

**SUMMIT**

**JBoss  
WORLD**

**PRESENTED BY RED HAT**

**LEARN. NETWORK.  
EXPERIENCE OPEN SOURCE.**

[www.theredhatsummit.com](http://www.theredhatsummit.com)

# NFS: The Next Generation

Steve Dickson

Kernel Engineer, Red Hat

Wednesday, May 4, 2011

**SUMMIT**

JBoss  
WORLD

PRESENTED BY RED HAT



# Overview

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

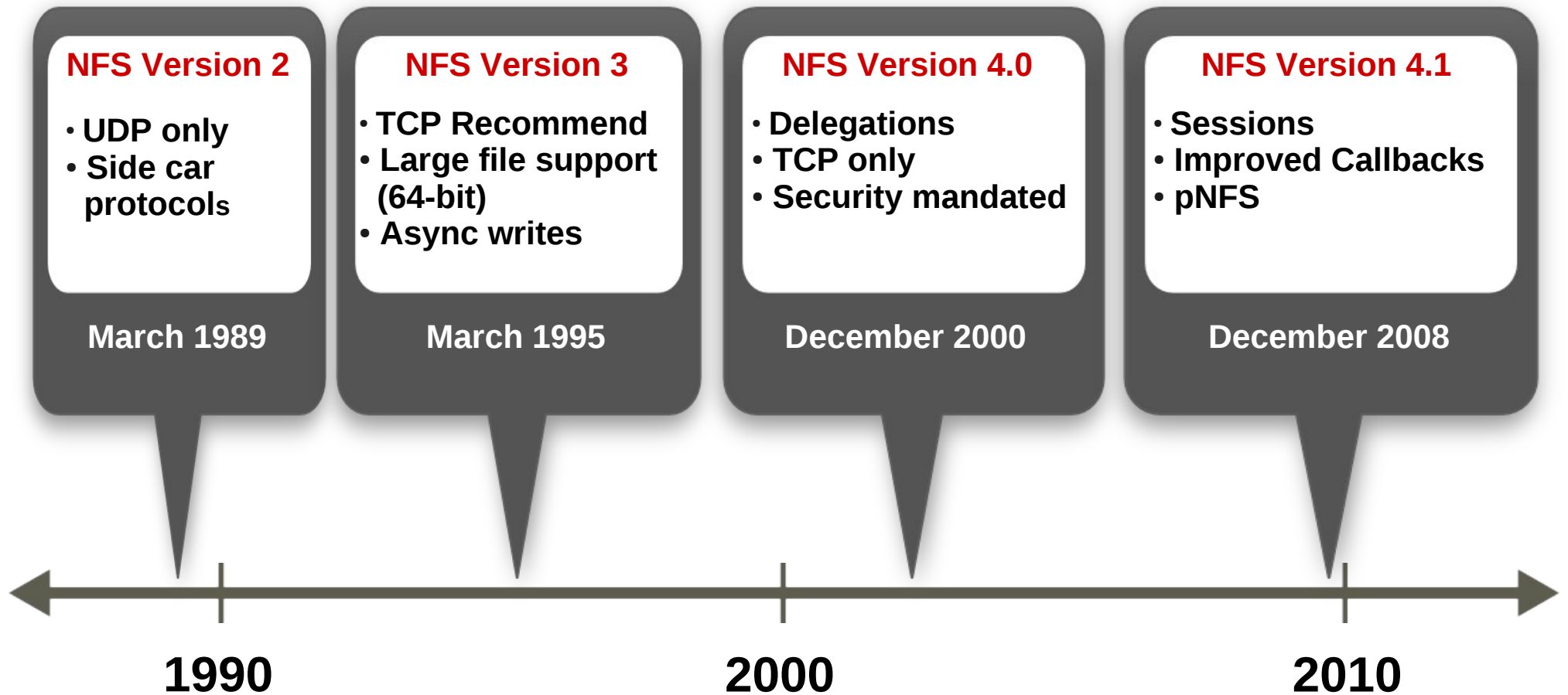
**SUMMIT**

**JBoss  
WORLD**

PRESENTED BY RED HAT



# Time Line



**SUMMIT**

**JBoss  
WORLD**

PRESENTED BY RED HAT



# 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](mailto:steved@redhat.com)>
  - Jeff Layton <[jlayton@redhat.com](mailto:jlayton@redhat.com)>
  - Bruce Fields <[bfields@redhat.com](mailto:bfields@redhat.com)>
  - David Howells <[dhowells@redhat.com](mailto:dhowells@redhat.com)>
- Top Contributors
  - Sachin Prabhu <[sprabhu@redhat.com](mailto:sprabhu@redhat.com)>
  - Flavio Leitner <[fleitner@redhat.com](mailto:fleitner@redhat.com)>

