Roberto Franchini
franchini@celi.it

Codemotion Milano
29/11/2014

CELI
LANGUAGE & INNOVATION

blogmeter

Cross Library

CELI

Adaptive Language Technology

Enabling

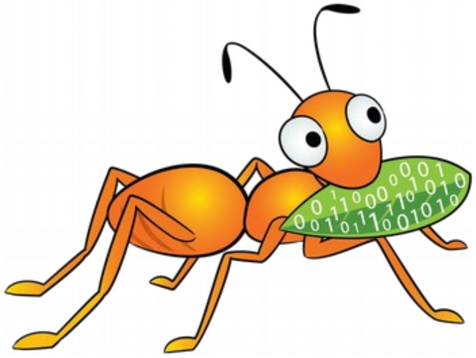
Speech Applications

Semantic Search

Text Analytics

Opinion Mining

Social Media Intelligence



GLUSTER

GlusterFS
A scalable distributed file system

whoami(1)

15 years of experience, proud to be a programmer

Writes software for information extraction, nlp, opinion mining (@scale), and a lot of other buzzwords

Implements scalable architectures

Plays with servers

Member of the JUG-Torino coordination team

franchini@celi.it

http://www.celi.it   http://www.blogmeter.it

github.com/robfrank github.com/uim-celi

twitter.com/robfrankie linkedin.com/in/robfrank

The problem

Identify a distributed and scalable file system

for today's and tomorrow's

Big Data

Once upon a time

2008: One nfs share

1,5TB ought to be enough for anybody

2010: Herd of shares

(1,5TB x N) ought to be enough for anybody

Nobody couldn’t stop the data flood

It was the time for something new

Requirements

Can be enlarged on demand

No dedicated HW

OS is preferred and trusted

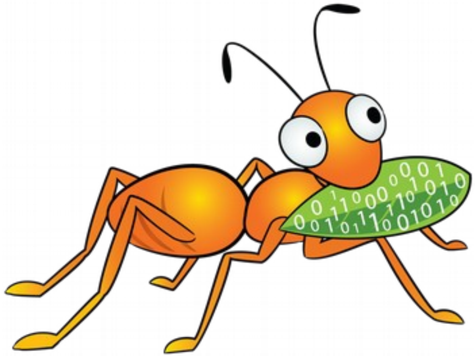
No specialized API

No specialized Kernel

POSIX compliance

Zilions of big and small files

No NAS or SAN (€€€€€)



GLUSTER

Clustered Scale-out **General Purpose** Storage Platform

- POSIX-y Distributed File System
- ...and so much more

Built on commodity systems

- x86_64 Linux ++
- POSIX filesystems underneath (XFS, EXT4)

No central metadata Server (NO SPOF)

Modular architecture for scale and functionality

Common use cases

Large Scale File Server

Media / Content Distribution Network (CDN)

Backup / Archive / Disaster Recovery (DR)

High Performance Computing (HPC)

Infrastructure as a Service (IaaS) storage layer

Database offload (blobs)

