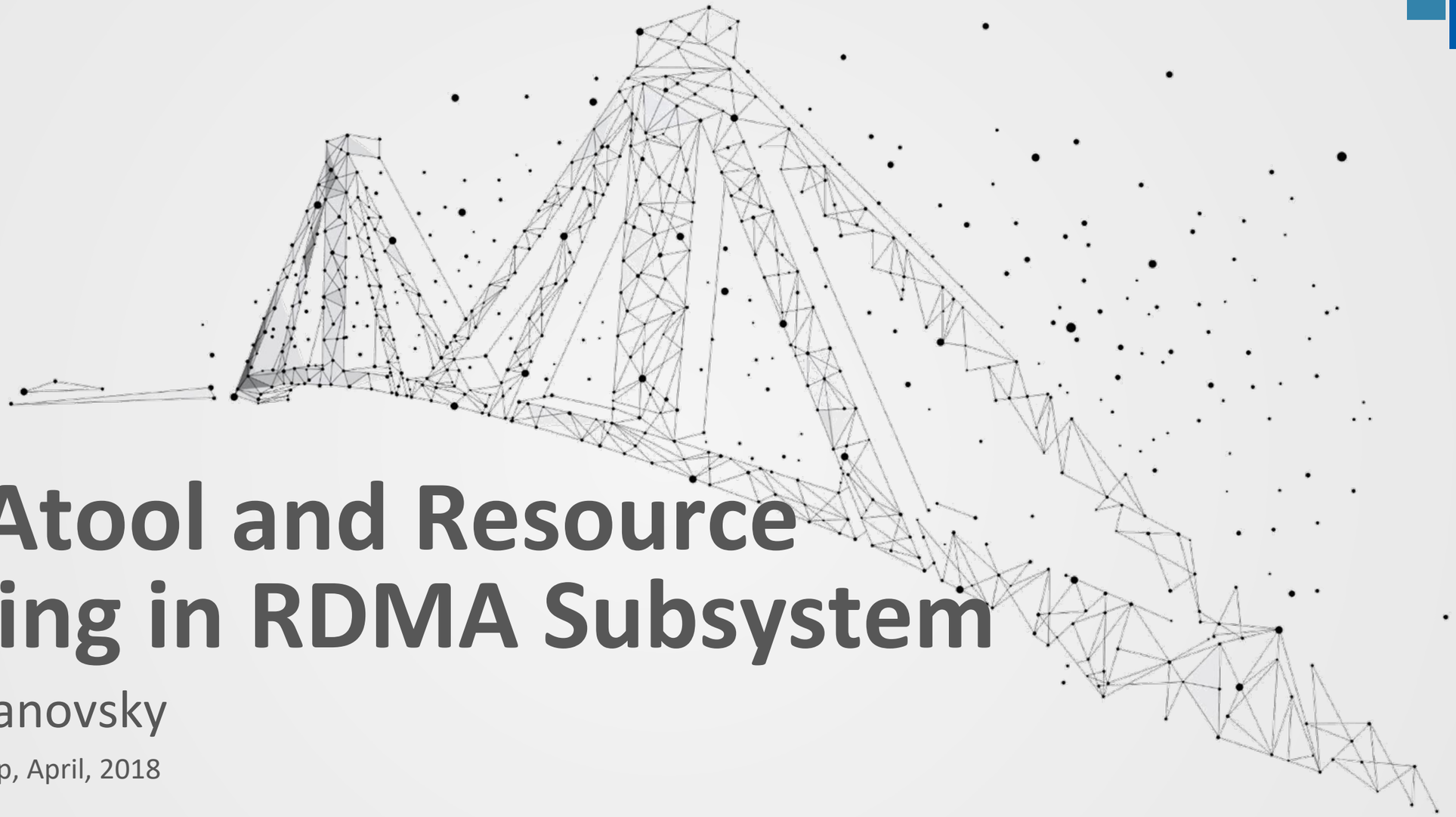


RDMAtool and Resource Tracking in RDMA Subsystem

Leon Romanovsky

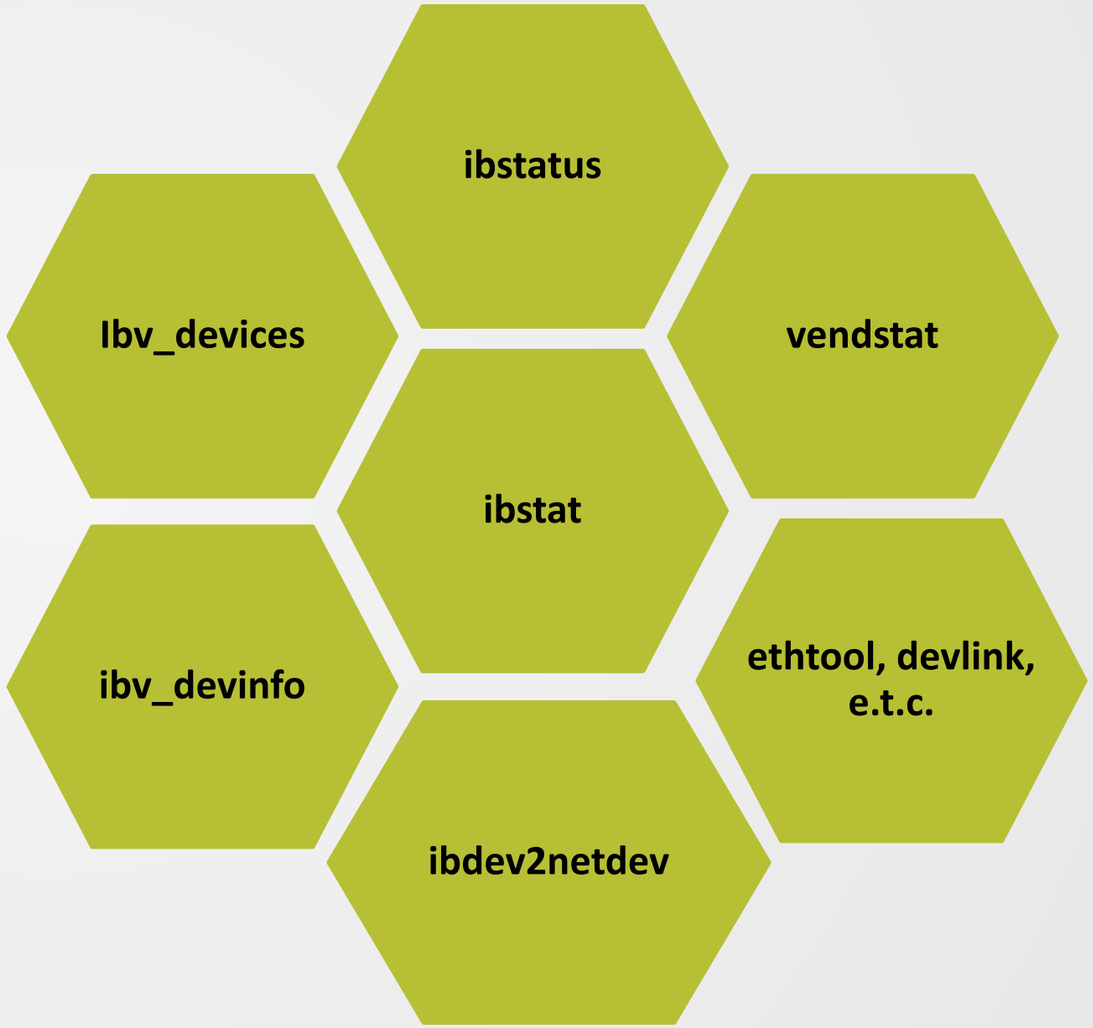
14th OFA Workshop, April, 2018

Agenda

- Why do we need another tool?
- Design goals
- Solution (Kernel <-> User)
- Solution (Kernel)
- Resource tracking
- RDMAtool
- Future plans

Why another tool?

