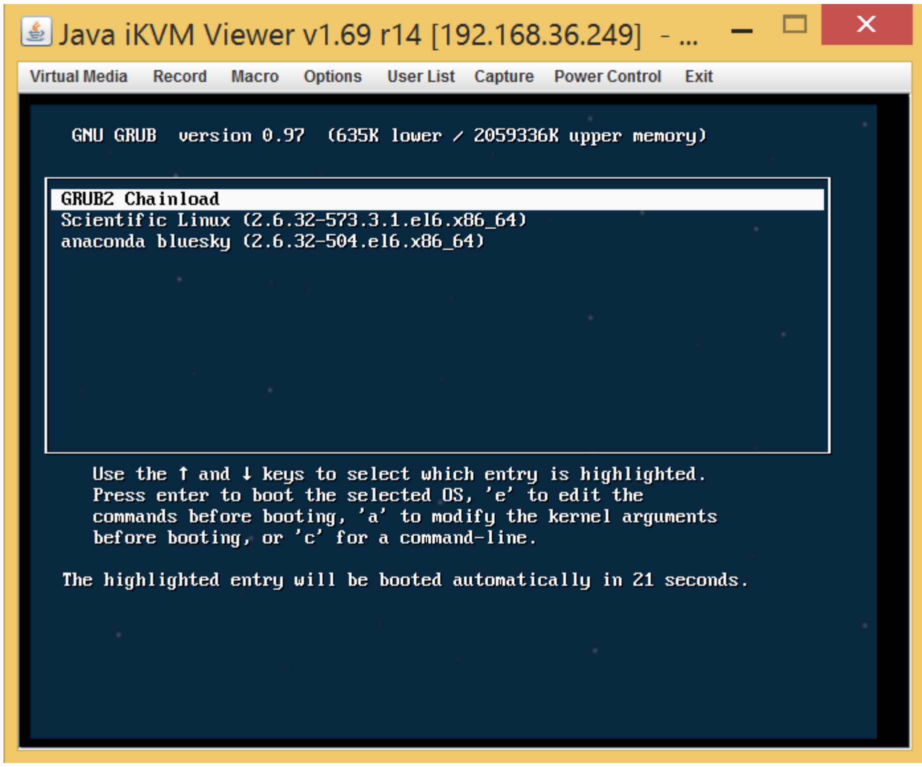
Now we will try to chain load GRUB2 from legacy GRUB. We will edit `/boot/grub/grub.conf`

```
# grub.conf generated by anaconda
# Note that you do not have to rerun grub after making changes to this file
# NOTICE:  You have a /boot partition.  This means that
#          all kernel and initrd paths are relative to /boot/, eg.
#          root (hd0,0)
#          kernel /vmlinuz-version ro root=/dev/sdb3
#          initrd /initrd-[generic-]version.img
#boot=/dev/sdb1
default=0
timeout=30
splashimage=(hd0,0)/grub/splash.xpm.gz
```

[13] https://wiki.gentoo.org/wiki/GRUB2_Migration

```
# hiddenmenu

title GRUB2 Chainload
root (hd0,0)
kernel /grub/i386-pc/core.img
boot

title Scientific Linux (2.6.32-573.3.1.el6.x86_64)
root (hd0,0)
kernel /vmlinuz-2.6.32-573.3.1.el6.x86_64 ro root=UUID=2c6a38dc-04e7-41c3-ae85-623dbefa2b1e
rd_NO_LUKS rd_NO_LVM LANG=en_US.UTF-8 rd_NO_MD SYSFONT=latarcyrheb-sun16 crashkernel=auto
KEYBOARDTYPE=pc KEYTABLE=us rd_NO_DM rhgb quiet
initrd /initramfs-2.6.32-573.3.1.el6.x86_64.img

title anaconda bluesky (2.6.32-504.el6.x86_64)
root (hd0,0)
kernel /vmlinuz-2.6.32-504.el6.x86_64 ro root=UUID=2c6a38dc-04e7-41c3-ae85-623dbefa2b1e
rd_NO_LUKS rd_NO_LVM LANG=en_US.UTF-8 rd_NO_MD SYSFONT=latarcyrheb-sun16 crashkernel=auto
KEYBOARDTYPE=pc KEYTABLE=us rd_NO_DM rhgb quiet
initrd /initramfs-2.6.32-504.el6.x86_64.img
```

And after reboot we must see



```
  GNU GRUB  version 0.97  (635K lower / 2059336K upper memory)

 GRUB2 Chainload
 Scientific Linux (2.6.32-573.3.1.el6.x86_64)
 anaconda bluesky (2.6.32-504.el6.x86_64)

    Use the ↑ and ↓ keys to select which entry is highlighted.
    Press enter to boot the selected OS, 'e' to edit the
    commands before booting, 'a' to modify the kernel arguments
    before booting, or 'c' for a command-line.

 The highlighted entry will be booted automatically in 21 seconds.
```

Figure 20: GRUB legacy chainloading GRUB2.

And GRUB2 shall chain load our OS.

Now we make GRUB2 default loader.

System launch on Blade

We used a network loading procedure developed at Hybrilit. It is based on traditional network load, modified for fast client OS updates, user work-process safety in case of server node failed.

Hybrilit administrators provided us with need files and instructions involving DHCP, SYSLINUX, TFTP, HTTP configurations, installation images for initd (called nanoramfs) and rootfs (called ramfs).

