Thesis Background Presentation

Emerson Ford

October 5, 2020

University of Utah School of Computing

Containers

Container Overview

- Increasingly popular framework to distribute and deploy applications.
- Tools like **Kubernetes** have become popular for container orchestration.

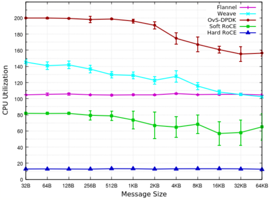
Container Requirements

- Isolation
    - namespaces
    - cgroups
    - network policy
- Portability
    - migration
- Performance
    - low isolation overhead

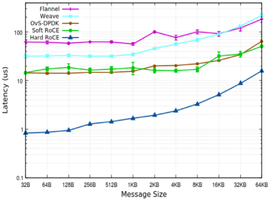
Container Networking Requirements

- Control Plane Policies
  - firewall
  - routing
  - vlans
- Data Plane Policies
  - QoS
  - metering
  - fairness

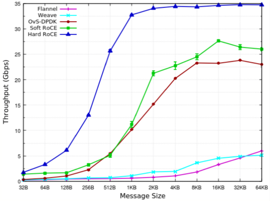
Container Network Overlay Performance



(a) CPU utilization.



(c) Latency.



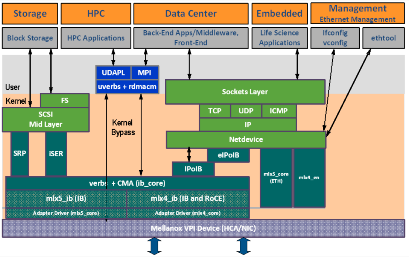
(e) Throughput.

- Current networking isolation requires pretty significant performance sacrifices.
- Less than ideal for HPC applications.

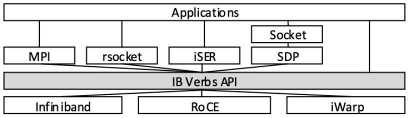
source: A Performance Comparison of Container Networking Alternatives by Ubaid Abbasi, El Houssine Bourhim, Mouhamad Dieye, and Halima Elbiaze

RDMA Overview

- Form of kernel bypass networking
- `libibverbs` is the “narrow waist” of RDMA operations
- Extremely low latency, high throughput

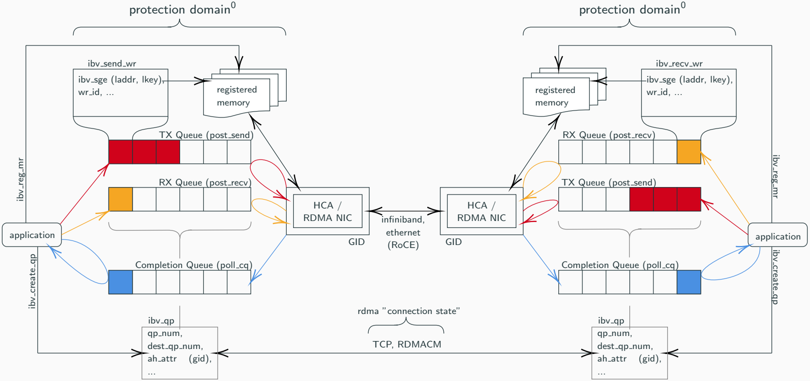


source: NVIDIA MLNX_OFED Documentation Rev 5.1-0.6.6.0



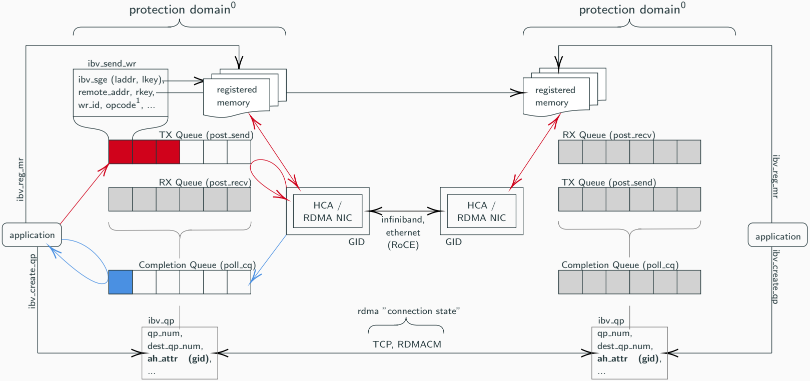
source: FreeFlow Paper Figure 3

RDMA Two-Sided Ops



[0] Every object in the protection domain is mapped in the application's virtual address space. The HCA can access every object in the protection domain.

RDMA One-Sided Ops



[0] Every object in the protection domain is mapped in the application's virtual address space. The HCA can access every object in the protection domain.

[1] opcode is one of `IBV_WR_RDMA_WRITE`, `IBV_WR_RDMA_READ`, `IBV_WR_ATOMIC_CMP_AND_SWP`, `IBV_WR_SEND`

Status Quo

- RDMA significantly improves HPC application performance.
- Containers are quickly becoming a common framework for application distribution and deployment, but container networking isolation is slow.
- **Note**: similar research is being done for RDMA use in VMs in the cloud

Problem Statement

How can we enable the use RDMA in containers while preserving container requirements and performance?

