PRESENTED BY RED HAT

LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.

www.theredhatsummit.com

NFS Version 4

Features & Benefits

Steve Dickson
Consulting Software Engineer
Red Hat, Inc
Thursday, June 24

Overview

- NFS version 4 Protocol
- NFS V4 Protocol Extensions
- Secure NFS
- Debugging Tools
- Trace Points with SystemTap
- NFS Metric Tools

History of NFS

- Mid 1980's Sun Developed NFS
  - Version 1 was never released.
- Mid 90's NFS version 3 was released
  - Large file support – 64bit file sizes
  - Asynchronous I/O - commits
  - Read Directories with Attributes - READDIRPLUS

NFSv4 Advantages

- Performance
  - Read/Write Delegations
- Server maintains client state
  - Callbacks to Clients
- Multi-Component Messages
  - Less Network traffic
- Mandates strong security architecture
  - Available on **ALL** versions
- Elimination of 'side-car' protocols
  - No rpc.statd or In-kernel lockd
  - Only port 2049

Why NFS Version 4?

- Performance
- Elimination of 'side-car' protocols
- Multi-Component Lookups
- Mandates strong security architecture
- Server maintains client state

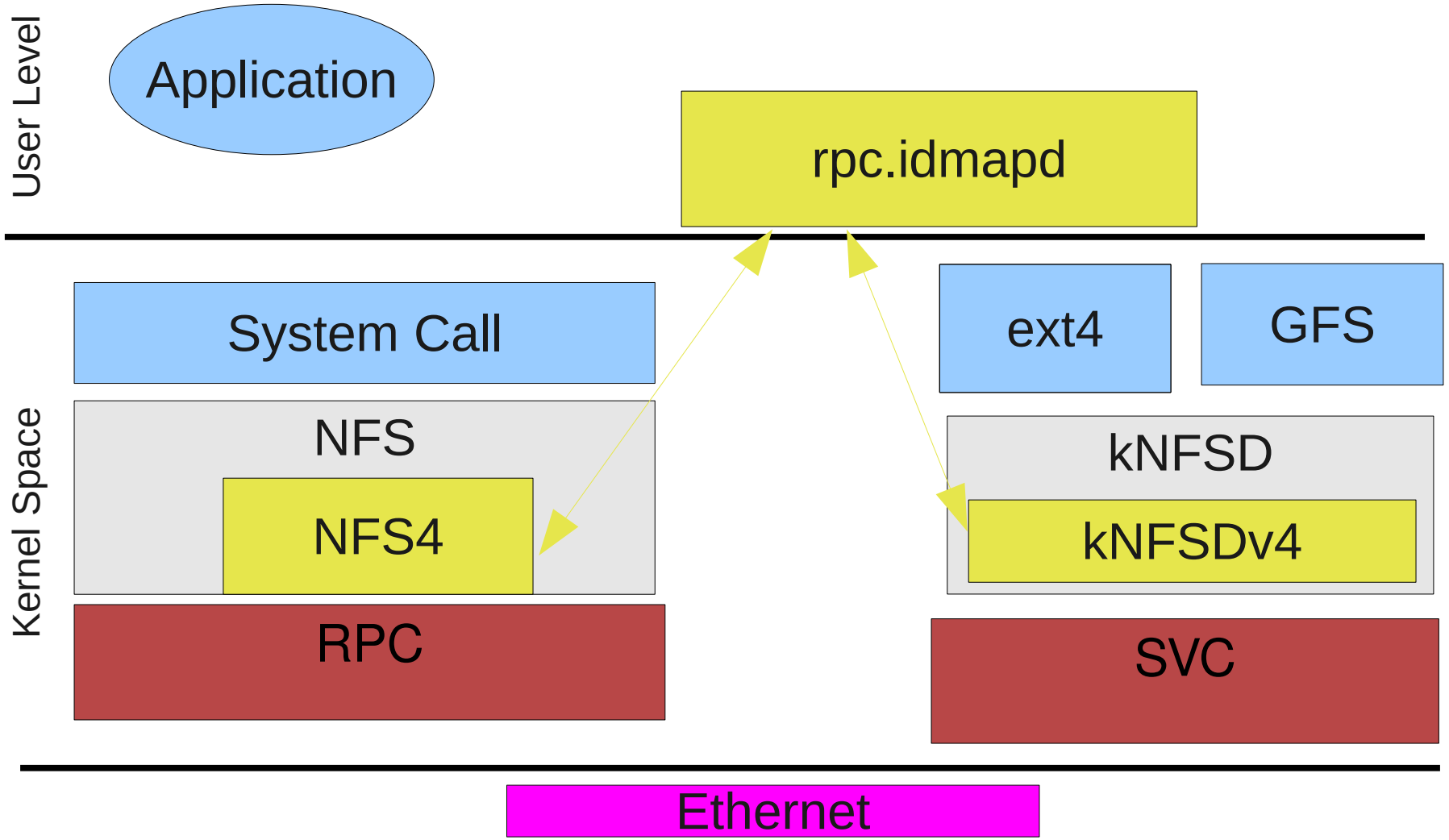
NFSv4 Protocol Feature List

- Compound Procedures
  - Multiple operations sent in one Over-The -Write message.
- Firewall Friendlier
  - Mount and locking protocols are integrated into protocol
  - Only TCP is supported
- Open and Close Operations
  - Atomic creates supported
- Pseudo File System
  - Shared server namespace

NFSv4 Feature List (cont'd)

- UTF-8 Strings are used for User/Group ids
  - Allow for Internationalization support
  - rpc.idmapd – maps user@domain to Linux UIDs on server and client.
- Integrated Access Control List (ACL) support
  - NT style ACLs
- File System Referrals
- Designed for future protocol extensions

