Replacing Your Proprietary Scale-out NAS With GlusterFS

Jacob Shucart

SA, Red Hat

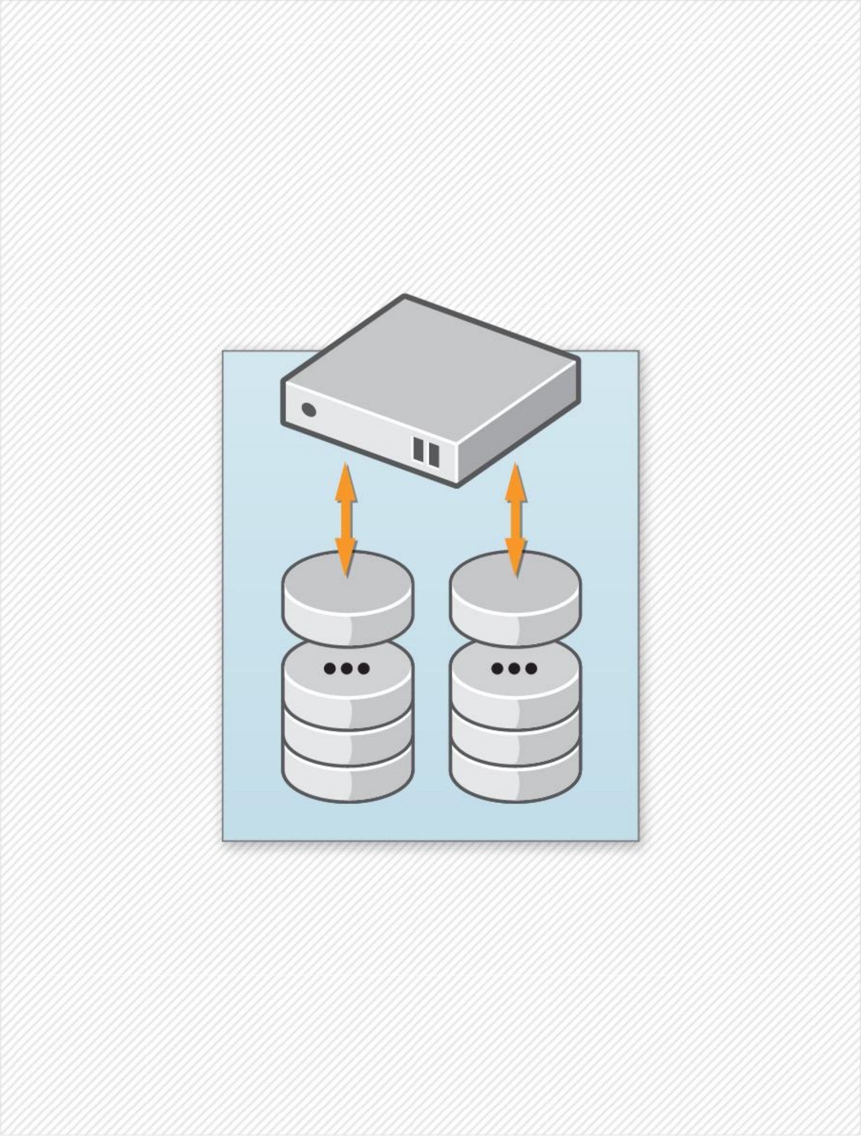
January 2012

Agenda

- Introduction
- Technology overview
    - High-level overview of what an implementation looks like
    - Understand the data flow
- Demonstration
- Q&A

Technology Overview -
Queue the marketing slides

What is GlusterFS?



Scale-out storage software for

- Unstructured / file data
- Objects
- Big data

Scalable

- Scales linearly and non-disruptively
- Performance
- Capacity
- Petabytes and beyond