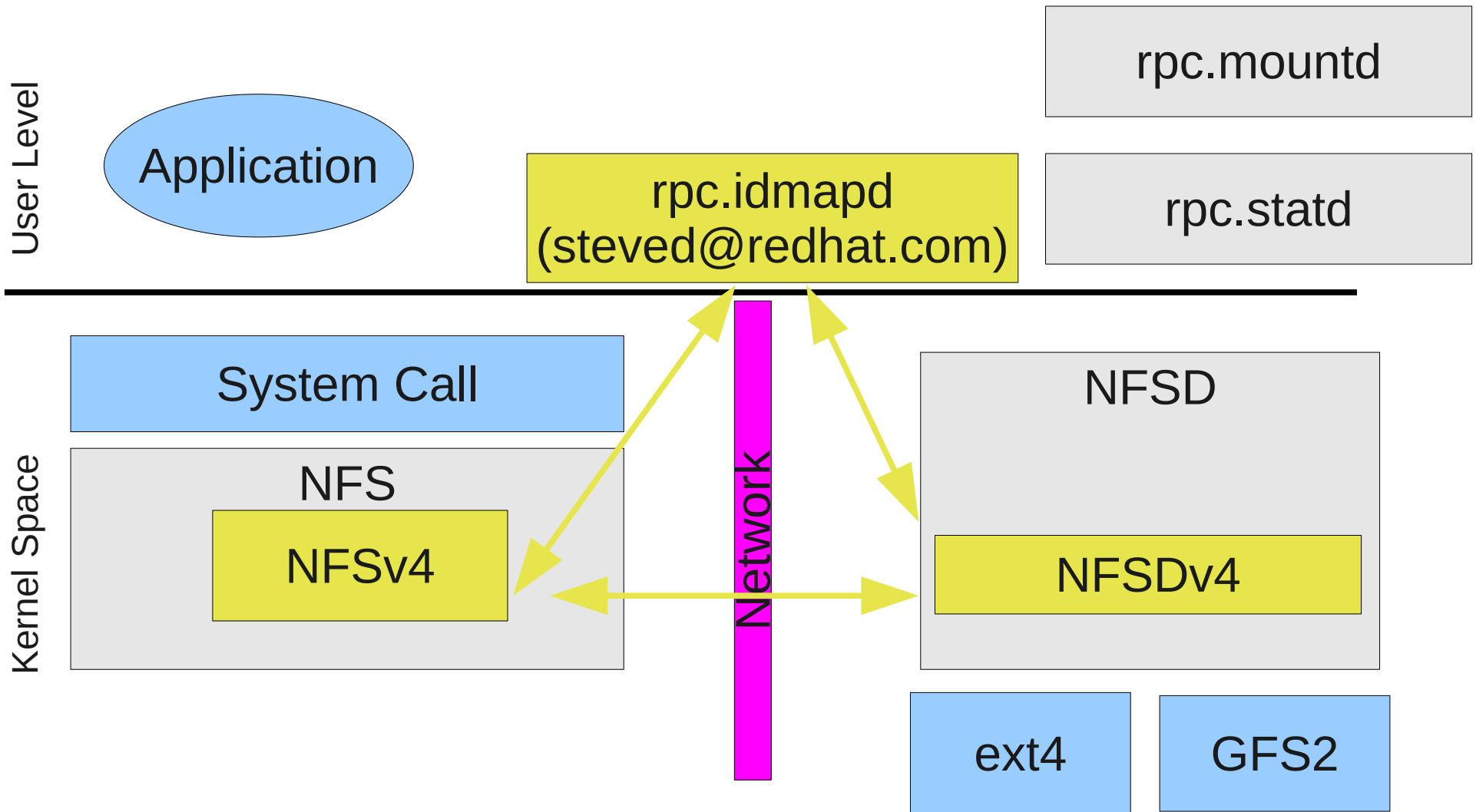
**SUMMIT**

**JBoss  
WORLD**

PRESENTED BY RED HAT



# NFSV4 Architecture



**SUMMIT**

**JBoss  
WORLD**

PRESENTED BY RED HAT



# 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.



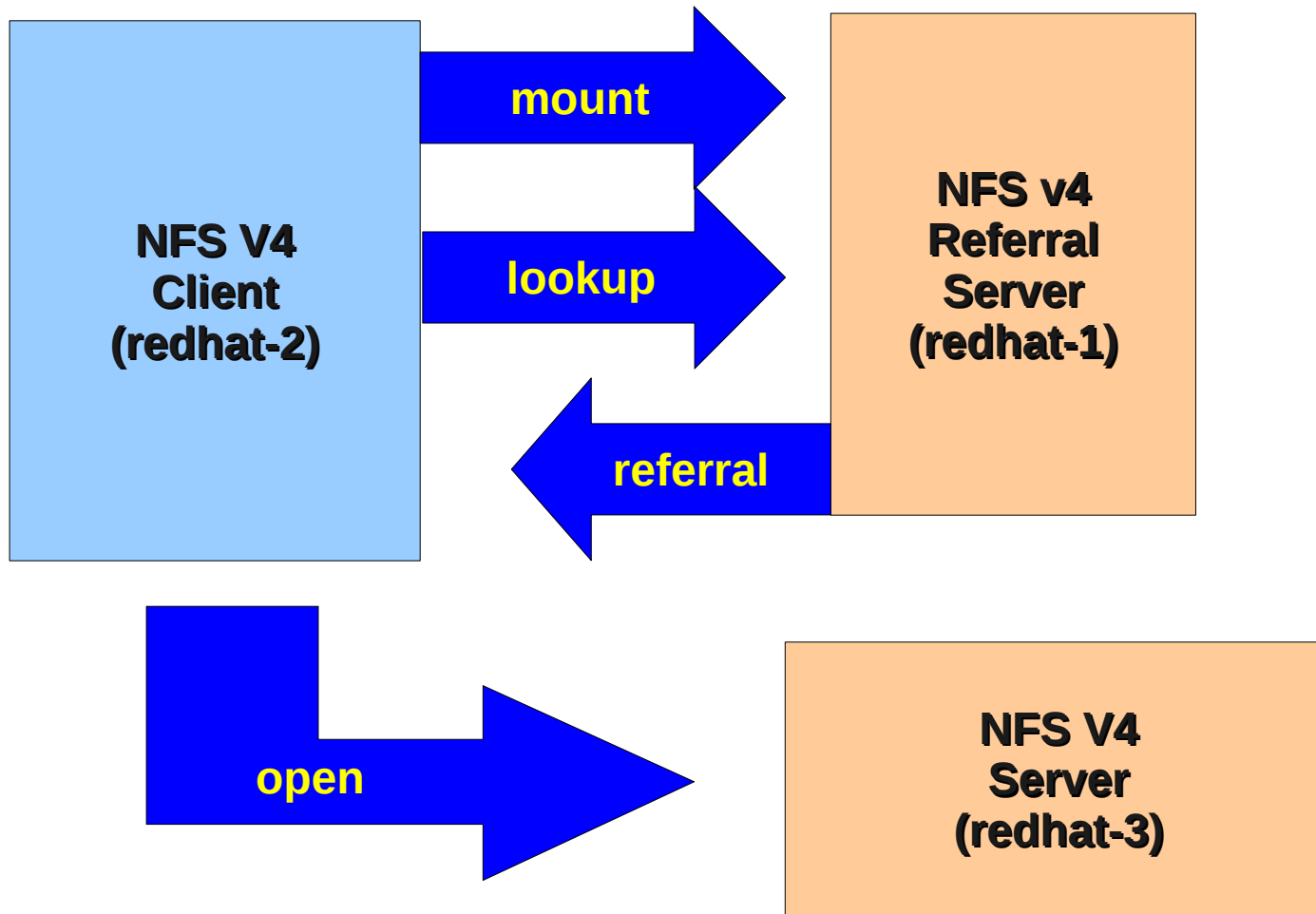


