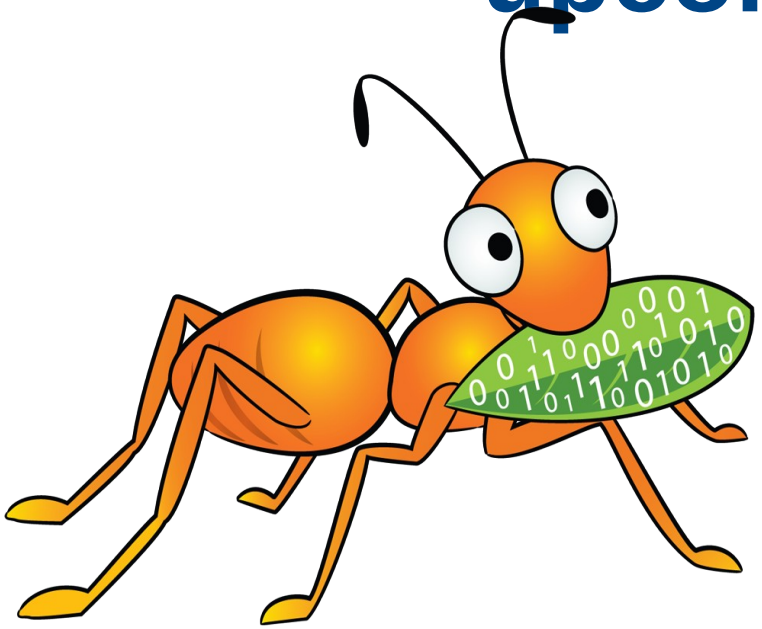
Gluster roadmap: Recent improvements and upcoming features

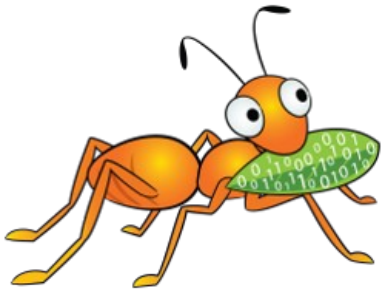


Niels de Vos
GlusterFS co-maintainer

ndevos@redhat.com

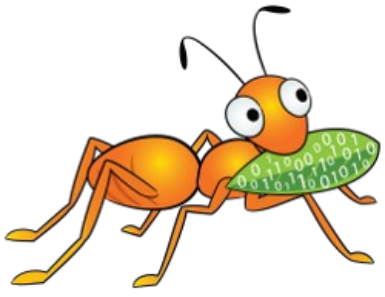
Agenda

- Introduction into Gluster
- Quick Start
- Current stable releases
- History of feature additions
- Plans for the upcoming 3.8 and 4.0 release
- Detailed description of a few select features



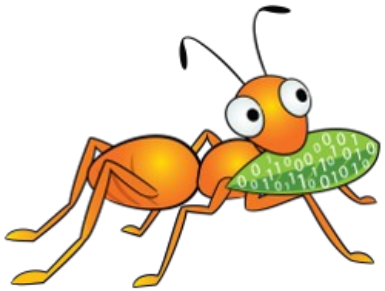
What is GlusterFS?

- Scalable, general-purpose storage platform
  - POSIX-y Distributed File System
  - Object storage (swift)
  - Distributed block storage (qemu)
  - Flexible storage (libgfapi)
- No Metadata Server
- Heterogeneous Commodity Hardware
- Flexible and Agile Scaling
  - Capacity – Petabytes and beyond
  - Performance – Thousands of Clients



