Debugging Gluster with Wireshark and SystemTap

Examples based on real user problems

Niels de Vos
Sr. Software Maintenance Engineer
Red Hat Global Support Services

FISL – 10 May 2014

GLUSTER COMMUNITY redhat.

Introduction

- Name: Niels de Vos
- Company: Red Hat
- Department: Global Support Services
- Job title: Sr. Software Maintenance Engineer
- Duties:
  - assist with solving complex customer support cases, write bugfixes/patches, document solutions
  - Sub-maintainer for Gluster/NFS, release-maintainer for glusterfs-3.5 (current stable version)

Agenda

- Gluster Overview
- Introduction in Wireshark
- Minimal explanation of SystemTap
- Use Cases
  - Mount failures
  - Hanging QEMU image access
  - Missing, or incorrect access time of files when writing through a CIFS/Samba mountpoint

Gluster Overview

Debugging Gluster with Wireshark and SystemTap

Examples based on real user problems

Terminology

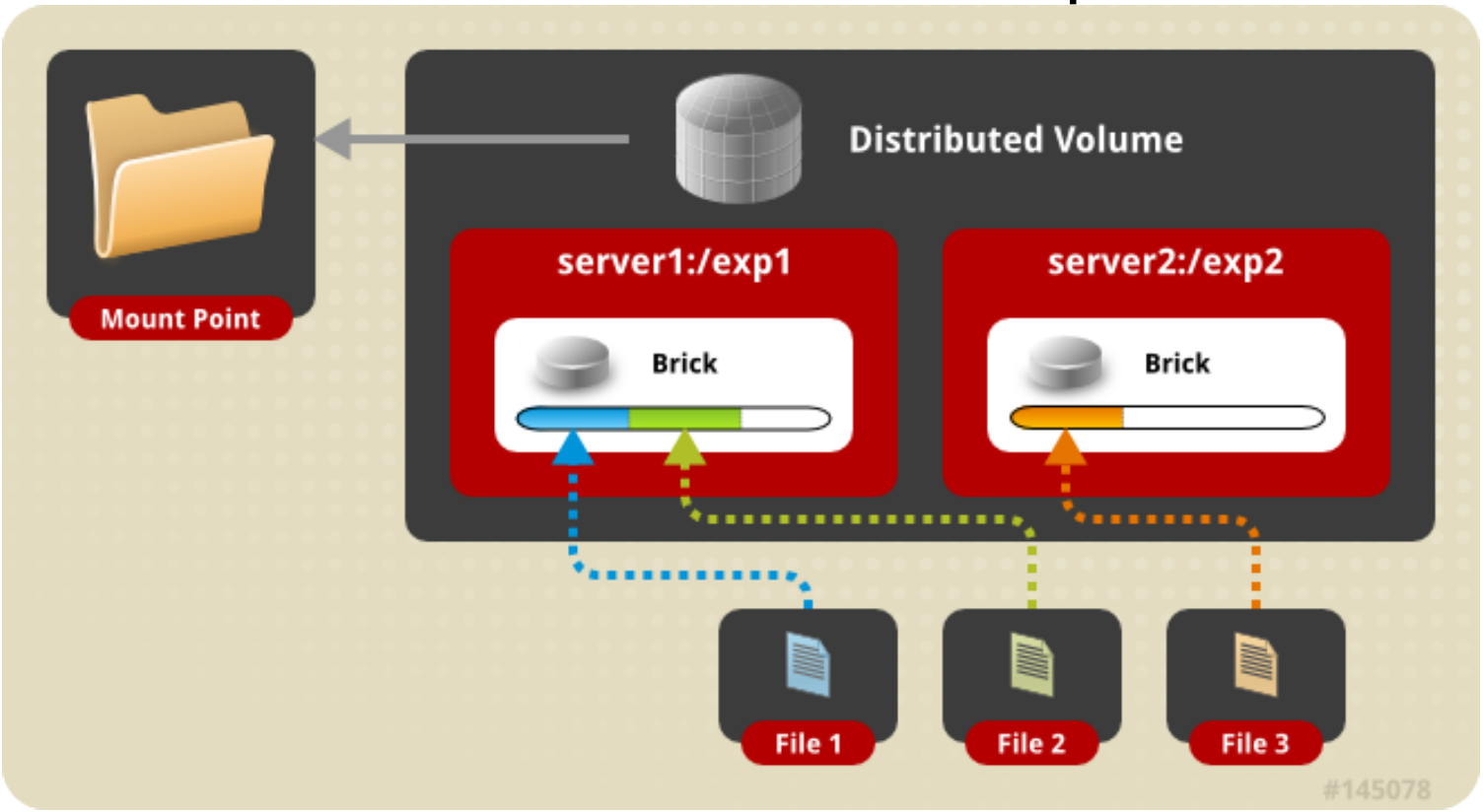
- Brick
  - Fundamentally, a filesystem mountpoint
  - A unit of storage used as a ***capacity*** building block
- Translator
  - Logic between the file bits and the Global Namespace
  - Layered to provide GlusterFS ***functionality***