After services configuration we were able to chroot into image that would be unpaceked into blade memory and install/configure new software and drivers.

Note that for drivers' installation one must have same kernel versions on Server OS and on Blade OS.

Drivers Installation

We installed Mellanox OFED InfiniBand drivers and applications. (Similarly to host server installation)

We installed required packages:

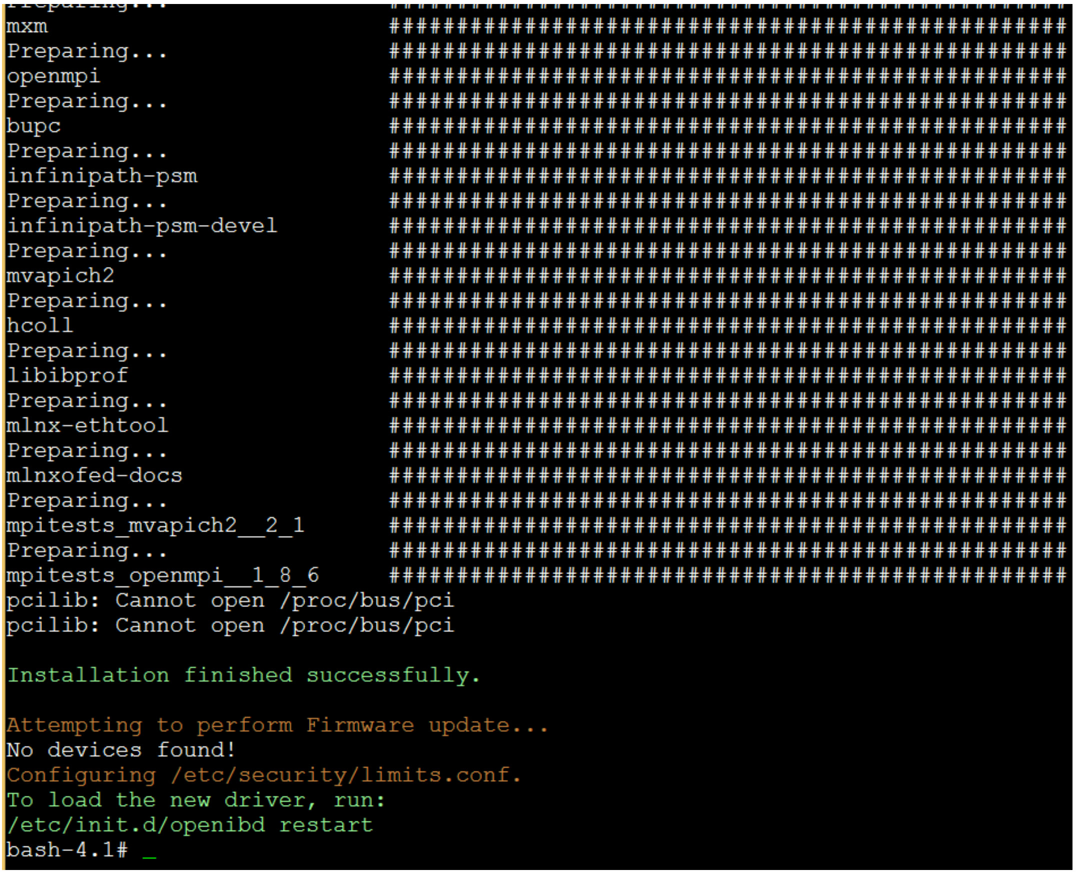
```
yum install -y net-tools mc libnl-devel libudev-devel libnl-devel tcl tk tcl-devel glib2-devel
yum install -y git patch kernel-devel gcc zlib-devel libuuid-devel libtool automake autoconf
yum install redhat-rpm-config gcc-gfortran bison flex
yum install kernel-headers-2.6.32-573.3.1.el6.x86_64
yum install kernel-sources-2.6.32-573.3.1.el6.x86_64
ln -s /usr/src/kernels/2.6.32-573.3.1.el6.x86_64 /lib/modules/2.6.32-573.3.1.el6.x86_64/build
ln -s /usr/src/kernels/2.6.32-573.3.1.el6.x86_64 /lib/modules/2.6.32-573.3.1.el6.x86_64/source
```

Downloaded and unpacked files, created kernel specific drivers

```
wget http://www.mellanox.com/downloads/ofed/MLNX_OFED-3.0-2.0.1/MLNX_OFED_LINUX-3.0-2.0.1-rhel6.6-x86_64.tgz
./mlnx_add_kernel_support.sh  --mlnx_ofed /usr/src/MLNX_OFED_LINUX-3.0-2.0.1-rhel6.5-x86_64/
```