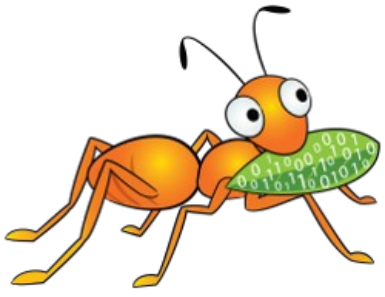
Terminology

- 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***

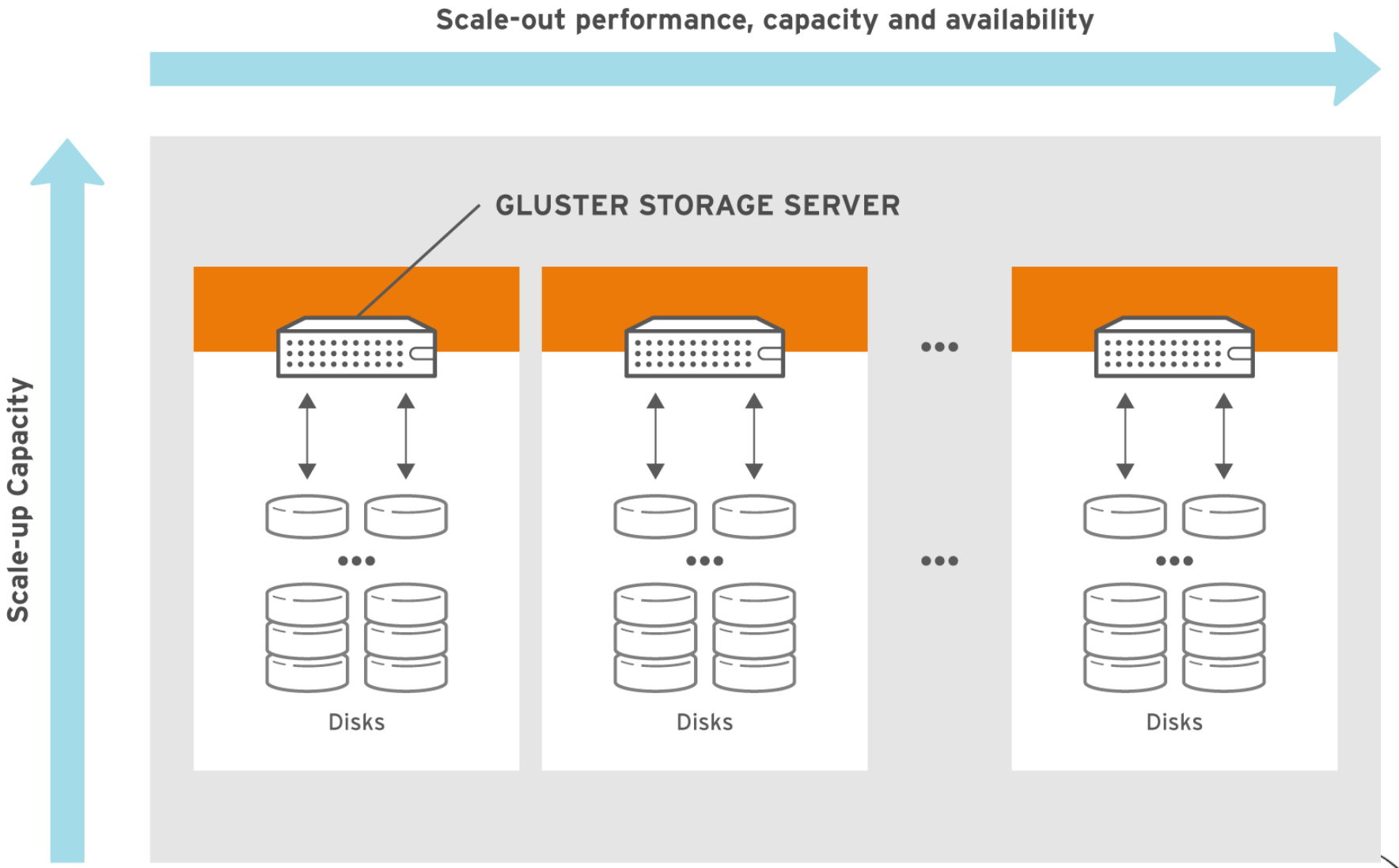


Terminology

- 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
- Trusted Storage Pool
  - A group of peers, like a “Gluster cluster”