# NFS v4 Referrals

- 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



**SUMMIT**

JBoss  
WORLD

PRESENTED BY RED HAT

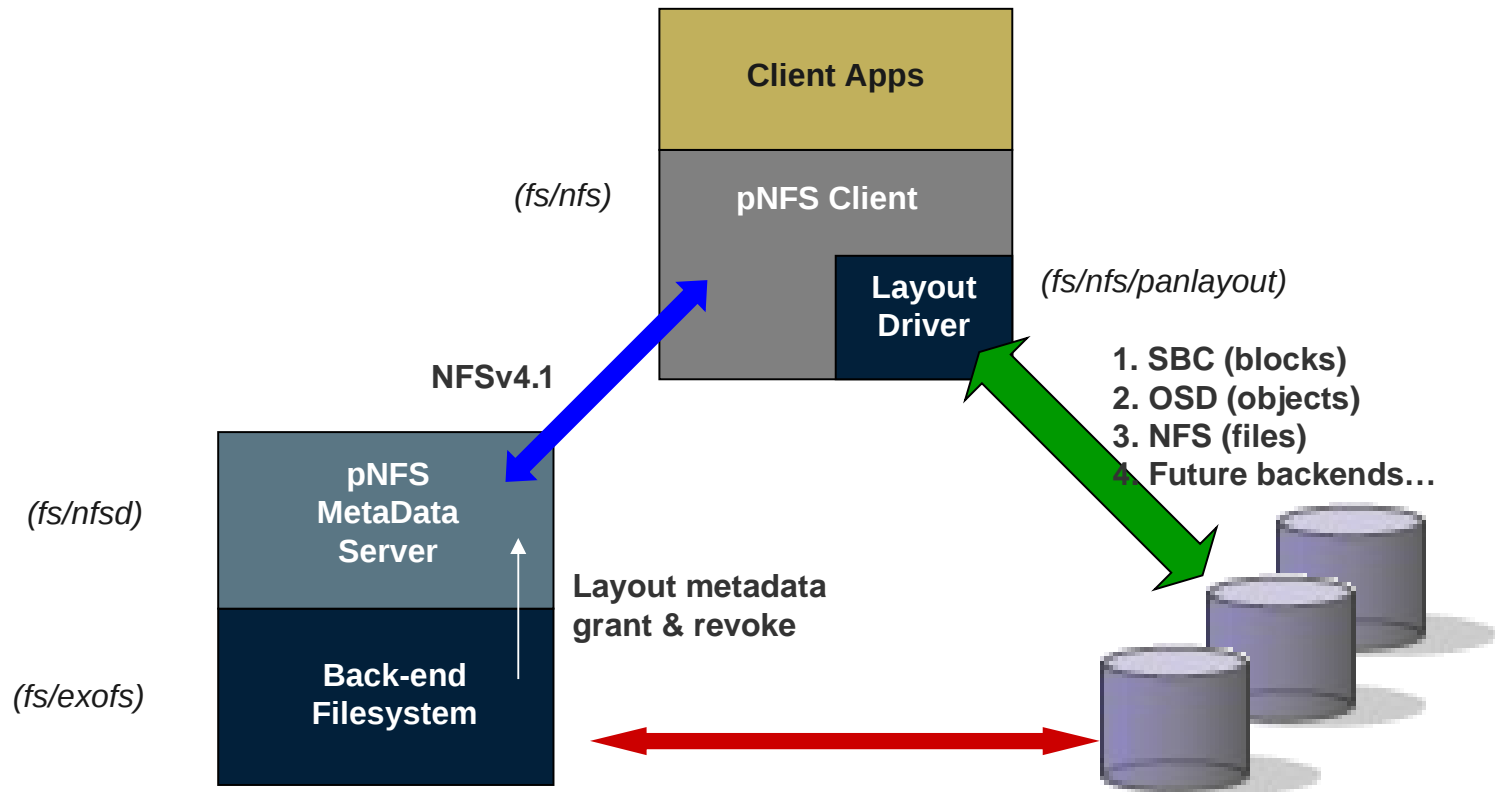


# NFS V4 Referrals - HOWTO

- 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



**SUMMIT**

**JBoss  
WORLD**