NFSV4 Architect



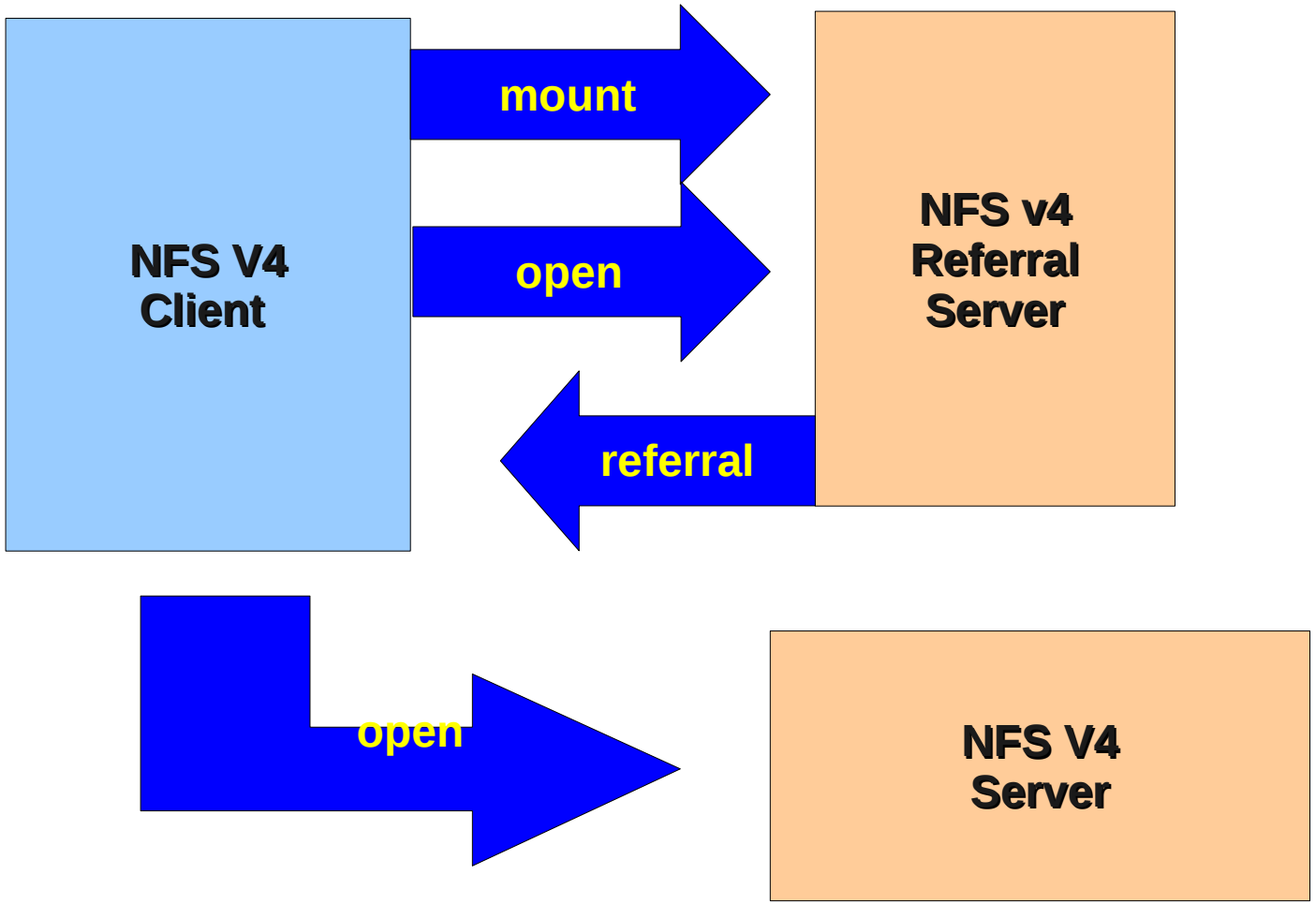
NFSv4 Default Protocol

- Server
  - Current exports will work seamlessly
    - No need for fsid=0 export
- Client
  - A mount configuration file (more later)
  - Mount to negotiate From V4
    - -t nfs4 option no longer needed.

NFS Mount Configuration File

- /etc/nfsmount.conf
  - Define mount options per mount point
  - Define mount options per server
  - Define mount options globally
- What Overrides What
  - Per per server options override globally options
  - Per mount point options override server options
  - Command line options override everything.

NFS V4 Referrals



NFS v4 Referrals

- On redhat-1 Server:
  - /refs/redhat-3 *(rw,refer=/export@redhat-3)
  - Mount –bind /refs/redhat-3 /refs/redhat-3
  - Service nfs start
- On the Client:
  - /sbin/nfs_cache_getent # used for DNS lookups
  - mount server:/refs /mnt # Do the v4 mount
  - cd /mnt/redhat-3 # jums to exported fs on redhat-3

Federated File System (FedFS)

- Main Job is to make referrals manageable
- Uses open protocols to create a scalable, cross-platform namespace accessible by unmodified NFS v4 clients.
- The three Protocols:
  - DNS – used to mount top of namespace
  - LDAP - used to store UUIDs of file systems.
  - RPC – used to administrate filesyservers.
- NFS first, but will be compatible with SMB/CIFS

