PRESENTED BY RED HAT

LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.

www.theredhatsummit.com

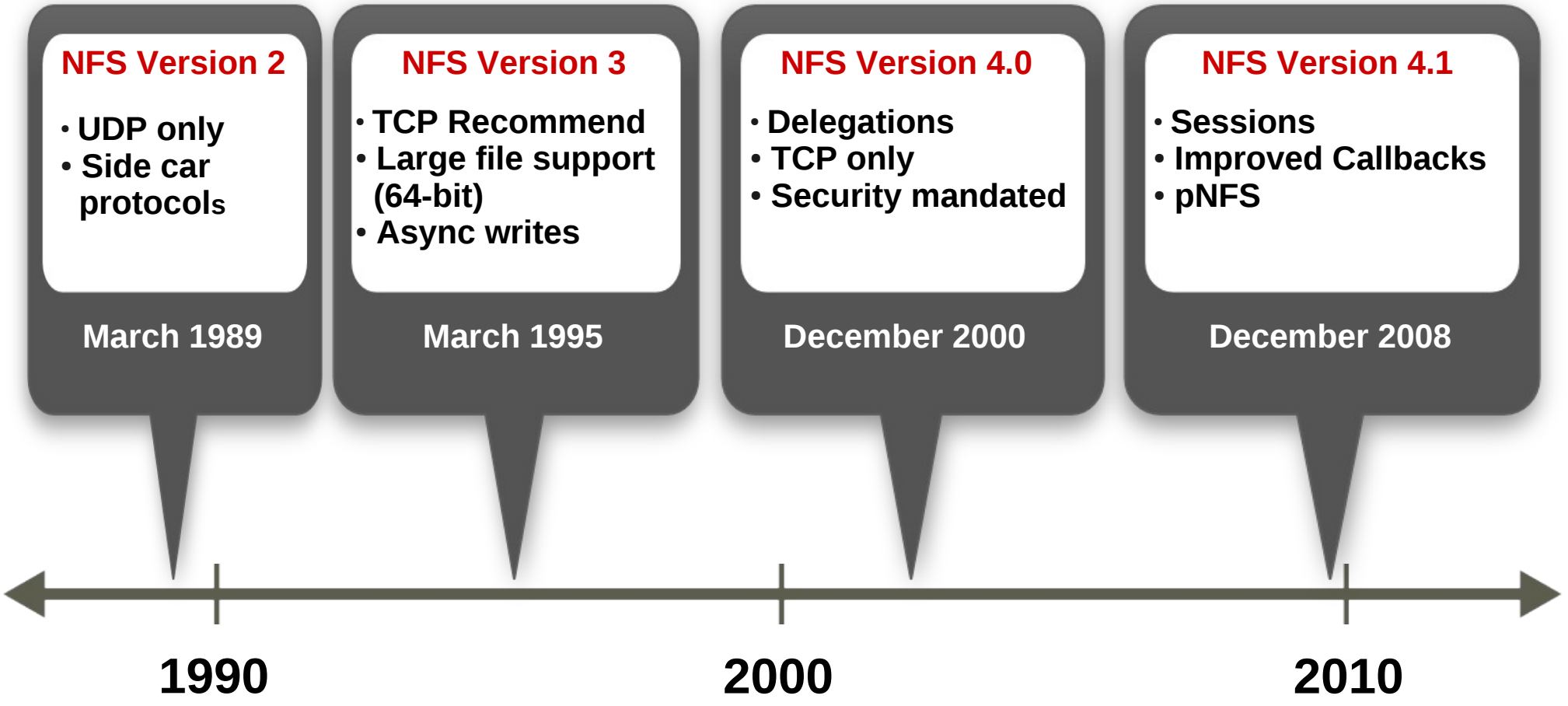
NFS: The Next Generation

Steve Dickson
Kernel Engineer, Red Hat
Wednesday, May 4, 2011

Overview

- Time Line
- What is in RHEL6
  - HOWTOs
- Debugging tools
- Debugging scenarios

Time Line


NFS Version 2
• UDP only
• Side car protocols
March 1989
NFS Version 3
• TCP Recommend
• Large file support (64-bit)
• Async writes
March 1995
NFS Version 4.0
• Delegations
• TCP only
• Security mandated
December 2000
NFS Version 4.1
• Sessions
• Improved Callbacks
• pNFS
December 2008
1990
2000
2010