Flexible

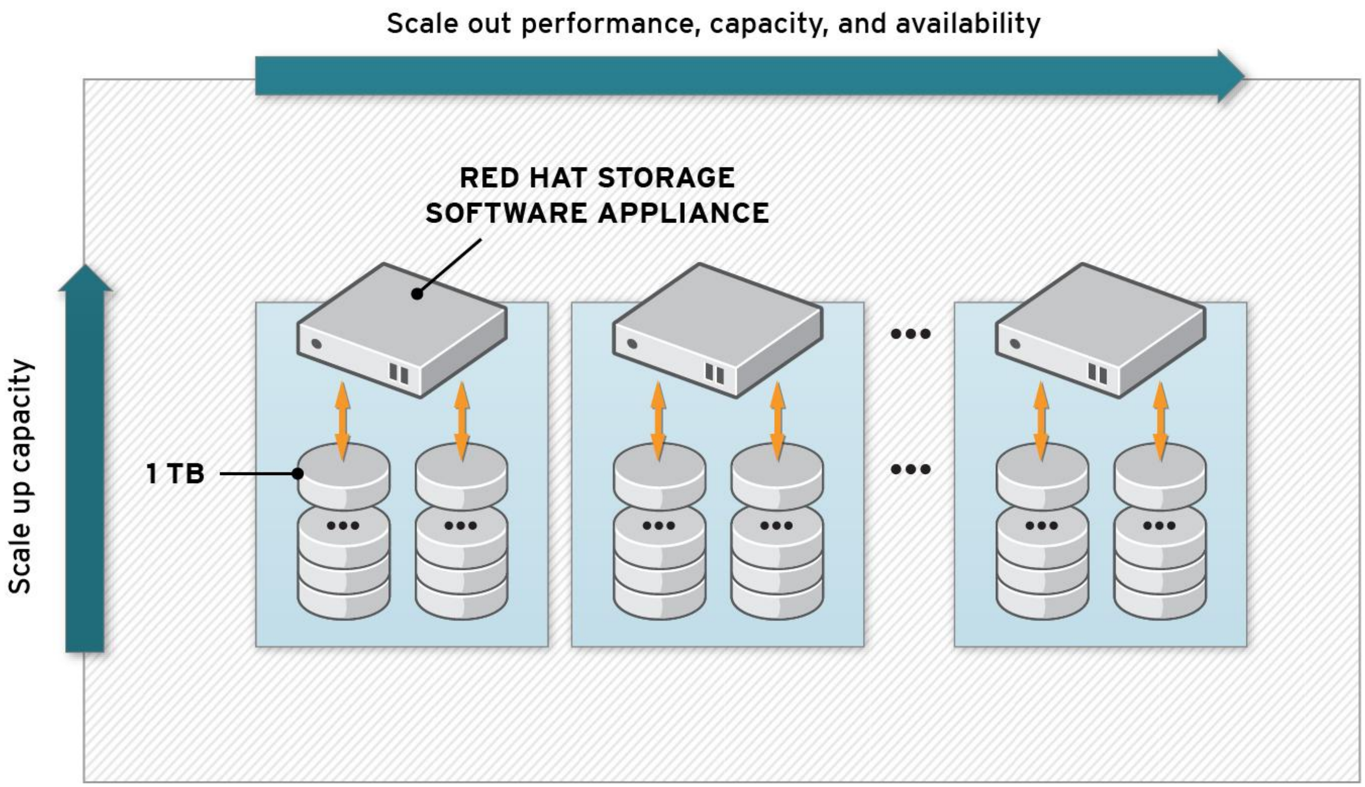
- Deploy anywhere
- Data center/private cloud
- Public cloud
- Hybrid cloud
- Multi-tenancy
- High Availability

