Exploring Linux NFS/RDMA

Shirley Ma and Chuck Lever, Oracle

The following is intended for information purposes, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.

Take-aways

- Performance and scaling opportunities
- How to harden Linux NFS/RDMA
- Moving forward together

Why NFS/RDMA?

- NFS on IPoIB works in Linux, but
  - Significant client-side resource requirements
  - Does not approach link speed

- Permanent storage advances
  - Better, larger caches
  - Persistent memory replacing spinning rust

- *Can NFS/RDMA deliver better reliability, performance, and efficiency?*

Known Implementations

- Linux NFS/RDMA is unmaintained
  - Enterprise distros may support NFS client
  - But upstream, client is now broken
  - Upstream Linux NFS/RDMA server has known panics

- Oracle Solaris 11 NFS/RDMA client and server
  - Actively supported and stable
  - No non-IB RDMA transports

Known Implementations

- Red Hat GlusterFS 3.2 server and client
  - No commercial support
  - NFSv3 only
- NFS-Ganesha server
  - 9p/RDMA, no NFS/RDMA
- Others?

Test Environment

- Hardware
  - 32GB, 6-core single socket, x86-64
  - Single ConnectX-2 QDR
- Software
  - NFS client: Linux 3.8.13 with NFS patches
  - NFS server: Solaris 11 update 1
- Switch
  - QDR InfiniBand

Functional Testing

- NFS functional tests
  - Basic functions - cthon04
  - Interoperability - cthon04, NFStests
    - IPv4/IPv6, endianness
  - Fuzz testing - xfstests

- Challenges:
  - Alternate memory registration modes
  - Common and uncommon HCAs and transports

Performance Testing

- Workload is IOzone
  - NFS share on tmpfs
  - Direct I/O
  - NUMA is disabled

- Metrics
  - Bandwidth
  - Round-trip latency
  - CPU efficiency
  - Interrupt load

