Webcast Presenter

Alex McDonald joined NetApp in 2005, after more than 30 years in a variety of roles with some of the best known names in the software industry.

With a background in software development, support, sales and a period as an independent consultant, Alex is now part of NetApp’s Office of the CTO that supports industry activities and promotes technology & standards based solutions, and is co-chair of the SNIA NFS Special Interest Group.

Gilles joined Cisco 18 years ago. For the last ten years, Gilles’ focus has been Storage & SAN extension technologies for designing and implementing Disaster Recovery Centers.

Gilles is now dedicated to Data Center Technologies like Unified Fabric, FCoE and Unified Computing System and is a member of the Cisco Europe Data Centre and Virtualisation Team as a Distinguished Engineer. He is a member of the Board of Directors of SNIA Europe (Storage Networking Industry Association) as Technical Chair.
The SNIA Ethernet Storage Forum (ESF) focuses on educating end-users about Ethernet-connected storage networking technologies.
SNIA’s NFS Special Interest Group

- NFS SIG drives adoption and understanding of pNFS across vendors to constituents
  - Marketing, industry adoption, Open Source updates
- NetApp, EMC, Panasas and Sun founders
  - NetApp, EMC and Panasas act as co-chairs
- White papers on migration from NFSv3 to NFSv4
  - An Overview of NFSv4; NFSv4.0, NFSv4.1, pNFS, and proposed NFSv4.2 features
  - Migrating from NFSv3 to NFSv4
Previous SNIA NFS Presentations

BrightTalk SNIA Channel NFS Mini Series

– Part1 – Four Reasons NFSv4
  - Discusses the reasons behind the development of NFSv4 and beyond, and the need for a better-than-NFSv3 protocol

– Part2 – Advances in NFS – NFSv4.1 and pNFS
  - An overview and some details on NFSv4.1, pNFS (parallel NFS), and FedFS (the Federated Filesystem); and a high level overview of proposed NFSv4.2 features

Slides available from

# The Four Reasons for NFSv4.1

<table>
<thead>
<tr>
<th>Functional</th>
<th>Business Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>Compliance, improved access, storage efficiency, WAN use</td>
</tr>
<tr>
<td>High availability</td>
<td>High Availability, Operations simplicity, cost containment</td>
</tr>
<tr>
<td>Single namespace</td>
<td>Reduction in administration &amp; management</td>
</tr>
<tr>
<td>Performance</td>
<td>Better network utilization for all NFS clients</td>
</tr>
</tbody>
</table>

- ACLs for authorization
- Kerberos for authentication
- Client and server lease management with fail over
- Pseudo directory system
- Multiple read, write, delete operations per RPC call
- Delegate locks, read and write procedures to clients
- Parallelised I/O

---

© 2012 Storage Networking Industry Association. All Rights Reserved.
We’ll cover

– Selecting the application for NFSv4.1
– Planning;
  ▪ Filenames and namespace considerations
  ▪ Firewalls
  ▪ Understanding statefulness
  ▪ Security
– Server & Client Availability
– Where Next
  ▪ Considering pNFS

This is a high level overview

– Use SNIA white papers and vendors (both client & server) to help you implement
Selecting the Parts

1 – An NFSv4.1 compliant server
   – Question; files, blocks or objects?

2 – An NSFv4.1 compliant client
   – Will almost certainly be *nix based; no native NFS4 Windows client
   – Some applications are their own clients; Oracle, VMware etc

3 – Auxiliary tools;
   – Kerberos, DNS, NTP, LDAP
Selecting an Application

First task; select an application or storage infrastructure for NFSv4.1 use

- Home directories
- HPC applications

Don’t select...

- Oracle; use dNFS built in to the Oracle kernel
- VMware & other virtualization tools; no support for anything other than NFSv3 as of this date
- “Oddball” applications that expect to be able to internally manage NFSv3 “maps” with multiple mount points, or auxiliary protocols like mountd, statd etc;
- Any application that requires UDP; NFSv4 doesn’t support anything except TCP
I18N Planning; Directory & File Names

Directory and File Names

- NFSv4 uses UTF-8
  - Backward compatible with 7 bit ASCII
- Check filenames for compatibility
  - NFSv3 file created with the name René contains an 8 bit ASCII
  - UTF-8 é indicates a multibyte UTF-8 encoding, which will lead to unexpected results

Action

- Review existing NFSv3 names to ensure that they are 7 bit ASCII clean
- These aren’t;
Planning; Namespace

- **Uniform and “infinite” namespace**
  - Moving from user/home directories to datacenter & corporate use
  - Meets demands for “large scale” protocol
  - Unicode support for UTF-8 codepoints

- **No automounter required**
  - Simplifies administration
Namespace Example

- Server exports
  - `/vol/vol0`
  - `/vol/vol1`
  - `/backup/archive`

Mount root `/` over NFSv3:
- Allows the client to list the contents of `vol/vol12`

Mount root `/` over NFSv4:
- If `/vol/vol12` has not been exported and the pseudo filesystem does not contain it; the directory is not visible
- An explicit mount of `vol/vol12` will be required.
Namespaces

Action

– Consider using the flexibility of pseudo-filesystems to permit easier migration from NFSv3 directory structures to NFSv4, without being overly concerned as to the server directory hierarchy and layout.