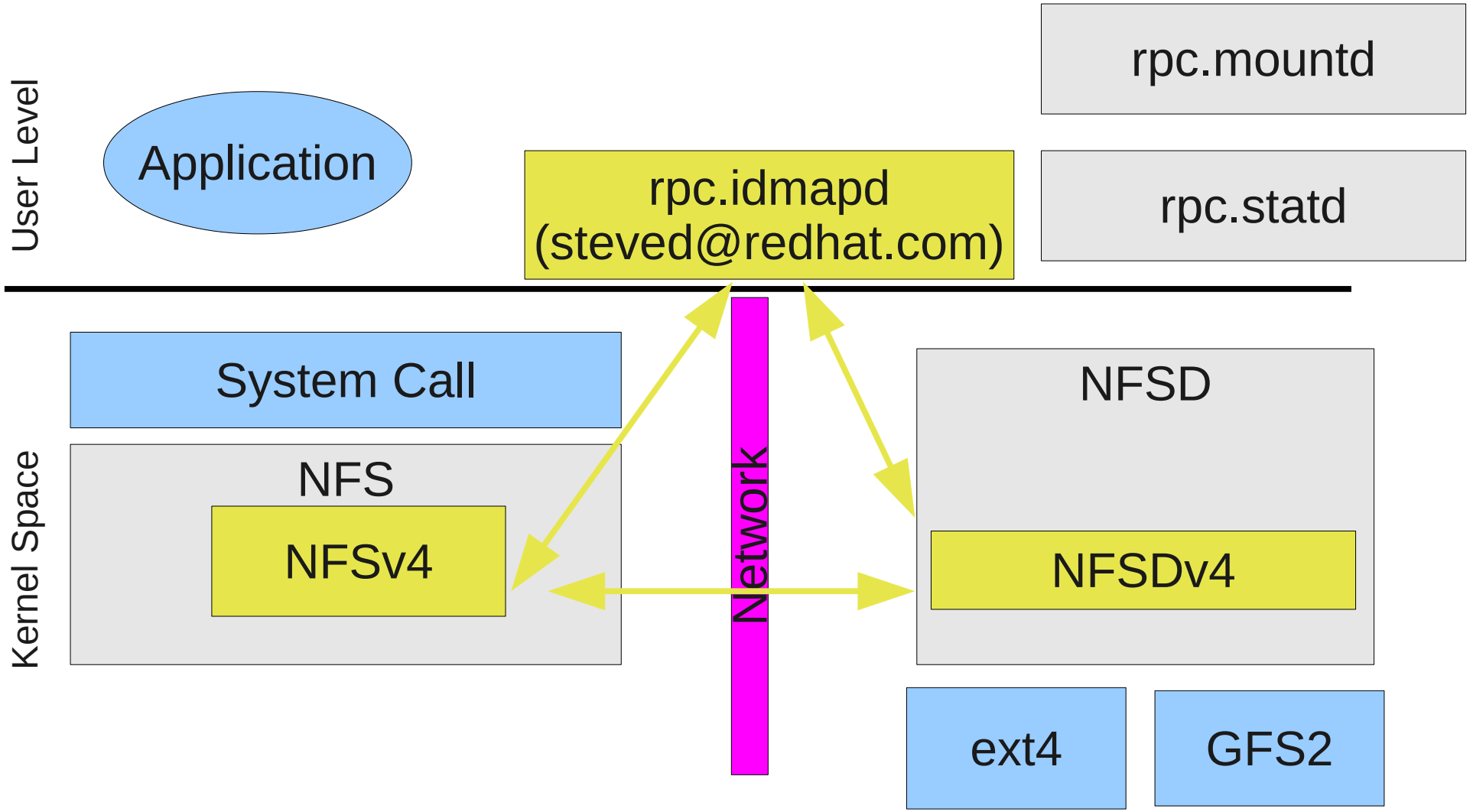
What is in RHEL6

- NFS v4 as the default protocol version
- V4 Referrals
- pNFS (Tech Preview in RHEL6.2)
- FSCache
- Secure NFS

Red Hat NFS Team

- Team Members
  - Steve Dickson <steved@redhat.com>
  - Jeff Layton <jlayton@redhat.com>
  - Bruce Fields <bfields@redhat.com>
  - David Howells <dhowells@redhat.com>
- Top Contributors
  - Sachin Prabhu <sprabhu@redhat.com>
  - Flavio Leitner <fleitner@redhat.com>

NFSV4 Architecture



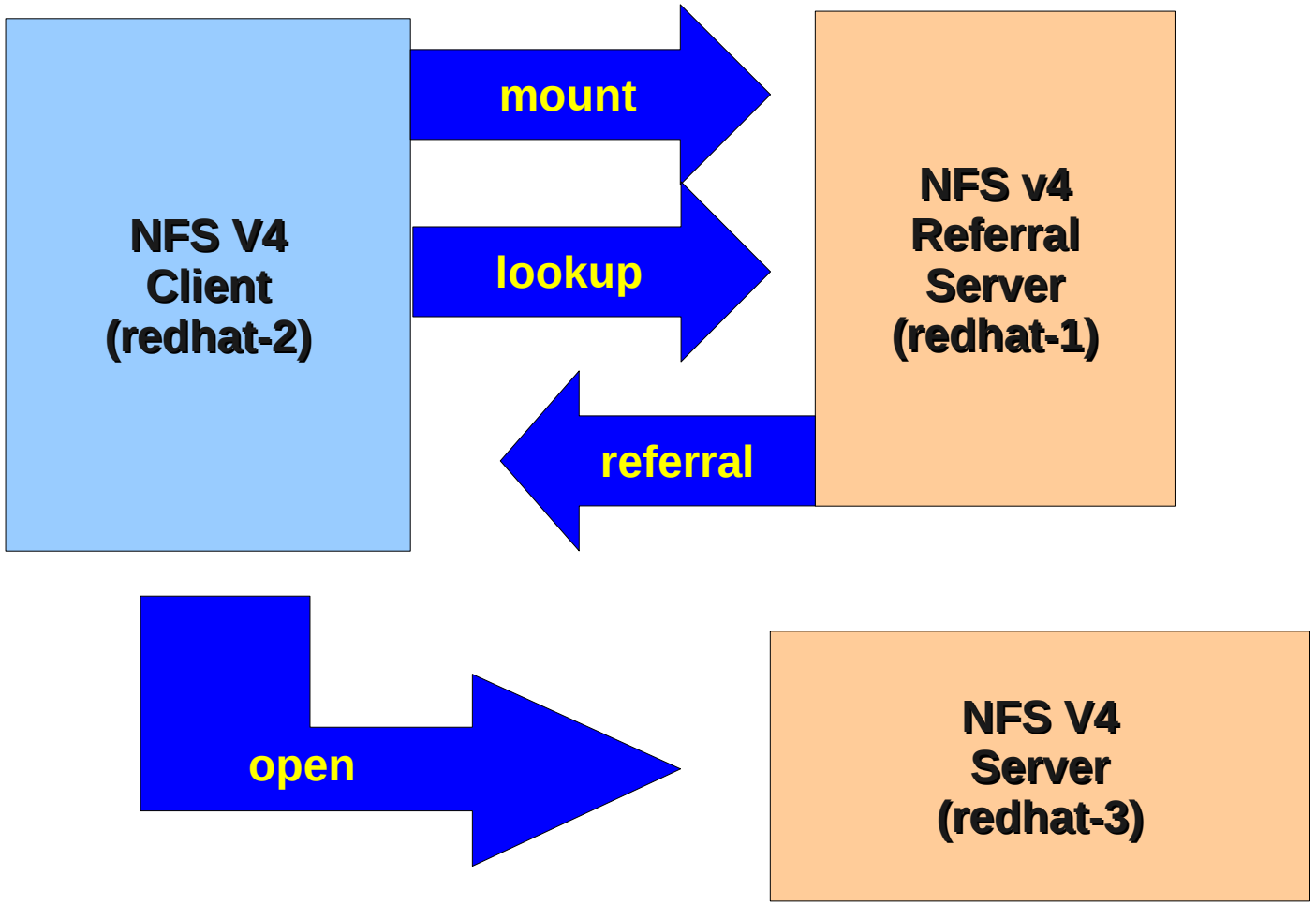
NFSv4 Default Protocol

- Mounts to negotiate from V4 down to v3 and then v2
- **/etc/nfsmount.conf**
  - Define mount options, such as protocol versions, by mount point, by server and globally
  - What Overrides What
    - Per server options override global options
    - Per mount point options override server options
    - Command line options override everything.

