Cray® DataWarp™ SSD Installation and Configuration Guide

S-2547-5202

Contents

1 Install and Configure Cray® DataWarp™ SSD

Two different families of the FusionIO SSD cards are supported by this installation procedure: older ioScale 2 cards and newer SX300 and PX600 cards. The installation procedures differ slightly depending on which type of card is installed. It is the administrator's responsibility to know which type is installed on the machine. Systems with more than one type of SSD are not supported.

Table 1. Installation differences based on SSD card family

Entity	ioScale2 SSD Cards	SX300/PX600 SSD Cards
Driver	VSL3	VSL4
RPM naming convention	fio-common-*version_vsl*.noarch.rpm fio-sysvinit-*version_vsl*.noarch.rpm fio-util-*version_vsl*.noarch.rpm iomemory-vsl-*version_vsl*.src.rpm	fio-common-*version_vsl4*.x86_64.rpm fio-sysvinit-*version_vsl4*.x86_64.rpm fio-util-*version_vsl4*.x86_64.rpm iomemory-vsl4-*version_vsl4*.x86_64.rpm
Service name	`iomemory-vsl`	`iomemory-vsl4`

1.1 Set `CLEinstall` Parameters for SSD

The `CLEinstall` program ensures that the proper RPMs are installed during the installation or upgrade of CLE software based on the following SSD-specific paramters in the `CLEinstall.conf` file:

`SSDtype` Specifies whether SSD cards are present in the system. Set `SSDtype=ioScale2` for FusionIO IoScale2 SSD cards. Set `SSDtype=ioMemory3` for FusionIO SX300 or PX600 SSD cards.

Default is `none`.

`CNL_swap` Specifies whether the SSD cards are configured for compute node memory swapping. Set `CNL_swap=yes` if the SSD cards are used for swapping.

Default is `no`.

Set these parameters prior to running `CLEinstall` during an installation or upgrade of CLE system software or when running `CLEinstall` as a separate event.

1.2 Install the SSD Driver

Prerequisites

- `CLEInstall` has successfully executed and installed the SSD RPMs in the shared root.
- If adding a new blade to the system, see "Updating the System Configuration After a Blade Change" in S-2393 prior to proceeding.

The SSD driver can be configured to use the SSD devices as scratch/temporary storage or for swapping. These procedures cover both options.

1. Log on to the boot node (use the currently booted system or boot the boot and SDB nodes).

```
smw:~#  ssh root@boot
```

2. Initiate `xtopview` and change directory to `/software`.

```
boot:~ # xtopview
default/:/ # cd /software
```

3. (VSL4 software only) Modify permissions for the `/usr/bin/fio-set-affinity` file.

```
default/:/software # chmod 0755 /usr/bin/fio-set-affinity
```

4. Turn off the `iomemory-vsl` service in the default view and verify the results.
   - For VSL3:

```
default/:/software # chkconfig iomemory-vsl off
default/:/software # chkconfig --list iomemory-vsl
iomemory-vsl      0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

   - For VSL4:

```
default/:/software # chkconfig iomemory-vsl4 off
default/:/software # chkconfig --list iomemory-vsl4
iomemory-vsl4     0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

5. Edit the LVM configuration file in the default shared root view to ensure that the following lines are included in the `advanced settings` area within the `device` section.

```
default/:/software # vi /etc/lvm/lvm.conf
```

```
types = [ "fio", 16 ]
issue_discards = 1
```

6. Exit `xtopview`

```
default/:/software # exit
```

7. Initiate the node specialization view, where *N* is an integer node ID or a physical ID.

```
boot:~ # xtopview -n N
```

8. Turn on the `iomemory-vsl` service and verify the results.

- For VSL3:

```
node/N:/ # chkconfig iomemory-vsl on
node/N:/ # chkconfig --list iomemory-vsl
iomemory-vsl      0:off   1:on   2:on   3:on   4:on   5:on   6:off
```

- For VSL4:

```
node/N:/ # chkconfig iomemory-vsl4 on
node/N:/ # chkconfig --list iomemory-vsl4
iomemory-vsl4     0:off   1:on   2:on   3:on   4:on   5:on   6:off
```

The following steps are only required for setting up scratch. If setting up swap, skip to step *13* on page 6.

9. (Scratch setup only) Create a mount point for the scratch file system.

```
node/N:/ # mkdir -p mount_point
```

For example:

```
node/N:/ # mkdir -p /flash/scratch1
```

10. (Scratch setup only) Specialize the `/etc/fstab` file for this node only.

```
node/N:/ # xtspec /etc/fstab
```

11. (Scratch setup only) Edit the `/etc/fstab` file and add an entry to mount the file system, where *vg_name* is the volume group and *lv_name* is the logical volume name, both of which will be created later.

