



# Datacenter Fabric Workshop



## OpenIB Release and Verification

Hal Rosenstock, Voltaire  
Amit Krig, Mellanox

---





# Agenda

---



- Initial Release Functionality
- Initial Release Process
- Verification Philosophy
- Missing Pieces



# Initial Release Functionality

---



- Kernel
  - mthca
  - Core
  - ULPs
    - IPoIB
    - SDP
    - SRP
    - kDAPL
    - iSER



# Initial Release Functionality

---



- Userspace
  - User verbs libraries
    - Ibverbs, mthca
  - User CM library (ibcm)
  - User AT library (ibat)
  - Management
    - OpenSM
    - Diagnostics
  - Flash burning (mstflint)



# Initial Release Functionality

---



- Userspace (cont'd)
  - ULPs
    - uDAPL
    - MPI
      - OpenMPI ?
      - MVAPICH
  - SDP library
  - AIO example
  - perftest



# Initial Release Process

---



- Patches on top of upstream kernel
  - Bug fixes
  - SDP
  - SRP
  - kDAPL & uDAPL kernel support
- Release candidate branches
  - Userspace



# Release Cycle

---



- Release candidate
- Verification & testing
- Bug reporting
  - Bugzilla



# Verification Philosophy

---



- Verify different levels of the stack
- “Smoke” Test
- API tests
- Functional tests
- Stress tests
- Performance tests
- Topology tests
- Error injection tests
- Automation of the above (directed tests) with a random test mode





# Verification

---



- Module Tests
- Tests for Functional Components including ULPs
- Automated Framework
- Must have test requirements
  - Readme file, Basic Help,
  - .config file for regression , .config for random, Local runner



## Verification

---



- Test tree for each layer (uDAPL for example)
  - Basic tests/API/smoke tests
  - Full flow tests
  - Performance
  - Bad machine
  - General running tool, can run 1 test, Dir , module or several modules



# Test Tools

---



- IBTP
  - InfiniBand Test Project
  - Proposed location in OpenIB svn tree
    - <https://openib.org/svn/gen2/trunk/ibtp/>



# Test Tools

---



- Other Parallel Test Tool “Threads”

- DART /Dashboard for framework

- Features

- Client/Server model for testing and reporting
      - Separation of data from presentation using XML and XSLT
      - Nightly dashboards with content updated hourly
      - Summarization of build logs to highlight any errors and warnings
      - Comparison of testing results across configurations
      - Multiple sites can contribute testing results
      - Empowers developers to test software in an extreme programming environment
      - Email notifications can be sent to developers when errors occur



# Test Tools

---



– LANL Test Harness



# Datacenter Fabric Workshop



Thank You

---