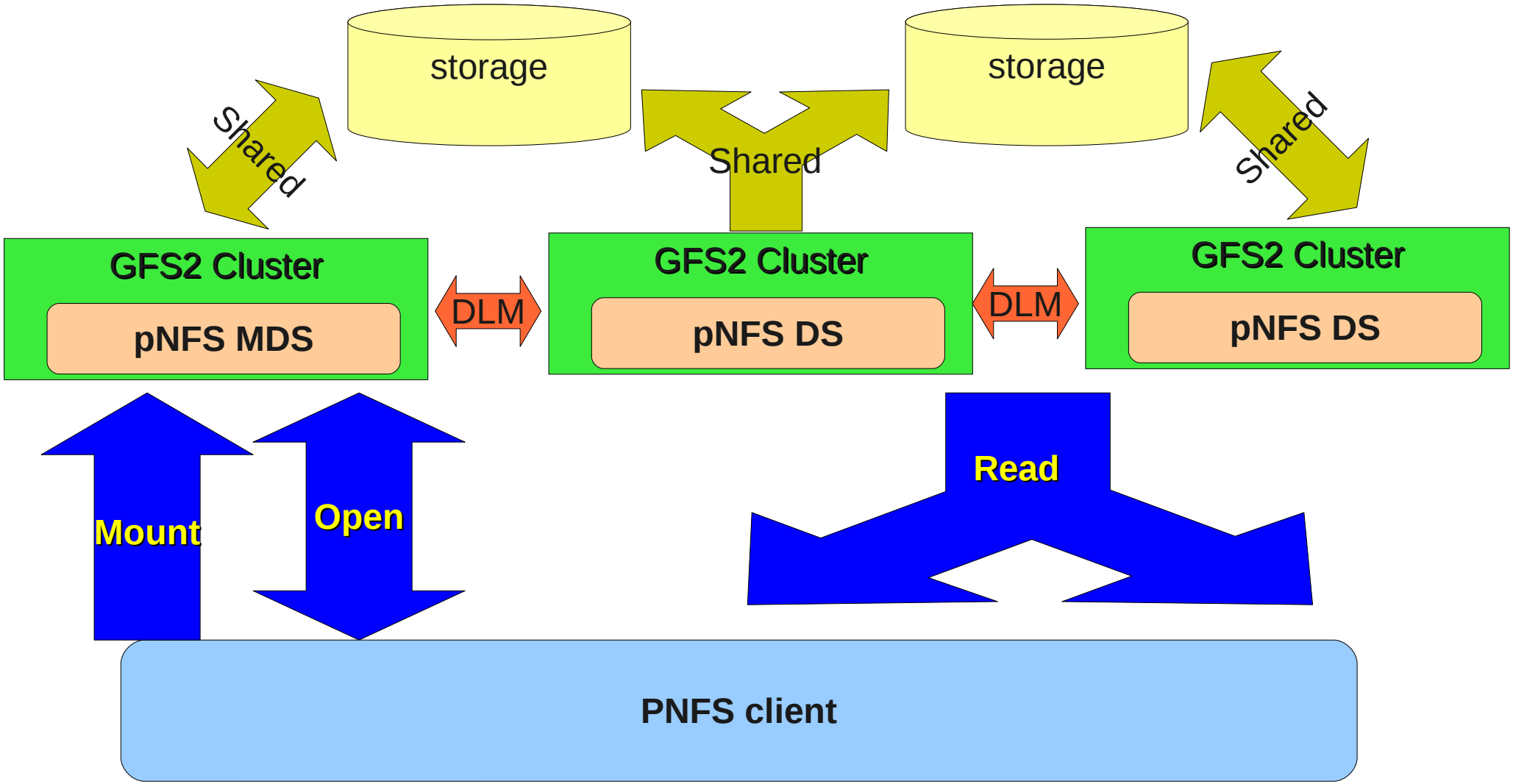
NFS minor version 1 (NFSv4.1)

- Sessions
  - Exactly-Once semantics
    - Duplicate Request Cache
  - Callbacks –
    - More Firewall friendly
      - Made on same connection as requests
      - Client initiated
  - Directory Delegations (Currently not supported)
  - Enabling pNFS

Parallel NFS (pNFS)