> **IMPORTANT:** The `/etc/fstab` entry must be tab-separated for LVM, and the entry must be entered on one line only.

```
node/N:/ # vi /etc/fstab
```

```
/dev/vg_name/lv_name mount_point  xfs  defaults,noauto,noatime,nobarrier,discard  0  0
```

For example:

```
/dev/scratch1_vg/scratch1 /flash/scratch1  xfs  defaults,noauto,noatime,nobarrier,discard  0  0
```

12. (Optional - NFS setup for scratch only) Set up NFS export so that scratch is accessible from the login or other service node.

a. Specialize the `/etc/exports` file for this node only.

```
node/N:/ # xtspec /etc/exports
```

b. Edit the `/etc/exports` file and add an entry for scratch.

```
node/N:/ # vi /etc/exports
```

```
mount_point *(rw,no_root_squash,no_subtree_check)
```

For example:

```
/flash/scratch1 *(rw,no_root_squash,no_subtree_check)
```

c. Turn on the NFS server.

```
node/N:/ # chkconfig nfsserver on
```

13. Specialize the `/etc/sysconfig/iomemory-vsl` file for this node only.

- For VSL3:

```
node/N:/ # xtspec /etc/sysconfig/iomemory-vsl
```

- For VSL4:

```
node/N:/ # xtspec /etc/sysconfig/iomemory-vsl4
```

14. Edit `/etc/sysconfig/iomemory-vsl` to modify the iomemory configuration for this node.

- For VSL3:

```
node/N:/ # vi /etc/sysconfig/iomemory-vsl
```

- For VSL4:

```
node/N:/ # vi /etc/sysconfig/iomemory-vsl4
```

a. Ensure that `ENABLED=1` is set and not commented out.

b. Define `LVM_VGS`, the LVM volume groups parameter.

```
LVM_VGS="/dev/vg_name"
```

For example:

```
LVM_VGS="/dev/scratch1_vg"
```

c. (Scratch setup only) Define the file system to be mounted.

```
MOUNTS="mount_point"
```

For example:

```
MOUNTS="/flash/scratch1"
```

15. Exit xtopview and `ssh` to the specialized node.

```
node/N:/ # exit
```

```
boot:~ # ssh cname
```

16. Use the `init` script to load the driver.

- For VSL3:

```
nid:~ # /etc/init.d/iomemory-vsl start
```

- For VSL4:

```
nid:~ # /etc/init.d/iomemory-vsl4 start
```

It will complain that the volume group was not found and that it could not mount in the console. This is expected, because the volumes have not been configured yet, and can safely be ignored.

17. Create the directory `/var/etclvm`.

```
nid:~ # mkdir /var/etclvm
```

18. Copy the `lvm.conf` file to the `/var/etclvm` directory. It includes the `issue_discards = 1` option, which configures LVM to send trim commands if volumes are removed.

```
nid:~ # cp -p /etc/lvm/lvm.conf /var/etclvm
```

19. Bind mount `/etc/lvm` to `/var/etclvm` to allow read/write capabilities, enabling LVM changes to be made to the storage node. This bind mount can also be added to the `/etc/fstab` file. Verify the mount.

```
nid:~ # mount -o bind /var/etclvm /etc/lvm
nid:~ # mount | grep lvm
10.131.255.254:/snv/29/var/etclvm on /etc/lvm type nfs (rw,relatime,vers=3,rsize=32768,
wsize=32768,namlen=255,hard,nolock,proto=tcp,timeo=600,retrans=2,sec=sys,
mountaddr=10.131.255.254,mountvers=3,mountport=37710,mountproto=tcp,local_lock=all,
addr=10.131.255.254)
```

20. Look for the new devices.

```
nid:~ # fdisk -l /dev/fio*
Disk /dev/fioa: 2600.0 GB, 2600000000000 bytes
255 heads, 63 sectors/track, 39512 cylinders, total 634765625 sectors
Units = sectors of 1 * 4096 = 4096 bytes
Sector size (logical/physical): 4096 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 32768 bytes
Disk identifier: 0x6901a7ad

Disk /dev/fioa doesn't contain a valid partition table
Note: sector size is 4096 (not 512)

Disk /dev/fiob: 2600.0 GB, 2600000000000 bytes
255 heads, 63 sectors/track, 39512 cylinders, total 634765625 sectors
Units = sectors of 1 * 4096 = 4096 bytes
Sector size (logical/physical): 4096 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 32768 bytes
Disk identifier: 0x00000000

Disk /dev/fiob doesn't contain a valid partition table
```