Terminology

- Volume
  - Bricks combined and passed through translators
  - Ultimately, what's presented to the end user
- Peer / Node
  - Server hosting the brick filesystems
  - Runs the Gluster daemons and participates in volumes

Distributed Volume

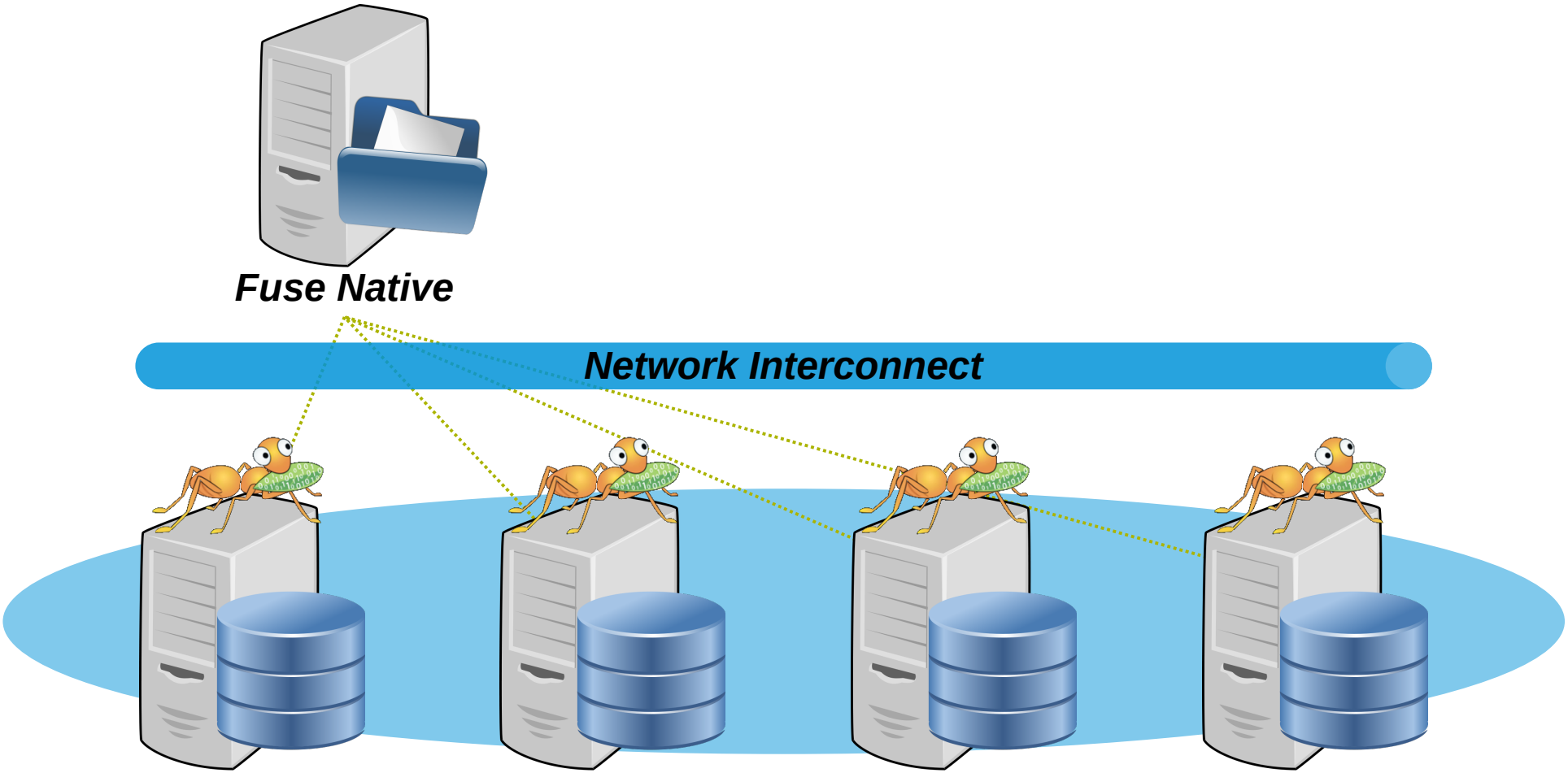
- Files “evenly” spread across bricks
- *Similar* to file-level RAID 0
- Server/Disk failure could be catastrophic