Unpacked them and installed:

```
./mlnxofedinstall -all
```

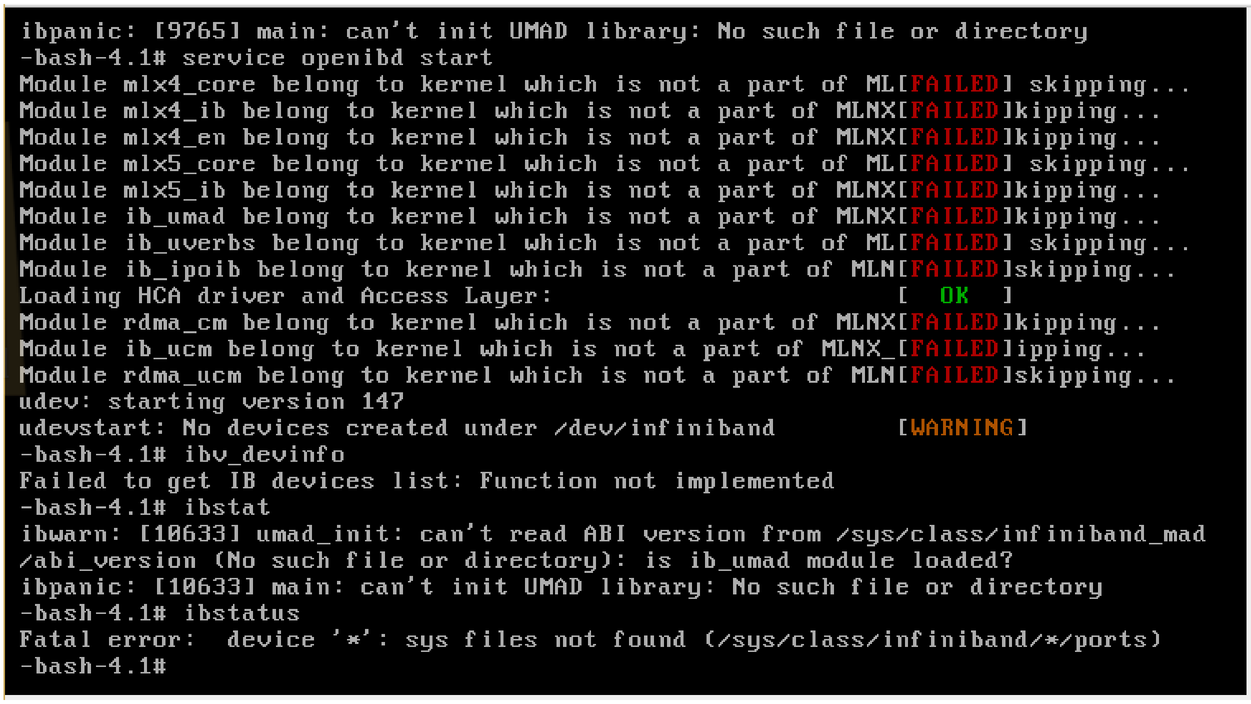


```
Preparing...                ##################################################
mxm                         ##################################################
Preparing...                ##################################################
openmpi                     ##################################################
Preparing...                ##################################################
bupc                        ##################################################
Preparing...                ##################################################
infinipath-psm              ##################################################
Preparing...                ##################################################
infinipath-psm-devel        ##################################################
Preparing...                ##################################################
mvapich2                    ##################################################
Preparing...                ##################################################
hcoll                       ##################################################
Preparing...                ##################################################
libibprof                   ##################################################
Preparing...                ##################################################
mlnx-ethtool                ##################################################
Preparing...                ##################################################
mlnxofed-docs               ##################################################
Preparing...                ##################################################
mpitests_mvapich2__2_1      ##################################################
Preparing...                ##################################################
mpitests_openmpi__1_8_6     ##################################################
pcilib: Cannot open /proc/bus/pci
pcilib: Cannot open /proc/bus/pci

Installation finished successfully.

Attempting to perform Firmware update...
No devices found!
Configuring /etc/security/limits.conf.
To load the new driver, run:
/etc/init.d/openibd restart
bash-4.1# _
```

Figure 21: Mellanox OFED installation success.