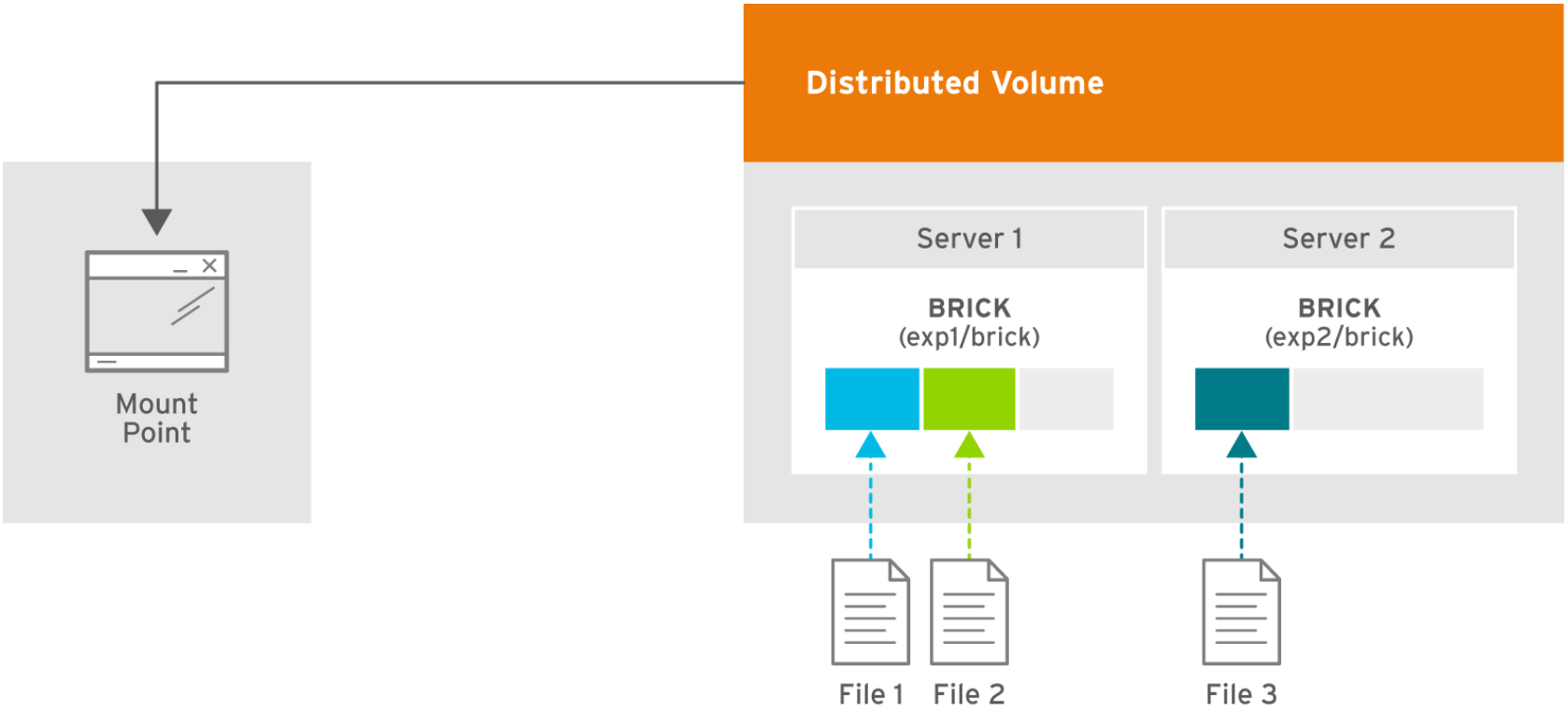


Scale-out and Scale-up



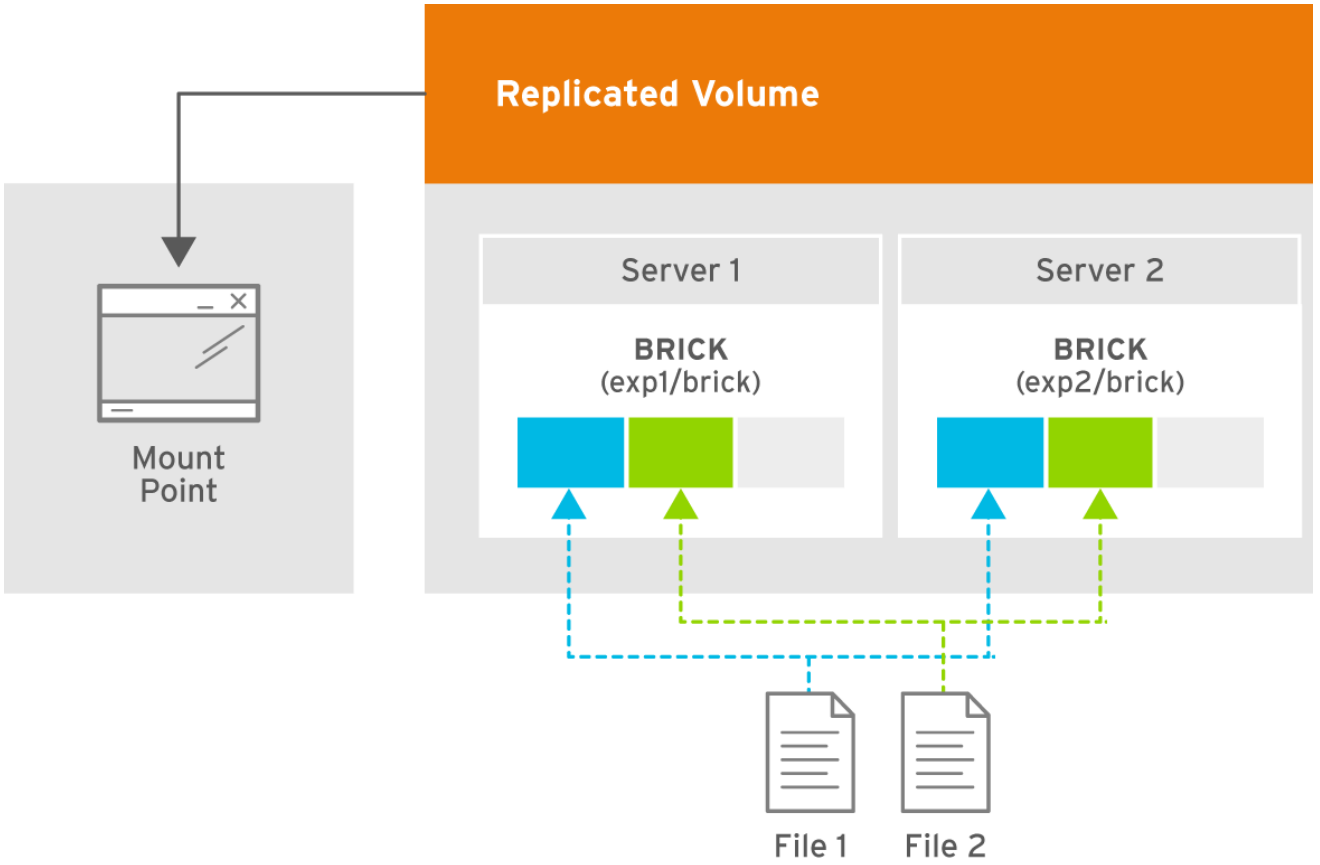
Distributed Volume

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