GlusterFS Native Client (FUSE)

- Specify mount to any GlusterFS server
- Native Client fetches volfile from mount server, then communicates directly with all nodes to access data
- The Virtual File System (VFS) from the Linux kernel communicates with the FUSE kernel module
- The FUSE kernel module has a connection (through `/dev/fuse`) with the GlusterFS-client daemon
- The GlusterFS-client relays the requests from the FUSE kernel module to the bricks

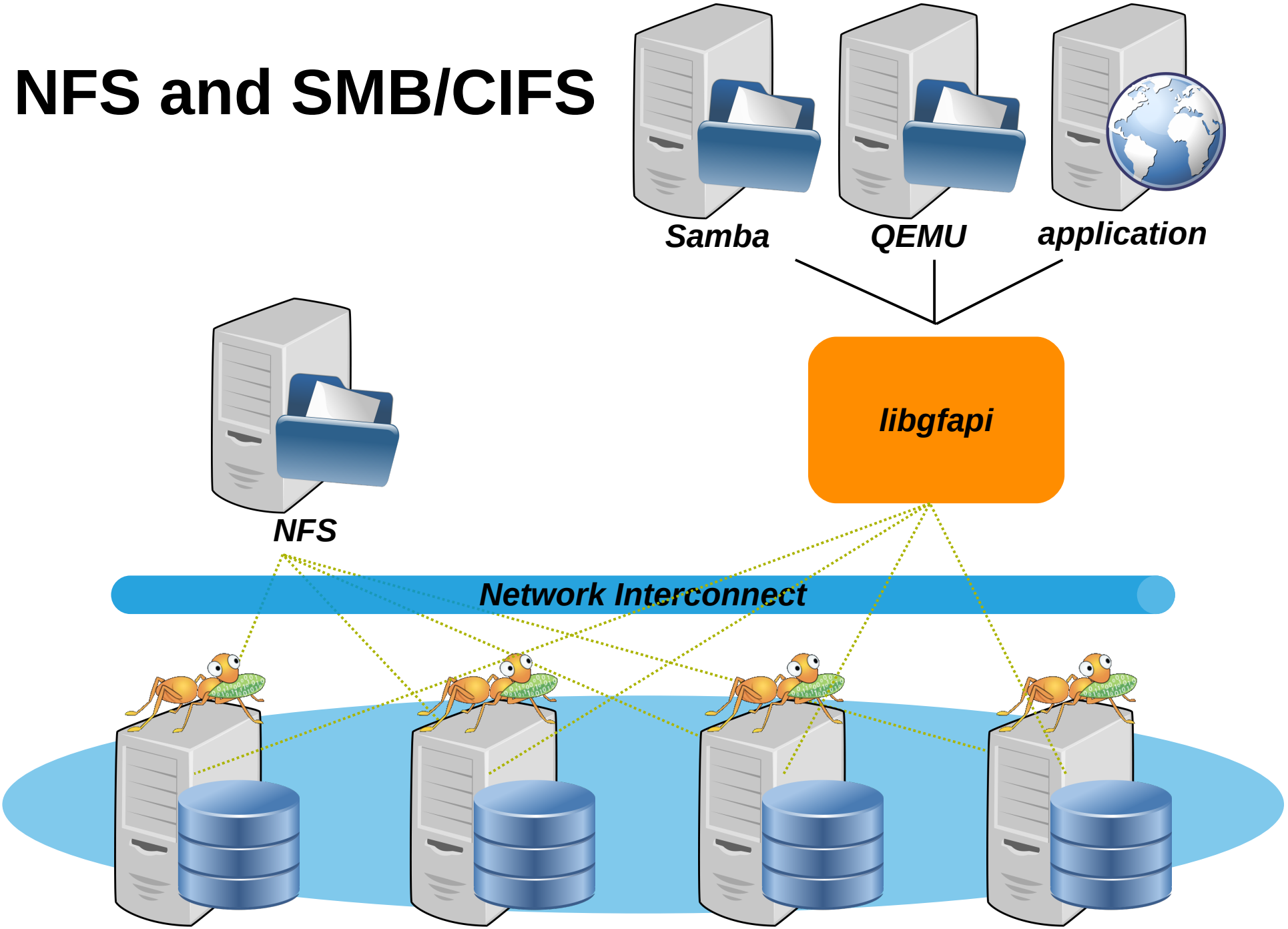
GlusterFS Native Client (FUSE)



NFS and SMB/CIFS

- Gluster/NFS:
  - Standard NFS v3 clients
  - Daemon as part of the glusterfs-server package
- SMB/CIFS:
  - Samba vfs_glusterfs plugin based on libgfapi
  - Configuration through the samba package
- A client mounts a single storage server
- The storage server acts like a GlusterFS-client and distributes/replicates the traffic
- Comparable with a gateway/proxy

NFS and SMB/CIFS



Introduction in Wireshark

Debugging Gluster with Wireshark and SystemTap

Examples based on real user problems

Introduction in Wireshark

- One of the most well known network protocol analyzers
- Can capture network traffic
- Can display hundreds of protocols
  - Version 1.8 and newer support GlusterFS
- Comes with several useful commandline tools
  - `tshark`, `editcap`, `capinfos`, ...
- Homepage: www.wireshark.org

Capturing network traffic

- Capture with Wireshark
  - Convenient, nice graphical interface
  - Analyze on the system used for capturing
  - Got (a recent) Wireshark on your server?
- Capture with `tcpdump`
  - Headless, no graphical environment needed
  - Separate production and analysis systems
  - Save in a file for off-line analysis
  - Can capture with rotating filenames

Capturing network traffic: examples

- Save to a file: `-w glusterfs.pcap`
- Capture on all interfaces: `-i any`
- Do not chop off packets: `-s 0`
- Filters:
  - Only TCP: `tcp`
  - Ports 24007 to 240100: `portrange 24007-240100`

Result:

```
# tcpdump glusterfs.pcap -i any -s 0 \
    tcp and portrange 24007-24100
```