**PRESENTED BY RED HAT**

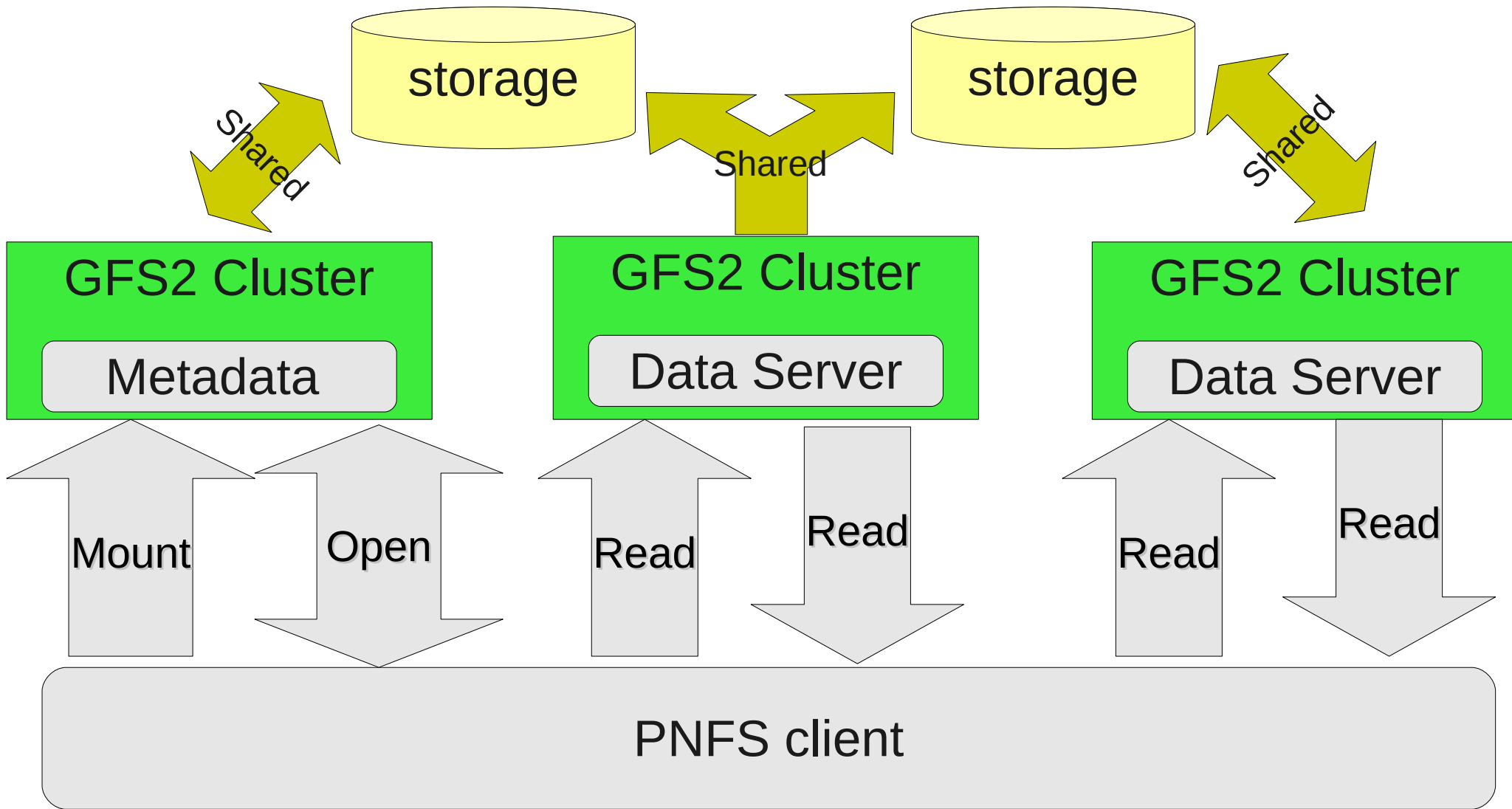


# Parallel NFS (pNFS)

- 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



**SUMMIT**

JBoss  
WORLD

PRESENTED BY RED HAT



# 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



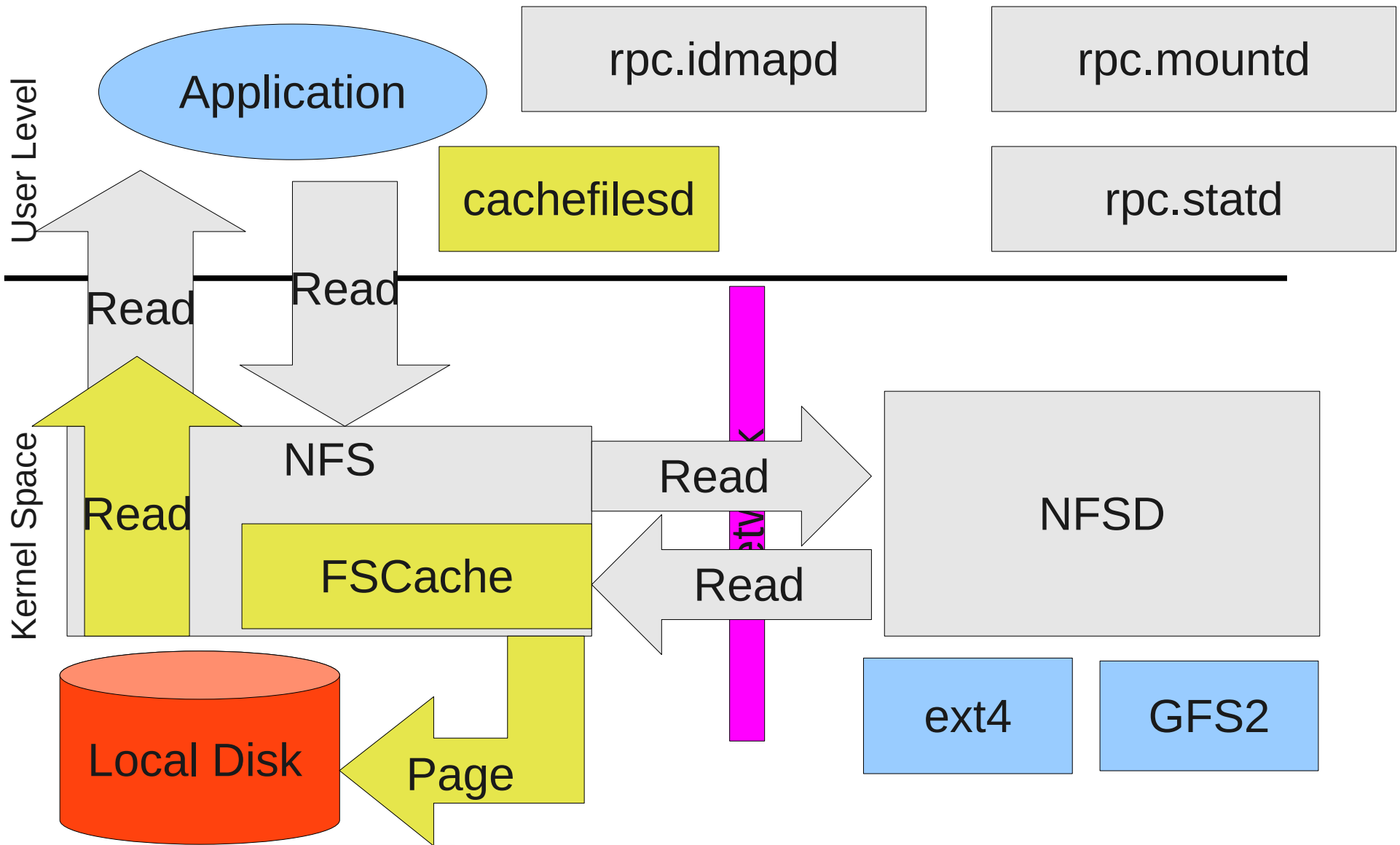
