Simulation for IB Management

OpenFabrics
Software
User Group
Workshop

Hal Rosenstock
Mellanox Technologies

Agenda

- Why simulation for IB management ?
- Simulators for IB management
- Building ibsim
- ibsim Architecture
- Running ibsim
  - ibsim command line options
  - ibsim example
- ibsim console and console commands
- Running libibumad based application with ibsim

Why simulation for IB Management ?

- Simulate real IB subnet
- So can run any IB management tool
  - OpenSM
  - infiniband-diags
  - etc.
- Simulators have built in agents like SMA, PMA, etc.
- Some limitations depending on IB management simulator
- Note: there is no simulation for libibverbs or librdmacm

Simulators for IB Management

- Two simulators
  - ibsim
    - Git tree: git://git.openfabrics.org/~halr/ibsim.git
    - Current version: 0.6
    - Master is one commit past this currently
      - ibsim: Allocate MFT according to number of switch ports
  - ibmgtsim
    - Part of ibutils
    - ibutils is unmaintained
      - Not libibumad based
      - Recompilation required for use with ibmgtsim
      - infiniband-diags not supported with ibmgtsim
- Focus of presentation is on ibsim

Building ibsim

- cd to unpacked simulator directory
- make ibsim and umad2sim wrapper:

```
$ make
```

Notes:

- By default, make will build ibsim against installed in /usr/local version of libib* libraries. If you want to build it against development tree, use IB_DEV_DIR variable (or export it into environment):

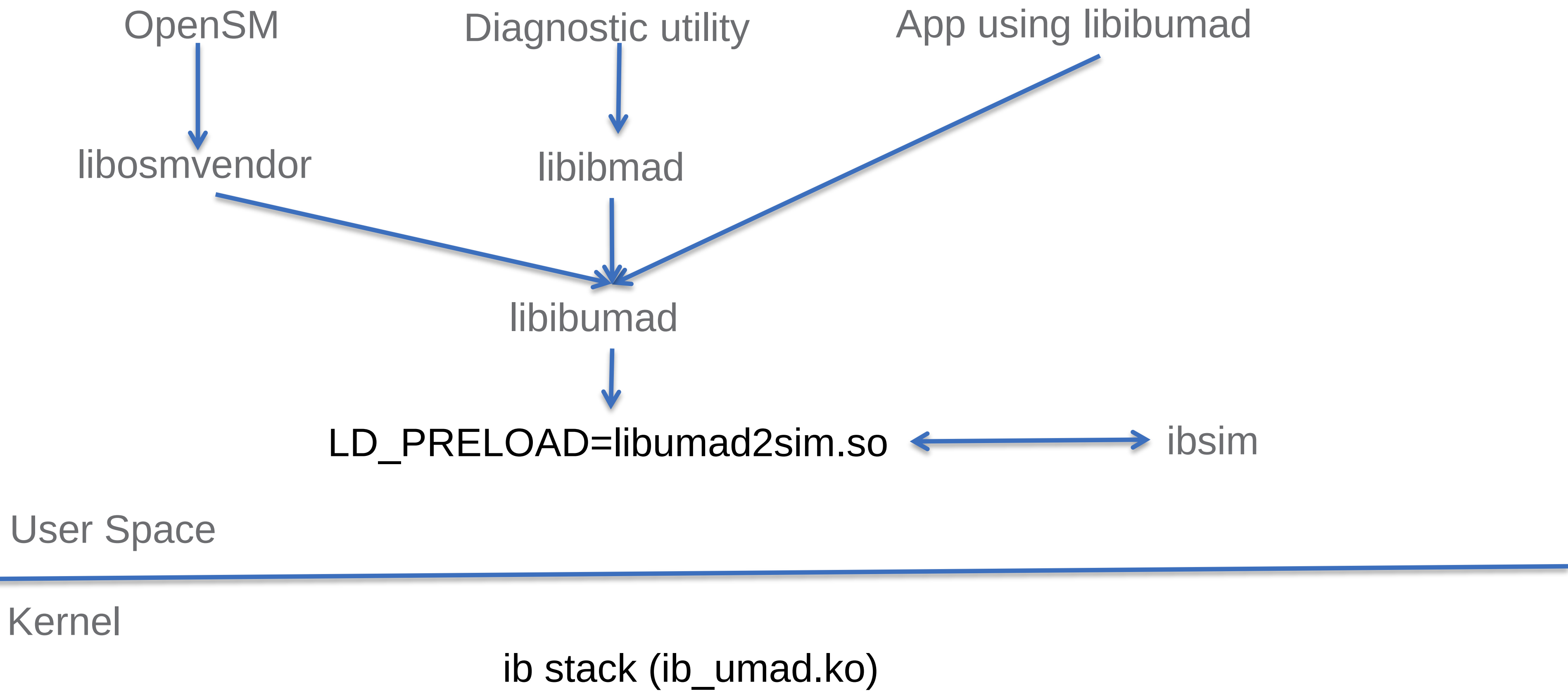
```
$ make IB_DEV_DIR=${HOME}/src/management
```