Unified Object Store + File Access

Features

ACL and Quota support

Fault-tolerance

Peer to peer

Self-healing

Fast setup up

Enlarge on demand

Shrink on demand

Snapshot

On premise phisical or virtual

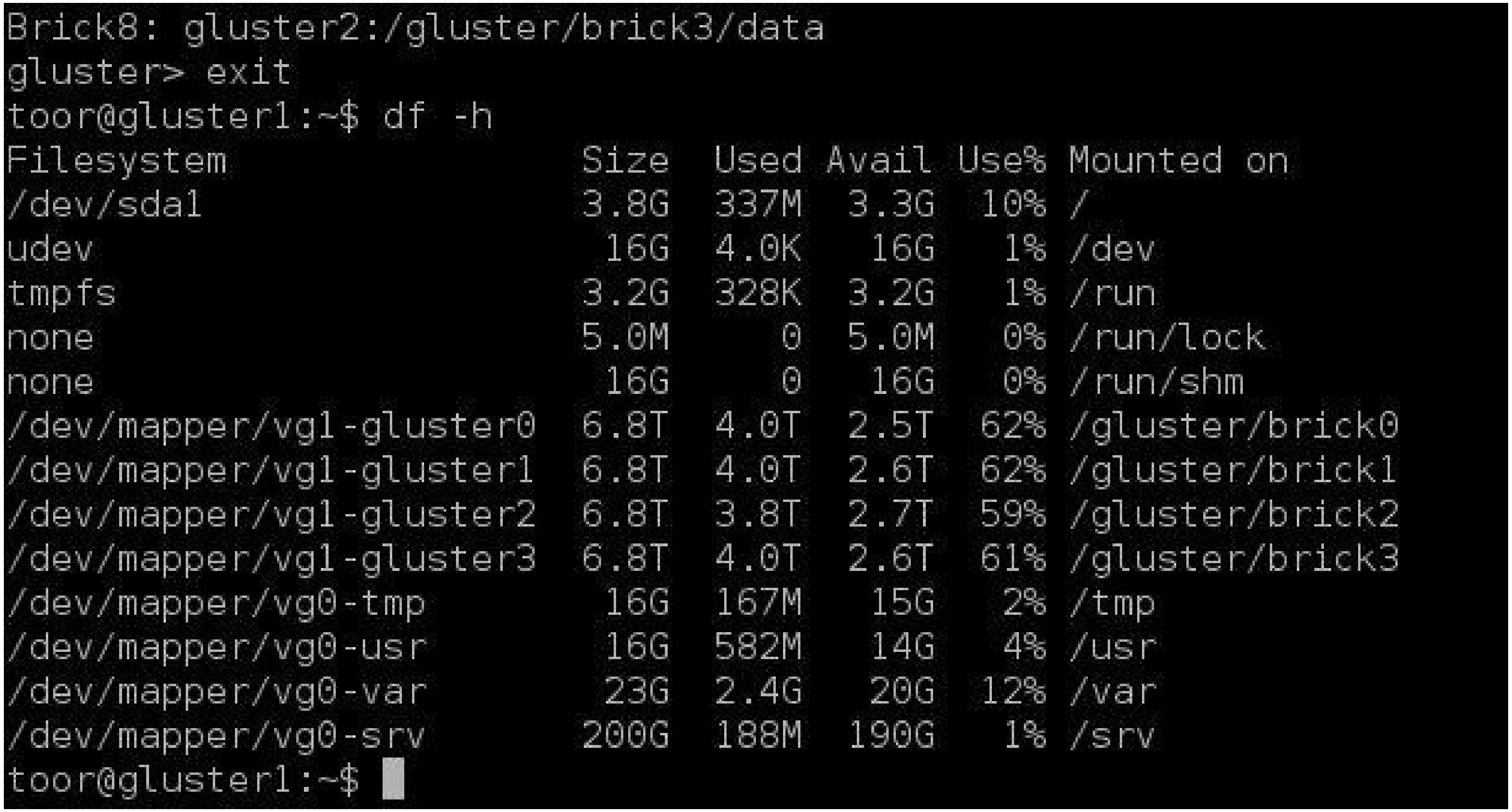
On cloud

Architecture

Architecture

- Peer / Node
  - cluster servers (glusterfs server)
  - Runs the gluster daemons and participates in volumes
- Brick
  - A filesystem mountpoint on servers
  - A unit of storage used as a ***capacity*** building block

