Experiences in Writing OFED Software for a New InfiniBand HCA

Knut Omang

[ April 6th, 2016 ]
Overview

- High level overview of Oracle's new Infiniband HCA
- Our software team's approach to the challenge
- Some reflections and experiences as a “newcomer” to OFED and Infiniband
My background

- CS background, high performance networking in the days of SCI and Myrinet + software engineering experience in other areas. New to Infiniband.
- Learned the programming trade from the inventors of object oriented programming.
- Motivated and inspired by good design that are pleasant to use.
- Backed by team with more than 20 years of RMA and RDMA experience...
Oracle has used Infiniband and OFED for over 10 years

- **OPEN Software Stack**
  - OFED verbs and ULPs
  - Enterprise stability
  - FW & BW compatibility, Long term commitment to interfaces

- **Industry Standard Specification**
  - Vendor Interoperability - Oracle EDR Infiniband HCA fully Interoperable with Mellanox and Intel/Qlogic
  - Solid Roadmap
  - Lowest Latency and Highest Throughput interconnect

- **Scaling of enterprise applications beyond what's available in the market**
Infiniband HCA designed to scale to a large number concurrent Processes and QPs
- Highly asynchronous usage model
- Supports SR/IOV with integrated vSwitches
- Each VF looks like a real HCA: Connected to (virtual) switch
- Integrated subnet management agent (SMA)
- Contemporary NIC Offloads (LSO, CSO, RSS, H/D split, etc) for Eth and IP
- On-chip MMU compatible with CPU page tables (SPARC and x86)

System integration
- SPARC SOC CPU: HCA integrated in SPARC SOC
- Standalone LP PCIe Gen3 Cards

All cases powered by the same Oracle IB HCA driver and user library
Support **lots** of processes
• 10s of thousands...

Support huge number of QPs
• Use case: 256K QPs
• Testing with > 1M QPs..

Support even huger number of MRs
• 6TB memory in 64k Mrs = 0x600.0000 MRs, > 24 bit, (> 100 million MRs)
• Support flat, larger MR space
The Oracle IB HCA SW opportunity

- Participate in making something new and cool!
- Start from clean sheets - little baggage - do it right!
- SW effort started early enough to influence HW!
- The adventure!

“...far, far away he could see something light and shimmering...”
The Oracle IB HCA SW challenge

- Adapting to changing environments...

- **Started while hardware still under development**

- **Models:**
  - Evolving executable RTL model of core components
  - Evolving SystemC high level model

- **Early goal: Be able to write “target” code from day 1!**
  - But still be flexible to handle changes

- **Required significant tools development**
  - Emulated PCIe front-end to simulators
  - Qemu patches (virtual IOMMU and SR/IOV support, simulator plugin framework)
  - Testing and test frameworks
  - Continuous integration
The Oracle IB HCA SW challenge

- Adapting to changing environments...

### Meeting/understanding OFED
- Infiniband standard vs defacto (OFED+existing implementations)
- OFED distribution vs upstream/distributions
- Aiming at a future target: Our need to use/test more bleeding edge kernels
- OFED version code x.y.z != tarball version x.y.z

### What is a good API?
- Expressibility: To what extent are all use cases expressible?
- Semantic well definedness and stability - extensibility..
- Simplicity, intuitivity: Numbers of lines needed for the “hello world” program...

### OFED APIs - strong points
- User and kernel level kept similar
- Transparent, flexible choice of kernel or user level implementation per call
- Core kept in correspondence with the standard
- Could (some of) the APIs have been included in the standard? (Like eg. SCI)
The Oracle IB HCA SW challenge

- Sophisticated On-Chip MMU..

- Supports SPARC and x86 page sizes
- HCA can point to any subtree of a page table.
- Also huge pages or Page Directory sized pages
The Oracle IB HCA SW challenge

- Get fast up to speed...

---

- Early impl. choice: Keep single code base as long as possible
  - Kernel/user mode similar - try to minimize maintenance!
  - Get something working ASAP - catch more HW bugs before tapeout..
  - Initial implementation: kernel mode + user mode as wrappers calling kernel
  - When basic feature complete: “Port” to fast path user mode
  - Keep ability to run tests in both modes!

- Proved very useful:
  - Testing aspects of kernel implementation with lower cost user test programs
  - Benefit from existing user test programs to test kernel implementation
  - Verifying functional equivalence of the two implementations
  - Short development cycle to add “fast path” user mode!

- But: OFED API not 100% up to this..
  - minor patches follows..
The Oracle IB HCA Software stack

- OFED: Some challenges..

- **Understanding process_mad**
  - Oracle IB HCA has an on-device SMA
  - Tried to understand if it was usable for debugging purposes..

- **Extensibility**
  - A tight API is good (as long as it has all that is needed..)
  - Debug/testing needs, avoid “hacks”..
  - How to allow two modes of operation from a program? (user/kernel)
  - QP, CQ, MR flags?
  - Extra variables (udata)

- **Defacto standard**
  - What is the “right way”? (“it works on existing HCAs..”)
  - Many applications ignore limits observable via query device etc
    - #of s/g entries
    - masks not handled appropriately by test applications
    - “API completeness”: Not all calls supports udata
The Oracle IB HCA SW challenge

- Stumbling blocks..

- **FMR**
  - Not part of the Infiniband standard
  - Semantics defined by implementation

- **XRC**
  - Even uglier implementation than FMR
  - Broken in recent kernels.. (?)

- **Virtualization and Integrated vSwitch**
  - Switch “always” connected to all vHCAs - ports always up
  - “How can the link be up when the cable is not connected?”

- **IP Offloads**
  - No Native IP offload APIs defined for IpoIB and EoIB
  - New OVG WG effort starts to address this!

- **GRH**
  - Wanted reusable test code for UD, RC, UC,…
  - Have to consider the implicit GRH bytes for UD..
Testing OFED providing devices

A case for an OFED compliance suite/test engine?

- Writing tests a major part of the SW work of a new HCA
  - ibv_*_pingpong examples: What do they actually test?
  - qperf: Nice stress test but *a lot* must work before it has value as test
  - testing without SM support: SM traffic generates a lot of noise

- Kernel side testing:
  - A loadable kernel unit test framework
  - Modeled after gtest
  - Uses netlink
  - Combined user/kernel tests (another case for extra flags)
  - Mainly for the parts not supported by/not easily tested by user mode

- User mode driver testing
  - Take an unmodified .ko - link it with user app?
  - Excellent for algorithms and memory usage testing (valgrind)
  - almost there using brute force regex approach + semi-automated mock interfaces..
  - but a more scientific approach better?
Conclusion/Moving forward

- **Overall good experience with OFED**
  - Good, mature API (with exceptions..)
  - Had (almost) what we needed
  - As with all APIs room for improvement

- **Minor patch sets for core/uverbs and libibverbs**
  - A total of about 10 small patches

- **Driver and user library to be open sourced**
  - Work to be started as soon as possible
  - Work with community now that it is public

- **A goal of contributing more as a team to upstream**
  - Already started with a few lpoIB related patches

- **Benefit for IB and OFED to have more vendors**
  - Oracle obviously into this for the long run – we are serious about infiniband!
Questions?
THANK YOU
Knut Omang
ORACLE