

SCSI sysfs representation

Hannes Reinecke
SUSE R&D

March 6, 2008



Novell.[®]

Current layout

Class devices all over

Current SCSI sysfs representation

struct device represents SCSI objects

- host / target / lun
- No transport specific objects

Upper layer drivers are scsi bus drivers

- sd / sr / st / ses
- Not sg as it has to attach to all devices

Everything else is stored as class devices

- Transport classes

sysfs object lifetime

SCSI host

- Created by the driver
- Lifetime managed by the underlying (PCI) device
- No problems there

SCSI target

- Total number controller by HBA
- Lifetime bound to the underlying devices

SCSI devices

- Lifetime controlled by underlying transport or userland
- Intermediate devices during scanning

Uevents during scanning

```
uevent path '/class/scsi_host/host2' action 'add'
uevent path '/class/spi_host/host2' action 'add'
uevent path '/bus/pci/drivers/mptspi' action 'add'
uevent path '/class/spi_transport/target2:0:0' action 'add'
uevent path '/class/spi_transport/target2:0:1' action 'add'
uevent path '/class/spi_transport/target2:0:1' action 'remove'
uevent path '/class/spi_transport/target2:0:2' action 'add'
uevent path '/class/spi_transport/target2:0:2' action 'remove'

uevent path '/class/spi_transport/target2:0:13' action 'remove'
uevent path '/class/spi_transport/target2:0:14' action 'add'
uevent path '/class/spi_transport/target2:0:14' action 'remove'
uevent path '/class/spi_transport/target2:0:15' action 'add'
uevent path '/class/spi_transport/target2:0:15' action 'remove'
uevent path
'/devices/pci0000:00/0000:00:10.0/host2/target2:0:0/2:0:0:0' action
'add'
uevent path '/class/scsi device/2:0:0:0' action 'add'
```

SCSI device attributes

No default attributes for target and device

- Attributes are created after objects are visible in sysfs
- Userland programs have to wait for them
- Racy

Convert to use default attributes

scanning lifetime rules

Scanning process

- Create object for each possible target
- Create scsi_device for LUN 0
- Delete if no response
- Get number of existing luns by Inquiry or from the HBA
- Create scsi_device for each found LUN
- Delete if no response
- Delete target if no LUNs connected

Problems during scanning

No uevents for targets

- targets have no bus, hence no uevents are generated
- No fancy udev matching rules possible
- Add `scsi_target` to the `scsi` bus will solve that

But:

- Intermediate targets may be created during scanning
- Targets are created and registered with `sysfs`, even for intermediate targets
- False positives for uevents

Rework to not register intermediate targets

Problems during scanning

Delayed SCSI device removal

- Removal of SCSI devices has to be delayed until I/O is finished
- SCSI device has to be kept active during that time
- Scanning process might hit upon a device scheduled for removal
- Removal is not reversible, so scanning process might return a (then invalid) device

Rework to make removal reversible

Fixed event ordering

```
uevent '/devices/pci0000:00/0000:00:10.0/host2' action 'add'
uevent '/devices/.../host2/scsi_host/host2' action 'add'
uevent '/devices/.../host2/spi_host/host2' action 'add'
uevent '/bus/pci/drivers/mptspi' action 'add'
uevent '/devices/.../host2/target2:0:0' action 'add'
uevent '/devices/.../host2/target2:0:0/spi_transport/target2:0:0'
action 'add'
uevent '/devices/.../host2/target2:0:0/2:0:0:0' action 'add'
uevent '/devices/.../host2/target2:0:0/2:0:0:0/scsi_device/2:0:0:0'
action 'add'
uevent '/devices/.../host2/target2:0:0/2:0:0:0/bsg/2:0:0:0' action
'add'
```

Further directions

scsi_target

scsi_target keeps own reference counter

- With rework reap_ref is in sync with kobj counter
- Removal desirable

integrate device_handler framework

- Convert device_handler to bus driver
- Match on targets only

scsi_device

Get rid of scsi_device class

- No information whatsoever
- All related information can be access via scsi bus

Convert 'sg' to a proper bus driver

- Requires multiple binding

Transport classes

Convert transport classes to buses?

- Either single bus 'transport' or per transport
- Matching provided by sysfs core
- Most of the attribute_container functionality already provided
- Might require multiple binding eventually

Novell.[®]

Unpublished Work of Novell, Inc. All Rights Reserved.

This work is an unpublished work and contains confidential, proprietary, and trade secret information of Novell, Inc. Access to this work is restricted to Novell employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of Novell, Inc. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. Novell, Inc., makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All Novell marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.