Figure 1

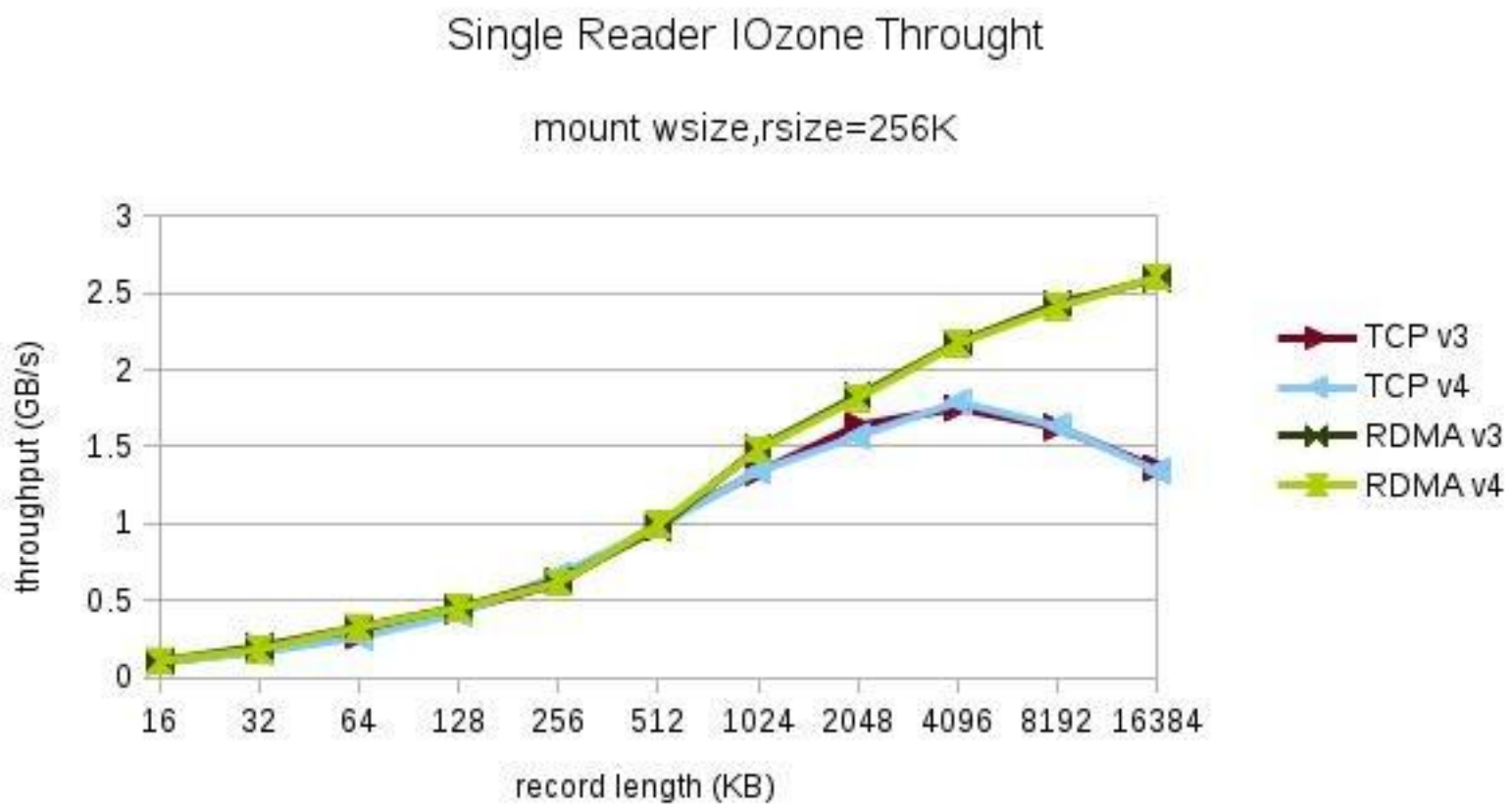

Single Reader IOzone Throught
mount wsize,rsize=256K
throughput (GB/s)
3
2.5
2
1.5
1
0.5
0
16
32
64
128
256
512
1024
2048
4096
8192
16384
record length (KB)
TCP v3
TCP v4
RDMA v3
RDMA v4

