RDMA-CORE UPDATE
Jason Gunthorpe, Dec 3 2020
COMMUNITY

- Maintaining solid velocity in 2020:
  - rdma-core: 749 commits, 22k LOC, from 66 contributors
  - Linux kernel RDMA: 1113 commits, 44k LOC, from 144 contributors
GENERAL
New functionality

- 2M Huge Page support for ODP MRs
  - User must ensure only huge pages are in the MR
- PCIe relaxed ordering bit in TLPs generated via MR (user space only)
- GID inspection API
  - General elimination of sysfs accesses from the library
- RoCEv2 IPv4 entropy bits derived from Flow Label
- More APIs converted to IOCTL format: get_context, get async fd, create/destroy qp/srq/wq
Kernel

- Tracepoints throughout the CM flow and other places
- More syzkaller bug fixes, clean on CM flows now
- Accelerated IPoIB for HFI1
- Deleted FMR support
Kernel Fork and MR

- Linux v5.11 will have improvements to fork and pinned for DMA pages
- Fork will ‘copy on fork’ any pages under DMA
- Ensures the physical page stays with the parent
- Eliminates the need for `ibv_fork_init()` and all the related overhead when working with MRs
- Needs test and confirmation from effected UCX community
RDMA-CORE
New Functionality

- RDMA CM automatic recovery from device hot plug/unplug
- CQ “parent domain” to control memory allocation of CQ rings
- IBA defined Extended Communication Establishment for RDMA CM
  - Allows drivers to exchange device specific details during QP setup. Eg detail about adaptive routing
- Universal query_device_ex
SHARED VERBS CONTEXT

- The ability to share an entire `ibv_context` between two processes
  - Not a security boundary, the whole thing is shared even if only some objects are in use

1. Transfer a `ctx->cmd_fd` to another process - fork, `SCM_RIGHTS`, etc
2. Call `ibv_import_device()` to create a local `ibv_context *` from the FD
   - FD and all resources any process creates exists until all processes using it close
3. Call `ibv_import_pd/mr()` to copy a PD or MR object into this process
   - Eg create a QP on a cross-process PD to allow sharing MR objects
4. Can’t share stateful objects like QP/CQ
DISTROS

- Continuing to support major Linux distributions
  - New rdma-core and kernel components being updated by distros
- GCC10 Link Time Optimization support
  - Becoming the default build mode for distributions
- ‘no man page install’ to support pandoc-less environments, eg spack
- Azure Pipelines CI tracks distros and modern compilers
  - Shared resource with UCX
PYVERBS TEST SUITE

- Growing collaborative effort
- Basic test coverage of verbs APIs
- Already exposing differences between providers
  - Many patches to close these deltas
- Easy to run, minimal setup
- 20 areas of test
DRIVER STUFF

- **Mlx5:**
  - VDPA net format QPs/CQs
  - UAR optimizations, in some cases fewer UARs consumed per context
  - Packet steering and mangling operations

- **Qedr**
  - XRC support