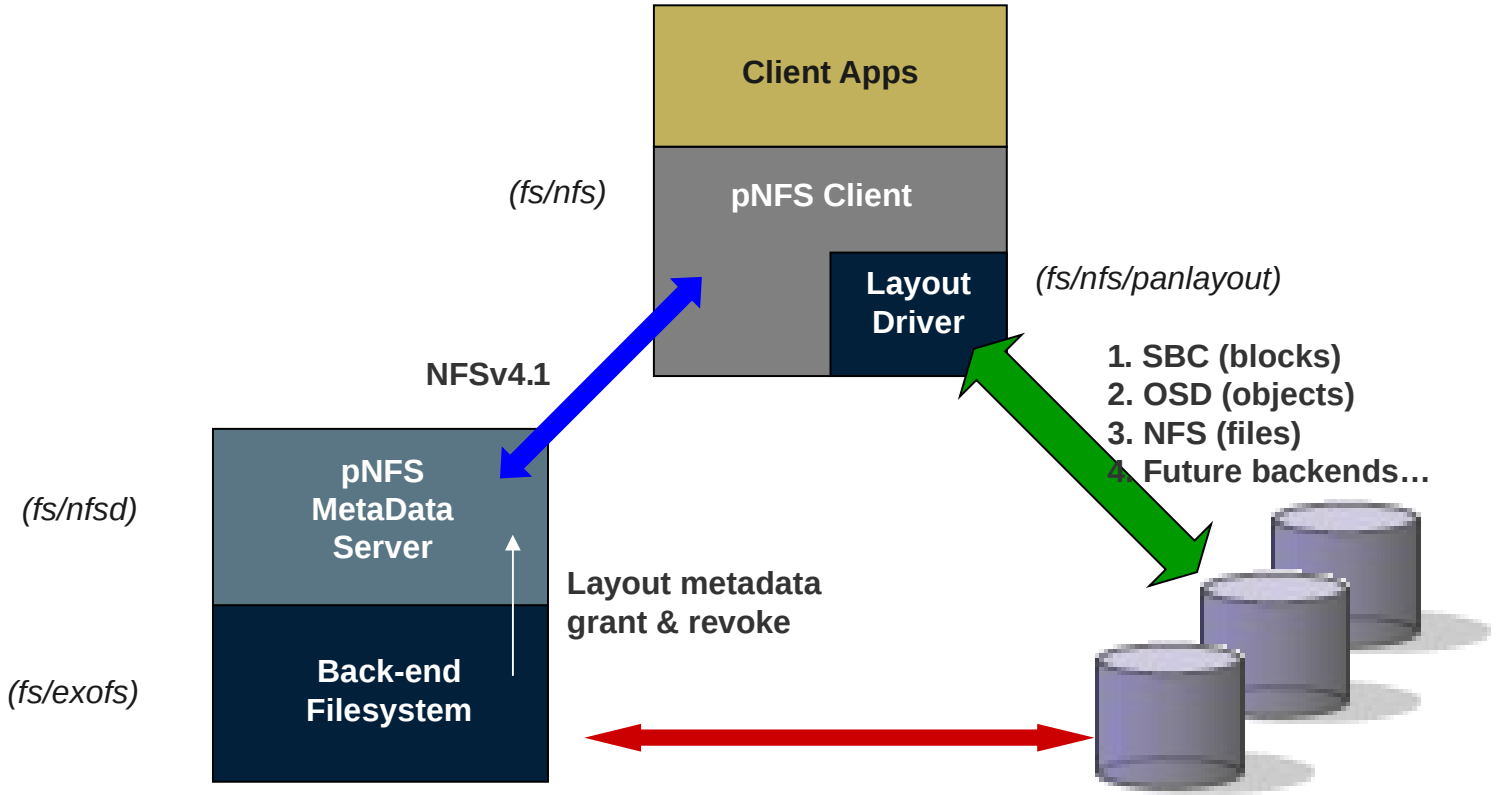
- Architecture
  - Metadata Server (MDS) – Handles all non-Data Traffic
  - Data Server (DS) – Responds directly to client I/O reqs
  - Multiple Clients – Access DS directly base on info from MDS
- Layout Types
  - File (Most common), Block and Object.
- Goal: Performance/Scalability
  - Network Appliance working with CITI at (UMICH)