Figure 2

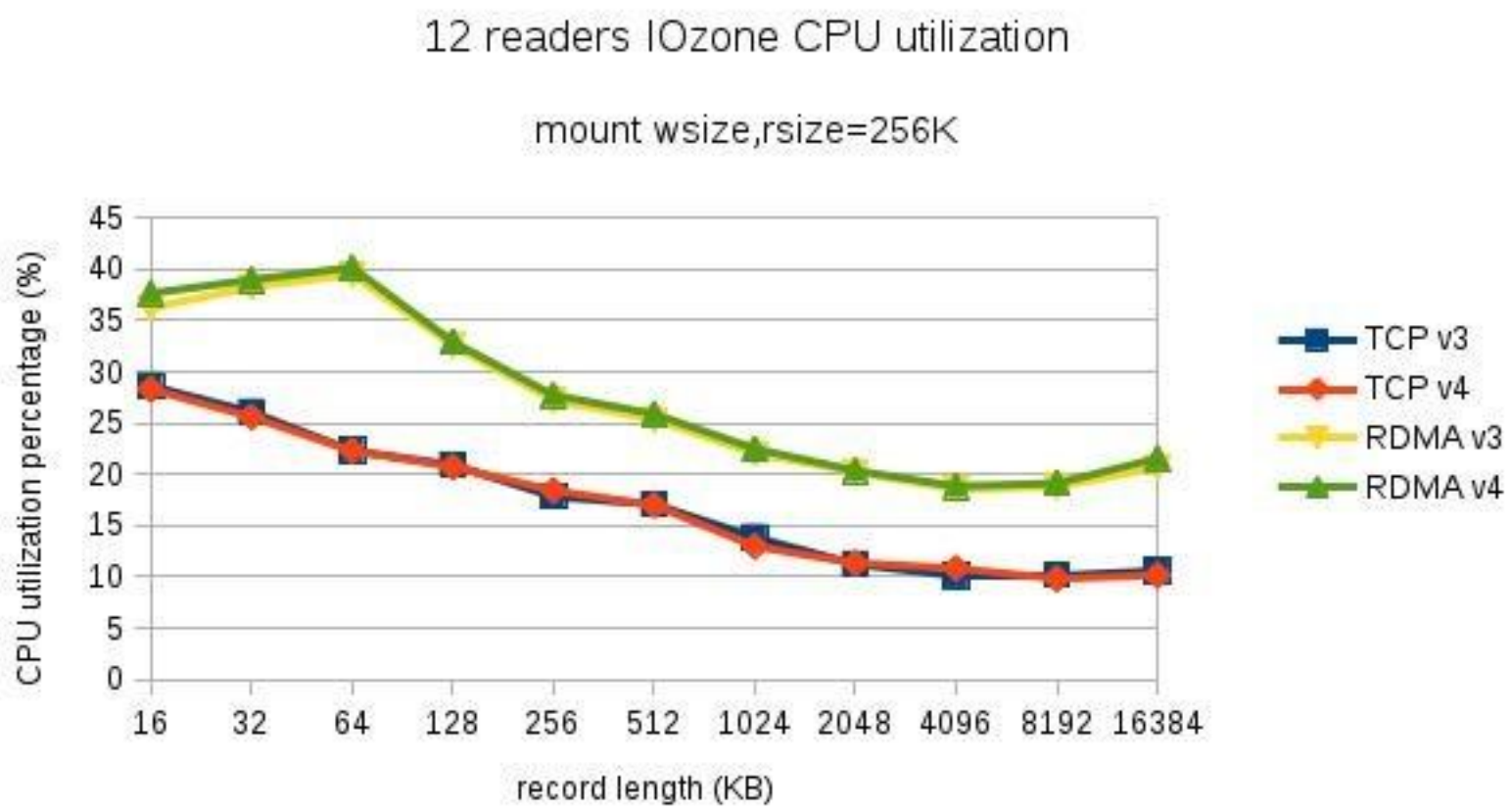
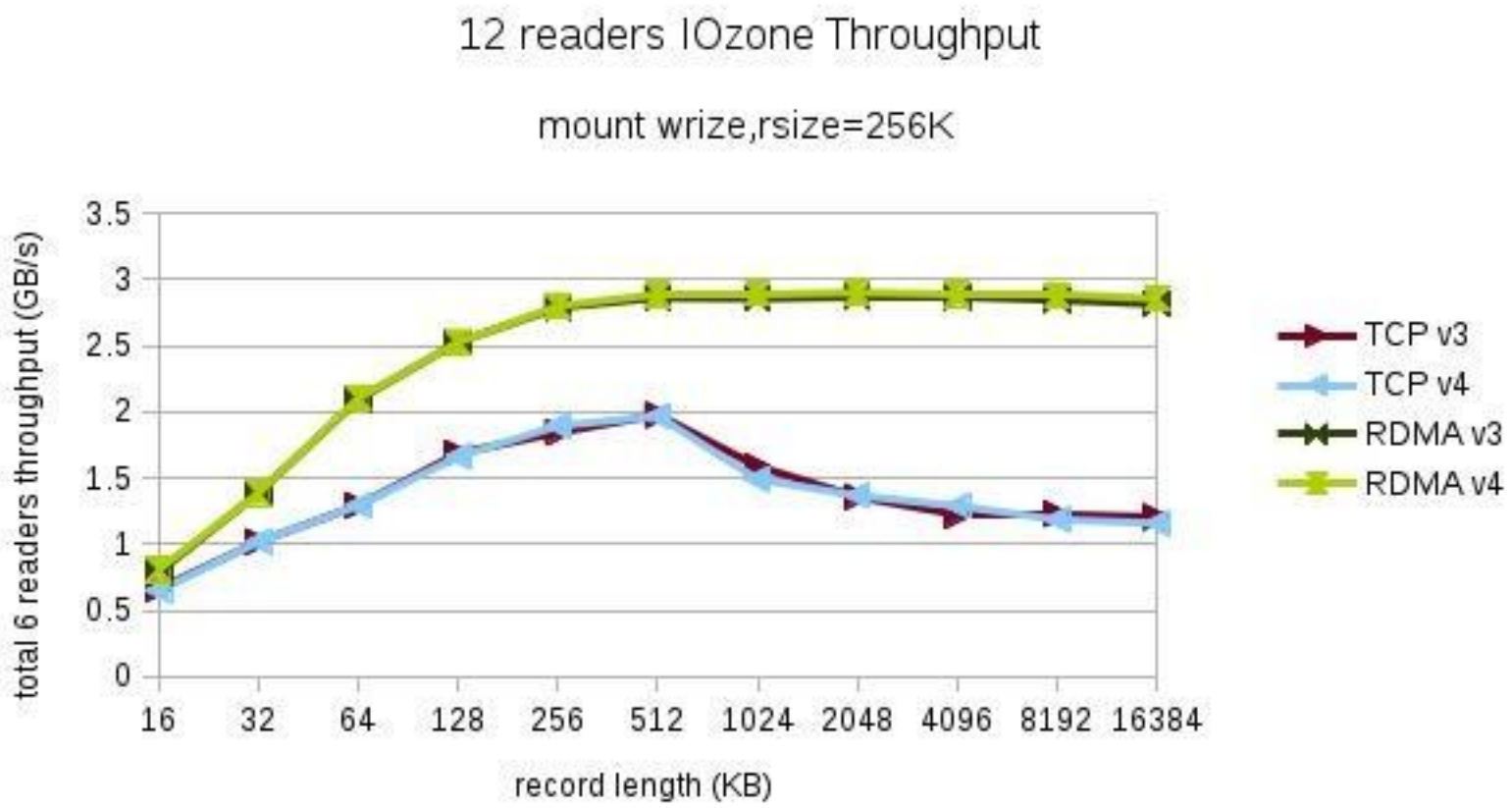