However;

– If there are applications that traverse the filesystem structure or assume the entire filesystem is visible, caution should be exercised before moving to NFSv4 to understand the impact presenting a pseudo filesystem

– Especially when converting NFSv3 mounts of / (root) to NFSv4
Planning; Stateful Clients & Sessions

- **Statefulness**
  - NFSv4 gives client independence
  - Previous model had “dumb” stateless client
  - Server had the “smarts”

- **Pushes work out to client through delegations & caching**
  - Compute nodes work best with local data
  - NFSv4 eliminates the need for local storage
  - Exposes more of the backend storage functionality
    - Client can help make server smarter by providing hints
Planning; Stateful Clients & Sessions

Sessions
- NFSv3 server never knows if client got reply message
- NFSv4.1 introduces Sessions
- A session maintains the server's state relative to the connections belonging to a client

Action
- None; use delegation & caching transparently; client & server provide transparency
- NFSv4 advantages include session lock clean up automatically
Firewalls

- NFSv3 promiscuously uses ports; including 111, 1039, 1047, 1048, and 2049 (and possibly more...)
- NFSv4 has no “auxiliary” protocols like portmapper, statd, lockd or mountd
  - Functionality built in to the protocol
  - Uses port 2049 with TCP only
- No floating ports required & easily supported by NAT

Action

- Open port 2049 for TCP on firewalls
Strong security framework
Access control lists (ACLs) for security and Windows® compatibility
Security with Kerberos
  – Negotiated RPC security that depends on cryptography, RPCSEC_GSS
NFSv4 can be implemented without implementing Kerberos security
  – Not advised; but it is possible
Implementing without Kerberos

- *No security is a last resort!*

NFSv3 represents users and groups via 32 bit integers
- UIDs and GIDs with GETATTR and SETATTR

NFSv4 represents users and groups as strings
- `user@domain` or `group@domain`

Requires NFSv3 UID and GID 32 bit integers be converted to all numeric strings
- **Client side:**
  - Run `idmapd6`
  - `/etc/idmapd.conf` points to a default domain and specifies translation service `nsswitch`.

  - Incorrect or incomplete configuration, UID and GID will display `nobody`
  - Using integers to represent users and groups requires that every client and server that might connect to each other agree on user and group assignments
Implementing with Kerberos

Find a security expert
- Requires to be correctly implemented
- Do not use NFSv4 as a testbed to shake out Kerberos issues!

User communities divided into realms
- Realm has an administrator responsible for maintaining a database of users
- Correct `user@domain` or `group@domain` string is required
- NFSv3 32 bit integer UIDs and GIDs are explicitly denied access

NFSv3 and NFSv4 security models are not compatible with each other
- Although storage systems may support both NFSv3 and NFSv4 clients, be aware that there may be compatibility issues with ACLs. For example, they may be enforced **but not visible** to the NFSv3 client.

Resources:
- [http://web.mit.edu/kerberos/](http://web.mit.edu/kerberos/)
Action

- Review security requirements on NFSv4 filesystems
- Use Kerberos for robust security, especially across WANs
- If using Kerberos, ensure it is installed and operating correctly
  - Don’t use NFSv4 as a testbed to shake out Kerberos issues

Consider using Windows AD Server
- Easy to manage environment, compatible

Last resort
- If using NFSv3 security, ensure UID and GUID mapping and translation is uniformly implemented across the enterprise
Linux Client and NFSv4.1

- **Upstream (Linus) Linux NFSv4.1 client support**
  - Basic client in Kernel 2.6.32
  - pNFS support (files layout type) in Kernel 2.6.39
  - Support for the 'objects' and 'blocks' layouts was merged in Kernel 3.0 and 3.1 respectively

- **Full read and write support for all three layout types in the upstream kernel**
  - Blocks, files and objects
  - O_DIRECT reads and writes supported
pNFS client support in distributions
  – Fedora 15 was first for pNFS files
  – Kernel 2.6.40 (released August 2011)

Red Hat Enterprise Linux version 6.2
  – “Technical preview" support for NFSv4.1 and for the pNFS files layout type
  – Full support in RHEL6.4, to be announced shortly

Other Open Source
  – Microsoft NFSv4.1 Windows client from CITI

No support in Solaris
  – Both server and client are NFSv4 only
NFSv4.1 (pNFS) can aggregate bandwidth

- Modern approach; relieves issues associated with point-to-point connections

- pNFS Client
  - Client read/write a file
  - Server grants permission
  - File layout (stripe map) is given to the client
  - Client parallel R/W directly to data servers

- Removes IO Bottlenecks
  - No single storage node is a bottleneck
  - Improves large file performance

- Improves Management
  - Data and clients are load balanced
  - Single Namespace

Diagram:
- NFSv4.1 Client (s)
- pNFS protocol
- Storage-access protocol
- Metadata Server
- Control protocol
- Data Servers
Start using NFSv4.1 today
- NFSv4.2 nearing approval

Planning is key
- Application, issues & actions to ensure smooth implementations

Next up; pNFS
- First open standard for parallel I/O across the network
- Ask vendors to include NFSv4.1 support for client/servers
- pNFS has wide industry support
- Commercial implementations and open source

Part 4 – Using pNFS
- Next BrightTalk on
  - Tuesday March 5, 16:00GMT, 17:00 CET
Question & Answer
To download this Webcast after the presentation, go to

http://www.snia.org/about/socialmedia/