Affordable

- Deploys on commodity hardware

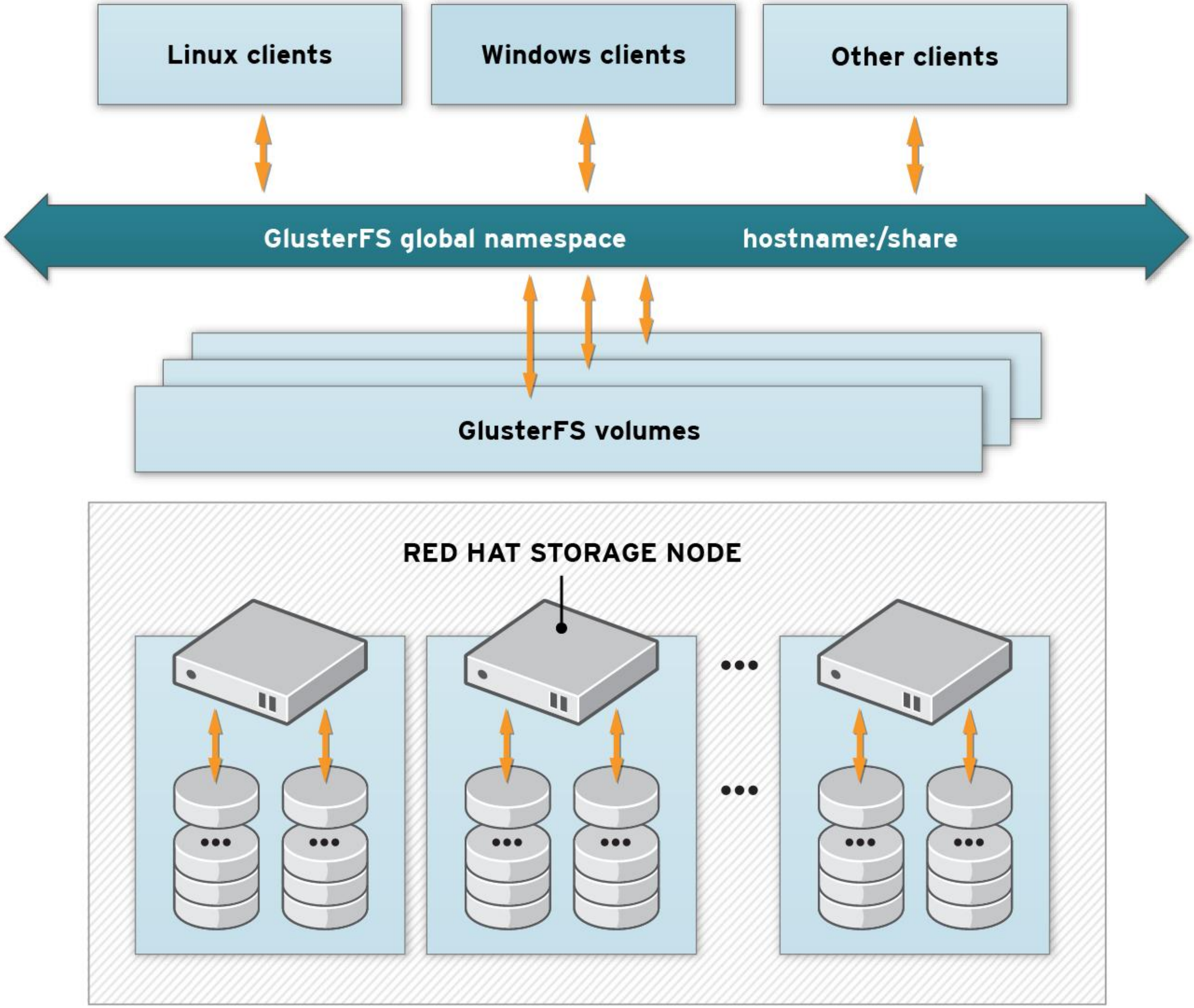
Use Case: Data Center / Private Cloud

Red Hat Storage Software Appliance


Scale out performance, capacity, and availability
RED HAT STORAGE SOFTWARE APPLIANCE
1 TB
Scale up capacity