- No open-source solution for RDMA device configuration
- All current query tools are based on verbs/sysfs
- No way to see kernel and user resources in use (a.k.a netstat)
- Need to expose object map for new kABI



Goals

- Unified and simple configuration interface
- Both human and machine input/output UI
- Able to perform batched configurations
- Suitable for containers (can’t change sysfs)
- PID and net namespaces aware
- Accessible to distributions
- Easy back/forward compatibility

Solution (Kernel <-> User)

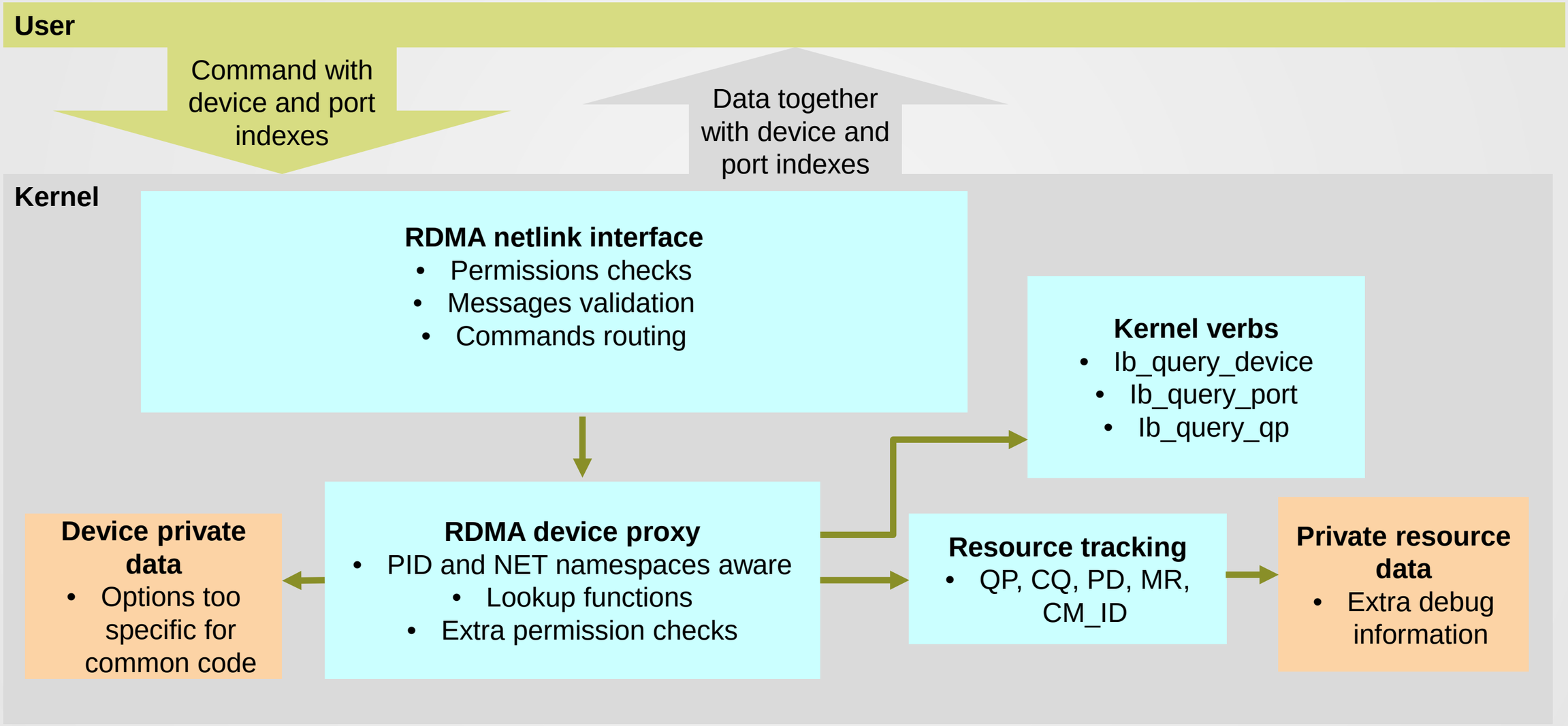
Communication Channel

- Netlink based communication
  - Asynchronous, local UDP messages
  - Supports type validation
  - Extensible by introduction of new attributes
  - Python, C/C++, Go language bindings

Userspace

- Part of iproute2 suite (ip, tc, devlink, e.t.c.)
  - Available in distros
- Follows iproute2 semantics
  - <tool name> <options> <object> <command>

Solution (Kernel)



