



Debuggability and Traceability

Leon Romanovsky

KS/LPC 2016

- Extensive usage of dynamic tracing
- Same errors prints with different levels
- No unified log convention to simplify parse tools
- No tracepoints usage (hfi1 is exception)
- Special performance counters tool (perfquery)
- Dead code placed for debug
- Special VERBOSE flag to increase debug level

- **ss tool for RDMA**
 - Connection statistics
 - Summary statistics (ss -s)
 - Per-application information
 - rdmastat vs. extend ss (new family)
- **Initial extension of RDMA CM was discussed at 2009 [1]**
 - Patches were never merged
 - FYI: netstat tool was marked as obsolete in 2012 [2]

- **ethtool tool for RDMA**
 - Vendor specific counters
 - Standard configuration options
 - Human interface
 - Provider specific plugins
 - Abstraction over different sysfs blobs
 - rdmtool vs. extend ethtool

- Hooks into running kernel for debugging and performance analysis
- Built-in filters (per-process, per-CPU, event triggering, e.t.c.)
- Allow access per counter
- Almost always available (debugfs)
- Plugged into perf utility
- and much more [3]
- RDMA possible usage
 - Remove and convert MAD snoop (Ira's patches)
 - Events debug framework for new ABI (TLP-based)
 - Memory registration analysis

- **Error prints handled in lowest possible layer**
 - Decrease driver printks
 - Canonized error paths
- **Conditional compilation**
 - Supported now for hfi1 only
- **General warning level**
 - Similar to all drivers, ULPs

1. <http://www.linuxquestions.org/questions/blog/vkmgeek-244791/rdma-remote-direct-memory-access-connection-statistics-2458/>
2. <https://sourceforge.net/p/net-tools/code/ci/77d0c1b2a55c1af31cce4df68da7bf93c8155111/>
3. <https://lwn.net/Articles/346487/>



Thank You