NFS/RDMA Implementation

Tom Talpey
Technical Director
Network Appliance, Inc.
tmt@netapp.com

Outline

- Overview/standards
- Linux implementation
- Development efforts

What is NFS/RDMA

- NFS/RDMA is:
  - An RPC-layer protocol that allows NFS to use RDMA networks (such as Infiniband and iWARP)
  - A transparent solution for applications, NFS protocol features, and NFS users
  - A significant performance boost to clients
    - Reduces client CPU overhead
    - Utilizes high-bandwidth, low-latency fabrics
  - A single-wire host cluster solution

Background

- Protocol is IETF NFS WG task
- Originally published in 2003
- Two specifications:
  - RPC/RDMA:
    - Defines RDMA transport
    - Specifies binding of any RPC protocol
  - NFS Direct:
    - Defines NFS v2, v3, v4 aspects
- See:
  - http://www.ietf.org/html.charters/nfsv4-charter.html

Implementations

- All on NFS**v3**
- Linux client
  - Infiniband and **iWARP**
- Linux server
  - (under development)
  - Infiniband and iWARP
- Network Appliance Server
  - Infiniband (prototype)
- Solaris 10 client and server
  - Infiniband

Dependencies

- RDMA fabric(s)
  - Infiniband
  - iWARP
- RDMA support in host
  - OpenIB, kDAPL, etc
- RPC/RDMA support in host
- Infrastructure
  - Server(s)
  - Fabric management, naming, admin, etc

RDMA Fabrics

- Infiniband
  - **Significant** presence in the recent market
  - Inexpensive, low-latency cluster “spine”
  - High adoption rate in HPC
- iWARP (TCP/IP RDMA)
  - **Emerging** presence in the market
  - Largely gated by 10GbE availability
  - Numerous vendors preparing products

Infrastructure

- Infiniband
  - Working in OpenIB and some Linux distros
  - Requires IB network, switches, subnet manager, etc.
  - A good deal of setup, but readily do-able
- iWARP
  - Working in vendor-provided packages
  - Working on OpenIB (as of 10/16!)
  - Reuses existing TCP/IP services/setup