21. Set up devices with LVM.

```
nid:~ # pvcreate /dev/fioa [/dev/fiob]
nid:~ # vgcreate vg_name /dev/fioa [/dev/fiob]
```

For example:

```
nid:~ # pvcreate /dev/fioa /dev/fiob
nid:~ # vgcreate scratch1_vg /dev/fioa /dev/fiob
```

22. (Scratch setup only) Create scratch volume.

```
nid:~ # lvcreate -l 100%VG -n lv_name -i2 vg_name
nid:~ # mkfs.xfs /dev/vg_name/lv_name
```

This example creates one volume using 100% of the capacity of both devices with data striped across the two (`-i2`) in 64K blocks (default stripe width).

```
nid:~ # lvcreate -l 100%VG -n scratch1 -i2 scratch1_vg
nid:~ # mkfs.xfs /dev/scratch1_vg/scratch1
```

23. (Swap setup only) Create swap volume. Create one logical volume per compute node using swap.

```
nid:~ # lvcreate --size 32G -n lv_name -i2 vg_name
nid:~ # mkswap /dev/vg_name/lv_name
```

This example creates one volume using 32GB of capacity of both devices with data striped across the two (`-i2`) in 64K blocks (default stripe width).

```
nid:~ # lvcreate --size 32G -n swap1 -i2 swap_vg
nid:~ # mkswap /dev/swap_vg/swap1
```

24. Gather the serial numbers for each PCI SSD card. These will be used to change the power limit for these cards within the configuration file.

```
nid:~ # fio-status | grep SN
```

25. Exit to the boot node and initiate `xtopview`.

```
nid:~ # exit
boot:~ # xtopview
```

26. Edit the configuration file and add an `options` entry to set the power limit for the PCI SSD cards to 40W in order to allow full performance, where *SN-value*:`40` represents an adapter serial number and the maximum amount of power that device should pull (in watts). Multiple pairs must be comma-separated.

- For VSL3:

  ```
  default/:/ # vi /etc/modprobe.d/iomemory-vsl.conf
  ```

  ```
  options iomemory-vsl external_power_override=SN-value:40
  ```

  For example:

  ```
  options iomemory-vsl external_power_override=1234G5678:40,0987G6543:40
  ```

- For VSL4:

  ```
  default/:/ # vi /etc/modprobe.d/iomemory-vsl4.conf
  ```

  ```
  options iomemory-vsl4 external_power_override=SN-value:40
  ```

For example:

```
options iomemory-vsl4 external_power_override=1234G5678:40,0987G6543:40
```

When this is properly configured, a message similar to the following will appear in the console log after a reload/reboot.

```
[12:37:46][c0-0c0s7n1]fioinf ioDrive 0000:03:00.0.0: PCIe power monitor enabled (master).
Limit set to 39.750 watts.
```

27. Exit `xtopview`.

28. For the new settings to take effect, invoke the `init.d` script to reload the driver or reboot the node. See *Proper Swap Server and SSD Shutdown* on page 13.

- For VSL3:

```
nid:~ # /etc/init.d/iomemory-vsl restart
```

- For VSL4:

```
nid:~ # /etc/init.d/iomemory-vsl4 restart
```

1.3 Configure Scratch Storage

Prerequisites

Complete the *Install the SSD Driver* on page 3 procedure before starting this procedure.

Two methods exist for configuring FusionIO SSDs for scratch storage. This method uses `init` scripts from FusionIO, and is likely the best choice because it handles shutdown on its own and does not require custom boot images after each update. The other method uses the standard Cray boot process. Both methods require that the kernel module is built and placed in the appropriate location after each CLE update.

1. Log on to the SMW as `root`.

2. Change directory to the boot image template directory.

```
smw:~#  cd /opt/xt-images/templates/cle_version-system_set
```

3. Create a mount point for the scratch file system.

```
smw:~#  mkdir -p mount_point
```

For example:

```
smw:~#  mkdir -p flash/scratch1
```

4. Edit the compute node `fstab` file and add a tab-separated entry (on one line) to mount the file system, where *cname* is the SSD node.

```
smw:~# vi etc/fstab
```

```
 device   mount_point  dvs path=mount_point,nodename=cname,blksize=1048576,ro_cache
```

For example:

```
/flash/scratch1   /flash/scratch1    dvs path=/flash/scratch1,nodename=c0-0c0s7n1,blksize=1048576,ro_cache
```

5. Invoke `shell_bootimage.sh` and create a new boot image using the site's regular procedure.

   Now that the storage is configured, the SSD service node and the compute nodes must be rebooted to pick up the changes. there are two options for how to proceed:

   - The first option is to exit from this procedure and reboot the system following the site-specific procedures.
   - The second option is to complete the steps in this procedure to reboot the SSD service node, reboot the compute nodes, then mount the scratch filesystem on the login nodes.