GlusterFS protocols

- Everything is TCP
- Based on SUN Remote Procedure Calls
  - RFC 5531
  - Data is encoded in XDR (RFC 4506)
  - Similarities with portmapper and NFS
- A number of sub-protocols are used
  - GlusterFS is the most important one (I/O)

Simple explanation of SystemTap

Debugging Gluster with Wireshark and SystemTap

Examples based on real user problems

SystemTap Introduction

- Capable of dynamically inserting instrumentation in the **Linux kernel**
- This also is possible for **userspace applications**
- Similar to running through gdb with breakpoints and displaying of structures
- Useful for gathering statistics on function calls, including delays and timings
- .stp scripts get compiled as kernel modules and automatically loaded (temporary)

Excellent SystemTap documentation

- Many existing functions (tapsets) available for re-use
  - Located in `/usr/share/systemtap/tapset`
- Many examples with different purposes can be found in the SystemTap wiki
  - `/usr/share/doc/systemtap*/examples`
  - https://sourceware.org/systemtap/examples/

User Problem: mount failures

Debugging Gluster with Wireshark and SystemTap

Examples based on real user problems

User problem: Mount failures

- Example capture file: `mount-failure.pcap.gz`
- Filter: `tcp.len > 0`
- Protocol outline
  1. Gluster Handshake – GETSPEC
  2. Gluster DUMP – DUMP (for each brick)
  3. Gluster Portmap – PORTBYBRICK (for each brick)
  4. *Should* connect to each brick
- Check reply packets
- Find area of the failure, remove filter

Solution: Mount failures

- The client does not receive any replies from the bricks
  - iptables target REJECT responds with ICMP replies
  - iptables target DROP does not cause any replies
- **Solution:** verify the firewall on the client, storage servers and possibly any systems inbetween

- Sometimes it is needed to capture multiple tcpdumps on different locations in the network traject

User Problem: hanging QEMU image access

Debugging Gluster with Wireshark and SystemTap

Examples based on real user problems

User Problem: Hanging QEMU image access

- Create a qcow2 image over the gluster:// protocol

```
# qemu-img create -f qcow2 \
  gluster://storage-1.example.com/fisl/vm.img \
  512M
```

- Wait for the hang

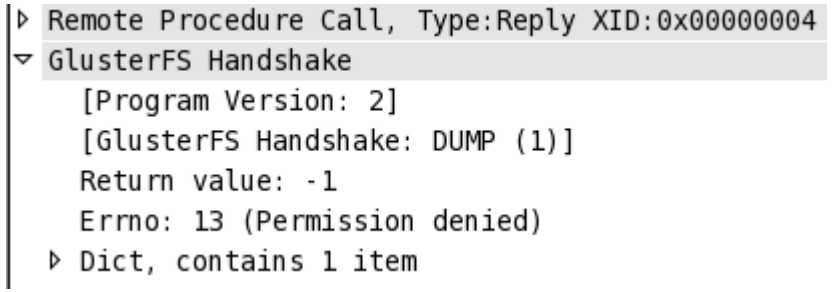
```
[nixpanic@vm122-229 ~]$ qemu-img create -f qcow2 gluster://storage-1.example.com/fisl/vm.img 512M
Formatting 'gluster://storage-1.example.com/fisl/vm.img', fmt=qcow2 size=536870912 encryption=off cluster_size=65536 lazy_refcounts=off
[2014-05-10 13:19:49.081034] E [client-handshake.c:1397:client_setvolume_cbk] 0-fisl-client-0: SETVOLUME on remote-host failed: Authentication failed
[2014-05-10 13:19:49.083312] E [client-handshake.c:1397:client_setvolume_cbk] 0-fisl-client-1: SETVOLUME on remote-host failed: Authentication failed
[2014-05-10 13:19:49.083773] E [client-handshake.c:1397:client_setvolume_cbk] 0-fisl-client-2: SETVOLUME on remote-host failed: Authentication failed
[2014-05-10 13:19:49.084734] E [client-handshake.c:1397:client_setvolume_cbk] 0-fisl-client-3: SETVOLUME on remote-host failed: Authentication failed
```

```
... SETVOLUME on remote-host failed:
Authentication failed ...
```