# FSCache

- 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



**SUMMIT**

JBoss  
WORLD

PRESENTED BY RED HAT

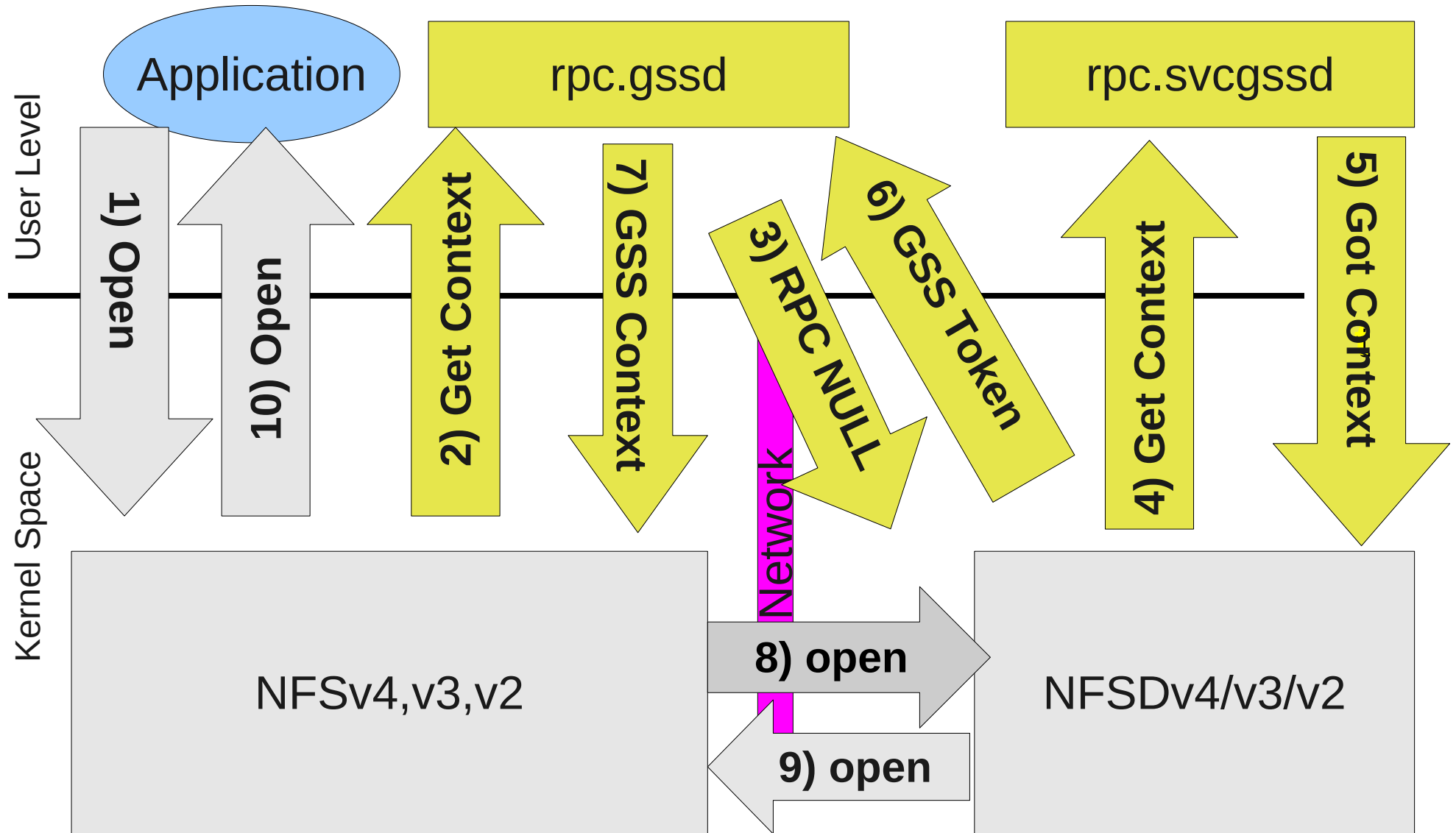


# FSCache – HOWTO

- 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



SUMMIT

JBoss  
WORLD

PRESENTED BY RED HAT



# 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
```

**SUMMIT**

**JBoss  
WORLD**

PRESENTED BY RED HAT



# 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 files 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/O 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

