NFS/RDMA Implementation(s) Update

Tom Talpey

Network Appliance, Inc.

tmt@netapp.com

NetApp®

Outline

- Review NFS/RDMA Protocol(s)
- Implementation on Linux
- Implementation on OpenSolaris
- Next steps

NetApp®

Review… mostly

What is NFS/RDMA

- A binding of NFS v2, v3, v4 atop RDMA transport such as Infiniband, iWARP
- A significant performance optimization
- An enabler for NAS in the high-end

NetApp®

Benefits of RDMA

- Reduced Client Overhead
- Data copy avoidance (zero-copy)
- Userspace I/O (OS Bypass)
- Reduced latency
- Increased throughput, ops/sec

Followon NFS/RDMA Benefits

- Protocol enhancements and extensions
  - Databases, cluster computing, etc
- Scalable cluster/distributed filesystem
- NFSv4/Sessions interaction and enhancement

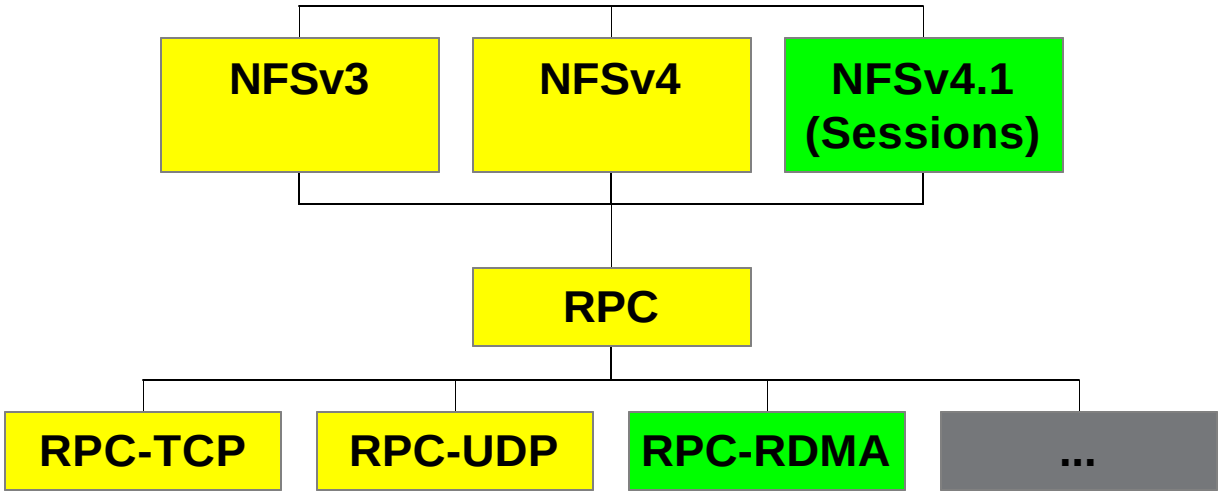
NetApp®

Active protocol specifications

- IETF NFSv4 Working Group
- From the bottom up:
  - RPC/RDMA
  - NFS RDMA binding
  - NFSv4 Transport enhancements
    - (a part of NFSv4.1)
    - Sessions
    - Exactly-once semantics

NetApp®

NFS-RDMA Protocol Stack



NetApp®

RPC/RDMA

- Core RDMA transport binding for RPC in general
- Provides
  - Encoding, etc
  - Inline and Direct (RDMA chunk) transfer
  - Credits
- http://www.ietf.org/internet-drafts/draft-nfsv4-rpcrdma-02.txt (October 2005)

NetApp®

NFS Direct

- NFS binding for RPC/RDMA
- Provides
  - Inline and Direct (RDMA) NFS RPC definitions
  - “What gets chunked”
- http://www.ietf.org/internet-drafts/draft-nfsv4-nfsdirect-02.txt (October 2005)

NetApp®

NFSv4 RDMA and Sessions

- Transport Enhancement for NFSv4
- Provides
  - Session concept
  - Exactly-once semantics
  - General for TCP and RDMA
- http://www.ietf.org/internet-drafts/draft-nfsv4-minorversion1-01.txt (Dec 2005)

NetApp®

NFS RDMA Problem Statement

- IETF Problem Statement for NFS over RDMA
- Provides
  - Rationale
  - Outlines requirements
  - IETF-chartered first step
- http://www.ietf.org/internet-drafts/draft-ietf-nfsv4-nfs-rdma-problem-statement-03.txt

Applying to NFSv3

- Immediate performance benefit
- Straightforward integration with existing implementation
- High market acceptance
- “NFS on Steroids”
- Side protocols (NLM) problematic