- Notice for the hang
- Also affects QEMU staring virtual machines

User Problem: Hanging QEMU image access

- Example capture file: `qemu-img.pcap.gz`
- Filter: `tcp.len > 0`
- Filter: `glusterfs.hndsk.proc`
- Filter: `glusterfs.hndsk.proc && rpc.msgtyp == 1`
- Expand the protocol tree of the RPC Reply



Solution: Hanging QEMU image access

- The storage servers (bricks) return Permission Denied
- **Solution:** follow the documentation, and pay special attention to 'stopping and starting the volume'. Stopping and starting the volume is not the same as rebooting the storage server.

- Configuration Guide for libvirt/QEMU with Gluster on the Gluster Community Wiki

  (http://www.gluster.org/community/documentation/index.php/Libgfapi_with_qemu_libvirt)

User Problem: wrong/missing access time

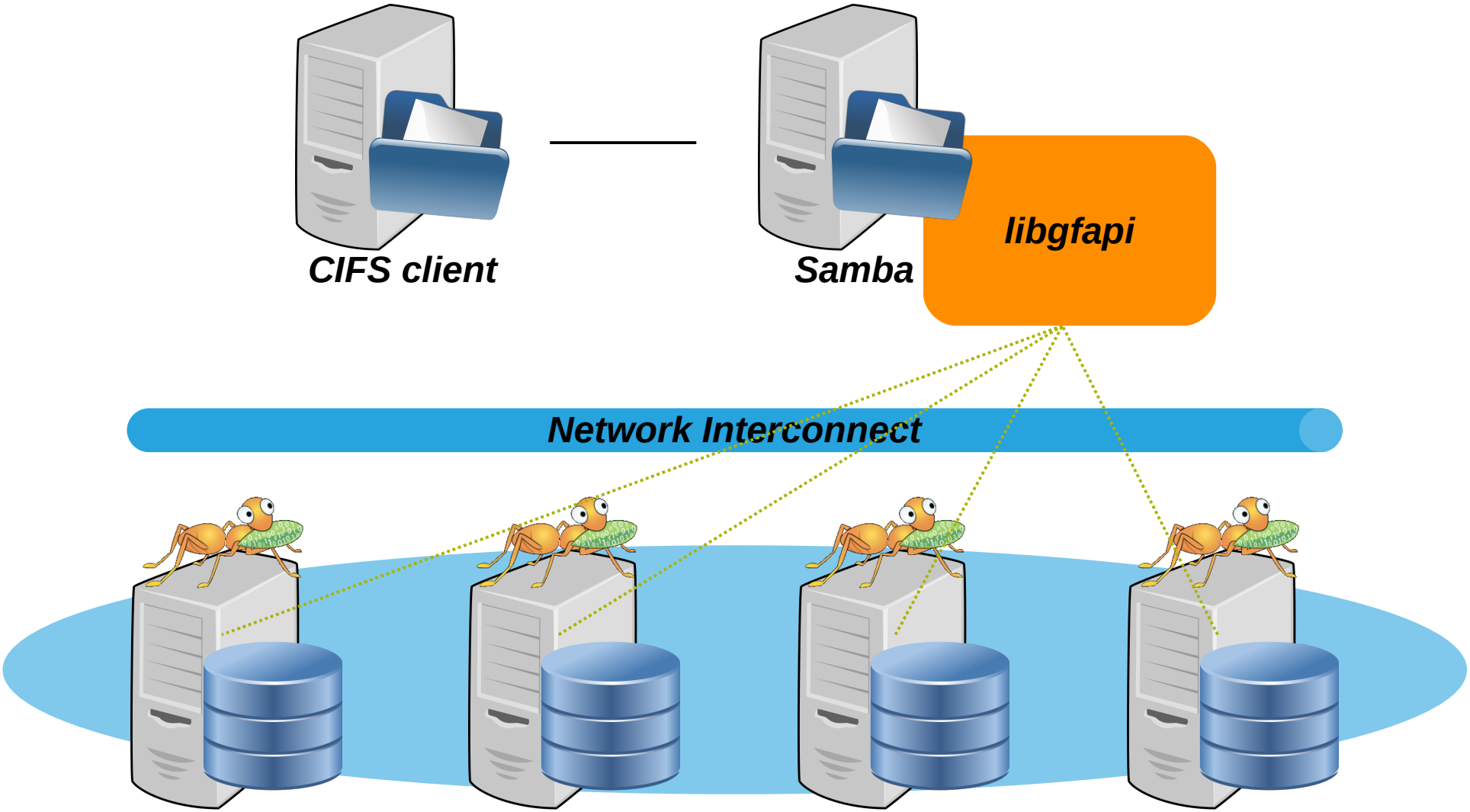
Debugging Gluster with Wireshark and SystemTap