- 'make dep', 'make clean' and 'make install' are available

ibsim Architecture

For generation and reception of MAD traffic, the /dev/umadX file descriptor interface between libibumad and ib_umad kernel module is replaced by using preloaded libumad2sim.so shared library (umad2sim wrapper is part of the ibsim distribution) which conveys MADs to/from IB management application (SM/SA/PerfMgr/diag) to ibsim.

ibsim Architecture



OpenSM

Diagnostic utility

App using libibumad

libosmvendor

libibmad

libibumad

LD_PRELOAD=libumad2sim.so

ibsim

User Space

Kernel

ib stack (ib_umad.ko)

ibsim Architecture

- Any libibumad based application will work with ibsim
  - Kernel support and userspace application recompilation are not required
- ibsim works locally via unix sockets or remotely via inet sockets
  - IBSIM_SERVER_NAME and IBSIM_SERVER_PORT environment variables

Topology Files

- ibsim takes ibnetdiscover style topology file
  - Can be generated from real cluster snapshot
- ibmgtsim takes hardware description file

ibsim command line options

- Usage: ibsim [-f outfile -d(ebug) -p(arse_debug) -s(tart) -v(erbose) -I(gnore_duplicate) -N nodes -S switches -P ports -L linearcap -M mcastcap -r(emote_mode) -l(isten_to_port) <port>] <netfile>

ibsim example

-
  ```
  ibsim –s ibnd.map
  parsing: ibnd.map
  ibnd.map: parsed 21080 lines
  ###########################
  Network simulator ready.
  MaxNetNodes     = 2048
  MaxNetSwitches = 256
  MaxNetPorts     = 13312
  MaxLinearCap   = 30720
  MaxMcastCap    = 1024
  ```

ibsim console

- ibsim has a simple console command interface and can simulate random packets drops and link up/down events. It is possible to run batch commands from file via pipe or named fifo.

ibsim console commands

- sim> help
- sim> Commands:
- !<filename> - run commands from the file
- Start network
- Dump ["nodeid"] : dump node information in network
- Route <from-lid> <to-lid>
- Link "nodeid"[port] "remoteid"[port]
- ReLink "nodeid" : restore previously unconnected link(s) of the node
- ReLink "nodeid"[port] : restore previously unconnected link
- Unlink "nodeid" : remove all links of the node
- Unlink "nodeid"[port]
- Clear "nodeid" : unlink & reset all links of the node
- Clear "nodeid"[port] : unlink & reset port
- Guid "nodeid" : set GUID value for this node
- Guid "nodeid"[port] : set GUID value for this port

ibsim console commands

- Error "nodeid"[port] <error-rate> [attribute]: set error rate for
- port/node, optionally for specified attribute ID
- Some common attribute IDs:
- NodeDescription : 16
- NodeInfo : 17
- SwitchInfo : 18
- PortInfo : 21
- PerformanceSet "nodeid"[port] [attribute].[field]=[value] : set perf. counters values
- Baselid "nodeid"[port] <lid> [lmc] : change port's lid (lmc)
- Verbose [newlevel] - show/set simulator verbosity
- 0 - silent
- 1 - debug verbose
- Wait <sec> : suspend simulator prompt
- Attached : list attached clients
- X <client num> : (force) disconnect client
- #... : comment line (for scripts) - ignored
- Help/?
- Quit

Running libibumad based application with ibsim

- Use preloaded umad2sim wrapper

```
$ LD_PRELOAD=./umad2sim/libumad2sim.so ibnetdiscover
```

  , or

```
$ LD_PRELOAD=./umad2sim/libumad2sim.so opensm -f -
```

- Point of attachment is indicated by SIM_HOST environment variable. If not specified, first entry in topology file is used. For OpenSM, if -g option is used, it must be the same node name as port indicated.
- In order to run OpenSM as non-privileged user you may need to export OSM_CACHE_DIR variable and to use '-f' option in order to specify writable path to OpenSM log file.
- Set SIM_SET_ISSM environment variable to 1 when running OpenSM if want to work with SM handover/failover

OPENFABRICS
ALLIANCE

OpenFabrics Software User Group Workshop