pNFS File Layout



PNFS Object Layout

- Clients have Direct access to back storage



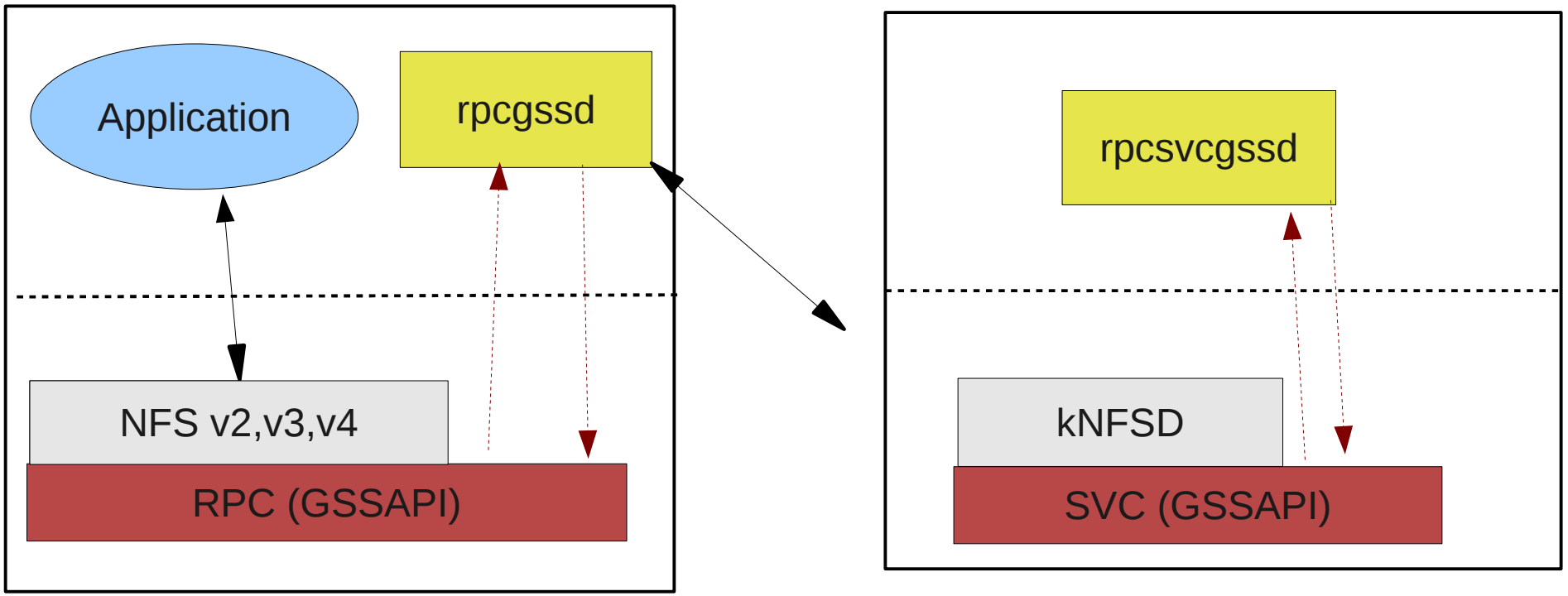
Secure NFS

- Used by ALL three NFS versions
  - Use the ‘-o sec=krb5’ mount option
- Uses GSS-API cryptographic method.
- Three Kerberos 5 security levels
  - krb5: Authentication (RPC header is signed)
  - krb5i: Integrity (Header and Body are signed)
  - Krb5p: Privacy (Header signed. Body encrypted)

Secure NFS (cont'd)

- User level daemons used to handle complicated context initiation phase
  - rpc.gssd – Client daemon that handles security contexts
  - rpc.svcgssd – Server daemon that handles security contexts
- Set SECURE_NFS in /etc/sysconfig/nfs
- Both daemons use files in the rpc_pipefs filesystem to get “upcalls” from the kernel.

Security Context Data flow



- Security Context Needed
- None cached; upcall to rpcgssd
- Server called; upcall to rpcsvcgssd

- rpcsvcgssd does gssapi magic
- Server returns gss context
- gss context cached in client

Debugging Tools

- rpcdebug – enable and disable kernel debugging
  - tail -f /var/log/messages # to see the debugging
- wireshark or tshark – analyze network traffic
  - tshark -w /tmp/data.pcap host <nfs_server>
- nfsstat- list nfs statistics
- System back traces for hung process
  - echo t > /proc/sysrq-tigger

NFS TracePoints

- Trace Points are availability in RHEL5.3 and beyond.
  - Generated “on the fly” from kernel header files.
  - 3 tracepoints used for NFS diagnostics
- rpc_call_status
  - Shows all errors that occur during NFS operations
- rpc_connect_status
  - Shows errors that occur during network connections
- rpc_bind_status
  - Show errors that occur during the binding of network connections

NFS and TracePoints (con't)

- Need to install kernel-devel rpm
  - yum install kernel-devel
- stap -L 'kernel.trace("*")'
  - Show all the available tracepoint
- The tracepoints can be accessd by systemtap script:

```
probe kernel.trace(“rpc_call_status”)
{
    terror =  task_status($task);
    If (terror) {
        printf(“%s[%d]:call_status::%s:%s: error %d(%s)\n”,
            execname(), pid(), cl_server($task), cl_prog($task),
            terror, errno_str(terror));
    }
}
```


NFS and Systemtap

- Probe all Client file operations

```
probe nfs.fop.entires {
    printf("%s: %s\n", name , argstr)
}
probe nfs.fop.return {
    printf("%s: %s\n", name, retstr)
}

probe begin { log("Starting NFS probes") }
probe end {log ("Ending NFS probes")}
```