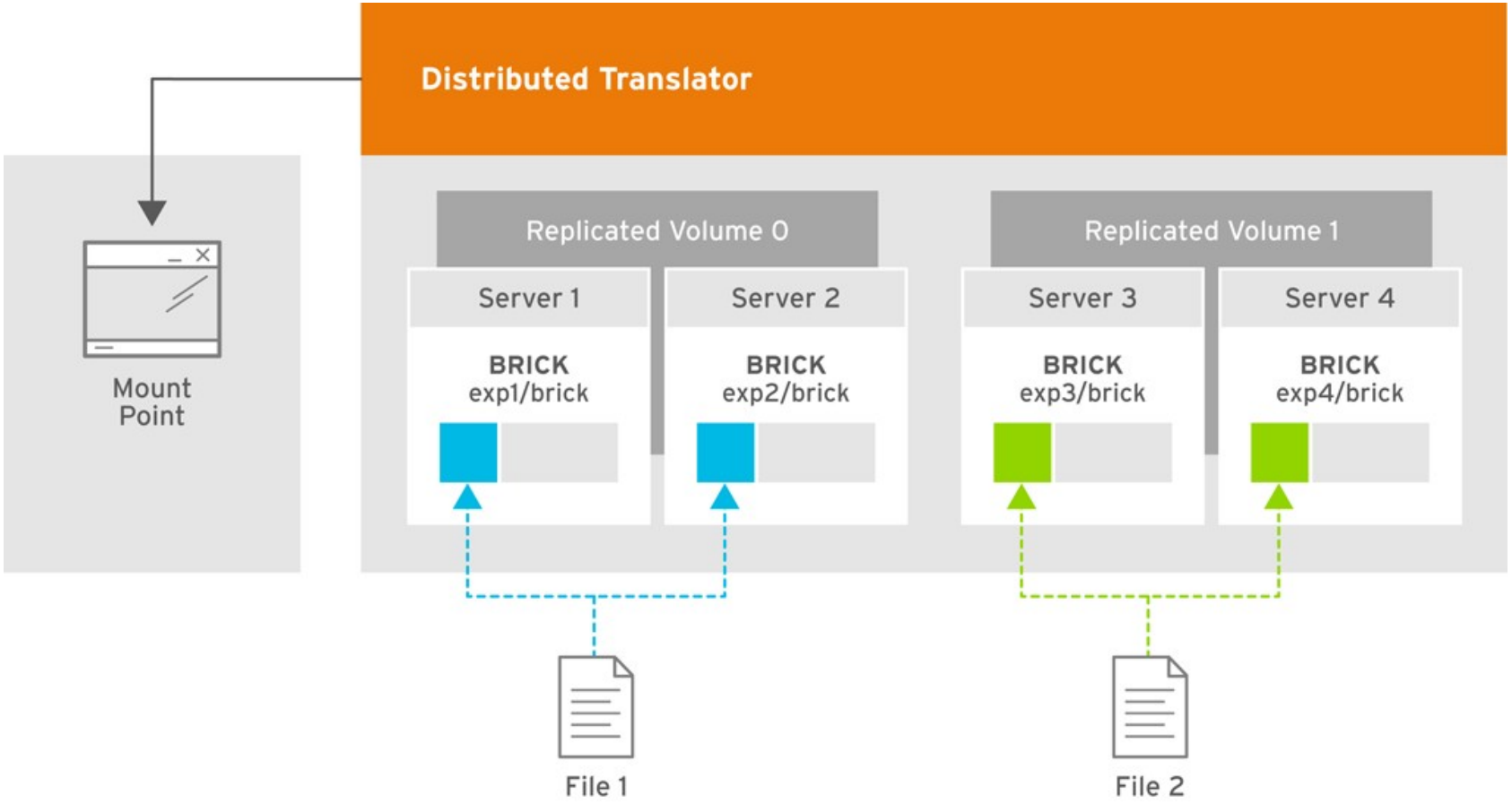
Replicated Volume

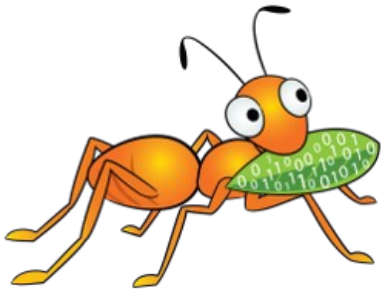
- Copies files to multiple bricks
- *Similar* to file-level RAID 1



Distributed Replicated Volume

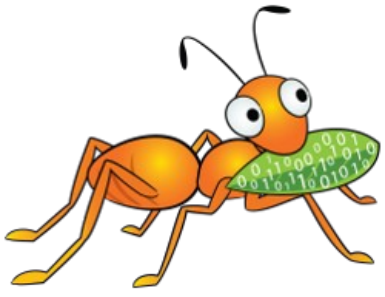
- Distributes files across replicated bricks