12 readers IOzone CPU utilization
mount wsize,rsize=256K
CPU utilization percentage (%)
45
40
35
30
25
20
15
10
5
0
16
32
64
128
256
512
1024
2048
4096
8192
16384
record length (KB)
TCP v3
TCP v4
RDMA v3
RDMA v4

Figure 3


12 readers IOzone Throughput
mount wrize,rsize=256K
total 6 readers throughput (GB/s)
3.5
3
2.5
2
1.5
1
0.5
0
16
32
64
128
256
512
1024
2048
4096
8192
16384
record length (KB)
TCP v3
TCP v4
RDMA v3
RDMA v4

Performance Opportunities

- Low-hanging fruit
  - Code path length and lock contention analysis
  - Larger maximum rsize and wsize
  - Interrupt mitigation

- Longer term
  - Multiple QPs per RPC transport instance
  - Predictable latency (NUMA)
  - New HCA capabilities

Potential NFS/RDMA Features

- NFSv4.1 - backchannel, pNFS
- NFSv4 referral and FedFS support
- Virtualization - containers, Xen, KVM, qemu

Potential Transport Features

- Alternate transports
  - InfiniBand
    - Legacy HCAs like mthca
    - Current and newer
  - iWARP
  - RoCE

- Connection and NFS server failure handling

Managing the Test Matrix

- Linux NFS/RDMA supports seven memory registration modes
  - Multiplies implementation complexity
  - Introduces administrative complexity
  - Test coverage challenges

- Possible solutions:
  - Remove some memory registration modes
  - Deprecate support for older HCAs

Observability Challenges

- Usual approaches for NFS field troubleshooting:
  - Capture and analyze wire traffic
  - Add code probes

- For NFS/RDMA:
  - ibdump works only for Mellanox HCAs
  - Analysis tools don’t yet dissect RPC/RDMA
  - Code probe bandwidth may be limited

Standards Work

- RFCs 5666 and 5667 (Talpey/Callaghan, 2010)
  - Implementation experience

- Potential protocol enhancements
  - Feature negotiation
  - More efficient READDIR
  - Allow more than one READ chunk per RPC

Opportunities To Contribute

- Continuous testing resources
- Observability tools
- Features, bug fixes
- Flush existing patches to upstream
- Support for upstream Linux NFS/RDMA server