IB problems we encountered during installation and testing

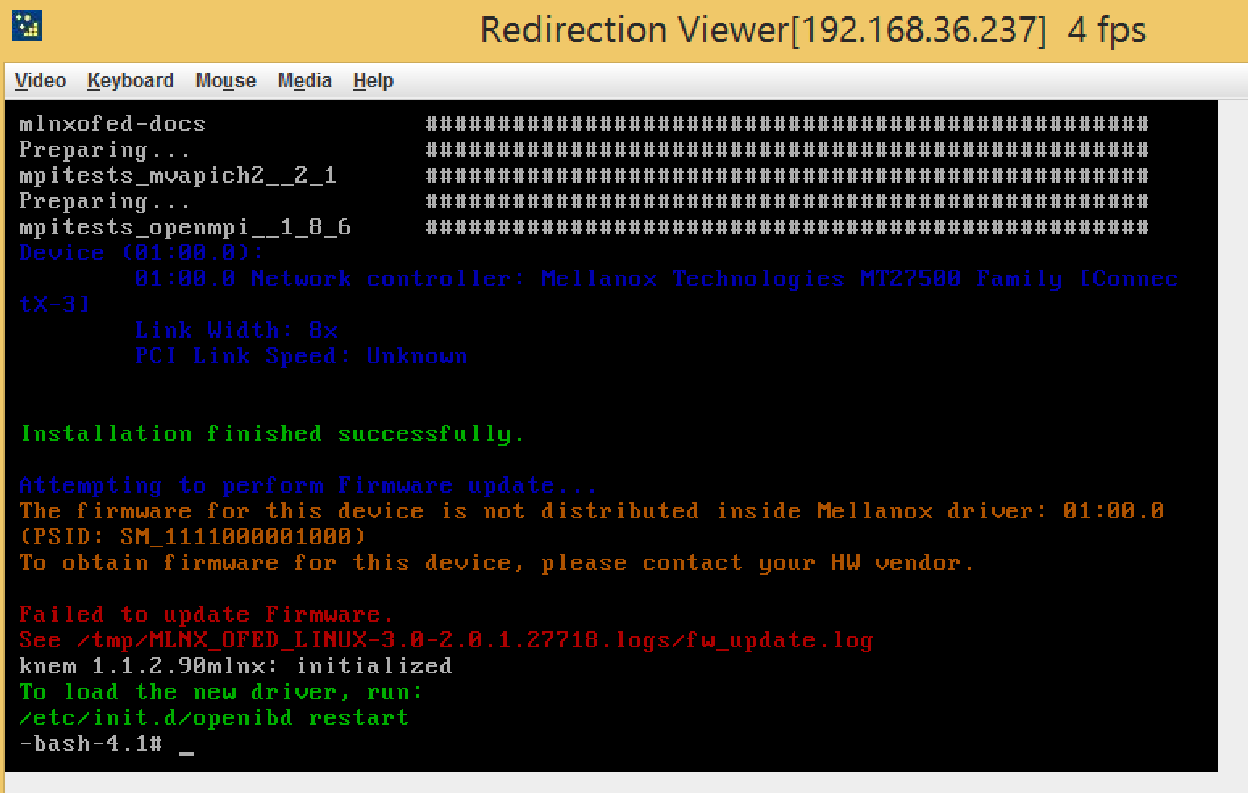
If compiled for one kernel and started on other drivers will fail:



```
ibpanic: [9765] main: can't init UMAD library: No such file or directory
-bash-4.1# service openibd start
Module mlx4_core belong to kernel which is not a part of ML[FAILED] skipping...
Module mlx4_ib belong to kernel which is not a part of MLNX[FAILED]kipping...
Module mlx4_en belong to kernel which is not a part of MLNX[FAILED]kipping...
Module mlx5_core belong to kernel which is not a part of ML[FAILED] skipping...
Module mlx5_ib belong to kernel which is not a part of MLNX[FAILED]kipping...
Module ib_umad belong to kernel which is not a part of MLNX[FAILED]kipping...
Module ib_uverbs belong to kernel which is not a part of ML[FAILED] skipping...
Module ib_ipoib belong to kernel which is not a part of MLN[FAILED]skipping...
Loading HCA driver and Access Layer:                       [  OK  ]
Module rdma_cm belong to kernel which is not a part of MLNX[FAILED]kipping...
Module ib_ucm belong to kernel which is not a part of MLNX_[FAILED]ipping...
Module rdma_ucm belong to kernel which is not a part of MLN[FAILED]skipping...
udev: starting version 147
udevstart: No devices created under /dev/infiniband        [WARNING]
-bash-4.1# ibv_devinfo
Failed to get IB devices list: Function not implemented
-bash-4.1# ibstat
ibwarn: [10633] umad_init: can't read ABI version from /sys/class/infiniband_mad
/abi_version (No such file or directory): is ib_umad module loaded?
ibpanic: [10633] main: can't init UMAD library: No such file or directory
-bash-4.1# ibstatus
Fatal error:  device '*': sys files not found (/sys/class/infiniband/*/ports)
-bash-4.1#
```

Figure 22Kernels mismatch.

