SolarFlare 10GigE Driver For RAMCloud

Behnam Montazeri Najafabadi
Stanford University

Overview

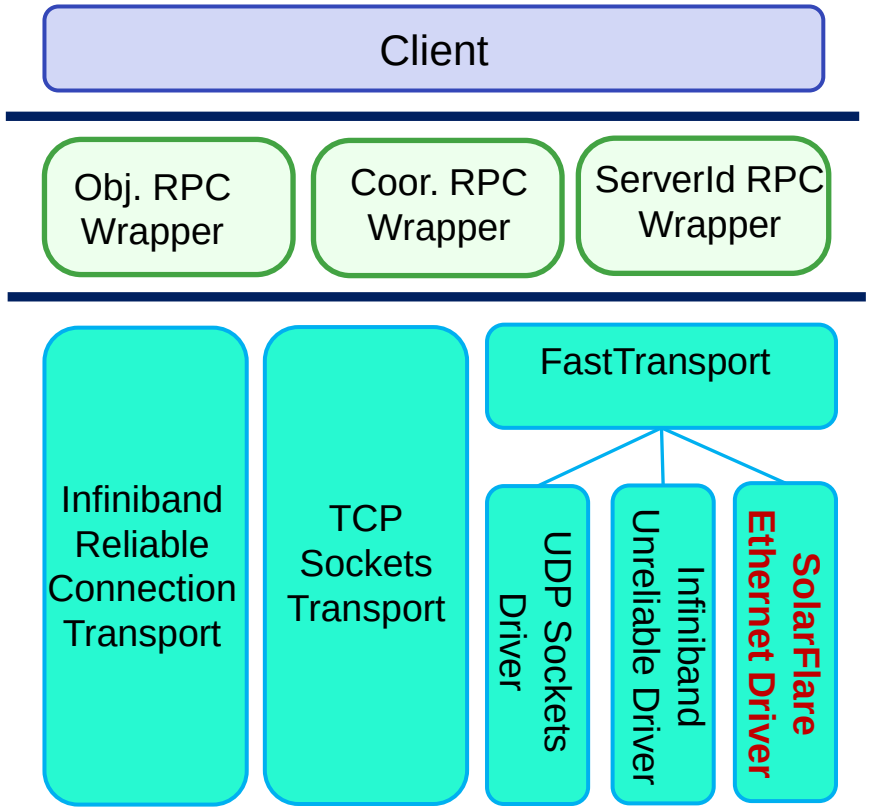
- **Infiniband as RAMCloud's native transport**
- **We want to add support for 10GigE**
  - ✓ Preliminary datagram that hooks into current system
  - ✓ No need to change the higher level code
- **Kernel bypass, our candidate for low latency**
  - ✓ SolarFlare provides kernel bypassed 10G Ethernet
  - ✓ Can use either raw Ethernet drivers or Onload TCP/UDP
- **SolarFlare in RAMCloud**
  - ✓ 9.7 us latency for 100B object, using our driver
  - ✓ 50 us latency for 100B object, using Kernel TCP

Outline

- **Transport layer in RAMCloud**
  - ✓ Overview of transport layer architecture
  - ✓ FastTransport, RAMCloud's home grown transport protocol
  - ✓ Drivers as the APIs to the network interfaces
- **SolarFlare 10Gb/s network interface**
  - ✓ SolarFlare features and hardware acceleration
  - ✓ SolarFalre NIC architecture
- **SolarFlare driver for RAMCloud**
  - ✓ SolarFlare Performance in RAMCloud and limitations
  - ✓ Onload TCP
- **Future work**
- **Conclusion**

Transport Layer in RAMCloud

- **Transport: API for Clients/Servers**
  - ✓ Goal: easy to support different networks and transports
  - ✓ Three-level diagram:
    - Transport:
      - reliable in-order message delivery
      - Client , sends requests
      - Server, replies to the client request
    - Wrappers
      - Collection of classes
      - Pack requests, unpack responses
      - Create sessions
      - Handle callbacks from transports
      - Implement synchronous waiting
      - Throw exception if necessary
    - Client Code



FastTransport

- One possible transport
- Can use any lower level (unreliable) datagram
- Implements reliable, in-order delivery
- Flow controlled protocol
- FastTransport sessions support multiple channels
  - ✓ Each channel supports one outstanding RPC
  - ✓ Multiple channels allow concurrent RPC to a server
- Clients send requests in one or more packets
- Servers respond similarly
- Explicit acks only for packet loss or long requests