NFS v4 Referrals

- Dynamic Namespace
- Directs v4 clients to file systems on other v4 server
- Not a Migration
- Federated File Systems (FedFS)
  - Manage the namespace

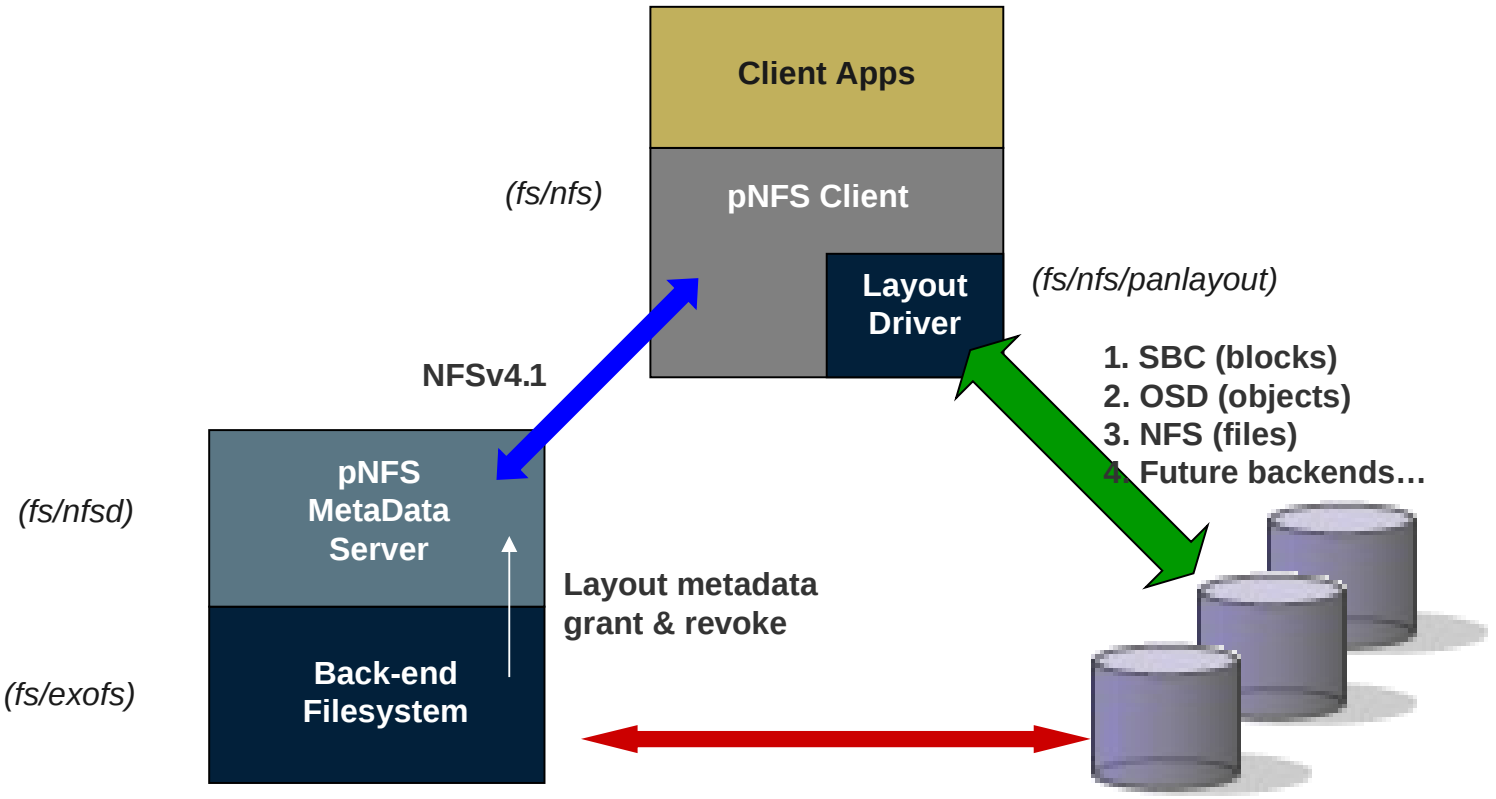
NFS V4 Referrals



NFS V4 Referrals - HOWTO

- On redhat-1 Server:
  - Export file system with: **refer=/export@redhat-3**
  - Bind mount file system: **mount –bind /export /export**
  - Start nfs server: **service nfs start**
- On the Client:
  - Mount file system: **mount server:/export /mnt/export**
  - Create the referral: **cd /mnt/export**