```
messages - System Log Viewer
File Edit View Filters Help
May 4 13:37:24 badhat kernel: [ 1183.888743] decode_attr_fs_locations: fs_locations done, error = 0
May 4 13:37:24 badhat kernel: [ 1183.888744] decode_attr_mode: file mode=01777
May 4 13:37:24 badhat kernel: [ 1183.888746] decode_attr_nlink: nlink=145
May 4 13:37:24 badhat kernel: [ 1183.888748] decode_attr_owner: uid=0
May 4 13:37:24 badhat kernel: [ 1183.888750] decode_attr_group: gid=0
May 4 13:37:24 badhat kernel: [ 1183.888751] decode_attr_rdev: rdev=(0x0:0x0)
May 4 13:37:24 badhat kernel: [ 1183.888753] decode_attr_space_used: space used=12288
May 4 13:37:24 badhat kernel: [ 1183.888754] decode_attr_time_access: atime=1304530348
May 4 13:37:24 badhat kernel: [ 1183.888756] decode_attr_time_metadata: ctime=1304530607
May 4 13:37:24 badhat kernel: [ 1183.888757] decode_attr_time_modify: mtime=1304530607
May 4 13:37:24 badhat kernel: [ 1183.888759] decode_attr_mounted_on_fileid: fileid=0
May 4 13:37:24 badhat kernel: [ 1183.888760] decode_getfattr: xdr returned 0
May 4 13:37:24 badhat kernel: [ 1183.888767] NFS: nfs_update_inode(0:1a/32770 ct=2 info=0x27e7f)
May 4 13:37:24 badhat kernel: [ 1183.888770] NFS: permission(0:1a/32770), mask=0x4, res=0
May 4 13:37:24 badhat kernel: [ 1183.888787] NFS: permission(0:1a/32770), mask=0x1, res=0
May 4 13:37:24 badhat kernel: [ 1183.888790] NFS: atomic_lookup(0:1a/32770), .Trash
May 4 13:37:24 badhat kernel: [ 1183.888792] NFS: lookup(/.Trash)
May 4 13:37:24 badhat kernel: [ 1183.888794] NFS call lookup .Trash
May 4 13:37:24 badhat kernel: [ 1183.888795] NFS call lookupfh .Trash
May 4 13:37:24 badhat kernel: [ 1183.888800] encode_compound: tag=
May 4 13:37:24 badhat kernel: [ 1183.888861] NFS reply lookupfh: -2
May 4 13:37:24 badhat kernel: [ 1183.888862] NFS reply lookup: -2
May 4 13:37:24 badhat kernel: [ 1183.888865] NFS: dentry_delete(/.Trash, 0)
May 4 13:37:24 badhat kernel: [ 1183.888892] NFS: permission(0:1a/32770), mask=0x1, res=0
May 4 13:37:24 badhat kernel: [ 1183.888895] NFS: atomic_lookup(0:1a/32770), .Trash-3606
May 4 13:37:24 badhat kernel: [ 1183.888897] NFS: lookup(/.Trash-3606)
May 4 13:37:24 badhat kernel: [ 1183.888898] NFS call lookup .Trash-3606
May 4 13:37:24 badhat kernel: [ 1183.888900] NFS call lookupfh .Trash-3606
May 4 13:37:24 badhat kernel: [ 1183.888903] encode_compound: tag=
May 4 13:37:24 badhat kernel: [ 1183.888955] NFS reply lookupfh: -2
May 4 13:37:24 badhat kernel: [ 1183.888957] NFS reply lookup: -2
May 4 13:37:24 badhat kernel: [ 1183.888959] NFS: dentry_delete(/.Trash-3606, 0)
```

SUMMIT

JBoss  
WORLD

PRESENTED BY RED HAT



# nfsiostat

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



# nfsiostat - (watch -n 0 nfsiostat)

```
Hagrid:pts/0
Every 0.1s: nfsiostat
Mon Apr 25 10:38:25 2011

RedHat:/home/tmp/Hagrid/nfsv4tcp mounted on /mnt/nfsv4tcp:

  op/s      rpc bklog
 664.75     0.00
read:      ops/s      kB/s      kB/op      retrans    avg RTT (ms)  avg exe (ms)
          4.750     30.784    6.481      0 (0.0%)    0.263         0.316
write:     ops/s      kB/s      kB/op      retrans    avg RTT (ms)  avg exe (ms)
          21.250    78.813    3.709      0 (0.0%)    0.471         0.529

RedHat:/home/tmp/Hagrid/nfsv3tcp mounted on /mnt/nfsv3tcp:

  op/s      rpc bklog
 734.75     0.09
read:      ops/s      kB/s      kB/op      retrans    avg RTT (ms)  avg exe (ms)
          13.250    6147.571  463.968    0 (0.0%)    122.283       179.962
write:     ops/s      kB/s      kB/op      retrans    avg RTT (ms)  avg exe (ms)
          36.250    7762.178  214.129    0 (0.0%)    2.310        105.697
```

**SUMMIT**

**JBoss  
WORLD**

PRESENTED BY RED HAT



# 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,rsiz=524288,wsiz=524288,namlen=255,acregmin=3,acregmax=60,acdirmin=30,at=0,timeo=600,retrans=2,sec=sys,clientaddr=192.168.62.7,minorversion=0
  NFS server capabilities: caps=0x7ffe,wtmult=512,dtsiz=4096,bsiz=0,namlen=255
  NFSv4 capability flags: bm0=0xfdfbfff,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 0_DIRECT read(2)
  applications wrote 0 bytes via 0_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
```

**SUMMIT**

**JBoss  
WORLD**

PRESENTED BY RED HAT





# mountstats – Proccotol 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 (milliseconds)
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)
```

**SUMMIT**

**JBoss  
WORLD**

PRESENTED BY RED HAT



# iostat

- **iostat -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:
badhat:/home/
      94890.67     85.33     0.00     0.00   95232.00  120490.67   211.00    93.00  117.67
Filesystem:
badhat:/home/
     103946.67     0.00     0.00     0.00  103765.33  128341.33   231.33   101.33  125.33
Filesystem:
badhat:/home/
     162634.67     0.00     0.00     0.00  162816.00     0.00   159.33   159.00    0.00
Filesystem:
badhat:/home/
     212202.67   232218.67     0.00     0.00  211861.33   14677.33   216.67   207.00   14.33
^C
Hagrid$
```