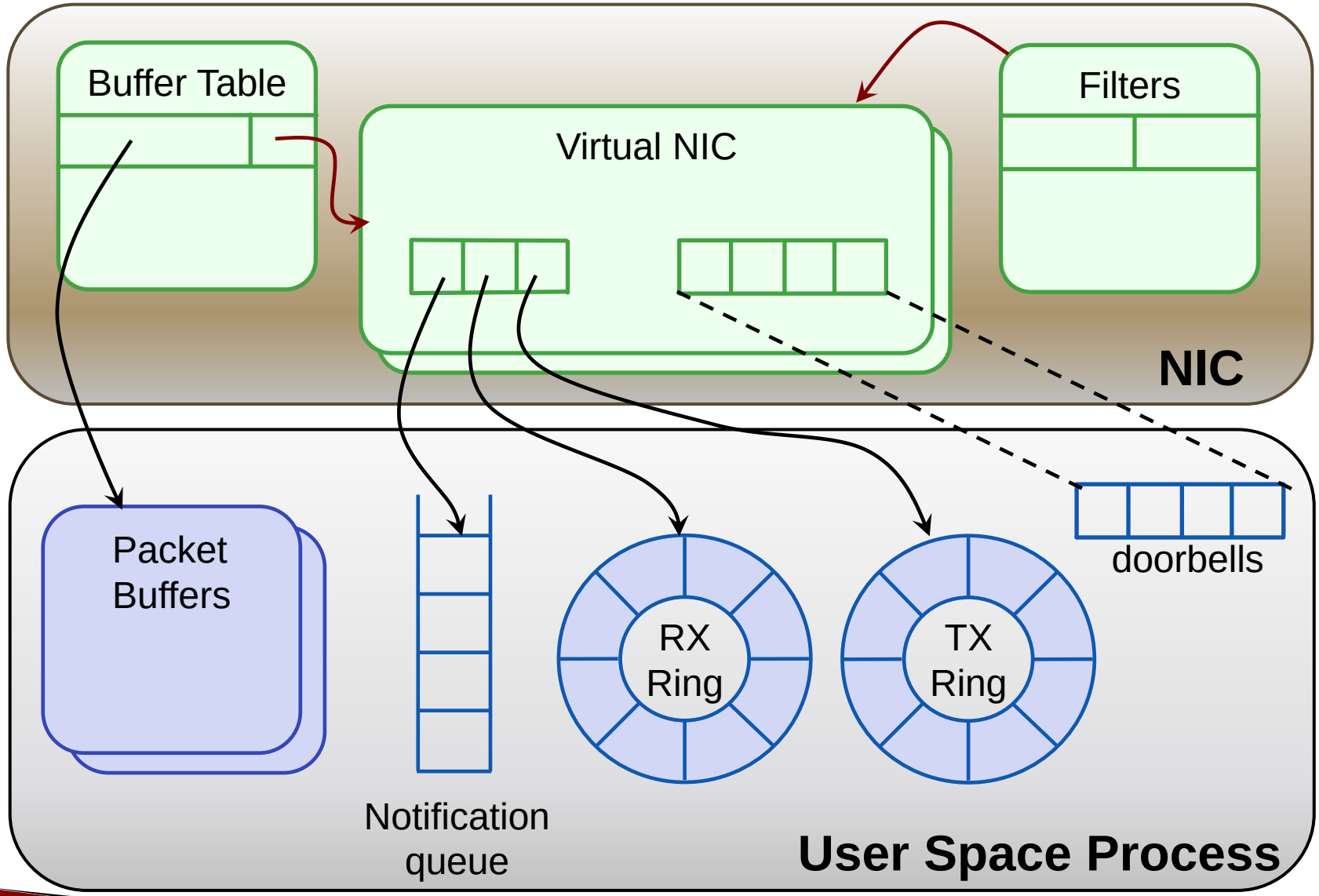
RAMCloud Drivers

- **Drivers are the APIs to the network interface**
- **Implements unreliable datagram for other transports**
- **Interface between FastTransport and Drivers:**
  - ✓ Connect(IncomingPacketHander*)
    - Invoked by transport to associate itself with the driver
    - Provides handler to be invoked when packets arrive
  - ✓ Disconnect(): removes association between driver and transport
  - ✓ sendPacket(address, header, payload)
  - ✓ Poller Object: uses polling approach to check for incoming packets
  - ✓ Address Object: used to name peers
  - ✓ Received Object: contains received packet and sender address
- **Different Drivers: UdpDriver, InfUdDriver, InfEthDriver, SolarFlareDriver**