Resource Tracking

- Generic code to track objects visible to kernel
- Implemented as 8-bit hashtable per-device
- CQ, QP, MR, PD, CM_ID are supported
  - QPs are limited to created by RDMA/core (not driver's internal) and doesn't include XRC
- Distinguish between kernel and user objects
- Provides information of leaked resources on device shutdown

```
[  438.421372] restrack: ------------[ cut here ]------------
[  438.423448] restrack: BUG: RESTRACK detected leak of resources on mlx5_2
[  438.425600] restrack: Kernel PD object allocated by mlx5_ib is not freed
[  438.427753] restrack: Kernel CQ object allocated by mlx5_ib is not freed
[  438.429660] restrack: ------------[ cut here ]------------
```

- Commit 03286030ac04 ("RDMA/restrack: Remove ambiguity in resource track clean logic")

RDMAtool

General

- Part of iproute2
  - https://git.kernel.org/pub/scm/network/iproute2/iproute2-next.git
- Single executable
- Works in place - no need to install anything to try it
- Accepts shortened variants of object names
- “Options” can be placed in any place in the command line

Invocation without arguments

```
➜  iproute2 git:(iproute2-next)  ./rdma/rdma
Usage: rdma [ OPTIONS ] OBJECT { COMMAND | help }
       rdma [ -f[orce] ] -b[atch] filename
where  OBJECT := { dev | link | resource | help }
       OPTIONS := { -V[ersion] | -d[etails] | -j[son] | -p[retty]}
```

RDMAtool Global Arguments

Options

- -b filename
  - Batched operation - every line is command to execute
- -f
  - Don’t stop in case of error during batched execution
- -j
  - Output in JSON format
- -p
  - Beatify JSON output
- -d
  - Detailed output, for example device and port supported capabilities are parsed

Example of Batched File

```
[root@server iproute2]# cat example-of-batch
dev
link show mlx5_1
link show mlx4_0/
```

RDMAtool Device Object

All devices

rdma dev

1: mlx5_0: node_type ca fw 3.4.9999 node_guid 5254:00c0:fe12:3455 sys_image_guid 5254:00c0:fe12:3455

2: mlx5_1: node_type ca fw 3.4.9999 node_guid 5254:00c0:fe12:3456 sys_image_guid 5254:00c0:fe12:3456

Detailed information for specific device

rdma dev **show mlx5_1 -d**

2: mlx5_1: node_type ca fw 3.4.9999 node_guid 5254:00c0:fe12:3456 sys_image_guid 5254:00c0:fe12:3456

caps: <BAD_PKEY_CNTR, BAD_QKEY_CNTR, AUTO_PATH_MIG, CHANGE_PHY_PORT, PORT_ACTIVE_EVENT, SYS_IMAGE_GUID, RC_RNR_NAK_GEN, MEM_WINDOW, UD_IP_CSUM, UD_TSO, XRC, MEM_MGT_EXTENSIONS, BLOCK_MULTICAST_LOOPBACK, MEM_WINDOW_TYPE_2B, RAW_IP_CS>

RDMAtool Link (IB) Object

All devices and all ports

rdma link

1/1: mlx4_0/1: subnet_prefix fe80:0000:0000:0000 lid 0 sm_lid 0 lmc 0 state DOWN physical_state POLLING

1/**2**: mlx4_0/2: subnet_prefix fe80:0000:0000:0000 lid 0 sm_lid 0 lmc 0 state DOWN physical_state POLLING

Detailed information for specific device and specific port

rdma link **-d show mlx4_0/2**

1/2: mlx4_0/2: subnet_prefix fe80:0000:0000:0000 lid 0 sm_lid 0 lmc 0 state DOWN physical_state POLLING

caps: <TRAP, AUTO_MIGR, SL_MAP, SYS_IMAGE_GUID, EXTENDED_SPEEDS, CM, DEVICE_MGMT, VENDOR_CLASS, CAP_MASK_NOTICE, CLIENT_REG>

Other variants

- rdma link show mlx4_0 <- Print all ports for specific device
- rdma link show mlx4_0/ <- Print all ports for specific device

RDMAtool Link (RoCE) Object

All devices and all ports

```
# rdma link
1/1: mlx5_0/1: state ACTIVE physical_state LINK_UP netdev ens4
2/1: mlx5_1/1: state ACTIVE physical_state LINK_UP netdev ens5
```