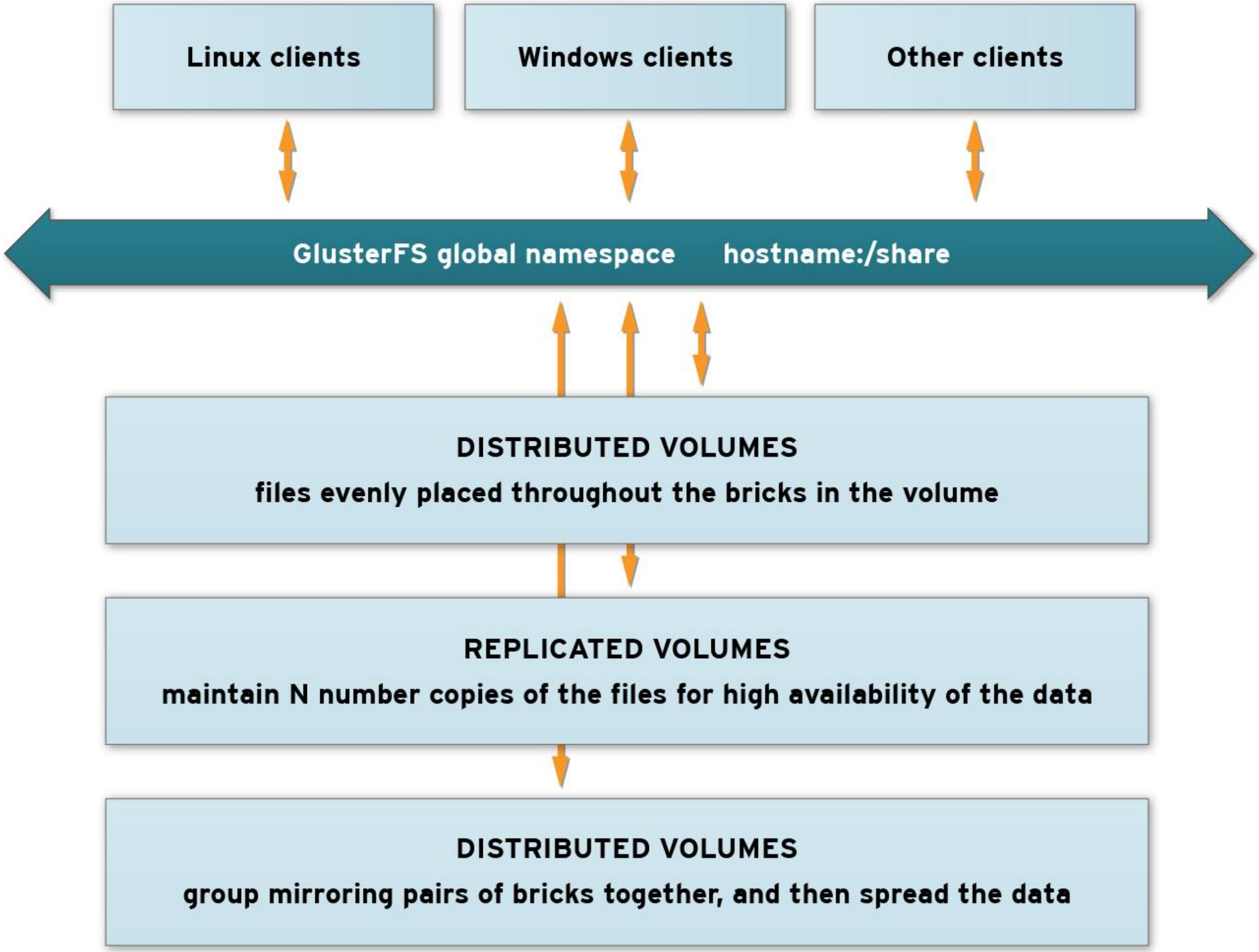
- Global namespace can span geographical distance
- GlusterFS file system
- Aggregates CPU, memory, network, capacity
- Deploys on Red Hat certified servers and underlying storage: DAS, JBOD.
- Scale-out linearly; performance and capacity as needed
- Replicate Synchronously and Asynchronously for high availability

Providing Access to Your Data



- GlusterFS enables you to create a Global Namespace
- On that namespace you can create volumes where data resides
- Clients access data from the volumes
- GlusterFS handles all volume-level policies
  - Distribute
  - Replicate
  - Geo-Rep
  - And more…

Gluster FS Handles Everything Else From There


Linux clients
Windows clients
Other clients
GlusterFS global namespace
hostname:/share
DISTRIBUTED VOLUMES
files evenly placed throughout the bricks in the volume
REPLICATED VOLUMES
maintain N number copies of the files for high availability of the data
DISTRIBUTED VOLUMES
group mirroring pairs of bricks together, and then spread the data

- Any GlusterFS node can handle client requests
- GlusterFS handles distributing, replicating, and remotely replicating the data
- Clients perceive volumes as being one share that they can read and write the data
- Everything that GlusterFS does behind that is transparent to the client