Open Discussion

OPENFABRICS ALLIANCE

10TH ANNUAL INTERNATIONAL DEVELOPER WORKSHOP

Appendix

OPENFABRICS ALLIANCE

10TH ANNUAL INTERNATIONAL DEVELOPER WORKSHOP

Figure 4

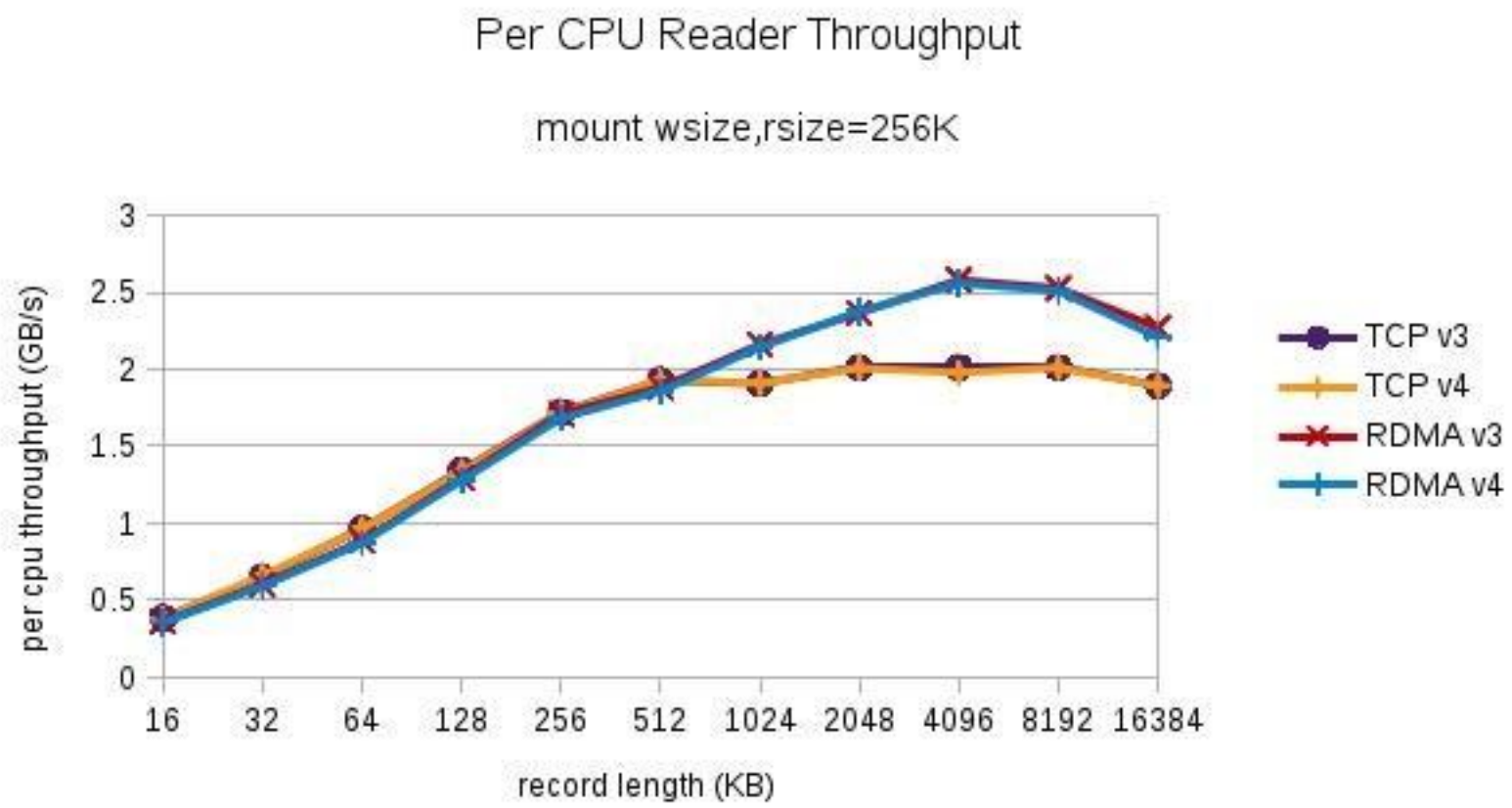
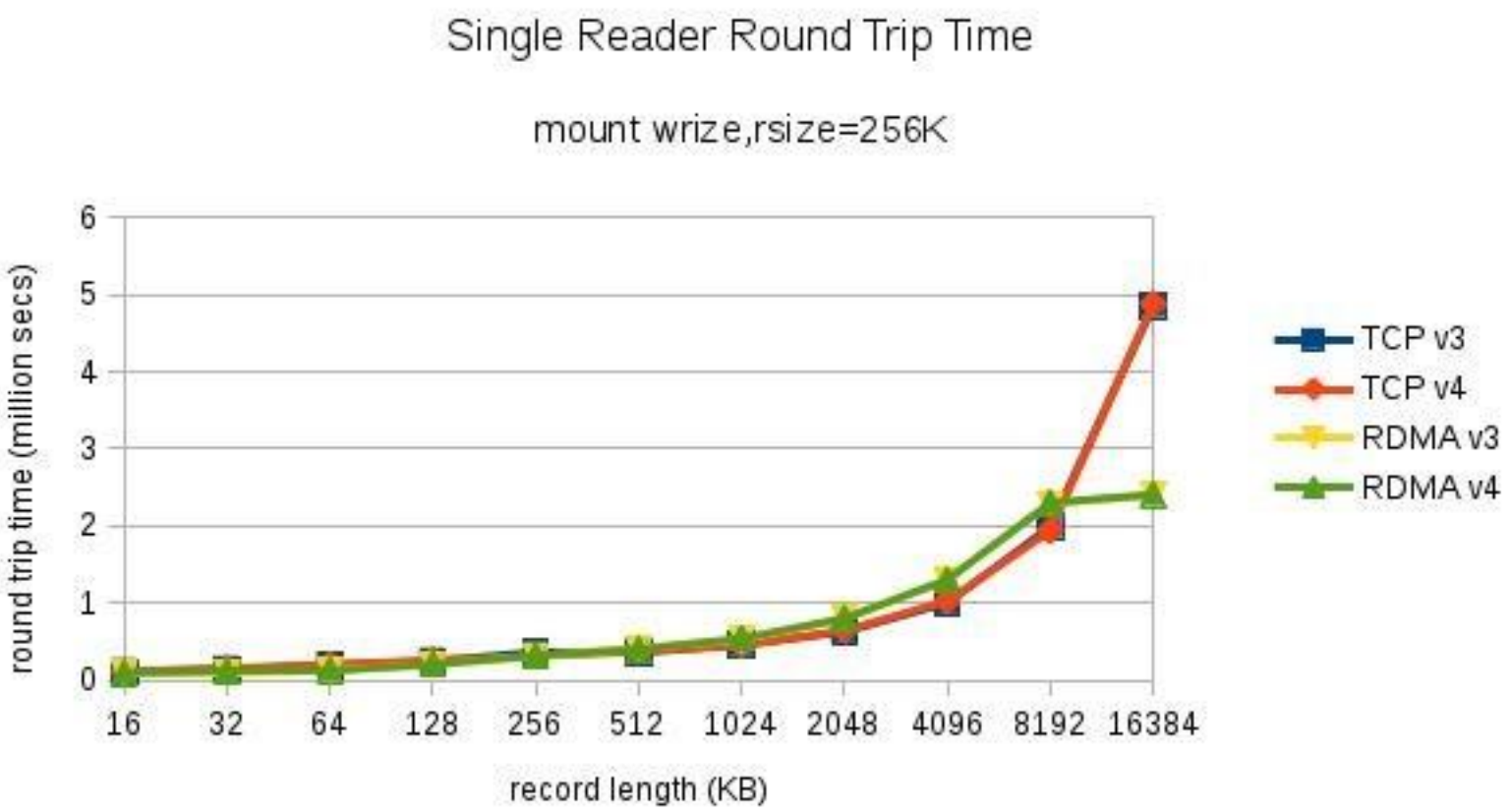

Per CPU Reader Throughput
mount wsize,rsize=256K
per cpu throughput (GB/s)
3
2.5
2
1.5
1
0.5
0
16
32
64
128
256
512
1024
2048
4096
8192
16384
record length (KB)
TCP v3
TCP v4
RDMA v3
RDMA v4

Figure 5


Single Reader Round Trip Time
mount wrize,rsize=256K
round trip time (million secs)
0
1
2
3
4
5
6
16
32
64
128
256
512
1024
2048
4096
8192
16384
record length (KB)
TCP v3
TCP v4
RDMA v3
RDMA v4