VirtuaLUN V7.5 Release Notes February 2017

This release is for the following:

- SANBlaze VirtuaLUN FCoE Systems
- SANBlaze VirtuaLUN iSCSI Systems
- SANBlaze VirtuaLUN Fibre Channel Systems
- SANBlaze VirtuaLUN SAS Systems
- SANBlaze VirtuaLUN NVMe Systems
- SANBlaze Value/Enterprise Software only installations

Feature Enhancements:

Non protocol specific product enhancements:

- Add Beta version of Test Manager (a new tool to automate the testing of NVMe and locally connected drives)
- Add more detailed error reporting
- Add support for QTL1521/QTL1574/QTL1675 quarch modules
- Add glitching controls to quarch hotswap modules
- Update web based shell program
- Show license update count and expiration date on maintenance page
- Add logging of all command line writes
- Allow for nicknaming a port

FC:

- Add ability to drop PLOGI responses
- Add ability to change number of credits on 32G ports
- Add hardware assisted T10 DIF support on 32G ports
- Implement VM_ID support on 32G ports

SAS:

- Add mechanism to send SMP commands
- Add format/unmap table

FCoE:

- Increase performance on Emulex based ports

NVMe:

- Implement dataset management
- Add SGL method of memory addressing
- Add support for drives formatted with MS=0
- Display version field on controller status page
- Add a CLI tool to do namespace management
- Add namespace management Identify commands to generic IO
- Implement UNH based conformance test suite
- Add number of IO queues/queue depth to controller status
- Add abrupt shutdown support
- Add SR-IOV support
- Add support for enable ‘save’ bit when doing SetFeatures calls

ISCSI:

- Add ability to login to all discovered portals at once
- Add LLDP/DCBX control on Emulex ports
- Add support for 25/50/100G adapters from Mellanox and Qlogic
- Trace all logins/logouts
- Implement BlasterMode for SW based ports
- Implement new error injection options on port actions page
- Allow iSCSI probe network to take a high to low range

Target:

- Implement SCSI Bind and UnBind commands
- Add support for split mode luns that are bound to a file
- Add ability to override log page values set via SMART error injection

Initiator:

- Add option to set alignment boundary when starting a test
- Add extra logging when final path to a device is lost
- Allow ability to manually define multiple targets as array members
- Allow WRITE_SAME and EXTENDED_COPY to be used in a test
- Support XCOPY between two different LUNs
- Allow non-integral deduplication and compression ratios
- Add ability to specify a percentage of LBA space as a test unit limit
- Add min/max to latency test
- Enhance SES support
- Add all zeros as a built in data patterns
- Decode test parameters in tests file for easier readability
- Add better indication in trace of command being retried on other paths
- Add TimeIOs feature to target and array levels
- Add tracking of path failover statistics

- Add VM_ID support
- Allow random values for skip blocks count in Rewrite test
- Add method to invoke custom CDBs from the command line

API:

- Add method to send a user defined CDB
- Add date/time to vlunSystemInfo XML call

Test Manager FAQs:

Q: Why are you releasing the code in Beta?
A: We have customers waiting for the Test Manager code for the NVMe compliance tests. SANBlaze has extensively qualified the Test Manager code, but there is a lot of new functionality and there may be some remaining bugs.

Q: Are there known bugs we should be aware of?
A: Yes, see the release notes for V7.5 Beta9 which are included in the software release.

Q: Will you make fundamental changes to the API or other functionality which will affect tests we may write for Test Manager?
A: Yes, but we will try to make changes backward compatible and if we find we need to change an API or function, we will publish it clearly and instruct you how to use the changed interface or function.

Q: I'm seeing a strange behavior, how do I debug what's going on?
A: The tests are controlled by a scheduler called sb_testmgr. It writes to a log file at /virtualun/logs/sb_testmgr. You can monitor the testing in progress using the command tail -f /virtualun/logs/sb_testmgr

Q: I hit the "stop / pause" button on the web page, but my test didn't stop.
A: If the test is in a transitional state (starting, stopping, pausing), state transitions are ignored.

Q: How do the live plots work? What data is displayed?
A: While SANBlaze IO tests are running, live plots show the performance of the tests that are currently running. After testing, live plots show the performance of the selected test. If you select more than one test, plots show data from the first selected test to the last regardless of if you have selected the tests between the first and last selected. If no tests have been run and none are running, the last 10 minutes of activity on the system is plotted.

Q: I am seeing a message that "No performance data is available" where the plotting should be. Why?

A: The plots currently only present data from the SANBlaze IO Tests. Choose an IO Test from the IO_Tests group on the Test Manager Configuration page. Also, the system must have a valid "Enhanced Data Collection" license. Contact SANBlaze if the system needs a license upgrade. To check the license, select the Maintenance page, and see if "Enhanced Data Collection" is available in the license attributes.

Q: Is there a way to assign a group of tests to all the LUNs?
A: Only at the CLI by running a script. This functionality will be added in a future release.

Q: I want to compare the performance of a device over two runs on the same plot. How do I plot yesterday's data and today's data on the same chart?
A: This is a feature that we are currently working on and will be available in a future release.

NVMe Conformance Testsuite Notes:

The tests were written based on v1.2.1 of the UNH IOL NVMe conformance testplan. Go to https://www.iol.unh.edu/testing/storage/nvme/test-suites to get a copy of the document.

The SANBlaze VirtuaLUN (VLUN) executes the test scripts through Test Manager.

All tests in all groups have been implemented except for group 8 (refer to the testplan for group headings).

Caveats:
- There are tests that require the use of an analyzer to completely verify test results. The VLUN doesn't have analyzer capabilities, so it can only infer that some things happened correctly (for example, the VLUN doesn't have access to the Submission or Completion queues, so when a command completes successfully, it can only infer that the information is correct in the appropriate queue).
- The VLUN doesn't have the ability to directly write to the controller registers; the NVMe driver doesn't allow it.
- Tests 5.4-5.6 define a number of return codes to be tested. Only the ones that the VLUN could reliably test have been implemented. The others will show a note in the test results saying that the test for the error code hasn't been implemented.
- Test 3.2 checks for REFTAG_CHECK and test 5.6 checks for GUARD_CHECK by doing a Write command in each test. If the disk can't handle these errors correctly, they can end up in an unresponsive state, resulting in the VLUN needing to be rebooted in order to get the disks back into an operating state.
- To skip REFTAG_CHECK: open the script for test 3.2, search for "WriteCmdBadDIF", and uncomment the "if" statement surrounding it.
- To skip GUARD_CHECK: open the script for test 5.6, search for "GUARD_CHECK", and uncomment the "if" statement surrounding it.