Detailed information for specific device and specific port

```
# rdma link show mlx5_1 -d
2/1: mlx5_1/1: state ACTIVE physical_state LINK_UP netdev ens5 netdev_index 7
    caps: <CM, IP_BASED_GIDS>
```

IPtool

```
# ip link show ens5
7: ens5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode
DEFAULT group default qlen 1000
    link/ether 24:8a:07:ad:77:b3 brd ff:ff:ff:ff:ff:ff
```

RDMAtool Resource Object

All devices and all ports

```
# rdma res
1: mlx5_0: pd 7 cq 4 qp 4 cm_id 1 mr 1
2: mlx5_1: pd 6 cq 3 qp 3 cm_id 1 mr 0
```

QP information

```
# rdma res show qp pid 0-100,654 type GsI,smi,uD,RC
link mlx5_0/1 lqpn 134 rqpn 0 type RC state INIT rq-psn 0 sq-psn 0 path-mig-state
MIGRATED pid 654 comm ibv_rc_pingpong
link mlx5_0/1 lqpn 132 type UD state RTS sq-psn 16 pid 0 comm [ib_core]
link mlx5_0/1 lqpn 1 type GSI state RTS sq-psn 0 pid 0 comm [ib_core]
link mlx5_0/1 lqpn 0 type SMI state RTS sq-psn 318 pid 0 comm [ib_core]
link mlx5_1/1 lqpn 132 type UD state RTS sq-psn 4 pid 0 comm [ib_core]
link mlx5_1/1 lqpn 1 type GSI state RTS sq-psn 0 pid 0 comm [ib_core]
link mlx5_1/1 lqpn 0 type SMI state RTS sq-psn 0 pid 0 comm [ib_core]
```

RDMAtool Resource Object

MR information

rdma res show **mr**

dev mlx5_0 rkey 0x1207 lkey 0x1207 iova 0x0 mrlen 4096 pid 654 comm ibv_rc_pingpong

PD information

rdma res show **pd** link mlx5_0

dev mlx5_0 local_dma_lkey 0x53503540 users **2** unsafe_global_rkey 0x0 pid **654** comm ibv_rc_pingpong

dev mlx5_0 local_dma_lkey 0x18598c users 0 pid **0** comm **[**rds_rdma**]**

dev mlx5_0 local_dma_lkey 0x18588b users 0 pid 0 comm [ib_srpt]

dev mlx5_0 local_dma_lkey 0x18578a users 0 pid 0 comm [ib_srp]

dev mlx5_0 local_dma_lkey 0xd67ab users 3 pid 0 comm [ib_ipoib]

dev mlx5_0 local_dma_lkey 0x1101 users 0 pid 0 comm [mlx5_ib]

dev mlx5_0 local_dma_lkey 0x1000 users 5 pid 0 comm [ib_core]

RDMAtool Resource Object

CQ information

rdma res show **cq** dev mlx5_0
dev mlx5_0 cqe 511 users 2 pid 654 comm ibv_rc_pingpong
dev mlx5_0 cqe 255 users 0 poll-ctx SOFTIRQ pid 0 comm [mlx5_ib]
dev mlx5_0 cqe 255 users 2 poll-ctx SOFTIRQ pid 0 comm [mlx5_ib]
dev mlx5_0 cqe 2047 users 6 poll-ctx WORKQUEUE pid 0 comm [ib_core]

CM_ID Information

rdma res show **cm_id**
link mlx5_0/**-** state LISTEN ps TCP pid 0 comm [rds_rdma] src-addr 0.0.0.0:18634
link mlx5_1/- state LISTEN ps TCP pid 0 comm [rds_rdma] src-addr 0.0.0.0:18634

RDMAtool Resource Filter Options

Object Type	Filters avaialble
QP	link, lqpn, rqpn, pid, sq-psn, rq-psn, type, path-mig-state, state
PD	dev, users, pid
CM_ID	link, lqpn, qp-type, state, ps, dev-type, tranport-type, pid, src-port, src-addr, dst-port, dst-addr
MR	dev, rkey, lkey, mrlen, pid
CQ	dev, users, poll-ctx, pid

Future plans

- Implement RDMA name persistence
- Configure SIW and RXE
- Various set options
- GID add/delete interface
- Provider specific information
- Convert rdma-core to do device discover over netlink

Thank You