

Target_Core_Mod v3.0, a ConfigFS enabled SCSI target infrastructure

Linux Storage and Filesystem Workshop, '09

Nicholas A. Bellinger, Linux-iSCSI.org

Changes from LIO v2.9 to v3.0 since LSF '08

- Code has been imported into `kernel.org/lio-core-2.6.git` which tracks `linux-2.6.git` (currently at v2.6.29)
- Generic Target Engine code and subsystem plugins (interface to Linux/SCSI, Linux/BLOCK and Linux/VFS) has been separated into `target_core_mod` that lives in `drivers/target` and `include/target`
- All IOCTL code for the Generic Target Engine and LIO-Target (iSCSI Target fabric module) has been converted to 100% upstream ConfigFS infrastructure
- `Target_Core_Mod/ConfigFS` has been submitted for review and inclusion in v2.6.30, `LIO-Target/ConfigFS` will be submitted separately.

Changes from v2.9 to v3.0, Continued

- Exhaustive support for SPC-3 compliant Persistent Reservations
- Initial support for implicit (out of band) Asymmetric Logical Unit Assignment Logical Unit and Target Port Groups through ConfigFS
- 4k sector support (physical and emulated support)
- Additional EVPD 0x83 Device Identifiers including NAA IEEE Registered Extended Assigned designator format from ConfigFS provided WWN information
- Unit Attention support

ConfigFS

- `target_core_mod` creates the ConfigFS group `/sys/kernel/config/target/core`
- `Linux/SCSI`, `Linux/BLOCK` and `Linux/VFS` storage HBA and objects/devices are registered/unregistered `mkdir(2)`, `rmdir(2)` and `echo` through `/sys/kernel/config/target/core/$HBA/$DEV`
- ConfigFS symlinks are used to create SCSI Target Ports from storage objects in `/sys/kernel/config/target/core/` to SCSI fabric modules in `/sys/kernel/config/target/$FABRIC`

ConfigFS layout

/sys/kernel/config/target/core/\$HBA/\$DEV groups and attributes:

alua_lu_gp/ : Used for ALUA logical unit groups

attrib/: Attributes like block_size, emulate_tas, emulate_ua_intrlck_ctrl, queue_depth, etc.

control : Used to pass parameters to subsystem plugins

enable : Used to enable storage object

fd/ : Used to pass file descriptor to subsystem plugins

pr/ : Used for SPC-3 persistent reservations information

wwn/ : Used for T10 world wide unique naming information

ConfigFS and Linux/SCSI

Registering a Linux/SCSI Storage Object:

```
mkdir -p /sys/kernel/config/target/core/pscsi_0/sdd
```

```
/* Using File Descriptor method, also can use UDEV path */
```

```
exec 3<>/dev/sdd
```

```
echo 3 > /sys/kernel/config/target/core/pscsi_0/sdd/fd
```

```
exec 3>&-
```

```
/* Or, using parameter method */
```

```
echo 'scsi_target_id=0,scsi_channel_id=0,scsi_lun_id=0' /
```

```
> /sys/kernel/config/target/core/pscsi_0/sdd/control
```

```
echo 1 > /sys/kernel/config/target/core/pscsi_0/sdd/enable
```

ConfigFS and Linux/BLOCK

Registering a Linux/Block LVM Storage Object:

```
mkdir -p /sys/kernel/config/target/core/iblock_0/lvm_test0
```

```
/* Using File Descriptor method, also can use UDEV path */
```

```
exec 3<>/dev/lilo-test/test0
```

```
echo 3 > /sys/kernel/config/target/core/iblock_0/lvm_test0/fd
```

```
exec 3>&-
```

```
/* Or, using parameter method */
```

```
echo 'iblock_major=252,iblock_minor=2' /
```

```
> /sys/kernel/config/target/core/iblock_0/lvm_test0/control
```

```
echo 1 > /sys/kernel/config/target/core/iblock_0/lvm_test0/enable
```

ConfigFS and Linux/VFS

Registering a Linux/VFS FILEIO storage object

```
mkdir -p /sys/kernel/config/target/core/fileio_0/my_file0
```

```
/* Using parameter method, also will automatically detect size  
   from fd_dev_name= that references underlying struct  
   block_device
```

```
*/
```

```
echo 'fd_dev_name=/tmp/my_file,fd_dev_size=10000000' /
```

```
> /sys/kernel/config/target/core/fileio_0/my_file0/control
```

```
echo 1 > /sys/kernel/config/target/core/fileio_0/my_file0/enable
```

SPC-3 Persistent Reservations: Whats implemented..?

