



Red Hat RDMA Integration and Testing Processes

Doug Ledford



Historical Releases



- Red Hat Enterprise Linux 4
 - Utilized OFED for both kernel code and user space packages
 - Took in all of the support present in OFED
 - The version of the openib rpm was == the version of OFED we pulled into any given release
- Red Hat Enterprise Linux 5
 - Utilized OFED for kernel code, but went directly to upstream releases for user space code
 - Did not take in all of OFED kernel code either, specifically excluded features not accepted upstream
 - The version of the openib rpm was == to the version of OFED we pulled the kernel support from

Current Releases



- Red Hat Enterprise Linux 6 and upcoming 7
 - Uses upstream kernel code.
 - Uses upstream user space packages.
 - Uses a new package named rdma as the base kernel configuration package. Starting with EL 6.3 and later, the version of the rdma package can be used to determine the version of the upstream kernel that we pulled the RDMA support from for the current EL kernel. For instance, in EL 6.3, the rdma package is version 3.3, so for that release the core RDMA kernel support came from the upstream version 3.3 Linus kernel. For EL 7, we also encode the release number into the rdma package version to ensure proper sorting. For instance, EL 7.0's rdma package is currently version 7.0_3.13_rc8-3.el7

Integration for current releases



- We perform a full kernel RDMA stack refresh with each point release, but we weed out patches that can't be taken due to need for other items that would break kABI in our kernels (the RDMA stack is exempt from the kABI list and has been since EL 4). The git whatchanged and git cherry-pick commands are essential for this work.
- We grab whichever user space packages have updated since the last point release, and rebuilt all dependent packages even if the dependent package didn't have it's own source update.

Testing Environment



- We have a specific cluster we use to test every update.
 - 56GBit/s Mellanox InfiniBand switch on one fabric
 - 40GBit/s Intel InfiniBand switch on another fabric
 - 40GBit/s Mellanox Ethernet switch on another fabric
 - 10GBit/s Dell Ethernet switch on another fabric
 - We test a fairly complete matrix of all possible combinations of InfiniBand, iWARP, RoCE/IBoE, P_Keys, VLANs, and SRIOV. And across this matrix we run high level tests (such as MPI tests) as well as low level tests (simple pings run overnight in a loop) and performance oriented tests.

Testing Environment (cont.)



- We test across a wide array of host adapters
 - mthca, mlx4, mlx5, qib, cxgb3, cxgb4, ocrdma
 - Where possible, we specifically attempt to test cross driver compatibility with each release
- Testing is mostly automated
 - We have an install environment that installs the latest release, installs our internal test harness, then runs a complete set of automated tests across the appropriate subset of machines for each test type. Failures are flagged for further analysis.

Questions?



- No? Awesome, I must have explained everything perfectly! 😊

