ACCELERATED IPOIB (AIP)

Mike Marciniszyn
Native IPoIB verbs implementation

- MTU limited to 4K for UD
  - OPA hardware supports 10K
- Connected mode can use 64K but poor scaling
- Additional overhead of verbs layer
- Single queue for RX/TX
  - Bottleneck

Requires RSS for spreading
OPA HARDWARE FEATURES

- 10K MTU
- 16 SDMA engines
- 256 Receive contexts
- Receive Side Matching (RSM)
  - Packet spreading
  - Key to the effort

Exploit native chip features!
IPOIB NATIVE NETDEV

- Core and IPoIB already enhanced to support grafting a netdev device data path onto IPoIB
- Data path allows for more than the single queue on send and receive
- Still uses UD QP for multicast and pathing

Exploit native RDMA features!
IPOIB ENHANCEMENTS

- Add 10k MTU support
- FM controls MTU
- IPoIB determines MTU on join

Extend IPoIB with minor enhancements
- **Support multiple TX/RX queues in hfi1**
  - Upper layer selects SDMA engine
  - RSM selects receive context
**RSM**

- Based on 9B (IB compatible) packet type
- Matching
  - Selects packet via Field1/Field2 and masks
  - Field2: Upper 8 bits of QP == 0x81
  - Field1: LNH == NO_GRH
- Receive context selection via inspection
  - Index: DETH hash
  - Upper entries in Map table contain netdev contexts
  - Offset: upper 8 entries of map
CODE CHANGES

- **IPoIB**
  - Add support for 10K MTU
  - detect IB_QP_CREATE_NETDEV_USE support when creating OPA UD QP

- **RDMAVT**
  - Advertise IB_QP_CREATE_NETDEV_USE support
  - React to IB_QP_CREATE_NETDEV_USE for UD create
  - Add 0x81 in upper 8 QPN bits
    - Handle restricted name space

- **HFI1 (most of the changes)**
  - Add TX/RX code
  - Support an additional netdev context type
  - Add AIP RSM rule

- **Upstream as of 5.7**
PERFORMANCE BEFORE

- Connected, 64K MTU, 4 iperf streams

max (avg) bandwidth, 4 streams

46.0 (40.9) Gbps “read”

50.8 (44.9) Gbps “write”
Performance After

- Datagram, 10K MTU, 2 iperf streams

Average bandwidth, 2 streams

91.9 Gbps “read”

98.5 Gbps “write”
FUTURES

- **ZERO copy**
  - Use page flipping techniques to replace context buffer pointers from available skbs

- **More flexible RX queues via ethtool**
  - Reserve more RSM map entries at the top end
  - Round robin entries based current count
  - Reduce/or increase contexts as necessary, adjusting round robin in map
THANK YOU

Mike Marciniszyn