SolarFlare Feature Set

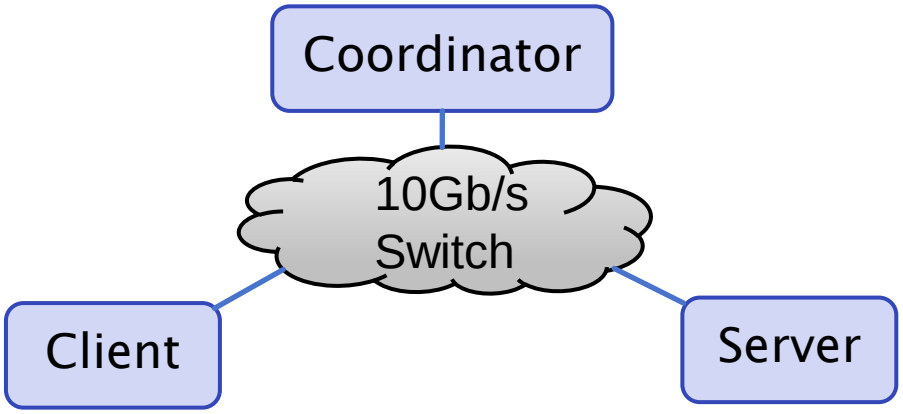
- **Virtual NIC**
  - ✓ Provide kernel bypassed access to the NIC for applications
  - ✓ IP/MAC filters on the NIC steer packets to the correct VNIC
  - ✓ Up to 1024 VNIC
- **Task offload**
  - ✓ Checksum offload
  - ✓ TCP Segmentation Offload (TSO)
- **Jumbo Frames**
- **Support for SR-IOV through virtual NIC**
- **Hardware Timestamps**
- **Receive Side Scaling**
  - ✓ Uses multiple receive queues

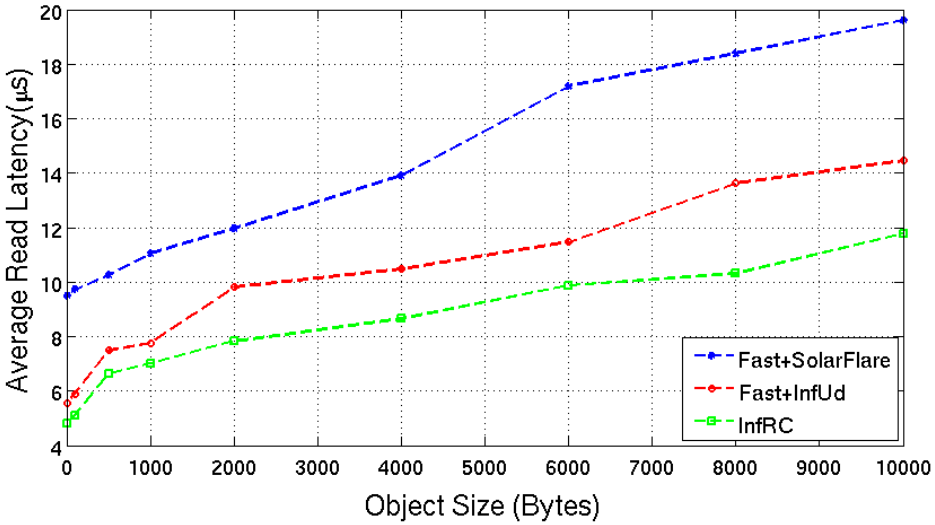
SolarFlare Network Interface

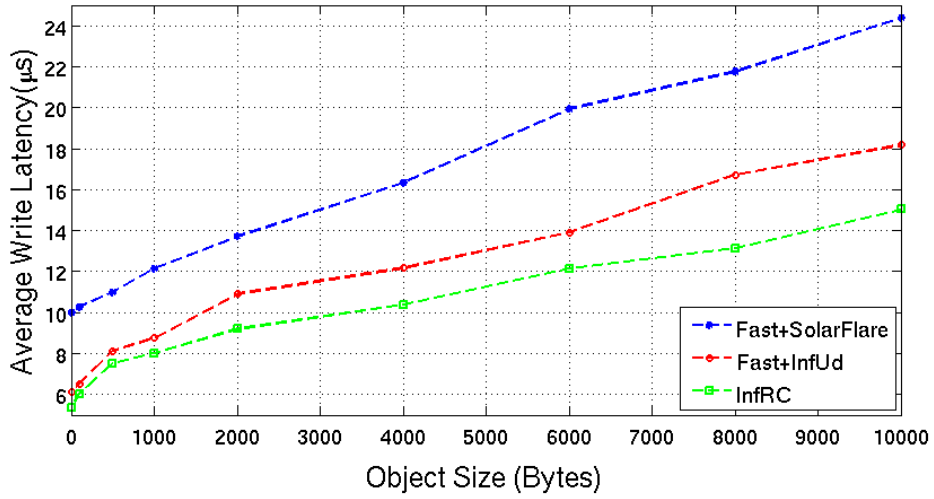


SolarFlare Performance in RAMCloud

- **RAMCloud SolarFlare**
  - ✓ For 100B objects:
    - Fast+SolarFlare latency 9.7us
    - 1.3 us comes from the switch