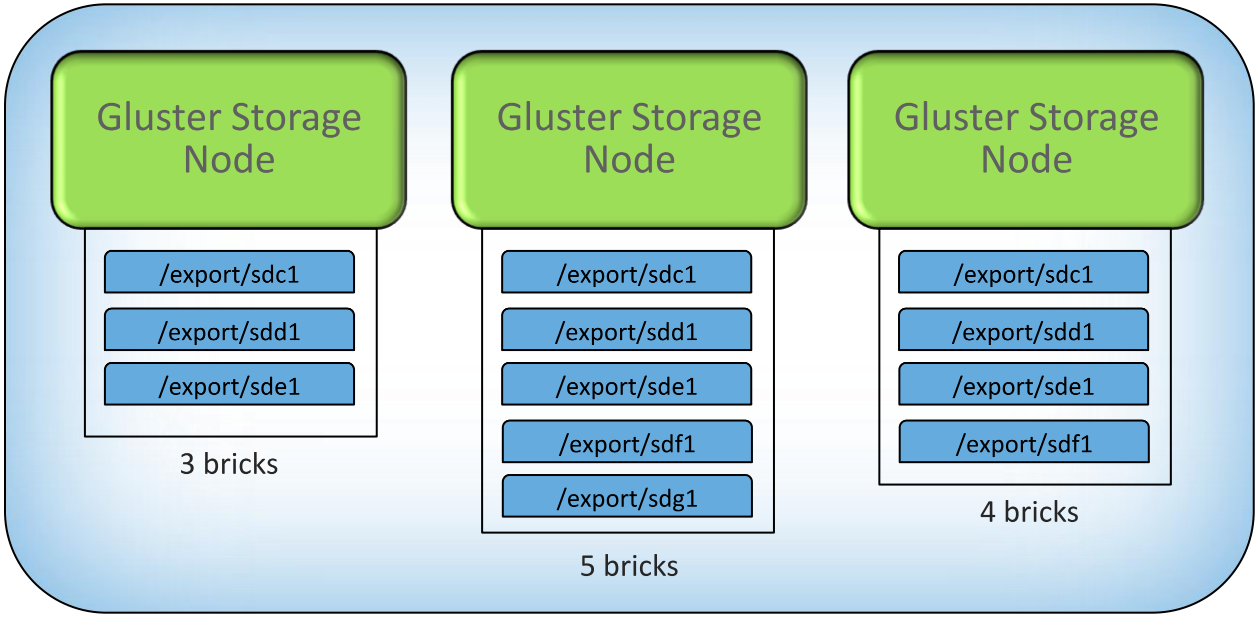
How Does GlusterFS Work Without Metadata?

- All storage nodes have an algorithm built-in
- All native clients have an algorithm built-in
- Files are placed on a brick(s) in the cluster based on a calculation
- Files can then be retrieved based on the same calculation
- For non-native clients, the server handles retrieval and placement

Gluster Technical Fundamentals

A Brick

- A brick is the combination of a node and a file system. hostname:/dir
- Each brick inherits limits of the underlying filesystem(ext3/ext4/xfs)
- No limit to the number bricks per node.
- Gluster operates at the brick level, not at the node level.
- Ideally each brick in a cluster should be the same size.