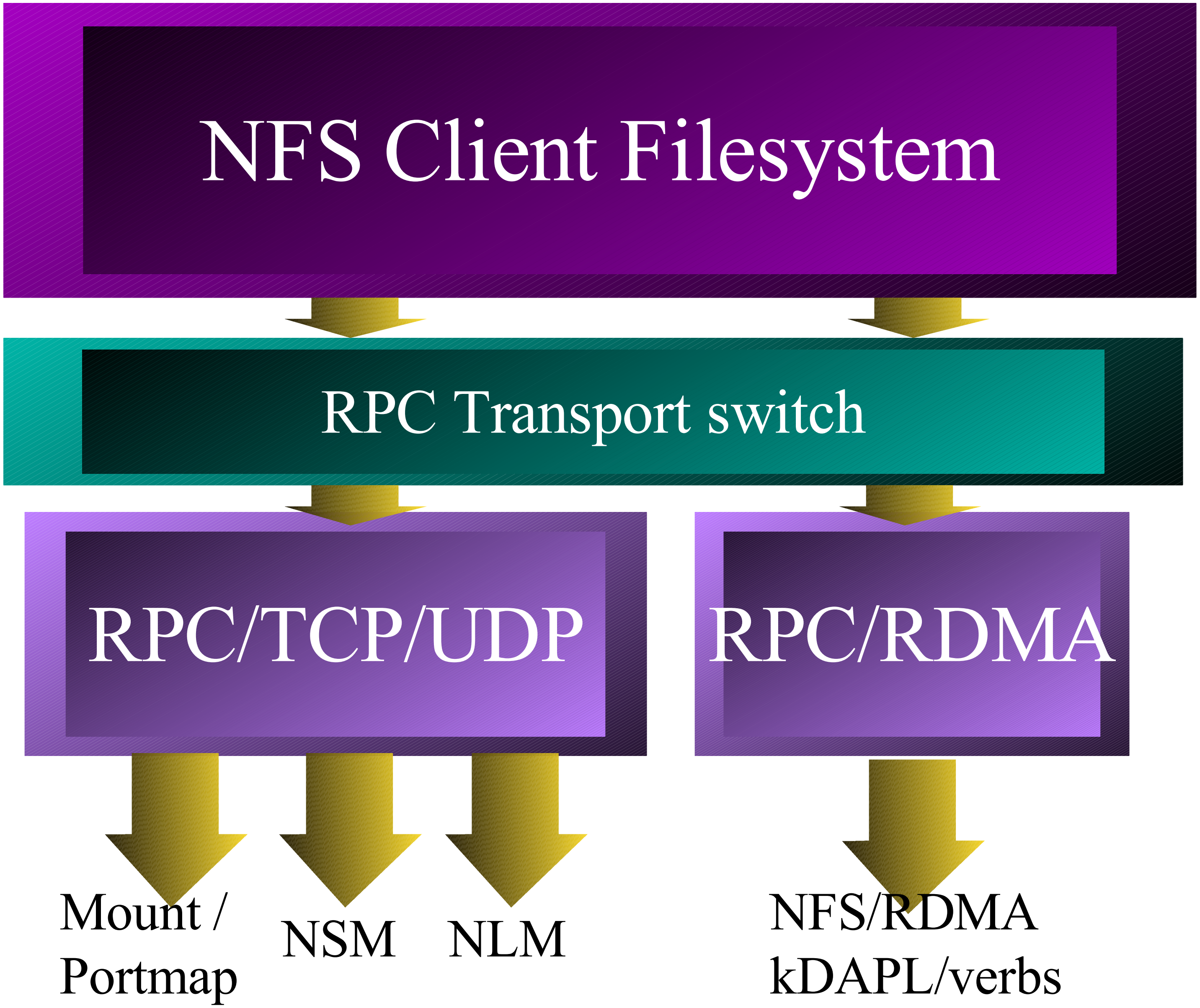
RDMA Framework

- Linux RDMA APIs
- OpenIB
  - Open-source Linux IB framework
  - Committed to 2.6.11+
  - Being extended to support iWARP
- Verbs
  - Low-level RDMA API
  - Kernel and user versions
- kDAPL
  - Historical RDMA portable API
  - Being phased out for Linux

RPC/RDMA Support

- Linux
  - Prototype (working!) client implementation since 2004
  - Server implementation under development
  - Currently developing on kDAPL
    - Works on Infiniband **and** iWARP
  - Plan to move to OpenIB verbs when iWARP is also supported
- Solaris
  - Solaris 10 supports client and server