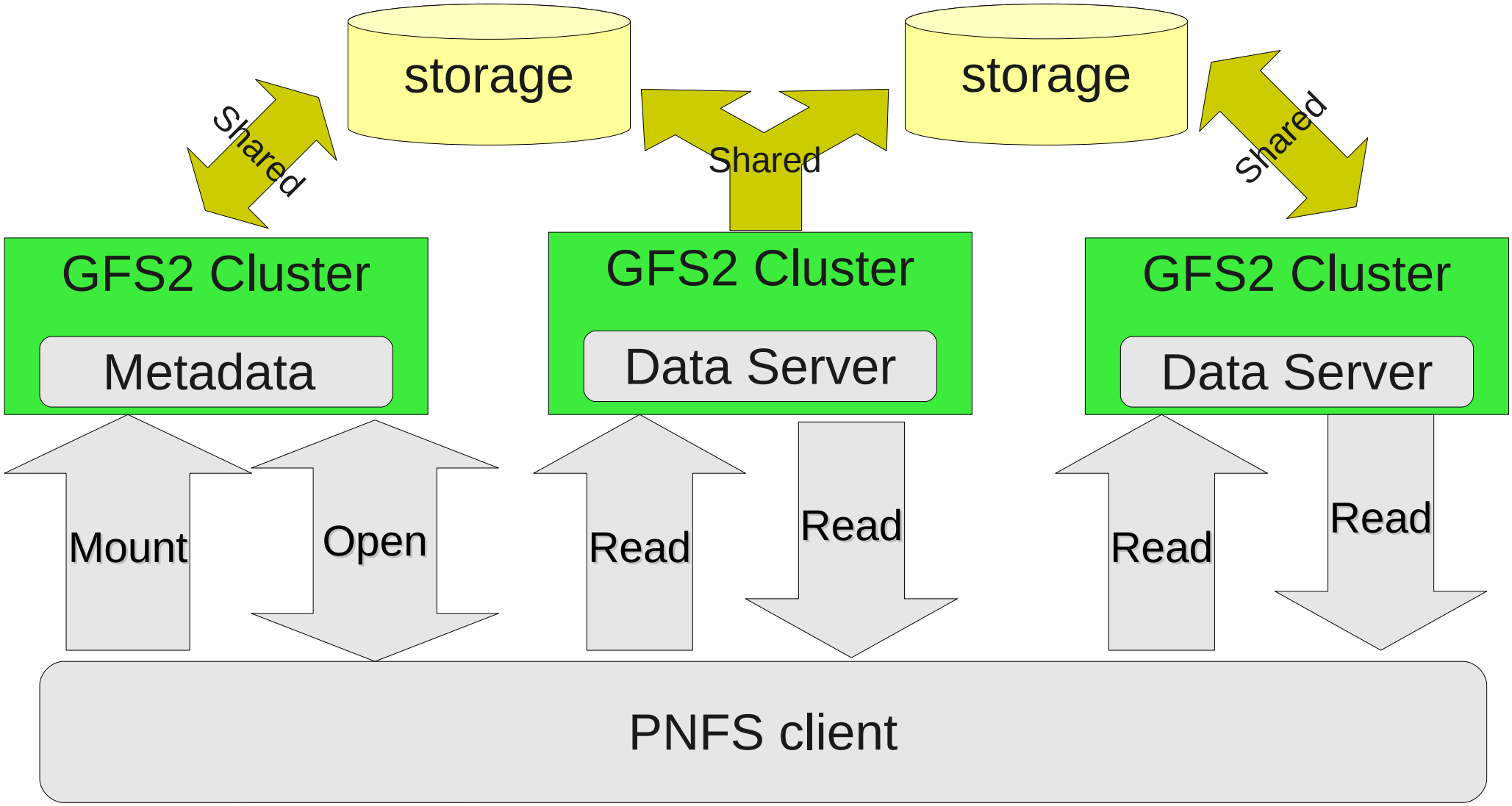
pNFS Object Layout



Parallel NFS (pNFS)

- Architecture
  - Metadata Server (MDS) – Handles all non-Data Traffic
  - Data Server (DS) – Responds directly to client I/O reqs
  - Shared Storage Between Servers
- There Layout Types
  - File Layout (Linux client support in 2.6.39)
  - Block Layout
  - Object Layout

PNFS - File Layout Architecture



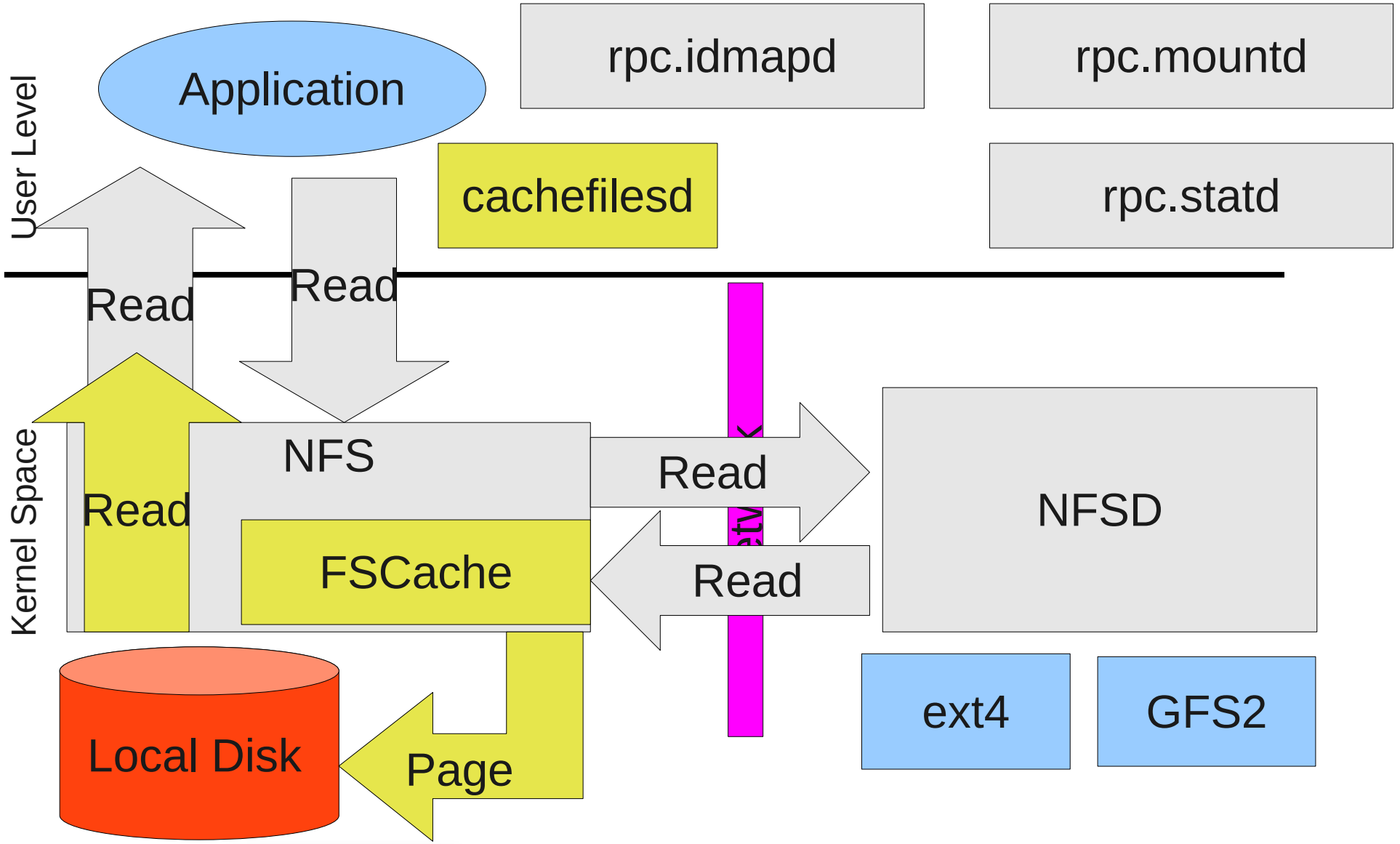
Parallel NFS (pNFS)