Software Approach

Software Approach Overview

Microkernel / Paravirtualized Approach:

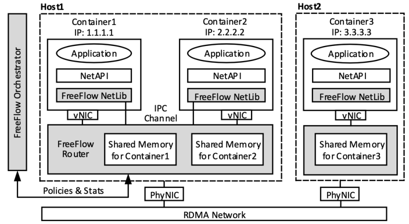
- FreeFlow
- MasQ

Virtualized RDMA:

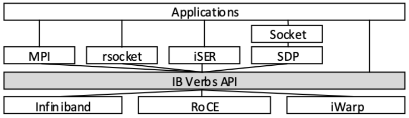
- SoftRoCE

FreeFlow Architecture

- RDMA client (FreeFlow Library / FFL)
- RDMA server (FreeFlow Router / FFR)
- Communicate with IPC and shared memory
- Only need `LD_PRELOAD` to make a FreeFlow compatible application

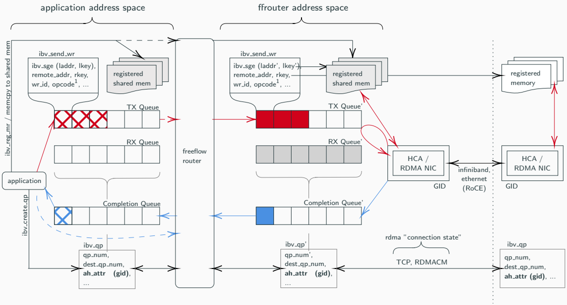


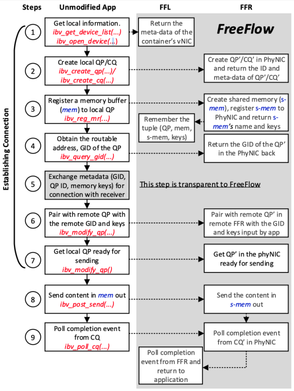
source: FreeFlow Paper Figure 4



source: FreeFlow Paper Figure 3

RDMA Send in FreeFlow





source: FreeFlow Paper Figure 5

FreeFlow Library — Router Communication

- IPC communication
  - Latency can be $\geq 5\mu s$
- Fastpath
  - Move `TX Queue` and `RX Queue` to shared memory with FreeFlow router
  - FreeFlow router spin reads these queue pairs (with cache flushes)

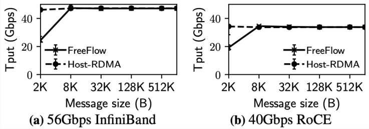
FreeFlow Challenges & Solutions

- Use shared memory to support and speed up one-sided operations.
  - Requires `malloc` highjack to page align memory.
  - New functions `ibv_malloc` and `ibv_free` to avoid this.
  - ffrouter must replace `laddr` with `laddr'`
- Utilize `libibverb`'s built in struct flattening to avoid deep copies in RPC.
- Multiple Unix sockets for parallel RDMA queue pairs to avoid head of line blocking.

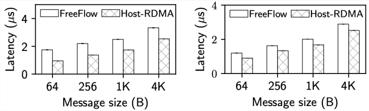
FreeFlow Benefits

- Control plane policy enforcement on queue pairs.
    - QoS and network overlay enforcements
- RDMA vNIC can be assigned a private IP, allowing for non-live container migration.
    - ffrouter can query network overlays to get private IP -> public IP translation.
- FreeFlow library can run TCP over RDMA using `rsocket` in `libibverbs`.

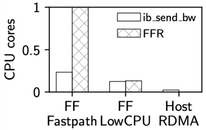
FreeFlow Performance



source: FreeFlow Paper Figure 9



source: FreeFlow Paper Figure 10

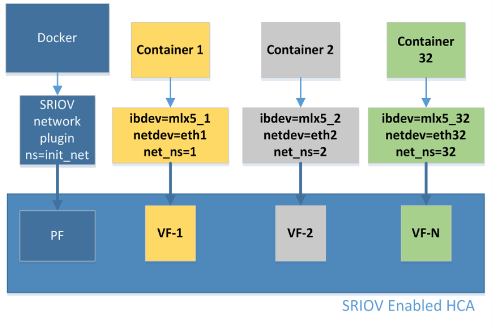


source: FreeFlow Paper Figure 12

- Approx 33% increase in latency for small messages
- Small message sizes bound in tput due to Fastpath single thread bottleneck
- Non-Fastpath CPU util overhead scales with actual load
  - Fastpath requires at least a single CPU core
- Applications finish at close to host RDMA speeds

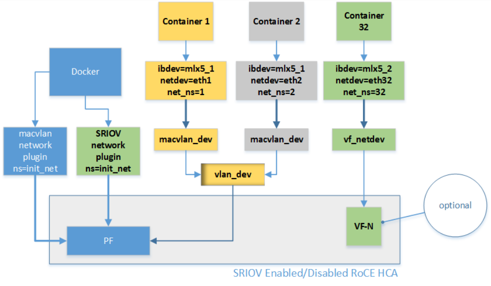
Hardware Approach

SRIOV


Docker
Container 1
Container 2
Container 32
SRIOV network plugin ns=init_net
ibdev=mlx5_1 netdev=eth1 net_ns=1
ibdev=mlx5_2 netdev=eth2 net_ns=2
ibdev=mlx5_32 netdev=eth32 net_ns=32
PF
VF-1
VF-2
VF-N
SRIOV Enabled HCA

- Not portable.
- Control plane policy enforcement relies on switch reconfiguration.

MACVLAN


Container 1
Container 2
Container 32
Docker
ibdev=mlx5_1 netdev=eth1 net_ns=1
ibdev=mlx5_1 netdev=eth2 net_ns=2
ibdev=mlx5_2 netdev=eth32 net_ns=32
macvlan network plugin ns=init_net
SRIOV network plugin ns=init_net
macvlan_dev
macvlan_dev
vf_netdev
vlan_dev
optional
VF-N
PF
SRIOV Enabled/Disabled RoCE HCA

- Control plane policy enforcement relies on switch reconfiguration?
- "GID table entries are created whenever an IP address is configured on one of the Ethernet devices of the NIC's ports."

RDMA Shared Device

- RDMA namespaces
- RDMA cgroups
- ConnectX6 NICs allow for hardware rules?