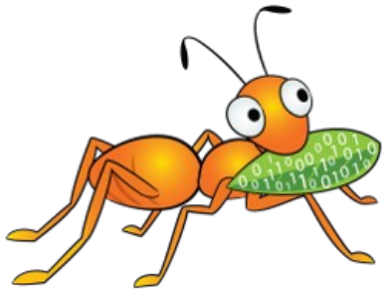
Data Access Overview

- GlusterFS Native Client
  - Filesystem in Userspace (FUSE)
- NFS
  - Built-in Service, NFS-Ganesha with libgfapi
- SMB/CIFS
  - Samba server required (libgfapi based module)
- Gluster For OpenStack (Swift-on-file)
  - Object-based access via Swift
- libgfapi flexible abstracted storage
  - Integrated with QEMU, Bareos and others



Quick Start

- Available in Fedora, Debian, NetBSD and others
- CentOS Storage SIG packages and add-ons
- Community packages in multiple versions for different distributions on http://download.gluster.org/
- Quick Start guides on http://gluster.org and CentOS wiki



Quick Start

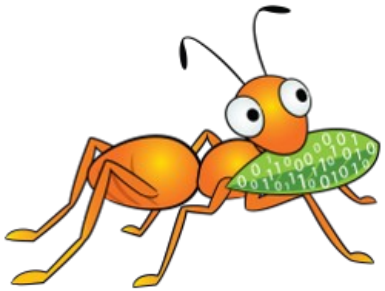
1. Install the packages (on all storage servers)
2. Start the GlusterD service (on all storage servers)
3. Peer probe other storage servers

4. Create and mount a filesystem to host a brick
5. Create a volume
6. Start the new volume

7. Mount the volume

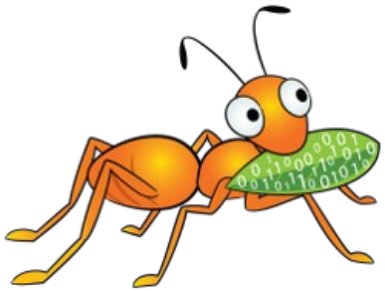
Current Stable Releases

- Maintenance of three minor releases
  - 3.7, 3.6 and 3.5
- Bugfixes only, non-intrusive features on high demand
- Approximate release schedule:
  - 3.5 at the 10th of each month
  - 3.6 at the 20th of each month
  - 3.7 at the 30th of each month
- Patches get backported to fix reported bugs



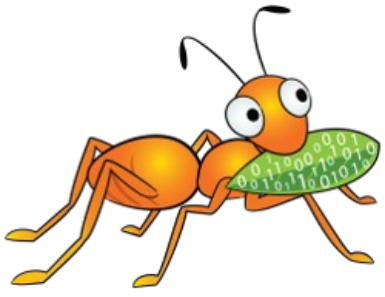
Features included in version 3.5

- File Snapshot for qcow2 files
- GFID access
- On-Wire (de)compression
- Quota Scalability
- Readdir ahead
- Zerofill
- Brick Failure Detection
- Parallel geo-replication



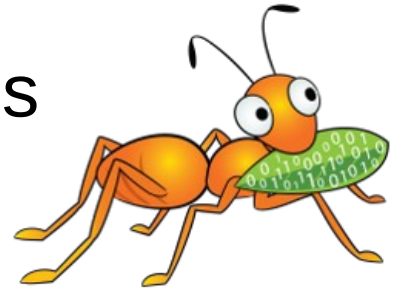
Features included in version 3.6

- Improved SSL support
- Heterogeneous bricks
- Volume wide locks for GlusterD
- Volume Snapshots
- User Serviceable Snapshots
- AFR refactor
- RDMA improvements
- Disperse translator for Erasure Coding



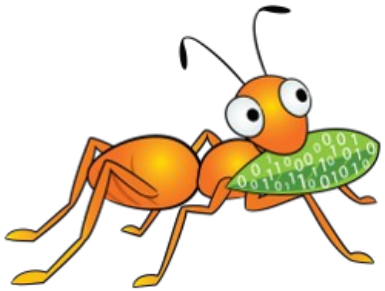
Features included in version 3.7

- Small-file performance enhancements
- Tiering for hot and cold contents
- Trash translator making undelete of files possible
- Netgroups and advanced exports configuration (NFS)
- BitRot detection
- Upcall infrastructure to notify client processes
- Support for NFS Ganesha clusters
- Arbiter volumes for 3-way replica, with only 2x the data
- Sharding to improve performance for VM images