6. Log on to the SSD service node.

```
smw:~#  ssh cname
```

For example:

```
smw:~#  ssh c0-0c0s7n1
```

7. Stop the file system safely, deactivate volume groups and unload the driver by invoking the `init` script.

   - For VSL3:
   - `nid:~ # /etc/init.d/iomemory-vsl stop`
   - For VSL4:

     ```
     nid:~ # /etc/init.d/iomemory-vsl4 stop
     ```

8. Exit the node.

```
nid:~ # exit
```

9. Reboot the node.

```
smw:~#  xtbootsys --reboot -L SNL0 cname
```

10. Log off as `root`.

```
smw:~#  exit
```

11. Reboot the compute nodes, where *cnames* is a comma-separated list of all compute nodes for this partition. Altternatively, shutdown and reboot the entire system if desired.

```
crayadm@smw:~#  xtbootsys --reboot -L CNL0 cnames
```

1.4 Configure Swap Storage

Prerequisites

Complete the *Install the SSD Driver* on page 3 procedure before starting this procedure.

1. Log on to the boot node and invoke `xtopview`.

```
smw:~# ssh root@boot
boot:~# xtopview
```

2. Create a mapping file `/etc/opt/cray/swap/swap.map` in the shared root that maps each compute node to its target swap server node and the logical volume.

   The file format is as follows, with one compute node per line and columns separated by white space.

```
Compute node(cname) swap server node(swap_cname) Logical Unit Number(LUN)
```

   This file should be visible from the compute nodes, and the path to this file will be an argument to a script that configures the swap space per compute node.

   For example:

```
c3-0c0s4n0 c3-0c0s2n2 0
c3-0c0s4n1 c3-0c0s2n2 1
```

3. Edit the `/etc/opt/cray/swap/swap.config` file to set the desired level of log messages.

   The valid logging levels are: `CRITICAL`, `ERROR`, `WARNING`, `INFO`, `DEBUG`, `NOTSET`. Default is `WARNING`. Log messages of the selected severity or greater will be outputted.

   For example:

```
LOG_LEVEL=INFO
```

4. Exit the `xtopview` default view.

5. Invoke `xtopview` for a swap server node. Each swap server node must be individually configured through `xtopview`.

```
default/:/ #exit
boot:~# xtopview -n swap_nid
```

6. Determine the path to each logical volume. Output shown is for example purposes only, each site's configuration will vary.

```
node/N:/ # lvdisplay | grep "LV Name"
LV Name /dev/swap_vg/swap-c0-0c0s4n0
LV Name /dev/swap_vg/swap-c0-0c0s4n1
LV Name /dev/swap_vg/swap-c0-0c0s4n2
```

7. Create the configuration file `/etc/ietd.conf` if it doesn't exist. Use `xtspec` to specialize the file.

```
node/N:/ # touch /etc/ietd.conf
node/N:/ # xtspec /etc/ietd.conf
```

8. Edit `/etc/ietd.conf` and add the following, where *cname* is the node on which the the swap server is running, and *swap_path* and *lv_path* are obtained from step *6* on page 11.

```
Target iqn.2001-01.com.cray:cname.swap
      Lun 0 Path=/dev/swap_path/lv_path
      Lun 1 Path=/dev/swap_path/lv_path
      ...
```

For example:

```
Target iqn.2001-01.com.cray:c0-0c0s7n2.swap
      Lun 0 Path=/dev/swap_vg/swap-c0-0c0s4n0
      Lun 1 Path=/dev/swap_vg/swap-c0-0c0s4n1
      Lun 2 Path=/dev/swap_vg/swap-c0-0c0s4n2
```

9. Exit `xtopview` and then ssh to the swap server node.

```
node/N:/ # exit
boot:~# ssh cname
```

10. Turn on the `iscsi-target` service so that it starts upon boot of the swap server node.

```
nid:~ # chkconfig iscsi-target on
```

11. Restart the iSCSI server by either executing `iscsi-target restart` or rebooting the swap server node.

```
nid:~ # /etc/init.d/iscsi-target restart
```