We have seen an interesting error when driver was recompiled on Blade

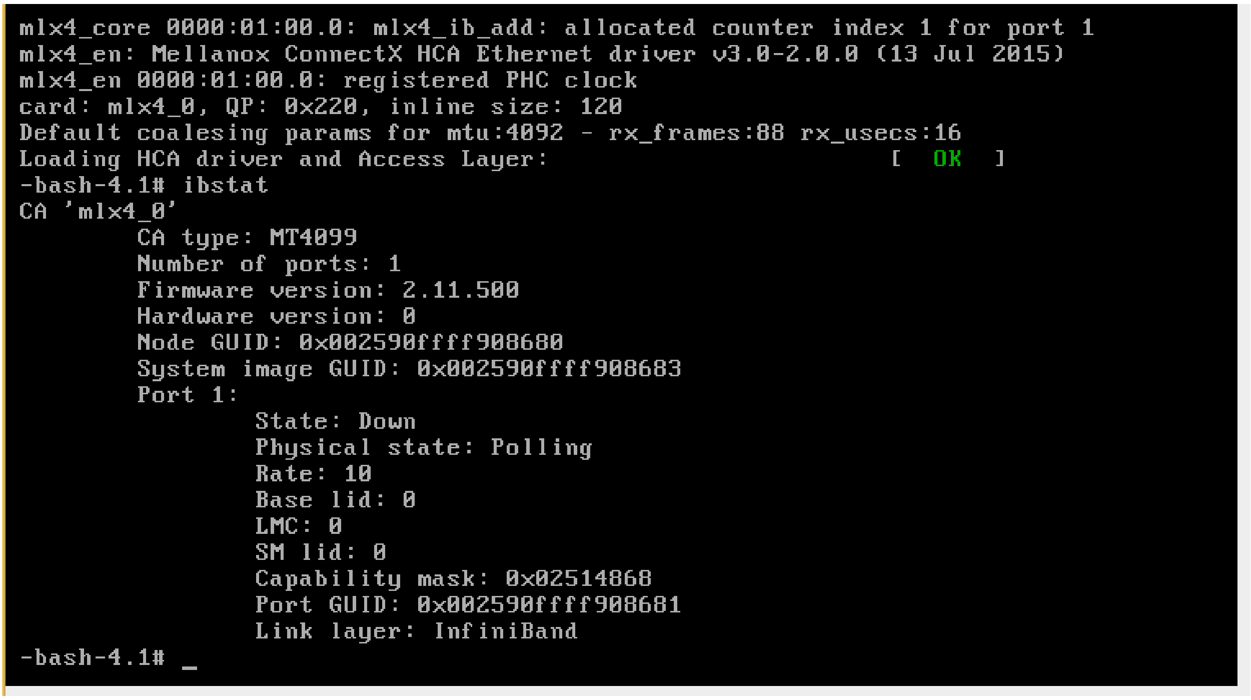


```
mlnxofed-docs                 ##################################################
Preparing...                  ##################################################
mpitests_mvapich2__2_1        ##################################################
Preparing...                  ##################################################
mpitests_openmpi__1_8_6       ##################################################
Device (01:00.0):
        01:00.0 Network controller: Mellanox Technologies MT27500 Family [Connec
tX-3]
        Link Width: 8x
        PCI Link Speed: Unknown


Installation finished successfully.

Attempting to perform Firmware update...
The firmware for this device is not distributed inside Mellanox driver: 01:00.0
(PSID: SM_1111000001000)
To obtain firmware for this device, please contact your HW vendor.

Failed to update Firmware.
See /tmp/MLNX_OFED_LINUX-3.0-2.0.1.27718.logs/fw_update.log
knem 1.1.2.90mlnx: initialized
To load the new driver, run:
/etc/init.d/openibd restart
-bash-4.1# _
```

Figure 23: Problems with on Blade Mellanox OFED drivers recompilation.

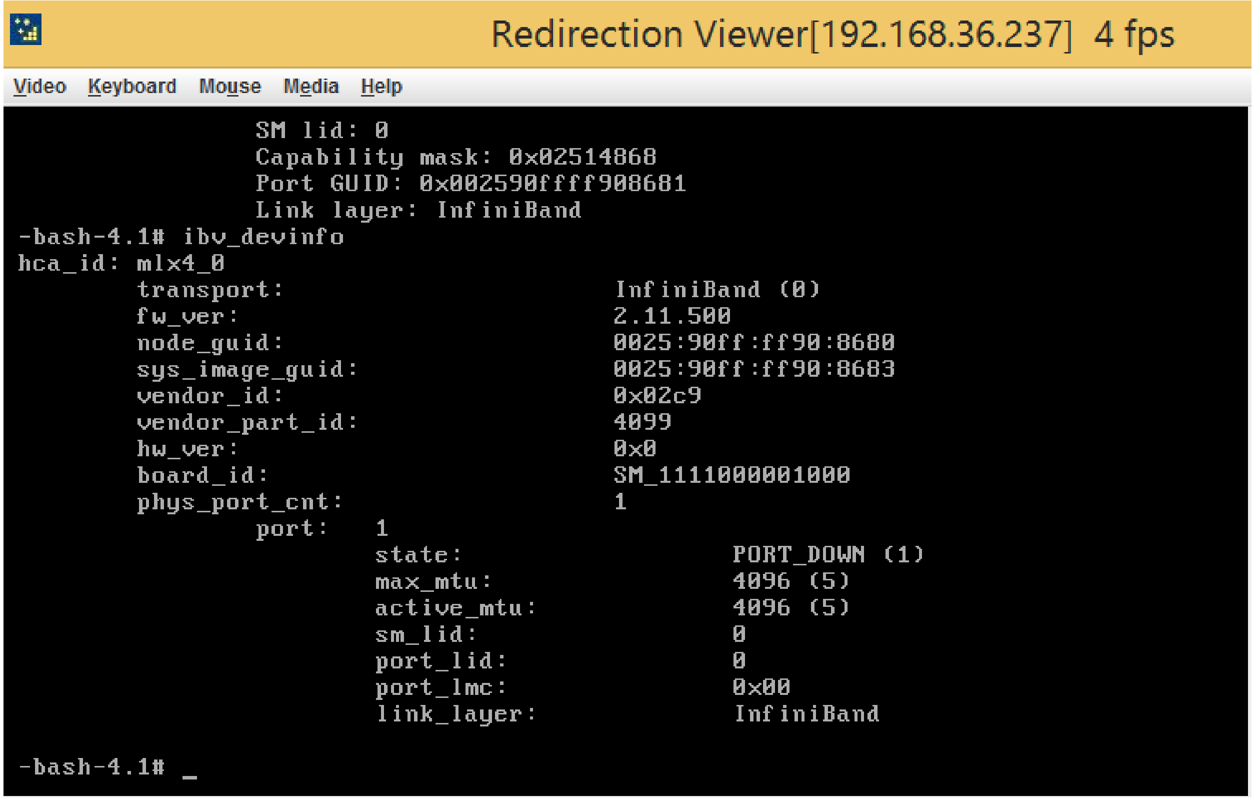
InfiniBand testing

We installed IB on two baldes, devices were found yet links were in DOWN state:



