A Roadmap for NFS on RDMA

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

- NFS/RDMA Background and Value
- Deployment: NFSv3/NFSv4/DAFS/NFSv4+
- A Brief Rant
- Timeline
What is NFS/RDMA
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
Benefits of NFS/RDMA

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

• Protocol enhancements and extensions
  - Databases, cluster computing, etc

• Scalable cluster/distributed filesystem

• As we raise the “NAS bar”, the protocol should express richer semantics
What has been proposed

- RPC/RDMA
- NFS binding
- NFS Transport enhancements
  - Sessions
  - Exactly-once semantics
Document: RPC/RDMA

- Core RDMA transport binding for RPC in general
- Provides
  - Encoding, etc
  - Inline and Direct (RDMA chunk) transfer
  - Credits
Document: NFS Direct

- NFS binding for RPC/RDMA
- Provides
  - Inline and Direct (RDMA) NFS RPC definitions
  - “What gets chunked”
Document: 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-talpey-nfsv4-rdma-sess-00.txt
Document: NFS RDMA Problem Statement

- IETF Problem Statement for NFS over RDMA
- Provides
  - Rationale
  - Outlines requirements
  - IETF-chartered first step
The Documents Together:

- Form the basis for a complete NFS over RDMA solution
- All NFS versions, and general RPC
- Do not fundamentally propose new NFS features (but enable a few)
NFS/RDMA Deployment
Where does it apply

- Well, everywhere, but especially…
- Datacenter apps
  - Databases
  - Clusters
  - Middle-tier
- Collaboration
- Scientific computing
- Extends the traditional NAS environment (upward)
Applying to NFSv3

- Immediate performance benefit
- Straightforward integration with existing implementation
- High market acceptance
- “NFS on Steroids”
- Side protocols (NLM) problematic
Applying to NFSv4

- Performance
- Enhanced correctness over v3
  - “The goodness of NFSv4”
- All side protocols over common transport
Applying to NFSv4+

- Further enhanced correctness
  - Exactly-once semantics ("EOS")

- Sessions
  - Trunking
  - Failover
  - Efficient resource management
  - Atomic Append (possible from EOS)
  - For both TCP and RDMA
The not-so-distant Futures

- Protocol support for
  - Databases
  - Local sharing applications
  - Clusters
  - Application semantics
    - Cache hints
    - Batch I/O
    - Open unlinked, Delete on Last Close
    - Fencing, etc etc
Relationship to DAFS

- There are two “perpendicular” aspects to DAFS, which we approach separately:
  - Performance (derived from RDMA)
  - Semantics (derived from Protocol)
Relationship to DAFS

- DAFS “borrowed” NFSv4 and has always promised to pay it back
- Performance is first step
- Semantics are the (longer) second
DAFS

• DAFS remains committed as the richest and highest performing NAS solution available today

• DAFS sets the standard for application NAS semantics
A Brief Rant on Client-side Caching
The Reasons for Caching

- Synchronous RPC avoidance
- Access prediction (Readahead)
- Writebehind
- Short read/write ops
- Sharing
The Cost of Caching

- Data copies
- Kernel memory
- Side protocols and heuristics for consistency checking
  - Attribute cache
  - Weak Cache Consistency (wcc)
  - Delegations
  - Locking
Caching and RDMA

- Data copies!
- RDMA/RPC reduces the latency and overhead
  - Synchronous ops are cheaper with RDMA
- Server caching improves further
  - Heuristics and explicit client cache hints
Server Implementers:

- Recognize client access
  - Sequential I/O
  - Pay attention to client advice
    - (When the protocol supports it)
- Caching makes perfect sense at the server
  - Reduced latency (in presence of hints)
  - Correctness is managed locally
VFS Implementers:

- Pay attention to uncached performance
  - User direct I/O, async I/O
  - Short ops – readthrough, writethrough

- Consider the local case:
  - Clustered sharing mediation
    - No need for attribute, wcc checks
  - Server caching

- Often makes sense to NOT cache
  - When application requirements (aio), RDMA (transport) and server latency (cache hints) can be optimized
Upcoming
Roadmap

- Early win: NFSv3 on IB
- Prepare the Transport: NFSv4 Sessions
- Enable the applications by extending the protocol
- Employ \textit{(and foster)} iWARP
NFSv3 on IB

- Hope to have several prototypes at the Bake-a-thon
NFSv4 Sessions

- Transport enhancements a part of NFSv4.1
- IETF process
NFSv4 Protocol extension

• Details TBD, but:
  - Cache hints
  - Batch I/O
  - Atomic append
  - Are all easily achievable

• IETF process
iWARP (future RDMA)

- A 1GbE NFS/RDMA/iWARP solution is very compelling
- Client CPU overhead of traditional NFS is often prohibitive for datacenter apps at this rate.
- The clear 10GbE partner for NFS
- Watch this space!
Questions?

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