- Execute probe

```
$ sudo stap nfs-probes.stp
```


NFS and Systemtap

- Kernel-devel rpm needed and usually kernel-debuginfo rpms are needed as well.
  - yum enablerepo=rhel-debuginfo install kernel-debuginfo*
- man -k stap – shows all the 'built in' tap scripts which live in **/usr/share/systemtap/tapset** directory
  - man tapset::.nfs – shows NFS scripts
- Systemtap home page: **http://sourceware.org/systemtap/wiki/HomePage**
- “Home grown” NFS tap scripts
  - git://fedorapeople.org/~steved/systemtap.git

NFS Metrics

- iostat -n
  - New '-n' flag to iostat command
    - yum install sysstat
    - Operations per sec
    - Reads and Writes per sec

```
Filesystem:  rBlk_nor/s   wBlk_nor/s   rBlk_dir/s   wBlk_dir/s   rBlk_svr/s   wBlk_svr/s   ops/s
rops/s    wops/s

tophat:/home   0.50        0.01        0.00        0.00        0.00        0.01      0.00      0.00      0.00
tophat:/home   15.71       0.01        0.00        0.00        0.00        0.01      0.00      0.00      0.00
```


NFS Metrics

- nfsiostat
  - NFS client per-mount I/0 statistics
    - Statistic per memory page
    - Statistics per directory operations
    - Statistics per file access
    - Uses /proc/self/mountstats

rawhide:/home mounted on /mnt/home:

op/s	rpc bklog
233.00	2.10

read:	ops/s	kB/s	kB/op	retrans	avg RTT (ms)	avg exe (ms)
	232.000	14908.719	64.262	0 (0.0%)	61.875	83.925

NFS Metrics

- mountstats
  - Overall NFS client per-mount statistics

```
GETATTR:
        3 ops (0%)    0 retrans (0%)      0 major timeouts
        avg bytes sent per op: 138 avg bytes received per op: 112
        backlog wait: 0.000000      RTT: 0.333333     total execute time: 0.333333 (milliseconds)
LOOKUP:
        4 ops (0%)    0 retrans (0%)      0 major timeouts
        avg bytes sent per op: 144 avg bytes received per op: 176
        backlog wait: 0.000000      RTT: 0.750000     total execute time: 0.750000 (milliseconds)
READ:
        8001 ops (20%)    0 retrans (0%)      0 major timeouts
        avg bytes sent per op: 140 avg bytes received per op: 65655
        backlog wait: 22.235471     RTT: 58.915511    total execute time: 81.165479 (milliseconds)
WRITE:
        14997 ops (37%)  0 retrans (0%)      0 major timeouts
        avg bytes sent per op: 35107    avg bytes received per op: 136
        backlog wait: 1892.769887       RTT: 51.310862    total execute time: 1944.124225 (millisecon
```


NFS and IPv6 Support

- **Client side (done):**
  - Hostname resolves to IPv6 address.
  - All NFS versions are now supported.
  - Current release target is Fedora 13.
- **Server side (experimental):**
  - Kernel pieces are mostly in-place, rpc.nfsd is finished
  - IPv6-capable mountd/exportfs prototype is now available

Acknowledgments

NFSv4.1: An update
Mike Eisler, Network Appliance February 23, 2009
http://www.connectathon.org/talks09/eisler_cthon_2009.pdf

Progress on NFSv4.1: Definition and a review of changes from NFSv4.
Dave Noveck, Network Appliance, February 5, 2007
http://www.connectathon.org/talks07/NFSv41update.pdf

NFSv4 Sessions Linux Implementation Experience
Jon Bauman & Mike Stolarchuk, CITI, U of Michigan
Center for Information Technology Integration
University of Michigan, Ann Arbor
http://www.connectathon.org/talks05/bauman.pdf

Parallel NFS (pNFS)
Garth Goodson, Network Appliance, February 28, 2006
http://www.connectathon.org/talks06/goodson.pdf

Acknowledgments

NFS Version 4 Minor Version 1
draft-ietf-nfsv4-minorversion1-25.txt
http://www.nfsv4-editor.org/draft-25/draft-ietf-nfsv4-minorversion1-25.html

Question

- Slides are available at:
  - http://people.redhat.com/steved/Summit10/Summit2010.pdf

FOLLOW US ON TWITTER

www.twitter.com/redhatsummit

TWEET ABOUT IT

#summitjbw

READ THE BLOG

http://summitblog.redhat.com/

Headline Here

Text with no bullets

- Bullets layer one
  - Bullets layer two
    - Bullets layer three