Examples based on real user problems

User Problem: wrong/missing access time

- After writing to an existing file located on a share provided by Samba, the access time is ...
  - ... not displayed in Windows Explorer
  - ... very far in the future on Linux
- Does not happen when the share is located on a FUSE mountpoint (exported by Samba)
- Only happens with the new vfs_glusterfs module for Samba (libgfapi integration)

Samba (CIFS)



User Problem: wrong/missing access time

- Use systemtap to print the received and send access time:

```
#!/usr/bin/stap

probe process("/usr/lib*/libgfapi.so.*").function("glfs_utimens"),
      process("/usr/lib*/libgfapi.so.*").function("glfs_utimens")
{
        printf("%s: atime->tv_sec=%ld\n", probefunc(), $times[0]->tv_sec);
        printf("%s: mtime->tv_sec=%ld\n", probefunc(), $times[1]->tv_sec);
}

probe
process("/usr/lib*/samba/vfs/glusterfs.so").function("vfs_gluster_ntimes")
{
        printf("%s: atime->tv_sec=%ld\n", probefunc(),
                $ft->atime->tv_sec);
        printf("%s: mtime->tv_sec=%ld\n", probefunc(),
                $ft->mtime->tv_sec);
}
```

User Problem: wrong/missing access time

- Resulting output:

```
vfs_gluster_ntimes: atime->tv_sec=-1
vfs_gluster_ntimes: atime->tv_nsec=0
vfs_gluster_ntimes: mtime->tv_sec=1389363885
vfs_gluster_ntimes: mtime->tv_nsec=0
glfs_utimens: atime->tv_sec=-1
glfs_utimens: atime->tv_nsec=0
glfs_utimens: mtime->tv_sec=1389363885
glfs_utimens: mtime->tv_nsec=0
glfs_utimens: atime->tv_sec=-1
glfs_utimens: atime->tv_nsec=0
glfs_utimens: mtime->tv_sec=1389363885
glfs_utimens: mtime->tv_nsec=0
```

Solution: wrong/missing access time

- The vfs_glusterfs module from Samba does not do any value checking of the access time
- In case the access time is -1, the access time should not get updated
- Libgfapi requires sending a access time

- **Solution:** in case the access time is -1, send the cached (in Samba) access time
- Upstream Samba report and patch:
  - http://thread.gmane.org/gmane.network.samba.internals/74524

Thank You!

Slides Available at: http://people.redhat.com/ndevos/talks/fisl15

- ndevos@redhat.com

  storage-sales@redhat.com
- RHS:

  www.redhat.com/storage
- GlusterFS:

  www.gluster.org
- Red Hat Global Support Services:

  access.redhat.com/support

@Glusterorg

@RedHatStorage

Gluster

Red Hat Storage

Debugging Gluster with Wireshark and SystemTap

Examples based on real user problems