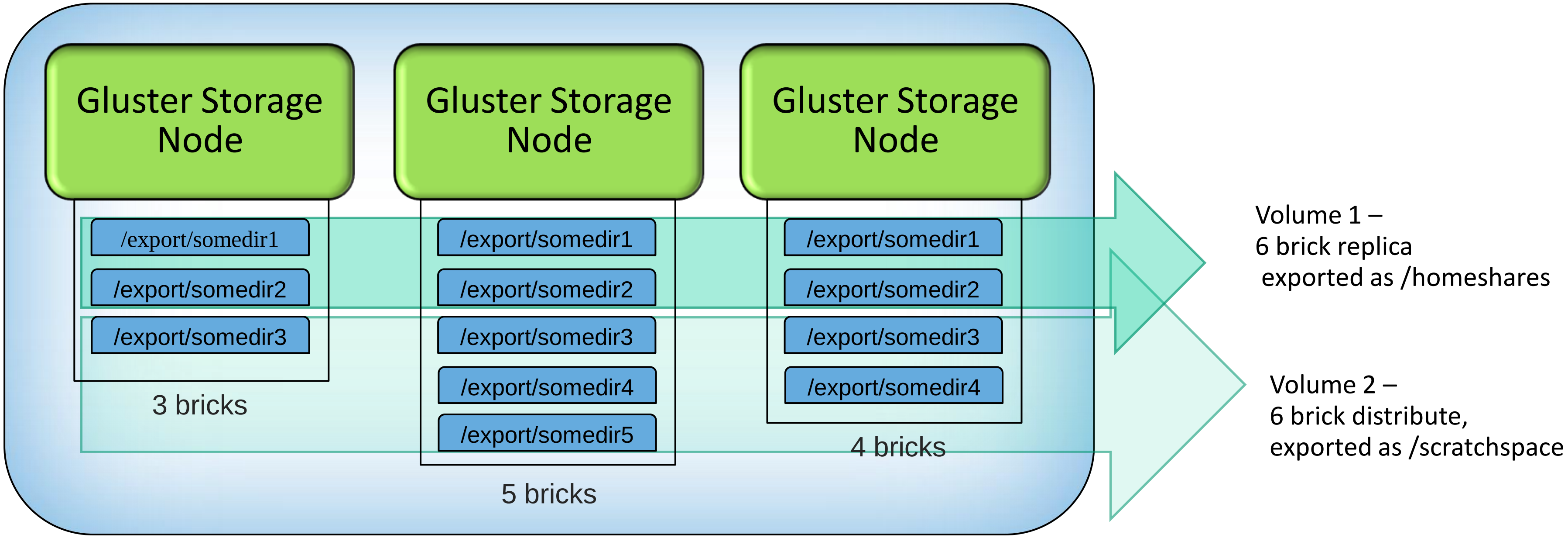
A Gluster cluster with 12 bricks.

Volumes

- A volume is some number of bricks => 2, clustered and exported with Gluster.
  - Volumes have administrator assigned names.
  - Volume name = export name.
  - A brick is a member of only one volume.

- A Gluster namespace can have 1 or more volumes.
  - A Gluster namespace can have a mix of replicated and distributed volumes.
  - Data in different volumes physically exists on different bricks.
  - Volumes can be sub-mounted on clients using NFS, CIFS and/or GlusterFS clients.

- The directory structure of the volume exists on every brick in the volume.

