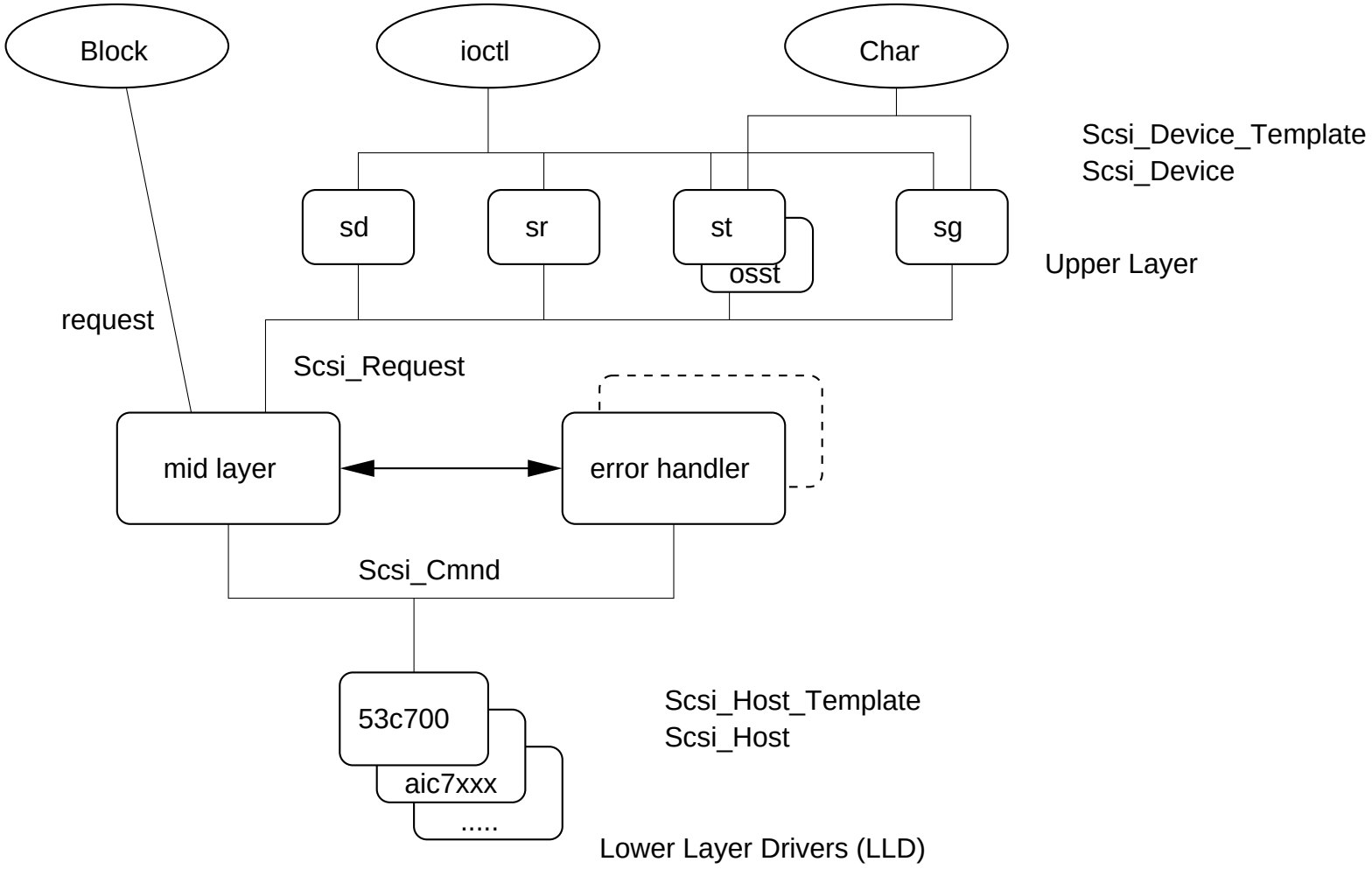
Incrementally Improving the SCSI Subsystem

James Bottomley

SteelEye Technology

28 June 2002

The SCSI Stack



Device Inquiry (scsi_scan.c)

- Static exception table inside scsi_scan.c is **wrong**
  - Already $\geq$100 lines long
  - Will grow without bound
- Solution: Move the exception table to user level.
- Use the existing infrastructure to do this.

Enter Hotplug

- Existing infrastructure to send device insertion (or detection) events back to the user
- Hotplug scripts may configure the device from user level.
- Scripts are also asynchronous—they may be used to trigger further device discovery.

Hotplug Inquiry

- Can scan entirely from user space using "scsi add-single-device"
- Can trigger scanning from host adapter hotplug
- add-single-device triggers minimal inquiry and then passes results up in device hotplug event.
- device hotplug configures all the exceptions (and also triggers LUN scanning if necessary)

Finding the Root Filesystem

- Can still use hotplug to probe the root device
- However, need small hotplug system on the initial ramdisk.