NetApp®

Applying to NFSv4+

- Performance
- Enhanced correctness
  - "The goodness of NFSv4"
  - Exactly-once semantics ("EOS")
  - No side protocols / side connections
- Sessions
  - Trunking
  - Failover
  - Efficient resource management
  - (Other benefits from EOS)
  - For both TCP and RDMA

NetApp®

Roadmap

- Early win: NFSv3 on IB
- Prepare the Transport: NFSv4 Sessions
- Employ (*and foster*) iWARP
- NFSv4/RDMA as cluster FS

NetApp®

Linux Implementation

Client Implementation Goals

- Support
  - TCP/UDP/IPv4/IPv6
  - NFS/RDMA
- Support other transports:
  - TOE
  - “Bypass” (pNFS)
- Integrate with core Linux - kernel.org

What's needed

- RPC transport abstraction
  - Allows adding RDMA (and others)
- RPC/RDMA implementation
  - Different for client, server
- Abstract RDMA API
  - Allowing RDMA providers to plug in without changing RPC and NFS each time

NetApp®

Historic Linux RPC support

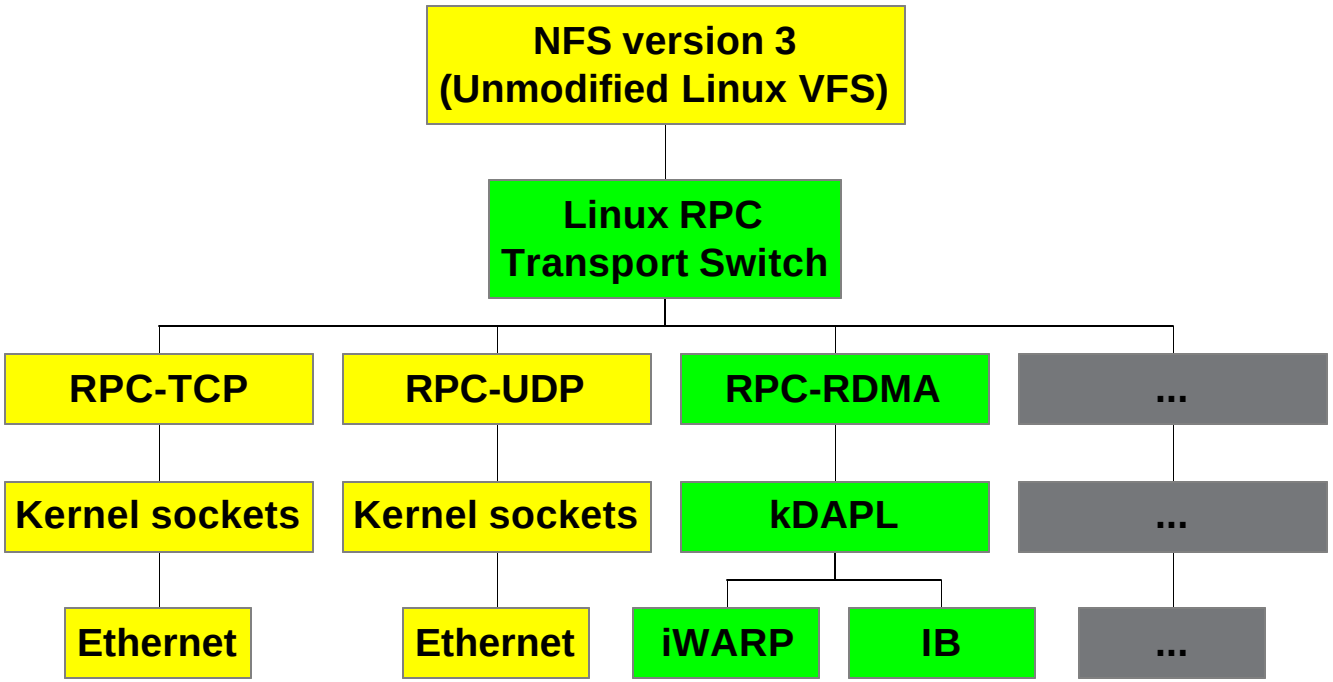
- Single module – sunrpc.o
- Only IPPROTO_{TCP,UDP}
- Only kernel sockets API
- Much specific knowledge roto-tilled:
  - Stream/dgram (framing needed)
  - Connection oriented (reconnect needed)
  - Reliable (retransmit needed)
- Endpoint is 1-1 per xprt (mount)

Solution: RPC Transport Switch

- Abstraction for transport type
- One each for
  - TCP/IPv4
  - UDP/IPv4
  - TCP/IPv6
  - UDP/IPv6
  - RDMA
- NFS mount API extensions

NetApp®

NFS-RDMA
Client Software Stack



NetApp®

New RPC transport switch

- Supports traditional TCP, UDP over IPv4
- Bull IPv6 (TCP, UDP)
- RDMA over OpenIB
  - Infiniband
  - iWARP

NetApp®

Current RPC switch

- Part of Linux 2.6.15
- http://troy.citi.umich.edu/~cel/linux-2.6/2.6.15/
  - [Many rolled-up NFS improvements in CEL_NFS-ALL.patch]
  - Simplified from earlier versions
- Abstracts transport type, address family, per-xprt parameters, etc.