12. Verify that the target and LUNS are present.

```
nid:~ # cat /proc/net/iet/volume
tid:1 name:iqn.2001-01.com.cray:c0-0c0s7n2.swap
   lun:0 state:0 iotype:fileio iomode:wt blocks:10000000 blocksize:512 path:/dev/swap_vg/swap-c0-0c0s4n0
   lun:0 state:0 iotype:fileio iomode:wt blocks:10000000 blocksize:512 path:/dev/swap_vg/swap-c0-0c0s4n1
   lun:0 state:0 iotype:fileio iomode:wt blocks:10000000 blocksize:512 path:/dev/swap_vg/swap-c0-0c0s4n2
```

13. Exit to the boot node and repeat steps *5* on page 11 through step *12* on page 12 for each swap server node.

```
nid:~ # exit
```

Client Configuration - Configure Multipath within the Shared Root.

14. Initiate `xtopview`.

```
boot:~# xtopview
```

15. Create the `/etc/multipath` directory prior to configuring the compute nodes.

```
default/:/ # mkdir /etc/multipath
```

16. Edit `/etc/multipath.conf`. Add the bolded items in the example shown if they are not already in the configuration file.

```
blacklist_exceptions {
       property (ID_WWN|ID_SCSI_VPD|UDEV_LOG)
       device {
```

```
        vendor "IET"
        product "VIRTUAL-DISK"
        }
}
defaults {
   polling_interval 10
   path_selector "round-robin 0"
   path_grouping_policy multibus
   prio const
   path_checker directio
   max_fds 8192
   rr_weight priorities
   failback immediate
   no_path_retry 30
   user_friendly_names yes
   getuid_callout "/lib/udev/scsi_id -g -u -d /dev/%n"
}
devices {
    device {
       vendor "IET"
       product "VIRTUAL-DISK"
       no_path_retry queue
    }
...
}
```

17. Exit `xtopview`.

```
default/:/ #exit
```

1.5 Proper Swap Server and SSD Shutdown

The swap servers must shut down prior to shutting down the FusionIO SSD devices, and the internal software running on the SSDs must properly shut itself down when the Cray system is shutting down. There are two methods, manual or automatic, of ensuring that this occurs.

Manual method:

1. Watch xtconsole to verify that the swap server has shut down; that the `/etc/init.d/iscsi-target stop` command executed successfully.
2. Watch xtconsole to verify that the SSD drivers unloaded. This must occur prior to answering `y/n` to the "clear alerts before halting" prompt.

Automatic method:

3. Modify the shutdown script `/opt/cray/hss/default/etc/auto.xtshutdown` and add the following two lines for each SSD node:

```
lappend actions { crms_exec_on_bootnode "root" "ssh cname /bin/umount mount_point }
lappend actions { crms_exec_on_bootnode "root" "ssh cname /etc/init.d/iscsi-target stop" }
lappend actions { crms_exec_on_bootnode "root" "ssh cname /etc/init.d/iomemory-vsl stop" }
```

For example:

```
...
set actions {}
lappend actions { crms_exec "xtcli shutdown $data(idlist)" }
lappend actions { crms_exec "xtcli shutdown $data(idlist)" }
lappend actions { crms_exec_on_bootnode "root" "ssh c0-0c1s7n1 /bin/umount /flash/scratch1 }
lappend actions { crms_exec_on_bootnode "root" "ssh c0-0c1s7n2 /bin/umount /flash/scratch1 }
lappend actions { crms_exec_on_bootnode "root" "ssh c0-0c1s7n1 /etc/init.d/iscsi-target stop" }
lappend actions { crms_exec_on_bootnode "root" "ssh c0-0c1s7n2 /etc/init.d/iscsi-target stop" }
lappend actions { crms_exec_on_bootnode "root" "ssh c0-0c1s7n1 /etc/init.d/iomemory-vsl stop" }
lappend actions { crms_exec_on_bootnode "root" "ssh c0-0c1s7n2 /etc/init.d/iomemory-vsl stop" }
lappend actions { crms_exec_on_bootnode "root" "xtshutdown -y" }
lappend actions { crms_sleep 10 }
lappend actions { crms_exec_on_bootnode "root" "/sbin/shutdown -h now" }
lappend actions { crms_ssh_close }
lappend actions { crms_sleep 20 }
lappend actions { my_halt }
```

IMPORTANT: Save a copy of `/opt/cray/hss/default/etc/auto.xtshutdown` as it may get overwritten when the SMW software is updated.