- RHEL6.2:
  - Only Client support (Tech Preview)
  - Only File Layouts supported
- pNFS mount:
  - **mount -o minorversion=1 server:/export /mnt/export**
- RHEL-Next
  - Block and Object layout support

FSCache

- Main Goal: Improve Server Scalability
  - Some short term performance degradation on client
- Only Reads are Cached.
  - Opening the file for writes flushes and disables cache
- Mount the file system with 'fsc' mount flag
  - **mount -o fsc server:/export /mnt**
- Cachefilesd – Cache Management Daemon
- Tech Preview in RHEL6

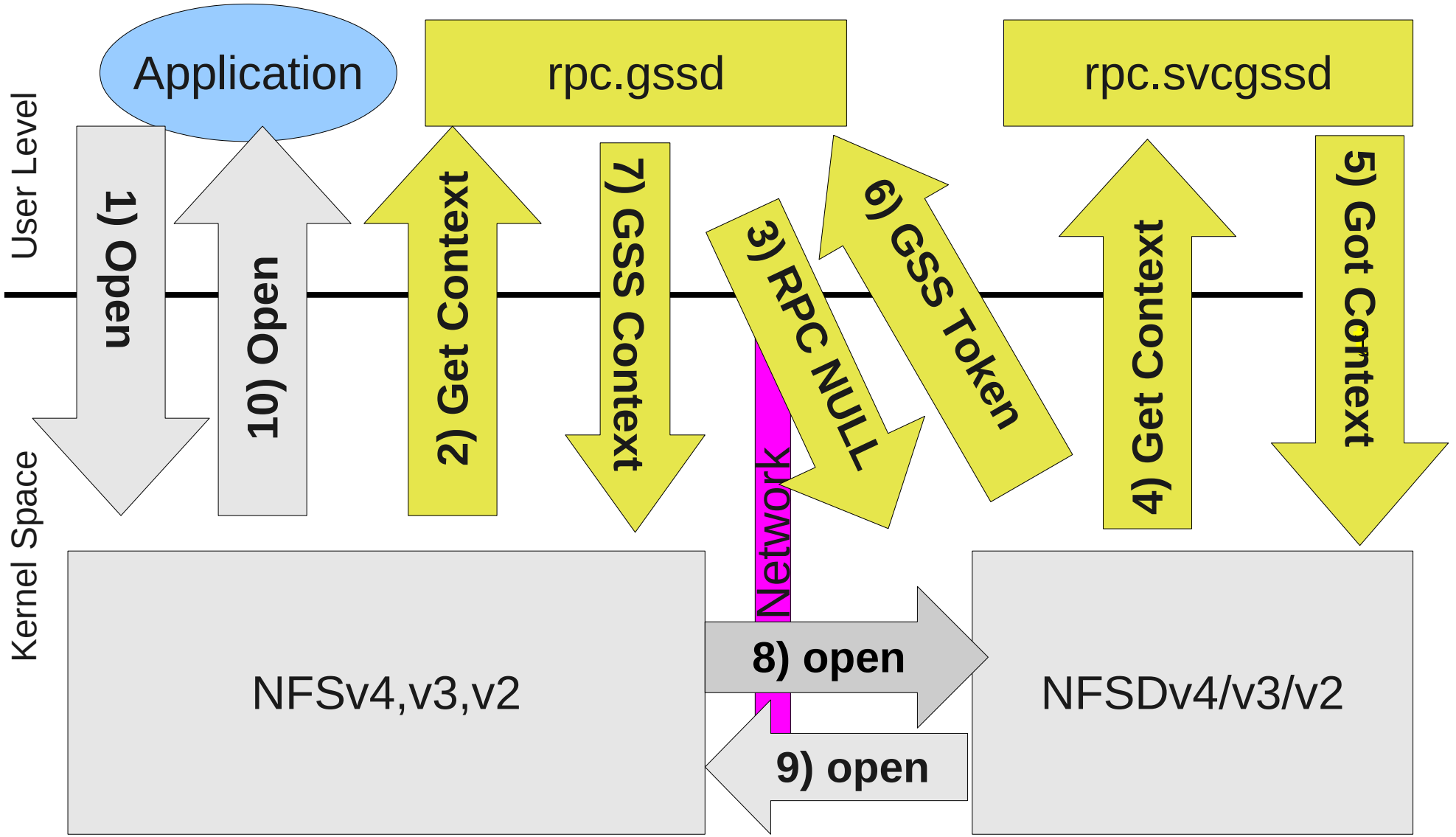
FSCache Architecture



FSCache – HOWTO

- Fill the cache by reading a 1Gig file
  - dd if=/mnt/home/tmp/1g of=/dev/null bs=16384
- Clear the page cache
  - umount /mnt
  - **mount -o fsc** server:/export /mnt
- Monitor the FSCache status
  - watch -n 0 cat **/proc/fs/fscache/stats**
- Re-read data out of the local cache
  - dd if=/mnt/home/tmp/1g of=/dev/null bs=16384

Secure NFS Architecture



Secure NFS

- 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 mount option
  - **mount -o sec=krb5 server:/export /mnt/export**
- Turn on SECURE_NFS
  - Added '**SECURE_NFS**=yes' to **/etc/sysconfig/nfs**

Secure NFS – HOWTO

- Setup kerberos configuration file, **/etc/krb5.conf**
  - [realms] section

```
STEVED.COM {
  kdc=kerberos.redhat.com:88
  admin_server = kerberos.redhat.com:749
}
```

  - [domain_realm] section

```
.steved.com = STEVED.COM
steved.com = STEVED.COM
```

- In cross-realm environments client mappings must be set up in the [domain_realm] section

```
pro5.redhat.com = STEVED.COM
pro1.redhat.com = STEVED.COM
```


Secure NFS – HOWTO

- Create machine keytabs on both the server and client
  - Use kadmin or kadmin.local to create a machine keytab in /etc/krb5.keytab
    - addprinc -randkey nfs/pro5.redhat.com
    - ktadd -k /tmp/keytab nfs/pro5.redhat.com
    - cp /tmp/keytab /etc/krb5.keytab
  - Use (as root) klist -k to verify the /etc/krb5.keytab is setup correctly.

```
# klist -k
Keytab name: FILE:/etc/krb5.keytab
KVNO Principal
----------------------------------------------------------------------
6 nfs/pro5.lab.boston.redhat.com@STEVED.COM
```