**SUMMIT**

**JBoss  
WORLD**

PRESENTED BY RED HAT



# 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

eth0 (host redhat) - Wireshark

Filter: **rpc** Expression... Clear Apply

| No. | Time     | Source       | Destination  | Protocol | Info   |
|-----|----------|--------------|--------------|----------|--|
| 198 | 0.034780 | 192.168.62.7 | 192.168.62.8 | NFS      | V3 ACCESS Call (Reply In 199), FH:0x0945a986               |
| 199 | 0.034953 | 192.168.62.8 | 192.168.62.7 | NFS      | V3 ACCESS Reply (Call In 198)                              |
| 200 | 0.035030 | 192.168.62.7 | 192.168.62.8 | NFS      | V3 LOOKUP Call (Reply In 202), DH:0x0945a986/Hagrid.test   |
| 201 | 0.035201 | 192.168.62.7 | 192.168.62.8 | NFS      | V2 GETATTR Call (Reply In 203), FH:0xa2401c5f              |
| 202 | 0.035216 | 192.168.62.8 | 192.168.62.7 | NFS      | V3 LOOKUP Reply (Call In 200) Error:NFS3ERR_NOENT          |
| 203 | 0.035314 | 192.168.62.8 | 192.168.62.7 | NFS      | V2 GETATTR Reply (Call In 201)                             |
| 204 | 0.035338 | 192.168.62.7 | 192.168.62.8 | NFS      | V2 LOOKUP Call (Reply In 205), DH:0xa2401c5f/Hagrid.test   |
| 205 | 0.035597 | 192.168.62.8 | 192.168.62.7 | NFS      | V2 LOOKUP Reply (Call In 204), FH:0x475415ec               |
| 209 | 0.036100 | 192.168.62.7 | 192.168.62.8 | NFS      | V4 NULL Call (Reply In 211)                                |
| 211 | 0.036220 | 192.168.62.8 | 192.168.62.7 | NFS      | V4 NULL Reply (Call In 209)                                |
| 213 | 0.036584 | 192.168.62.7 | 192.168.62.8 | NFS      | V4 COMP Call (Reply In 214) <EMPTY> PUTROOTFH PUTROOTFH;GE |

Frame 213: 206 bytes on wire (1648 bits), 206 bytes captured (1648 bits)

- Ethernet II, Src: IntelCor\_27:c8:b2 (00:27:0e:27:c8:b2), Dst: Dell\_2f:a8:7c (00:13:72:2f:a8:7c)
- Internet Protocol, Src: 192.168.62.7 (192.168.62.7), Dst: 192.168.62.8 (192.168.62.8)
- Transmission Control Protocol, Src Port: 720 (720), Dst Port: nfs (2049), Seq: 45, Ack: 29, Len: 140
- Remote Procedure Call, Type:Call XID:0x4a323320
- Network File System, Ops(3): PUTROOTFH GETFH GETATTR
  - [Program Version: 4]
  - [V4 Procedure: COMPOUND (1)]
  - Tag: <EMPTY>
  - minorversion: 0
  - Operations (count: 3)
    - Opcode: PUTROOTFH (24)
    - Opcode: GETFH (10)
    - Opcode: GETATTR (9)
      - GETATTR4args
        - attr request

```
0000 00 13 72 2f a8 7c 00 27 0e 27 c8 b2 08 00 45 00  ..r/.|. ' .....E.
0010 00 c0 a5 44 40 00 40 06 97 93 c0 a8 3e 07 c0 a8  ...D@.@. ....>...
0020 3e 08 02 d0 08 01 b8 30 3c 09 5d e6 d6 35 80 18  >.....0 <.]..5..
0030 00 2e 2e ec 00 00 01 01 08 0a 00 22 9e 9c 1e fa  ....."......
```

File: "/tmp/wiresharkXXXXNrvZyl"... Packets: 53656 Displayed: 26605 Marked: 0 Dropped: 10962 Profile: Default

SUMMIT

JBoss  
WORLD

PRESENTED BY RED HAT



# 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](http://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](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](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>

**SUMMIT**

**JBoss  
WORLD**

PRESENTED BY RED HAT



# Questions

Available at Campground 2 at 4:20pm

Slide Deck available at:

<http://people.redhat.com/steved/Summit11/>

Email Address:

[steved@redhat.com](mailto:steved@redhat.com)

**SUMMIT**

**JBoss  
WORLD**

PRESENTED BY RED HAT



**LIKE US ON FACEBOOK**

[www.facebook.com/redhatinc](http://www.facebook.com/redhatinc)

**FOLLOW US ON TWITTER**

[www.twitter.com/redhatsummit](http://www.twitter.com/redhatsummit)

**TWEET ABOUT IT**

#redhat

**READ THE BLOG**

[summitblog.redhat.com](http://summitblog.redhat.com)

**GIVE US FEEDBACK**

[www.redhat.com/summit/survey](http://www.redhat.com/summit/survey)

**SUMMIT**

**JBoss  
WORLD**

PRESENTED BY RED HAT