NFSv3/RDMA Client stack



NFS Client Filesystem

RPC Transport switch

RPC/TCP/UDP

RPC/RDMA

Mount / Portmap

NSM

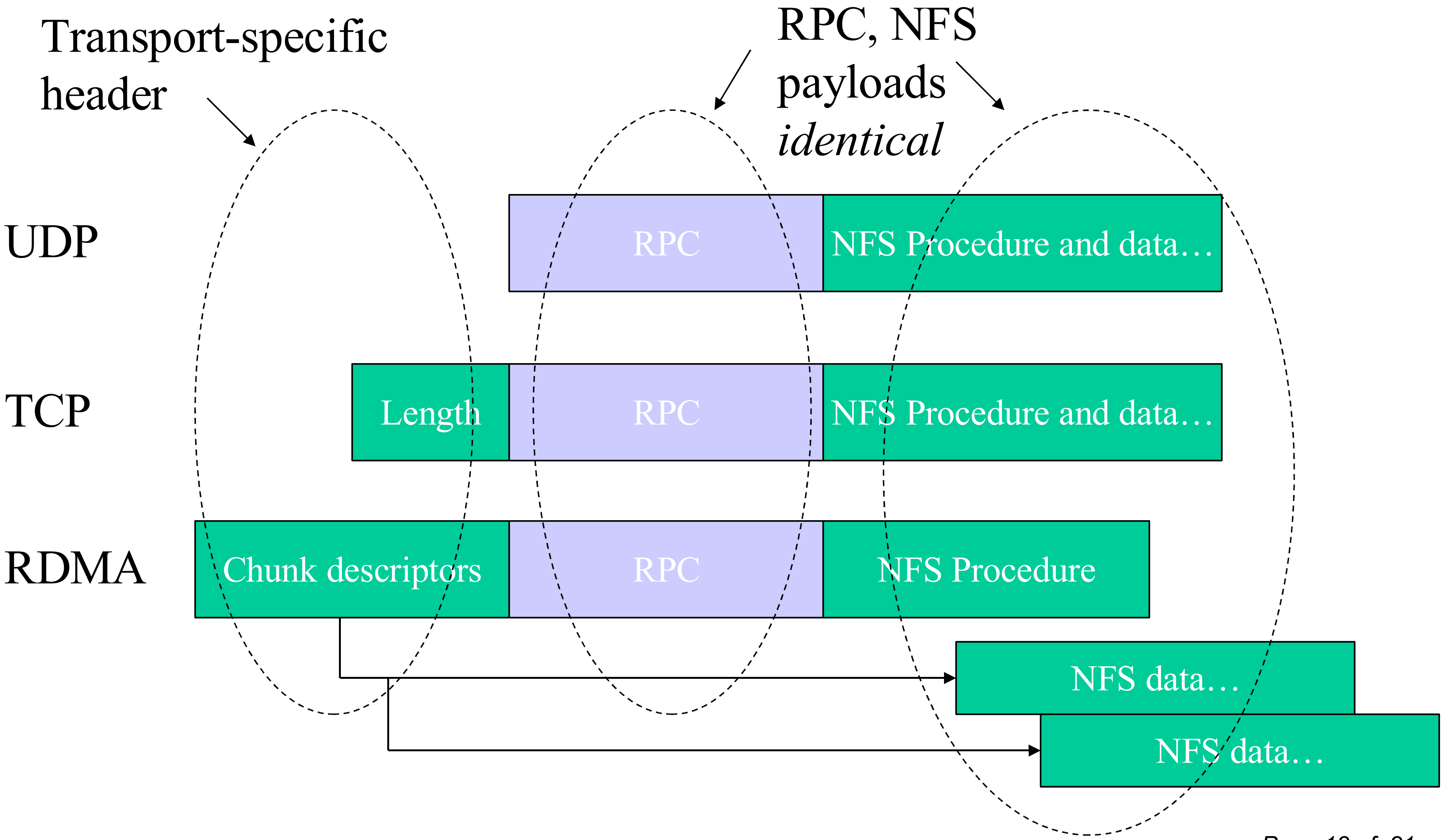
NLM

NFS/RDMA kDAPL/verbs

Linux RPC Transport Switch

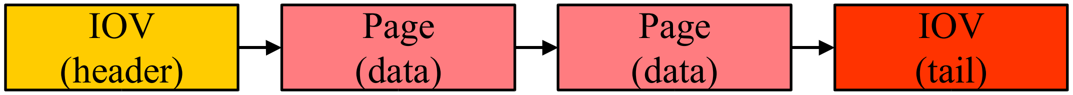
- Abstraction for multiple transports
  - IPv4 UDP/TCP
  - IPv6 UDP/TCP
  - RDMA, etc
- Extensible, configurable framework
  - Dynamically loadable, mount-driven
- Destined for mainline kernel
  - Implemented as patchset for many Linux versions
  - http://troy.citi.umich.edu/~cel/linux-2.6/2.6.13/release-notes.html

Transport RPC format



RPC/RDMA Transport

- Implemented transparently to NFS!
  - < 4K Lines Of Code
- Linux passes “iov” list down to xprt



IOV (header) → Page (data) → Page (data) → IOV (tail)

- RPC/RDMA module translates iov’s to chunks
  - RPC headers (first iovec) sent as inline
  - Data sent/received as RDMA chunk (0-copy 0-touch), or optionally copied inline
- Manages connections, etc

Linux Client status

- Fully functional client working on
  - 2.4.x
  - 2.6.x
  - Transparent VFS with caching
  - Also supports uncached zero-copy, zero-touch with O_DIRECT
- Implemented as RPC Transport
- Release is planned when Server and RPC Transport switch are available

Linux Server status

- Under development at UMich/CITI
  - Sponsored by NetApp and SGI
- Currently in “Phase 2”, basic connection and RPC exchange
  - Working on IB and iWARP
  - Just 2K Lines of Code!
- Phase 3 (functional completion) by end of year, with full RDMA Read and Write transfers
- Planned demo at Supercomputing 2005 in November (Seattle)

Linux Server Implementation

- Does not use a switch abstraction
  - Server must listen on all transports
  - Each request processed per-endpoint
- Does use iovec approach
  - Similar to client, decodes/encodes from/to RPC buffers
  - Minimal upper-layer changes
- Some transport ramifications
  - E.g. requires export IP address checking
  - OpenIB Infiniband API does not provide all these (yet – kDAPL does)

Other Linux Server stuff

- Multiple “credits” support is under development
  - No real performance numbers until this
- Currently developing on Ammasso iWARP hardware
  - Using Ammasso kDAPL
- Working status on OpenIB kDAPL
  - No major issues, but disruptive to our work
- Moving later to OpenIB native verbs

Opportunities

- Clusters!
  - High Performance Computing
  - Scientific computing
  - Financial apps
  - Databases
- NFS/RDMA provides a one-wire storage service
  - With full **transparency** and **sharing**
  - And very high performance

Summary

- The framework is (finally) coming into place to enable NFS/RDMA
- There are proven benefits
  - Performance, sharing, etc
- There is huge interest in Storage Services over cluster fabrics
- Linux will enable the adoption

Questions?

- tmt@netapp.com

- http://www.ietf.org/html.charters/nfsv4-charter.html
- http://www.citi.umich.edu/projects/rdma
- http://troy.citi.umich.edu/~cel/linux-2.6/2.6.13/release-notes.html
- Or, soon to a Linux kernel near you!