Secure NFS – HOWTO

- Setup kerberos configuration file (continued)
  - In multiple DNS domain environments client mappings must be set up in the [domain_realm] section
  - [domain_realm] section

    pro5.redhat.com = STEVED.COM
    pro1.redhat.com = STEVED.COM

NFSoverRDMA - HOWTO

- Bring up the IpoIB stack
  - On both machines
    - yum install rdma opensm
    - Server rdma start
    - Create ifconfig flies on both machine (needs to be done by hand)
  - On one machine
    - service start opensm

NFSoverRDMA - HOWTO

- Bring up the NFS server
  - Set **RDMA_PORT**=20049 in **/etc/sysconf/nfs**
  - Start the NFS server
    - service nfs start
- Configure the client
  - Load module
    - **modprobe xprtrdma**
- Mount the filesystem
  - **mount -o rdma,port=20049 server:/export /mnt/export**

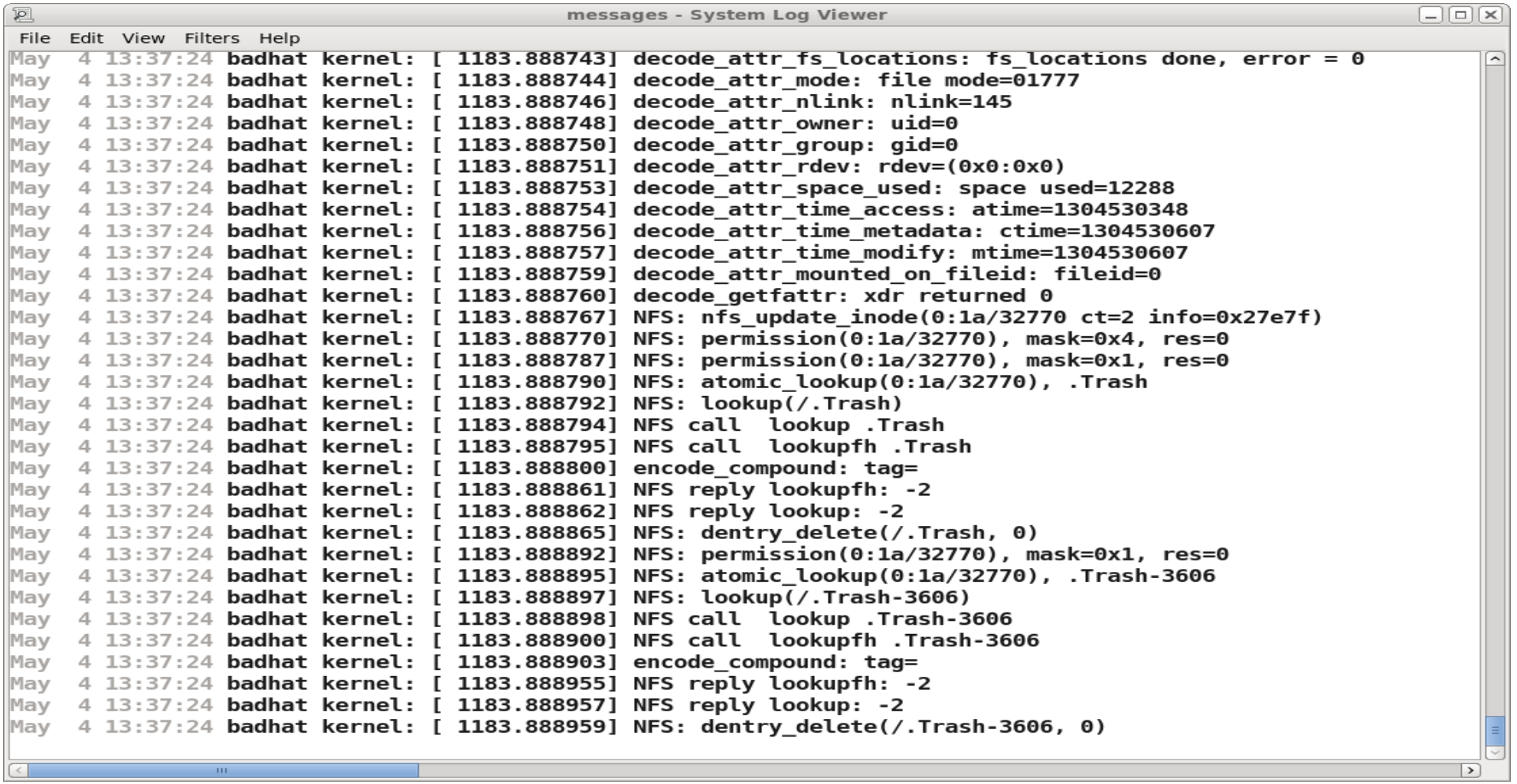
Debugging Tools

- rpcdebug – enable and disable kernel debugging
- nfsiostat – per-mount I/O statistics
- iostat -n – per-mount I/0 statistics
- Mountstats - stats per-protocol operation
- wireshark or tshark – analyze network traffic
- Trace Points
- SystemTap probs

rpcdebug

- Enables kernel debugging for nfs, nfsd and sunrpc modules
- Can be used to debug hanging mounts and/or processes
  - **rpcdebug -vh**
    - Shows list of modules and valid flags
  - **rpcdebug -m nfs -s all**
    - Enables all NFS client debugging
  - **rpcdebug -m rpc -s call**
    - Enables debugging when network connections are created.

