NVMe Compliance Suite

Please feel free to inject Q’s

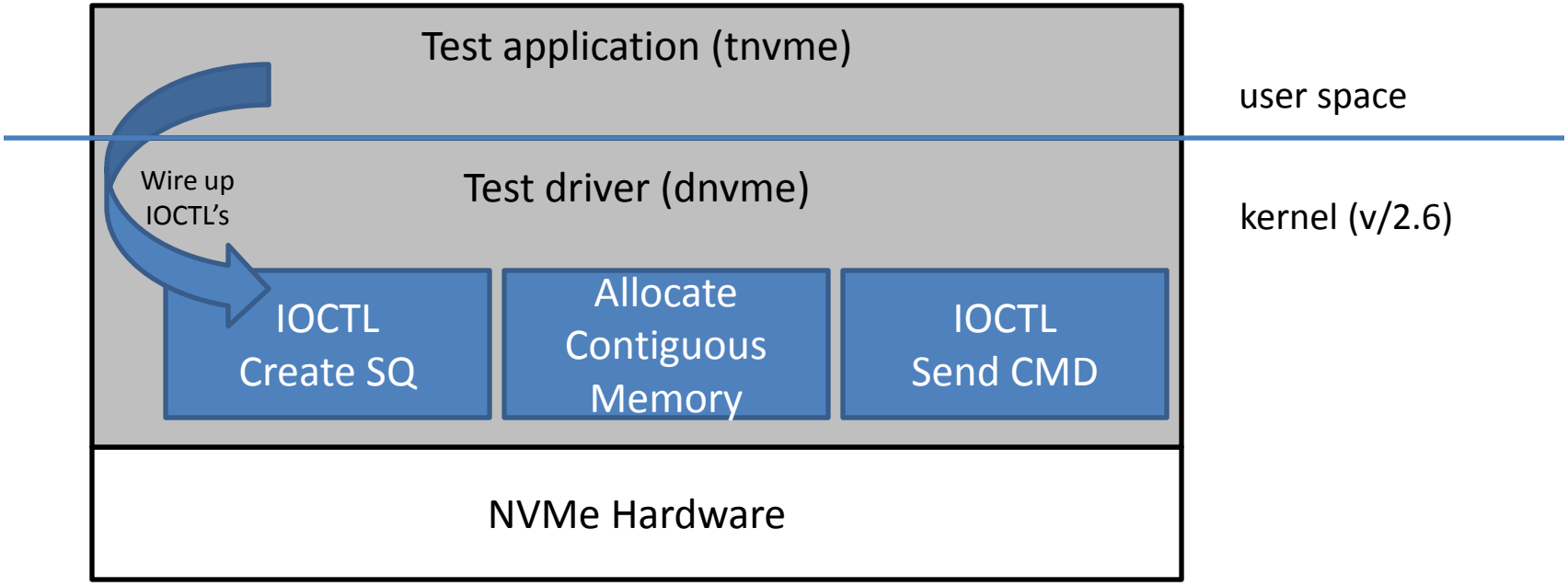
Objective

- To develop and release a software infrastructure allowing the creation of tests to target prospective NVMe hardware devices against the NVMe Working Group's released set of specifications.
  - Initially targeting v/1.0b
    - URL: http://www.nvmexpress.org/
  - Infrastructure allows other revisions to be coded which filters in only the relevant test cases.

Design Methodology

- To place most of the test logic in user vs. kernel space
  - Programmatic mistakes more lenient, and easier to debug
  - Not all developers are capable of safe kernel level coding
  - Use hybrid approach when necessary
- To create user space framework guiding test devel
  - Facilitates easily adding new tests
  - Enforces common documentation practices
  - Guide developers away from kernel crashes by disallowing “shall not” statements in NVMe spec.
- It is a compliance tool, not a benchmark tool
  - Chose functionality, not speed, when these conflicted

Overview



- tnvme "wires-up" components in dnvme to create a test
- Think of dnvme as a conduit which generically exports the majority of kernel level responsibilities into user space.

State of the Software

- Interrupts not handled
  - Currently polling CQ's must be done
  - Expect to support by 2-17-2012
- Only 7 Admin cmds supported
  - Expect to support on an as needed basis
- Only 2 NVM cmds support
  - Expect to support on an as needed basis

Get Started

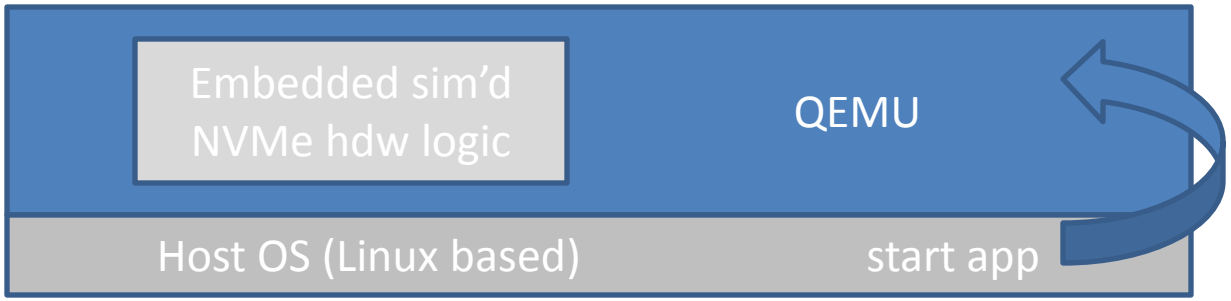
- Get the software
  - Released: https://github.com/nvmecompliance
- Vital to developers understanding
  - tnvme (all developers)
    - URL: https://github.com/nvmecompliance/tnvme/wiki
  - dnvme (advanced development)
    - https://github.com/nvmecompliance/dnvme/blob/master/Doc/readme.api.docx
- Need answers?
  - email: nvmecompliance@intel.com

Your Workflow

- Create an account
  - URL: https://github.com/plans
- Fork the repo(s) of interest
  - URL: http://help.github.com/fork-a-repo/
    - After a fork you need to clone it locally
- Modify code, commit, and push all locally
- Contribute your logic to the mainline
  - URL: http://help.github.com/send-pull-requests/

Real or Sim'd Hardware?

- Using real hardware
  - Clone tnvme and dnvme repo's only
- Using simulated hardware
  - Write tests w/o real hdw present using QEMU
  - QEMU is a process on some host OS
    - Supply a HD image; it virtualizes the containing OS.



QEMU Details

- Parent QEMU project
  - URL: http://wiki.qemu.org/Main_Page
- Intel cloned QEMU project
  - Added sim'd NVMe hardware
  - Continually adding new NVMe features
  - Released: https://github.com/nvmeqemu
- Run NVMe Compliance Suite within QEMU
  - https://github.com/nvmecompliance/manage/blob/master/readme.startHere.odt
    - Explains how to completely configure a raw system.

Simulated Hdw Overview