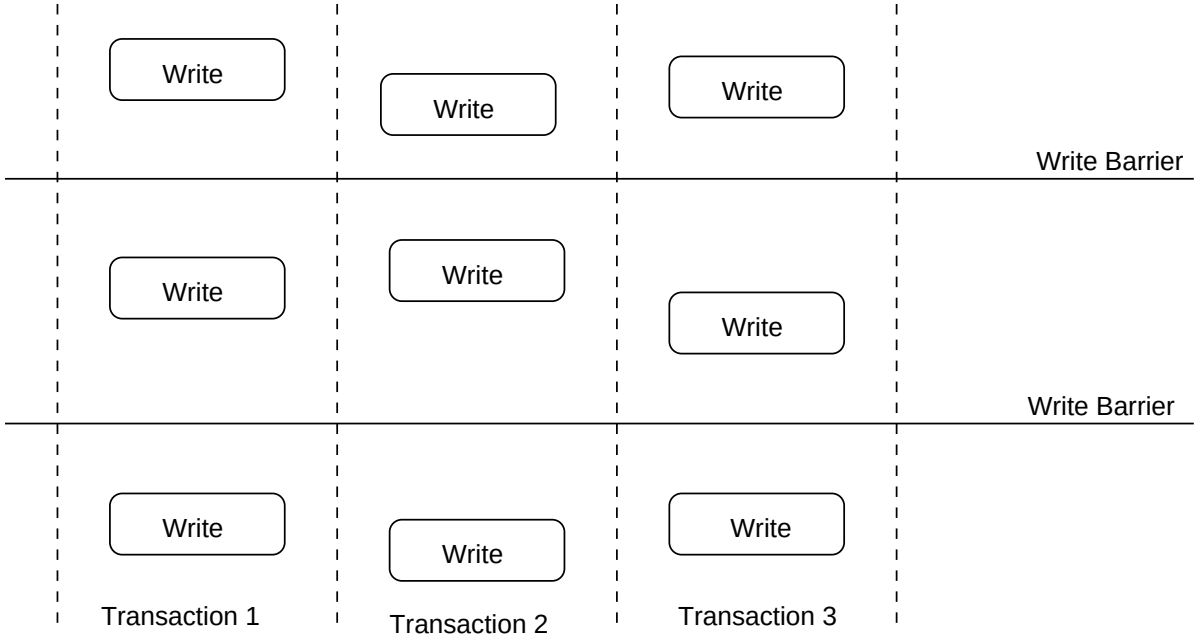
Tag Command Queueing

- Currently every driver (that does TCQ) has its own implementation in the LLD.
- Obviously, this leads to code duplication and makes finding bugs hard.
- The block layer now has generic tag handling which can be implemented by the LLDs.
- Write barriers come naturally with this implementation.
- patches to do this already exist (but error handler still needs fixing).

Tag Starvation

- Tag starvation occurs when the SCSI device ignores a tagged command for a long period of time while executing others.
- This condition is essentially a recoverable error $\Rightarrow$ treat in the error handler
- Several ways of fixing:
  1. Periodically send down ordered tags to force flushing
  2. When the condition occurs, throttle the drive queue until it executes the starved tag

Write Barriers



- Nine writes ordered as three transactions
- Previously, fs or database took care to wait on results of writes in a transaction to avoid overlap
- Now, send writes down with a barrier between them.
- Barrier must be preserved to the media surface

Write Barriers Continued

- Enforce the barrier to the media surface using ordered tags.
- Error handling now becomes a problem
- If the error is on the barrier (ordered tag), as soon as the SCSI device completes it, the barrier is gone from the device queue.
- **Cannot** abort any command and retry (crosses barrier).
- Instead, must trash all pending I/O (reset) and request that the block layer reissue the uncompleted writes in the correct order.
- QUEUE_FULL conditions can still cause slight barrier leakage.

Error Handler Changes

- Don't use Abort (see write barriers)
- Use correct reset (LUN, device or bus)
- Be aware of the actions of the reset (i.e. assume all outstanding I/O that should be affected is failed as well)
- Improve the error handler API—Instead of four separate function pointers for specific actions, implement a messaging interface to send arbitrary action requests.

Further Error Handler Changes

- Error handler should be stackable
- Entities at the top of the stack (i.e. volume managers, software raid) may wish to influence how errors at the bottom are handled
- Error information is (optionally) translated at each layer and passed up.
- Disposition advice is passed back down.
- Provides natural cancellation point for I/O.
- Block layer should provide this interface

Multi-Path Devices

- What level is appropriate for multi-path?
- Currently implementations at three levels
  - qla2x00—implemented in LLD
  - IBM Multi-path patches—implemented in mid layer.
  - md, lvm—implemented above upper layer drivers.
- Barrier Preservation for multi-path needs careful consideration
- Solution: Implement multiple paths in the block layer (therefore, would apply to IDE too).

Killing the Mid Layer

- Slow process—via starvation.
- Move to giving the upper layer device request prep functions for command translation
- Make the LLDs speak requests instead of Scsi_Cmnd structures
- Give the LLDs request functions.
- The mid layer still isn't dead, just disconnected: it now exists only as a set of helper functions for upper and lower layer drivers.

Advantages of Killing the Mid Layer

- Code which can be held in common between say IDE and SCSI is now in the block layer where it belongs
- The ide-scsi LLD can now die quietly as well
- You will record your ATAPI CD by connecting a SCSI cdrom request prep function to an IDE request function.

Conclusions

- Hotplug inquiry can allow us to eliminate most of the exceptional cases from the kernel
- The scsi-mid layer will be merged into the block layer as much as possible
- device connections (between upper and lower layers) will be done in the block queue.