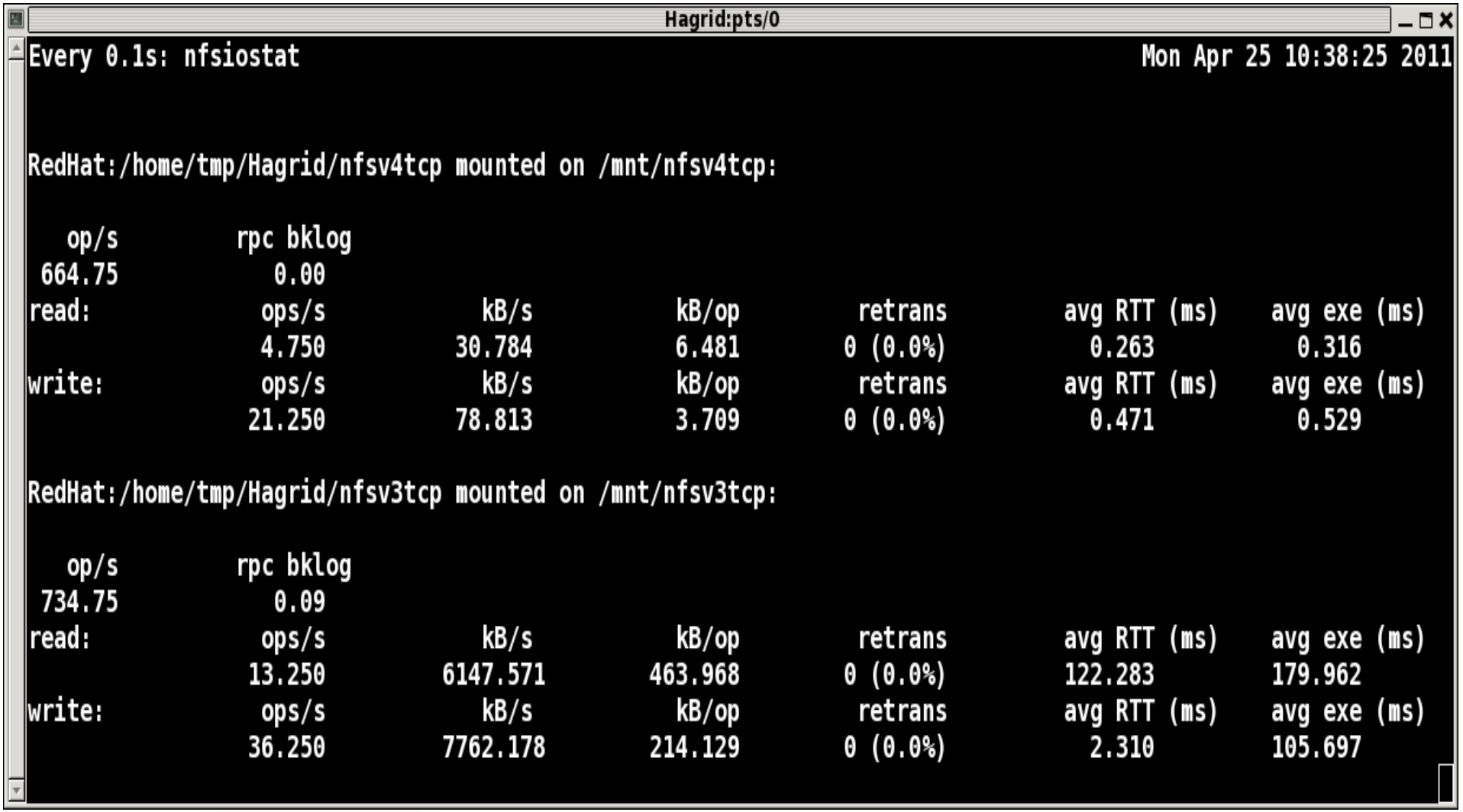
rpcdebug -m nfs -s all



nfsiostat

- NFS client per-mount I/0 statistics
  - **watch -n 0 nfsiostat**
    - Very handy way to monitor NFS traffic on all NFS mount points

nfsiostat - (watch -n 0 nfsiostat)



mountstats

- Overall NFS client per-mount statistics
  - --nfs – shows the number of calls into the VM subsystem
  - --rpc – shows stats on a per protocol bases

Mountstats – mount info, I/O counts, RPC stats

Hagrid:pts/0

```
Hagrid$ mountstats /mnt/home
Stats for badhat:/home/ mounted on /mnt/home:
  NFS mount options: rw,vers=4,rsize=524288,wsize=524288,namlen=255,acregmin=3,acregmax=60,acdirmin=30,a
t=0,timeo=600,retrans=2,sec=sys,clientaddr=192.168.62.7,minorversion=0
  NFS server capabilities: caps=0x7ffe,wtmult=512,dtsize=4096,bsize=0,namlen=255
  NFSv4 capability flags: bm0=0xfdffbfff,bm1=0xf9be3e,acl=0x3
  NFS security flavor: 1  pseudoflavor: 0

NFS byte counts:
  applications read 1073741824 bytes via read(2)
  applications wrote 2147483648 bytes via write(2)
  applications read 0 bytes via O_DIRECT read(2)
  applications wrote 0 bytes via O_DIRECT write(2)
  client read 1073741824 bytes via NFS READ
  client wrote 2147483648 bytes via NFS WRITE

RPC statistics:
  6197 RPC requests sent, 6197 RPC replies received (0 XIDs not found)
  average backlog queue length: 257
```


mountstats – Procotol times

Hagrid:pts/0

```
READ:
        2050 ops (33%)  0 retrans (0%)  0 major timeouts
        avg bytes sent per op: 184      avg bytes received per op: 523836
        backlog wait: 68.031220         RTT: 64.609756  total execute time: 133.095122 (milliseconds)
WRITE:
        4097 ops (66%)  0 retrans (0%)  0 major timeouts
        avg bytes sent per op: 524364   avg bytes received per op: 132
        backlog wait: 1798.099829       RTT: 4.648767   total execute time: 1803.023188 (milliseconds)
COMMIT:
        6 ops (0%)      0 retrans (0%)  0 major timeouts
        avg bytes sent per op: 184      avg bytes received per op: 124
        backlog wait: 320.000000        RTT: 5074.333333        total execute time: 5394.833333 (millise
OPEN:
        4 ops (0%)      0 retrans (0%)  0 major timeouts
        avg bytes sent per op: 269      avg bytes received per op: 420
        backlog wait: 0.000000  RTT: 0.500000   total execute time: 0.500000 (milliseconds)
OPEN_CONFIRM:
        2 ops (0%)      0 retrans (0%)  0 major timeouts
        avg bytes sent per op: 176      avg bytes received per op: 68
        backlog wait: 0.000000  RTT: 98.500000  total execute time: 98.500000 (milliseconds)
```


iostat

- **ioststat -n** shows NFS stats
- **watch -n 0 iostat -hn**
  - Very handy way to monitor NFS I/O on all NFS mount points

iostat - (iostat -hn 3)