Client Transport Switch Vector

```
struct rpc_xprt_ops {
    void        (*set_buffer_size)(struct rpc_xprt *xprt, size_t sndsize, size_t rcvsize);
    void        (*print_addr)(struct rpc_xprt *xprt, unsigned int format, char *buffer, size_t size);
    int         (*reserve_xprt)(struct rpc_task *task);
    void        (*release_xprt)(struct rpc_xprt *xprt, struct rpc_task *task);
    void        (*rpcbind)(struct rpc_task *task, struct rpc_clnt *clnt);
    void        (*set_port)(struct rpc_xprt *xprt, unsigned short port);
    void        (*connect)(struct rpc_task *task);
    void *      (*buf_alloc)(struct rpc_task *task, size_t size);
    void        (*buf_free)(struct rpc_task *task);
    int         (*send_request)(struct rpc_task *task);
    void        (*set_retrans_timeout)(struct rpc_task *task);
    void        (*timer)(struct rpc_task *task);
    void        (*release_request)(struct rpc_task *task);
    void        (*close)(struct rpc_xprt *xprt);
    void        (*destroy)(struct rpc_xprt *xprt);
    void        (*print_stats)(struct rpc_xprt *xprt, struct seq_file *seq);
};
```

Transport Hooks

- Each transport registers with switch
- NFS mount (and others) specify transport type and per-transport create data
- Transport gets control via xprt_procs, and network events
- Can unregister/unload

Transport switch mount API extensions

- At a minimum, pass transport type and addresses, NFS generic mount parameters
- Maintain per-transport arguments passed separately, and extensibly
- Mount API already under way as part of core NFS work

Client RDMA Implementation

- RPC/RDMA module
  - 3000 lines of code, 2 headers, 3 C files
- RDMA hardware providers
  - Available for several vendors
  - Also under way within OpenIB

NetApp®

Linux Client RDMA Implementation

- Available as open source
  - Dual GPL/BSD-style license
  - http://www.sourceforge.net/projects/nfs-rdma
- Linux 2.6.15 supported
  - Requires additional RPC switch patch – 2.6.15
- 2.4 Linuxes (“old” RPC switch):
  - RedHat 7.3 (2.4.18)
  - SuSE 8 Enterprise (2.4.19)
  - RHEL 3.0 (2.4.21)

Linux Server RDMA Implementation

- Has moved from CITI to Open Grid Computing
- Separate code from client
  - Not transport-switch based
- NFSv3 initially
  - v2, v4 to be supported
  - “should work”
- Full RDMA semantics

NetApp®

Server codelines

- UMich CITI early code available
  - Linux 2.6.9
  - http://www.citi.umich.edu/projects/rdma
  - No further development on this code

Server codelines

- OGC code to be released
  - Linux 2.6.15+
  - Fully functional protocol implementation
  - Early prototype for comment!
  - Hopefully next week
- http://www.sourceforge.net/projects/nfs-rdma

NetApp®

Open Solaris Implementation

Ohio State University

- Joint supported effort between Sun Microsystems and Network Appliance
- Network-Based Computing Lab
  - http://nowlab.cse.ohio-state.edu/ (Dr Panda's group)
  - MPI over RDMA, Infiniband, etc

OpenSolaris work scope

- Fully functional and conformant client and server implementations
- Operation on Infiniband
- Interoperable with Linux, etc.
- To be released as an OpenSolaris public project

NetApp®

OpenSolaris Server project status

- Currently in design phase
- Original Solaris prototype available as reference, but requires changes
  - Relies on inefficient (and illegal for NFS) “read-read” RPC/RDMA transfer
  - Needs large (>=256KB) transfer sizes to yield major benefit
  - RPC API efficiency restrictions

NetApp®

OpenSolaris Server project status

- New analysis of Open Solaris RPC/XDR API and marshaling
- Implement RPC/RDMA protocol optimally
- Use new OpenSolaris Infiniband framework
- Infiniband connection model updates
  - ATS (Address Translation Service)
  - IPoIB
  - IBTA SWG port / serviceid mapping

NetApp®