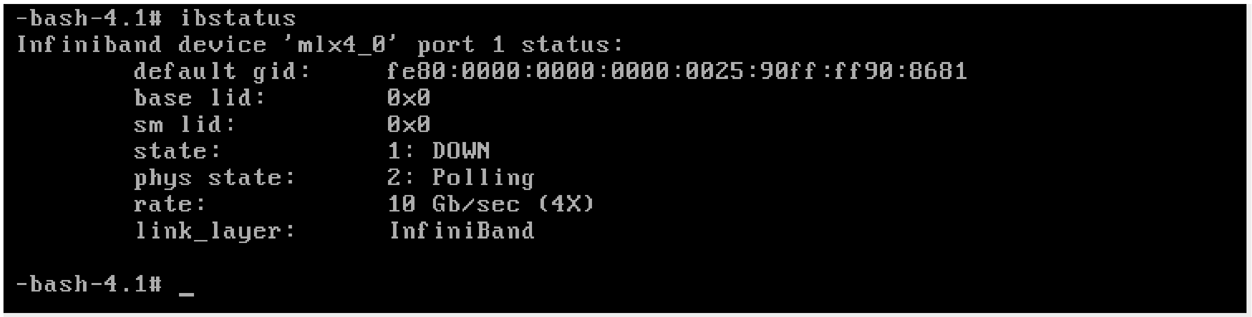
```
mlx4_core 0000:01:00.0: mlx4_ib_add: allocated counter index 1 for port 1
mlx4_en: Mellanox ConnectX HCA Ethernet driver v3.0-2.0.0 (13 Jul 2015)
mlx4_en 0000:01:00.0: registered PHC clock
card: mlx4_0, QP: 0x220, inline size: 120
Default coalesing params for mtu:4092 - rx_frames:88 rx_usecs:16
Loading HCA driver and Access Layer:                         [  OK  ]
-bash-4.1# ibstat
CA 'mlx4_0'
        CA type: MT4099
        Number of ports: 1
        Firmware version: 2.11.500
        Hardware version: 0
        Node GUID: 0x002590ffff908680
        System image GUID: 0x002590ffff908683
        Port 1:
                State: Down
                Physical state: Polling
                Rate: 10
                Base lid: 0
                LMC: 0
                SM lid: 0
                Capability mask: 0x02514868
                Port GUID: 0x002590ffff908681
                Link layer: InfiniBand
-bash-4.1# _
```

Figure 24: ibstat call.



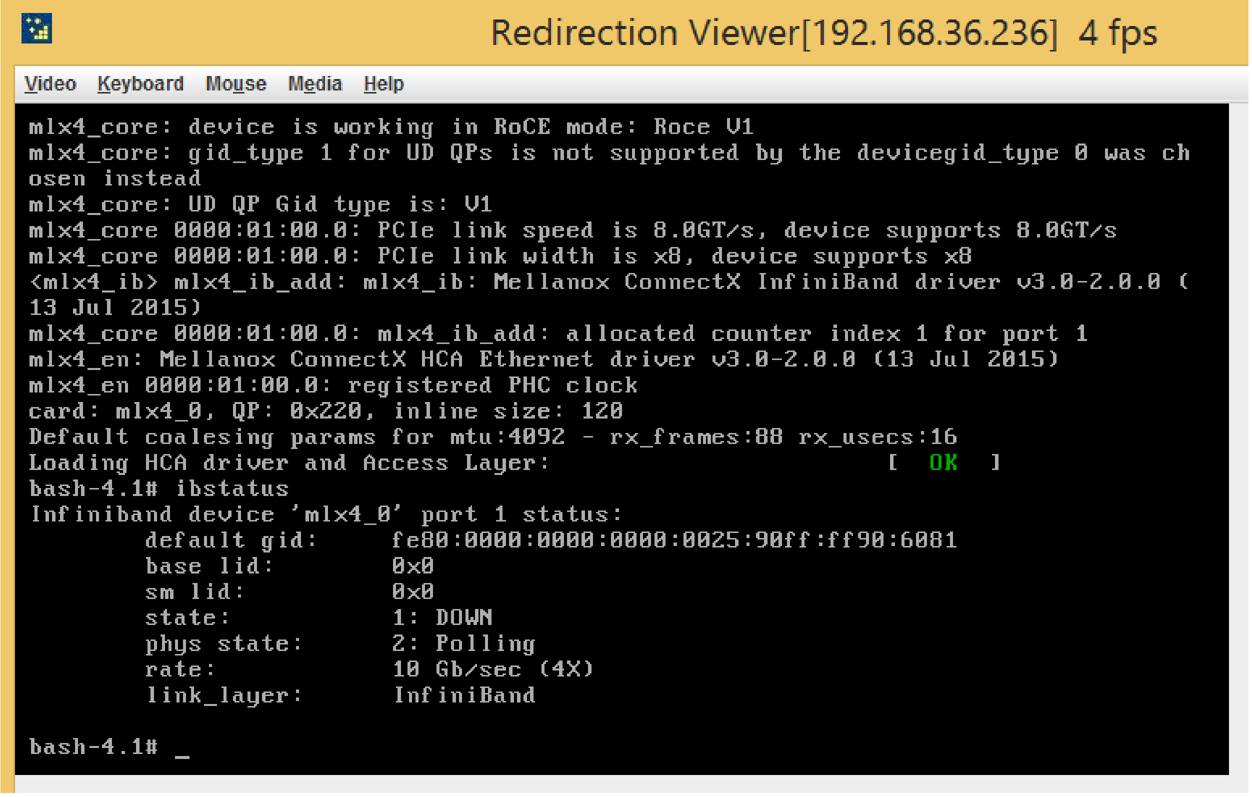
```
                SM lid: 0
                Capability mask: 0x02514868
                Port GUID: 0x002590ffff908681
                Link layer: InfiniBand
-bash-4.1# ibv_devinfo
hca_id: mlx4_0
        transport:                      InfiniBand (0)
        fw_ver:                         2.11.500
        node_guid:                      0025:90ff:ff90:8680
        sys_image_guid:                 0025:90ff:ff90:8683
        vendor_id:                      0x02c9
        vendor_part_id:                 4099
        hw_ver:                         0x0
        board_id:                       SM_1111000001000
        phys_port_cnt:                  1
                port:   1
                        state:                  PORT_DOWN (1)
                        max_mtu:                4096 (5)
                        active_mtu:             4096 (5)
                        sm_lid:                 0
                        port_lid:               0
                        port_lmc:               0x00
                        link_layer:             InfiniBand

-bash-4.1# _
```