BitRot support in 3.7

- Lazy checksum calculation after file close
- BitD daemon utilizes the changelog API
- Detection options for rotten data:
  - Upon open() and read() (disabled by default)
  - Periodic scan
- Detection only, manual repair needed



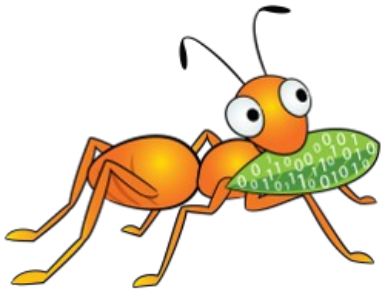
Sharding in 3.7

Split files into shards that get distributed by DHT

- Smaller shards help to
  - decrease time when healing is needed
  - make geo-replication faster

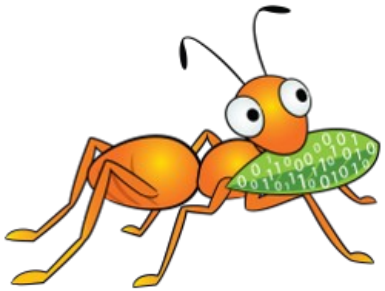
- More even distribution over bricks improve
  - utilization of space
  - client distribution, and performance

- Allows single files to be bigger than the bricks



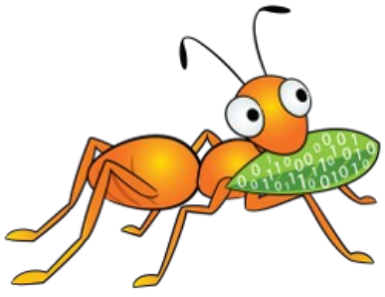
NFS-Ganesha support in 3.7

- Optionally replaces Gluster/NFS
- Supports NFSv4 with Kerberos
- Modifications to Gluster internals
  - Upcall infrastructure
  - Gluster CLI to manage NFS Genesha
  - libgfapi improvements
- High-Availability based on Pacemaker and Corosync



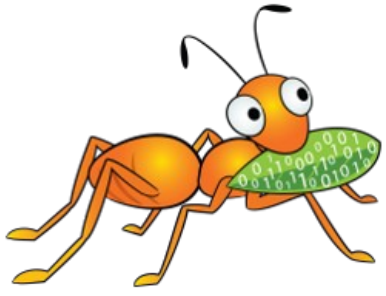
Plans for the next 3.8 release

- Scale out/in support with Tiering
- REST Management APIs for Gluster
- Manageable by Heketi
  - Easier integration in OpenStack, Kubernetes, …
- Subdirectory mounting for the FUSE client
- Converged High-Availability
  - Pacemaker managed NFS-Ganesha and Samba
- Quota for users/groups
- SEEK_DATA/SEEK_HOLE for sparse files



… more plans for the next 3.8 release

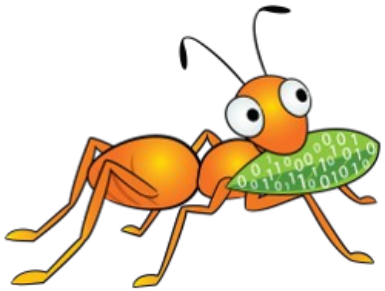
- Geo replication improvements
  - Tiering aware
  - Sharding support
- Multi-threaded self heal
- Throttling of clients doing excessive I/O
- inotify like functionality
- Kerberos for the Gluster protocols
- ... and much more



Preparations for multi-protocol support

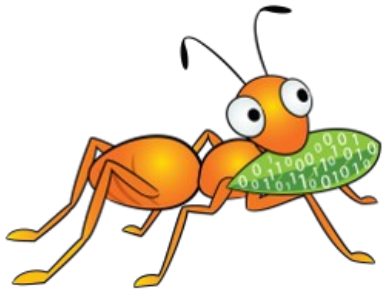
Simultaneous access of files through SMB, NFS and FUSE protocols:

- RichACL support
- Coherent client-side caching
  - Leases for SMB
  - Delegations for NFSv4
  - Layout recall for pNFS
- Mandatory lock support
- …



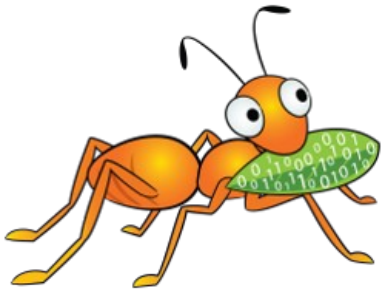
Plans for the next 4.0 release

- Scalability and manageability improvements
  - New Style Replication
  - Improved Distributed Hashing Translator
  - GlusterD 2.0 – aims to manage 1000 storage servers
- Composite operations in the GlusterFS RPC protocol
- Eventing framework for monitoring
- ... and much more



DHTv2 design

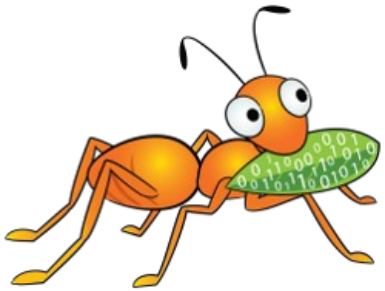
- Improve scalability, reduce performance impact
  - DHTv1 places all directories on all bricks
- Separate data and metadata in their own subvolumes
- Handle files and directories the same way
- Different on-disk layout, upgrades not possible/planned



New Style Replication

- Server-side replication
- Full data journal, can be placed on SSD
- More throughput for many workloads
- More precise, faster repair and healing
- Timebased journals provides the ability to implement snapshots of files

NSR would like a new name, suggestions welcome!



Resources

Mailing lists:

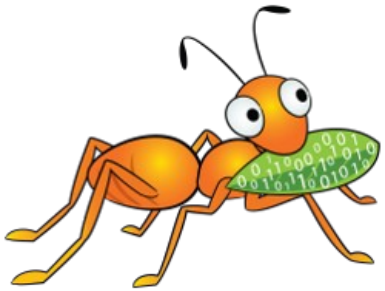
gluster-users@gluster.org
gluster-devel@gluster.org

IRC:

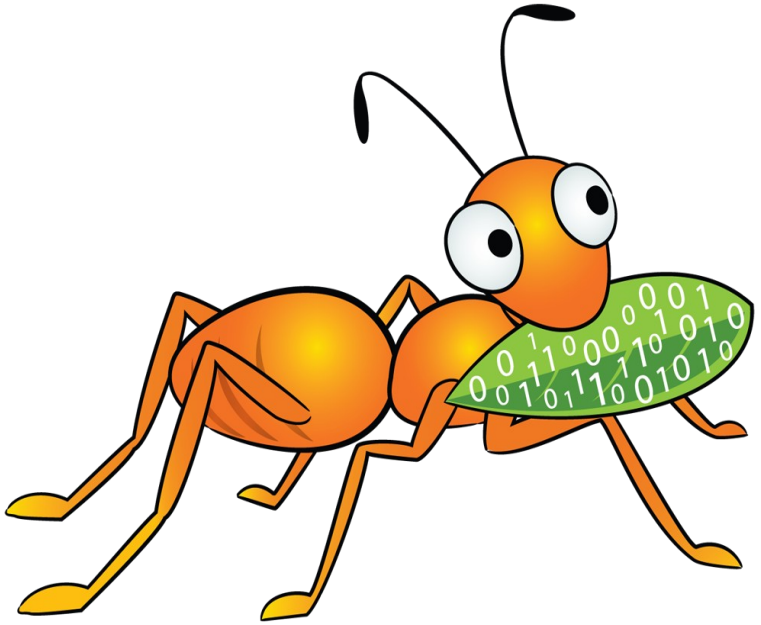
#gluster and #gluster-dev on Freenode

Links:

http://gluster.org/
http://gluster.readthedocs.org/
https://github.com/gluster/



Thank you!



Niels de Vos
ndevos@redhat.com
ndevos on IRC