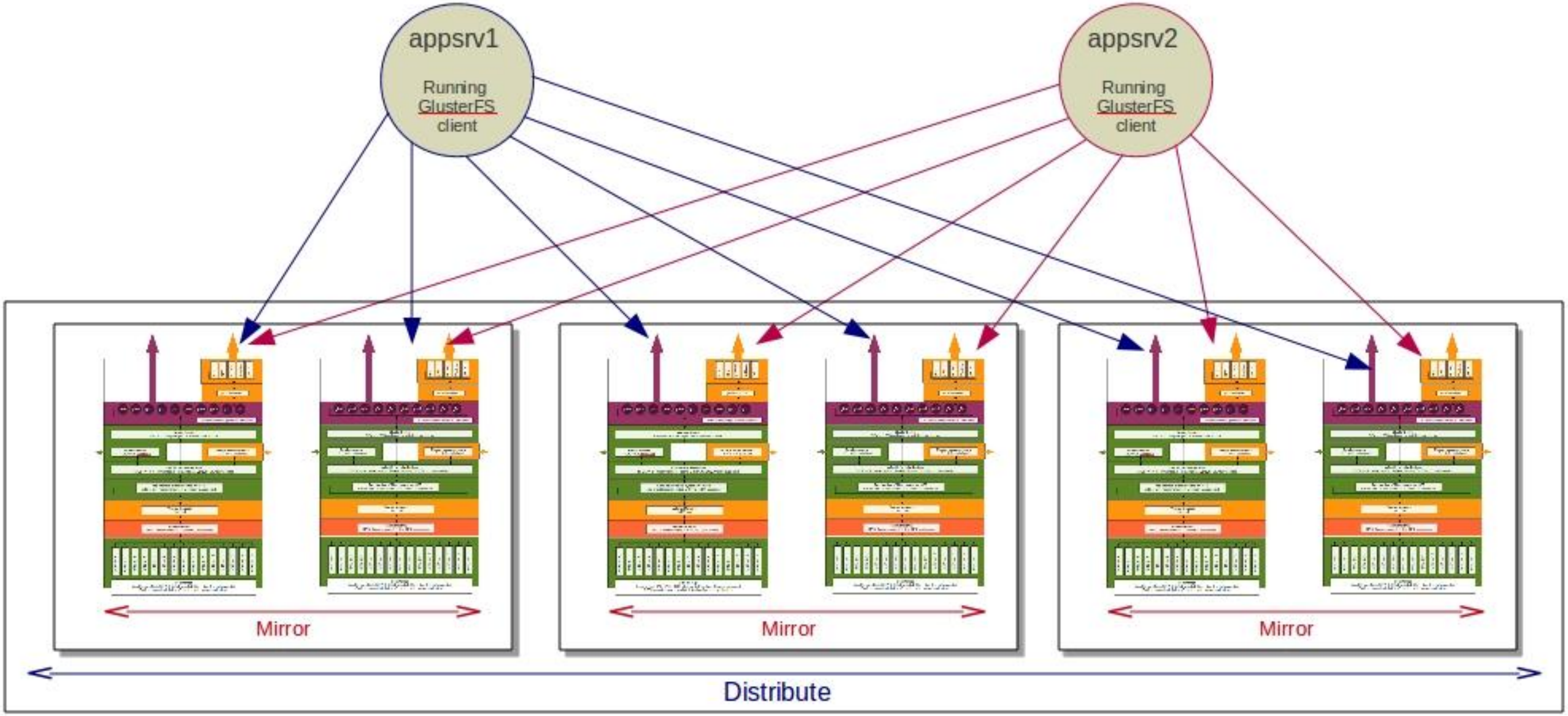


Volume Types

- Distribute
  - No data redundancy
  - Failure of a brick results in data access issues
- Replicate(or distribute + replicate)
  - Redundant at the brick level through synchronous writes
  - High availability
  - N replicas are supported
- Stripe(or distribute + stripe)
  - Limited use case(scratch space, very large files, some HPC)
  - Problems with small files