- PROUT Service Actions: REGISTER, RESERVE, RELEASE, CLEAR, REGISTER_AND_IGNORE, PREEMPT, and PREEMPT_AND_ABORT
- All PROUT Reservation Types are supported: Write Exclusive, Exclusive Access, Write Exclusive Registrants Only, Exclusive Access Registrants Only, Write Exclusive All Registrants, Exclusive Access All Registrants
- All PRIN Service Actions: READ_KEYS, READ_RESERVATION, REPORT_CAPABILITIES, READ_FULL_STATUS

SPC-3 Persistent Reservations: What clients have been tested?

- RHEL v5u3 using SCSI Fencing (uses Write Exclusive, Registrants Only and PREEMPT_AND_ABORT) using ext3 mounts. Testing with GFS (with multiple writers) is underway
- MSFT Cluster 2008 (uses Write Exclusive, Registrants Only and PREEMPT) using the MSFT domain validation suite

SPC-3 Persistent Reservations: Whats left to implement..?

- Activate Persist Through Power Loss (APTPL) using `/var/target/$HBA/$DEV/persist` from ConfigFS storage object layout for registration/reservation metadata
- PROUT REGISTER_AND_MOVE Service Action (Register and move reservation)
- SPEC_I_PT (Allows multiple Initiators to be registered with a single PROUT Service Action)

Asymmetric Logical Unit Assignment: Whats implemented?

- Logical Unit Groups (per storage object)
- Target Port Groups (per SCSI target port)
- Implicit ALUA (through ConfigFS)
- Optimized and Non Optimized ALUA access states
- REPORT_TARGET_PORT_GROUPS

Asymmetric Logical Unit Assignment: What clients..?

- Linux using the Open/iSCSI Initiator with the generic ALUA handler (scsi_dh_alua)
- OpenSolaris using their iSCSI Initiator with ZFS LUNs and MPxIO

Asymmetric Logical Unit Assignment: Whats left?

- Explicit ALUA using SET_TARGET_PORT_GROUPS
- Transitions between different ALUA access states for both implicit (via ConfigFS) and explicit (via SET_TARGET_PORT_GROUPS)
- ALUA access states: STANDBY, UNAVAILABLE, and OFFLINE

Future Work:

- Upstream inclusion for Target_Core_Mod/ConfigFS v3.0 in v2.6.30
- Cleanup and submission of LIO-Target/ConfigFS v3.0 traditional iSCSI target fabric module
- OpenFCOE/ConfigFS fabric module against upstream Fcoe code
- iSER/ConfigFS fabric module against upstream OFA code
- PCIe IOV (I/O Virtualization) 10 Gb/sec Ethernet hardware

Future Work, continued

- Integration with the OpenFiler project (in progress)
- Slick CLI interface on top of ConfigFS for day-to-day administration

Thank you!

- Douglas Gilbert (SPC-3 PR support, and many other features would not have been possible w/o sg3_utils)
- Joel Becker (For creating ConfigFS, and answering many questions early on)
- Ming Zhang (For recommending ConfigFS in the first place!)
- Mike Christie (For making quick Open-iSCSI patches when we found bugs, and creating STGT)
- Fujita Tomonori (For creating STGT and his IOMMU work)
- Dr. Hannes Reinecke (For creating scsi_dh_alua, and all his Linux/SCSI work)
- James Bottomley (For maintaining Linux/SCSI, and answering obscure SCSI spec questions)

Thank you! continued,

- Al Tobey (For endless OpenSolaris MpxIO ALUA testing)
- Brad Fennel and Jason Hodges (For endless MSFT Cluster 2008 PR testing)
- Michael Kukat (For VirtualBox iSCSI testing)
- Leonid Grossman from Neterion (For excellent Linux support of Neterion's next-generation IOV enabled 10 Gb/sec hardware)
- Phillip Reisner from Linbit (For DRBD, and his upstream efforts)
- H. Peter Anvin (For all his open source work)
- Marc Fleischmann (For co-founding Rising Tide)