Python RDMA

OpenFabrics Software User Group Workshop

#OFSUserGroup
Jason Gunthorpe
What is it?

• 'ibtool' program to replace nearly the entire infiniband-diags
• Python library to implement InfiniBand diagnostics and RDMA
• Open Source and GPL'd
What is in it

• RDMA device discovery:
  
  ```python
  for I in rdma.get_devices(): print I.name;
  ```

• RDMA verbs
  
  ```python
  with rdma.get_verbs(path.end_port) as ctx:
      print ctx.query_device();
  ```

• IB Management
  
  ```python
  cpi = umad.SubnAdmGet(IBA.MADClassPortInfo);
  ```

• Plus More!
Package Contents

- RDMA Device Discovery
- Definitions from the IBA
- IB MAD RPC handling and parallelism
- IB subnet topology database
- libibverbs interface (Pyrex)
- Ibtool command line program
- Codegen'd and hand written documentation
- Test suite
Ibtool

- Re-implementation of infiniband-diags using Python as the implementation language:
  - One language
  - Greater consistency
  - Higher performance
- Also:
  - Test the Python RDMA core library
  - Access the unique features of the Python RDMA via the command line
  - Serve as programming examples
  - 45 commands implemented > 90% complete

March 15 – 18, 2015  #OFADevWorkshop
• Mostly looks the same:

```bash
$ ibtool ibaddr 7
GID fe80::17:77ff:feb6:2ca4 LID start 7 end 7
$ ibtool ibswitches
Switch : 0017:77ff:feb6:2ca4 ports 2 "Obsidian Longbow X100 - LBXR43D1FF" base port 0
  lid 7 lmc 0
Switch : 0017:77ff:fef9:6e79 ports 2 "Obsidian Longbow X100 - LBXREAF28B" base port 0
  lid 9 lmc 0
$ ibtool smpquery -P 2 NI -D 0,2
# Node info: DR Path (0, 2)
BaseVers:.........................1
ClassVers:........................1
NodeType:........................2
NumPorts:........................2
SystemGuid:......................0017:77ff:fef9:6e79
Guid:............................0017:77ff:fef9:6e79
PortGuid:........................0017:77ff:fef9:6e79
PartCap:.........................1
DevId:...........................0x0009
Revision:........................0x00010001
LocalPort:........................1
VendorId:.........................0x001777
```
• Some are new:

$ ibtool subnet_diff ref
Current subnet has 4 end ports, reference subnet has 4 end ports
All end ports in the current subnet are in the reference subnet.
All end ports in the reference subnet are in the current subnet.
Current subnet has 3 nodes, reference subnet has 3 nodes
All nodes in the current subnet are in the reference subnet.
All nodes in the reference subnet are in the current subnet.
Current subnet has 3 links, reference subnet has 3 links
All links in the current subnet are in the reference subnet.
All links in the reference subnet are in the current subnet.
All links in the current subnet have the same rate in the reference subnet.
Current subnet has 4 LIDs, reference subnet has 4 LIDs
All LIDs in the current subnet are the same as the reference subnet.
Ibtool (4)

• Section 8 of the Python RDMA manual details the various differences between **ibtool** and **infiniband-diags**:
  • Greater alignment with the IBA, PR usage, timeout computations, support for routed GIDs, etc
  • Everything supports GID/GUID/LID/DR path as a TARGET
  • Better diagnostics and debug output, including packet decodes
  • --sa and support for GMP over verbs lets **ibtool** return info without access to /dev/umad
  • LID and SA based subnet discovery options
  • Consistent support for a discovery caching file
Ibtool (5)

• Everything can be decoded and dumped:

```shell
$ ibtool ibaddr 9 -dd
D: Reply_MAD_METHOD_GET_RESP(129) SMPFormat(1.1) SMPNodeInfo(17)
   0 01010181 baseVersion=1,mgmtClass=1,classVersion=1,method=129
   4 00000000 status=0,classSpecific=0
   8 000079FF transactionID=134139628569652
 12 D0E94434
     + data SMPNodeInfo
 64 01010202 baseVersion=1,classVersion=1,nodeType=2,numPorts=2
 68 001777FF systemImageGUID=GUID('0017:77ff:fef9:6e79')
 72 FEF96E79
 76 001777FF nodeGUID=GUID('0017:77ff:fef9:6e79')
 80 FEF96E79
 84 001777FF portGUID=GUID('0017:77ff:fef9:6e79')
 88 FEF96E79
 92 00010009 partitionCap=1,deviceID=9
 96 00010001 revision=65537
100 02001777 localPortNum=2,vendorID=6007
```
Introspection

- Dynamic language with introspection makes this dead easy

```bash
$ ibtool query SubnAdmGetTable SANodeRecord \
  -f nodeInfo.systemImageGUID=0017:77ff:fef9:6e79
```

Reply structure #0

- LID..............................9
- nodeInfo.NumPorts...............2
- nodeInfo.SystemImageGUID........0017:77ff:fef9:6e79
- nodeInfo.PortGUID...............0017:77ff:fef9:6e79
- nodeInfo.VendorID..............0x001777
- nodeDescription.NodeString.....'Obsidian Longbow X100 - LBXREAF28B'

- 45 LOC! Perform any RPC, with any arguments and pretty print the result. Widely used in implementing ibtool.
SA query support

- Ibtool commands can transparently switch to use the SA rather than SMP queries:

```bash
$ ibtool ibnetdiscover --sa -d
D: Performing discovery using mode 'SA'
D: RPC MAD_METHOD_GET_TABLE(18) SAFormat(3.2)
    SANodeRecord(17) completed to 'Path 8 -> 8 SL=0 PKey=0xffff DQPN=1' len 504.
D: RPC MAD_METHOD_GET_TABLE(18) SAFormat(3.2)
    SAPortInfoRecord(18) completed to 'Path 8 -> 8 SL=0 PKey=0xffff DQPN=1' len 568.
D: RPC MAD_METHOD_GET_TABLE(18) SAFormat(3.2)
    SALinkRecord(32) completed to 'Path 8 -> 8 SL=0 PKey=0xffff DQPN=1' len 104.
```
Summary

• Great for writing management tools
• Very time efficient for test development, training and prototyping
• **Ibtool** is an improved, simpler and more maintainable version of the diags programs
Get it!

- Get it at GitHub
  http://github.com/jgunthorpe/python-rdma
- Read the manual!
- Try it out!
Thank You

OpenFabrics Software User Group Workshop