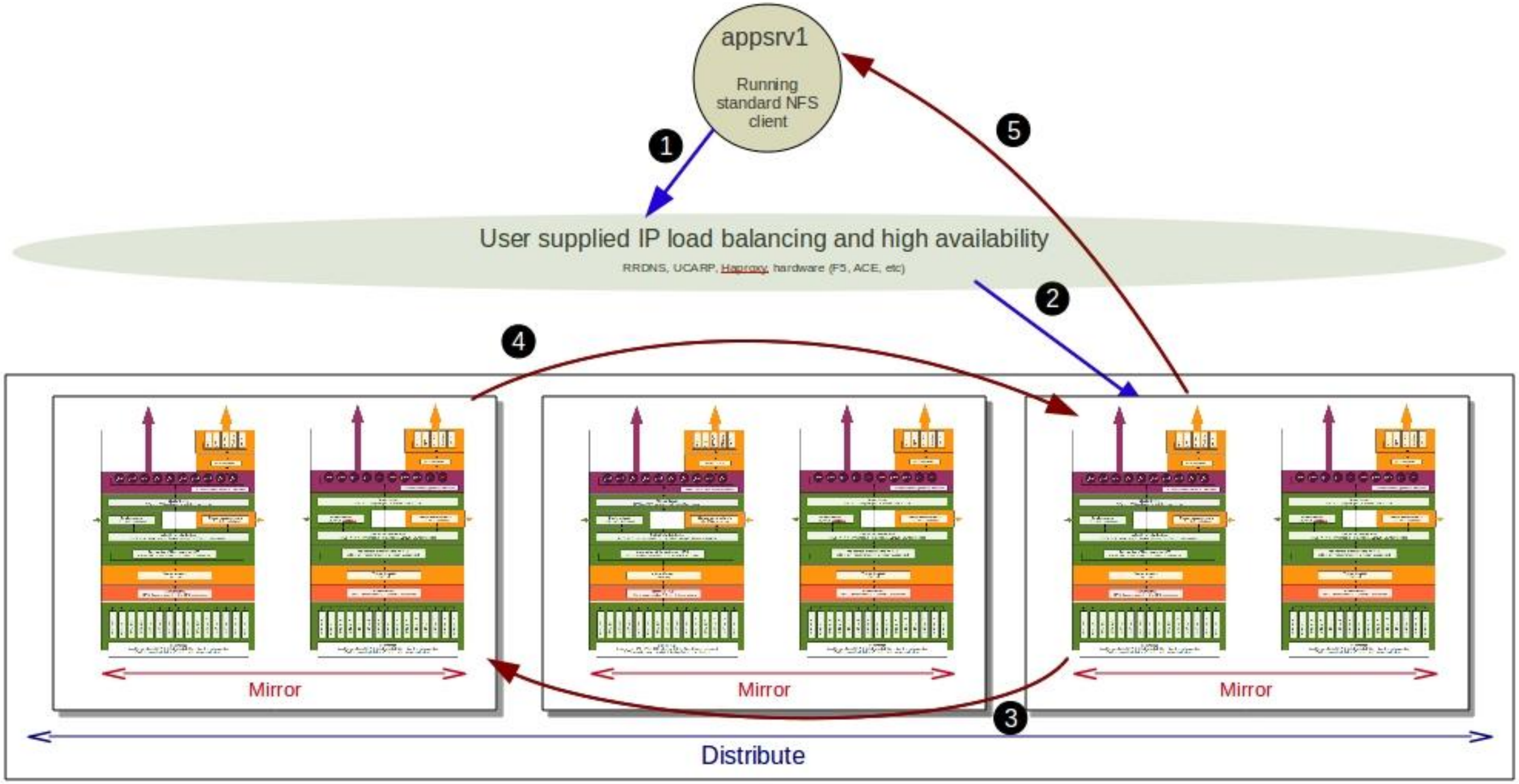
Gluster Technical Fundamentals

GlusterFS Native client data flow



Gluster Technical Fundamentals

NFS, CIFS dataflow


appsrv1
Running standard NFS client
1
5
User supplied IP load balancing and high availability
RRDNS, UCARP, Haproxy, hardware (F5, ACE, etc)
2
4
Mirror
Mirror
Mirror
3
Distribute

HA for NFS and CIFS

Any IP failover tool can work for NFS

- Appliance based load balancers with heartbeat such as F5
- Linux heartbeat, ucarp, CTDB
- Not all failover works for CIFS as that requires some session handling

CTDB is what we use

- It is very simple to configure
- Works for NFS
- Works for CIFS
- Is very robust and configurable

Round robin DNS for load balancing

- You can use any load balancer you want
- RRDNS is simple to configure and works well
- Prevents hot spots of activity

Sizing and Architecture

- **Gluster performance relies on hardware**
  - Number of systems depends on performance and capacity
  - There are many ways to meet customer needs
  - 2U & 4U DAS systems and JBODS are great building blocks
- **Capacity-centric environments**
  - 2U & 4U DAS systems with multiple JBODS
  - Lower RAM and CPU requirements
  - Lower network requirements
- **Mixed capacity and performance environments**
  - 2U & 4U DAS systems with 1-2 JBODS max
  - Higher RAM and CPU requirements
  - Low to high network requirements
- **High performance environments**
  - 1U or 2U systems with no JBODS
  - Highest RAM and CPU requirements
  - Fast disks and fast network

Checking System Requirements

Red Hat SSA 3.2 Configuration Guidelines

- Document link: https://access.redhat.com/kb/docs/DOC-66207

Client Dependency Packages

- Install required prerequisites on the client using the following command:

```
$ sudo yum -y install openssh-server  wget  fuse  fuse-libs  openib  libibverbs
```

- For Infiniband support, install **openib** and **libibverbs** packages.
- Portmapper for NFS

Gluster Packages

- http://download.gluster.com/pub/gluster/glusterfs/3.2/LATEST/
- **glusterfs-core** and **glusterfs-fuse** are required for Gluster Native Client
- **glusterfs-geo-replication** if you are using geo-replication
- **glusterfs-rdma** for Infiniband

Demonstration

Q&A and THANK YOU

Jacob Shucart

jshucart@redhat.com