Figure 25: ibv_devinfo call.



```
-bash-4.1# ibstatus
Infiniband device 'mlx4_0' port 1 status:
        default gid:     fe80:0000:0000:0000:0025:90ff:ff90:8681
        base lid:        0x0
        sm lid:          0x0
        state:           1: DOWN
        phys state:      2: Polling
        rate:            10 Gb/sec (4X)
        link_layer:      InfiniBand

-bash-4.1# _
```

Figure 26: ibstatus call.

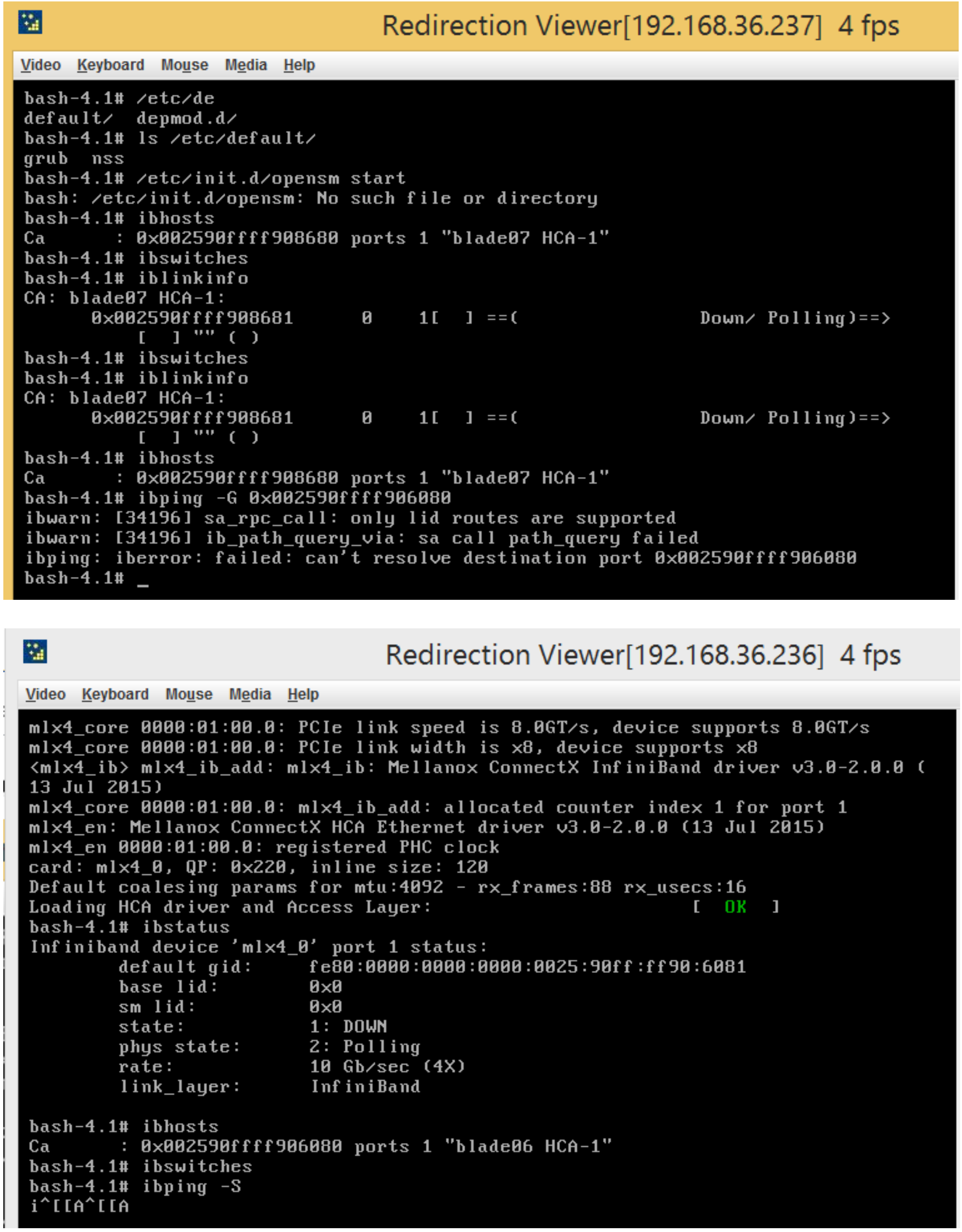


Redirection Viewer[192.168.36.236] 4 fps
Video Keyboard Mouse Media Help

```
mlx4_core: device is working in RoCE mode: Roce V1
mlx4_core: gid_type 1 for UD QPs is not supported by the devicegid_type 0 was ch
osen instead
mlx4_core: UD QP Gid type is: V1
mlx4_core 0000:01:00.0: PCIe link speed is 8.0GT/s, device supports 8.0GT/s
mlx4_core 0000:01:00.0: PCIe link width is x8, device supports x8
<mlx4_ib> mlx4_ib_add: mlx4_ib: Mellanox ConnectX InfiniBand driver v3.0-2.0.0 (
13 Jul 2015)
mlx4_core 0000:01:00.0: mlx4_ib_add: allocated counter index 1 for port 1
mlx4_en: Mellanox ConnectX HCA Ethernet driver v3.0-2.0.0 (13 Jul 2015)
mlx4_en 0000:01:00.0: registered PHC clock
card: mlx4_0, QP: 0x220, inline size: 120
Default coalesing params for mtu:4092 - rx_frames:88 rx_usecs:16
Loading HCA driver and Access Layer:                       [  OK  ]
bash-4.1# ibstatus
Infiniband device 'mlx4_0' port 1 status:
        default gid:     fe80:0000:0000:0000:0025:90ff:ff90:6081
        base lid:        0x0
        sm lid:          0x0
        state:           1: DOWN
        phys state:      2: Polling
        rate:            10 Gb/sec (4X)
        link_layer:      InfiniBand

bash-4.1# _
```

Figure 27ibstatus call on another node.

We tried ping-pong nodes over IB yet attempt failed with nodes not seeing one another.


Redirection Viewer[192.168.36.237] 4 fps
Video Keyboard Mouse Media Help

```
bash-4.1# /etc/de
default/  depmod.d/
bash-4.1# ls /etc/default/
grub  nss
bash-4.1# /etc/init.d/opensm start
bash: /etc/init.d/opensm: No such file or directory
bash-4.1# ibhosts
Ca      : 0x002590ffff908680 ports 1 "blade07 HCA-1"
bash-4.1# ibswitches
bash-4.1# iblinkinfo
CA: blade07 HCA-1:
      0x002590ffff908681      0     1[   ] ==(                Down/ Polling)==>