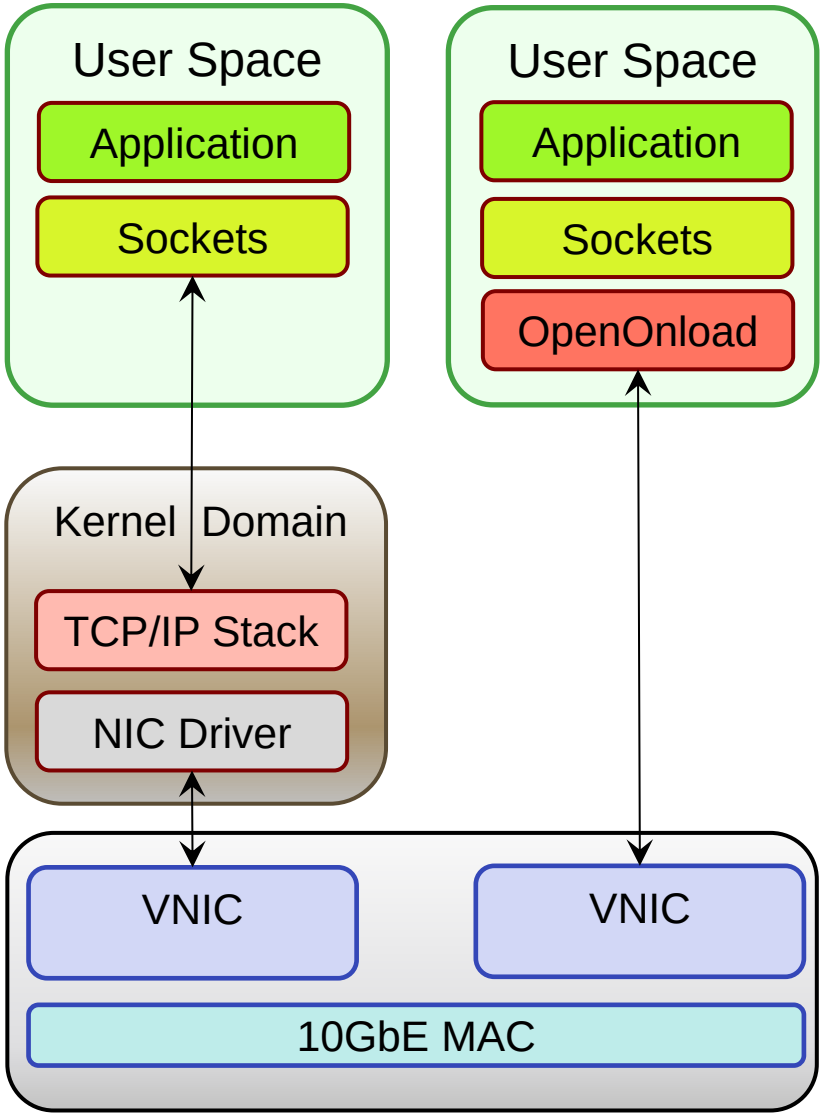


Limitations

- **The current SolarFlare driver is layer 2**
  - ✓ The clients have to provide MAC address
  - ✓ Cluster nodes must be on the same subnet
- **FastTransport is not yet the ideal transport**
  - ✓ Still needs to be improved
  - ✓ Must be tested in a large cluster
- **Next step is to move on to the layer 3**
  - ✓ We need to implement SolarFlare UDP driver
  - ✓ Need to implement ARP and RARP for RAMCloud
  - ✓ Alternatively, we can make use Onload TCP/UDP

Onload TCP

- **Onload**
  - ✓ SolarFlare accelerated network middleware
  - ✓ Dynamically linked to the user address space (kernel bypass)
  - ✓ Implements TCP and UDP over IP
- **Onload TCP ping-pong test outside RAMCloud:**
  - ✓ For 100B object:
    - TCP latency: 8.5us
    - UDP latency: 7.4us



Future Work

- **SolarFlare driver, from layer 2 to layer 3**
  - ✓ Adding ARP and RARP to the driver code
  - ✓ Making use of SolarFlare hardware accelerations (as appropriate)
- **Importing Onload TCP to RAMCloud**
  - ✓ Need to write a new TCP transport code
  - ✓ Must be Onload tunable
- **Analyzing latency overheads in RAMCloud transport**
  - ✓ Driver code and Transport layer
  - ✓ Higher level codes
- **Rethinking RAMCloud RPC**
  - ✓ General purpose
  - ✓ Scale: millions of sessions per server

Conclusion

- **User space, our only candidate for low latency**
- **Preliminary 10GigE support for RAMCloud**
  - ✓ The latency results are promising
- **Lots of room to improve**
  - ✓ FastTransport must be improved and maybe redesigned
  - ✓ Drivers must support layer 3 networking
  - ✓ Onload TCP to be added as a reference