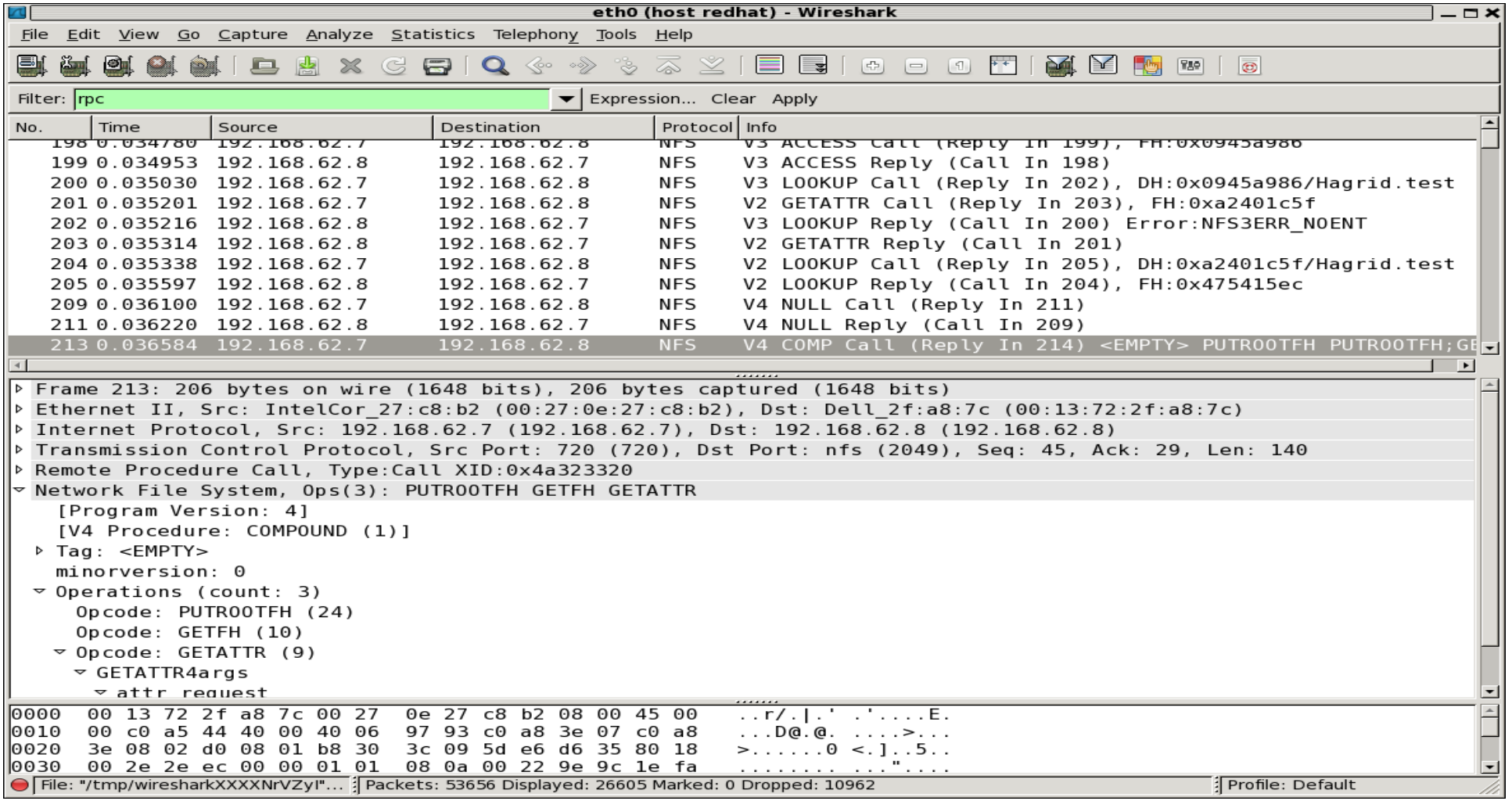
Hagrid:pts/0

```
badhat:/home/
                  125376.00    379600.00         0.00         0.00    125376.00     69290.67    195.33    123.00     67.67

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
badhat:/home/
                   94890.67        85.33         0.00         0.00     95232.00    120490.67    211.00     93.00    117.67

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
badhat:/home/
                   103946.67         0.00         0.00         0.00    103765.33    128341.33    231.33    101.33    125.33

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
badhat:/home/
                   162634.67         0.00         0.00         0.00    162816.00         0.00    159.33    159.00      0.00

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
badhat:/home/
                   212202.67    232218.67         0.00         0.00    211861.33     14677.33    216.67    207.00     14.33

^C
Hagrid$
```


Tshark/Wireshark

- Binary traces are much more useful than ASCII
  - **tshark -w /tmp/data.pcap**
  - **wireshark -r /tmp/data.pcap**
- Display only NFS traffic from a particular server
  - **tshark -R rpc -host <server>**

wireshark



NFS TracePoints

- 3 tracepoints used for NFS diagnostics
  - **rpc_call_status** - Shows 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
- Need to install kernel-devel rpm
  - yum install kernel-devel

NFS TracePoints

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

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


Systemtap Probes

- **kernel-devel** and **kernel-debuginfo** rpms are needed.
  - yum enablerepo=rhel-debuginfo install kernel-debuginfo*
- **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

Systemtap Probes

- Probe all Client file operations

```
probe nfs.fop.entries {
    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
```


Debugging – Mounts failing or hanging

- Hanging mount
  - Use '-vvv' flag to make mount verbose
  - Turn on kernel debugging with rpcdebug
    - **rpcdebug -m nfs -s mount**

Debugging - Hung Processes

- System Request Debugging
  - Create a system back trace
    - **echo t > /proc/sysrq-trigger**

References

- NFS version 4.1

  http://www.snia.org/events/storage-developer2007/.../SShepler_NFSv4_1_rev3.pdf

- Network File System - Wikipedia

  http://en.wikipedia.org/wiki/Network_File_System_%28protocol%29

- RFC 5661 - Network File System (NFS) Version 4 Minor Version 1 Protocol

  http://www.ietf.org/rfc/rfc5661.txt

- Linux Kernel Documentation :: filesystems : nfs-rdma.txt

  http://www.mjmwired.net/kernel/Documentation/filesystems/nfs-rdma.txt

Questions

Available at Campground 2 at 4:20pm

Slide Deck available at:
http://people.redhat.com/steved/Summit11/

Email Address:
steved@redhat.com

LIKE US ON FACEBOOK

www.facebook.com/redhatinc

FOLLOW US ON TWITTER

www.twitter.com/redhatsummit

TWEET ABOUT IT

#redhat

READ THE BLOG

summitblog.redhat.com

GIVE US FEEDBACK

www.redhat.com/summit/survey