          [  ] "" ( )
bash-4.1# ibswitches
bash-4.1# iblinkinfo
CA: blade07 HCA-1:
      0x002590ffff908681      0     1[   ] ==(                Down/ Polling)==>
          [  ] "" ( )
bash-4.1# ibhosts
Ca      : 0x002590ffff908680 ports 1 "blade07 HCA-1"
bash-4.1# ibping -G 0x002590ffff906080
ibwarn: [34196] sa_rpc_call: only lid routes are supported
ibwarn: [34196] ib_path_query_via: sa call path_query failed
ibping: iberror: failed: can't resolve destination port 0x002590ffff906080
bash-4.1# _
```

Redirection Viewer[192.168.36.236] 4 fps
Video Keyboard Mouse Media Help

```
mlx4_core 0000:01:00.0: PCIe link speed is 8.0GT/s, device supports 8.0GT/s
mlx4_core 0000:01:00.0: PCIe link width is x8, device supports x8
<mlx4_ib> mlx4_ib_add: mlx4_ib: Mellanox ConnectX InfiniBand driver v3.0-2.0.0 (
13 Jul 2015)
mlx4_core 0000:01:00.0: mlx4_ib_add: allocated counter index 1 for port 1
mlx4_en: Mellanox ConnectX HCA Ethernet driver v3.0-2.0.0 (13 Jul 2015)
mlx4_en 0000:01:00.0: registered PHC clock
card: mlx4_0, QP: 0x220, inline size: 120
Default coalesing params for mtu:4092 - rx_frames:88 rx_usecs:16
Loading HCA driver and Access Layer:                          [  OK  ]
bash-4.1# ibstatus
Infiniband device 'mlx4_0' port 1 status:
        default gid:     fe80:0000:0000:0000:0025:90ff:ff90:6081
        base lid:        0x0
        sm lid:          0x0
        state:           1: DOWN
        phys state:      2: Polling
        rate:            10 Gb/sec (4X)
        link_layer:      InfiniBand

bash-4.1# ibhosts
Ca      : 0x002590ffff906080 ports 1 "blade06 HCA-1"
bash-4.1# ibswitches
bash-4.1# ibping -S
i^[[A^[[A
```

Figure 28: ibping failed.

Conclusion

During period of summer program, we explored technologies that were new for me such as: IPMI, MegaPCLI RAID, ZFS, installation of a system on a disc less system. InfiniBand was not configured during given time period, yet attempts were made and we accomplished some progress.

A presentation of the obtained results was given at a public LIT JINR seminar. It was accepted well and spawned a productive discussion.

We hope to continue our work remotely for further exploration of cluster technologies.