Bricks on a node

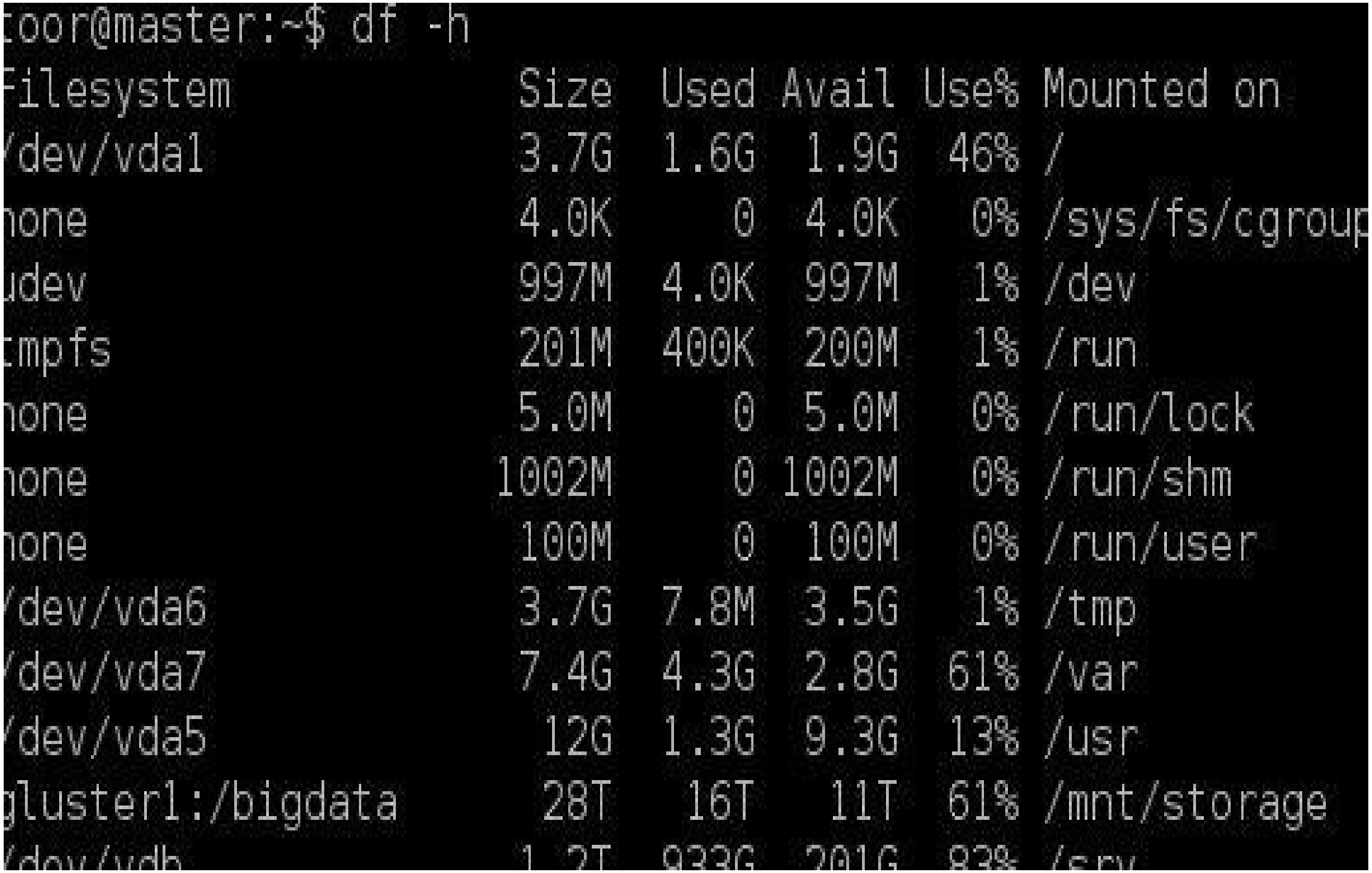


```
Brick8: gluster2:/gluster/brick3/data
gluster> exit
toor@gluster1:~$ df -h
Filesystem                 Size  Used Avail Use% Mounted on
/dev/sda1                  3.8G  337M  3.3G  10% /
udev                        16G  4.0K   16G   1% /dev
tmpfs                      3.2G  328K  3.2G   1% /run
none                       5.0M     0  5.0M   0% /run/lock
none                        16G     0   16G   0% /run/shm
/dev/mapper/vg1-gluster0   6.8T  4.0T  2.5T  62% /gluster/brick0
/dev/mapper/vg1-gluster1   6.8T  4.0T  2.6T  62% /gluster/brick1
/dev/mapper/vg1-gluster2   6.8T  3.8T  2.7T  59% /gluster/brick2
/dev/mapper/vg1-gluster3   6.8T  4.0T  2.6T  61% /gluster/brick3
/dev/mapper/vg0-tmp         16G  167M   15G   2% /tmp
/dev/mapper/vg0-usr         16G  582M   14G   4% /usr
/dev/mapper/vg0-var         23G  2.4G   20G  12% /var
/dev/mapper/vg0-srv        200G  188M  190G   1% /srv
toor@gluster1:~$ █
```

Architecture

- Translator
  - Logic between bricks or subvolume that generate a subvolume with certain characteristic
  - distribute, replica, stripe are special translators to generate simil-RAID configuration
  - perfomance translators
- Volume
  - Bricks combined and passed through translators
  - Ultimately, what's presented to the end user

Volume



```
toor@master:~$ df -h
Filesystem              Size  Used Avail Use% Mounted on
/dev/vda1               3.7G  1.6G  1.9G  46% /
none                    4.0K     0  4.0K   0% /sys/fs/cgroup
udev                    997M  4.0K  997M   1% /dev
tmpfs                   201M  400K  200M   1% /run
none                    5.0M     0  5.0M   0% /run/lock
none                   1002M     0 1002M   0% /run/shm
none                    100M     0  100M   0% /run/user
/dev/vda6               3.7G  7.8M  3.5G   1% /tmp
/dev/vda7               7.4G  4.3G  2.8G  61% /var
/dev/vda5                12G  1.3G  9.3G  13% /usr
gluster1:/bigdata        28T   16T   11T  61% /mnt/storage
/dev/vdb                1.2T  933G  201G  83% /srv
```

Volume types

Distributed

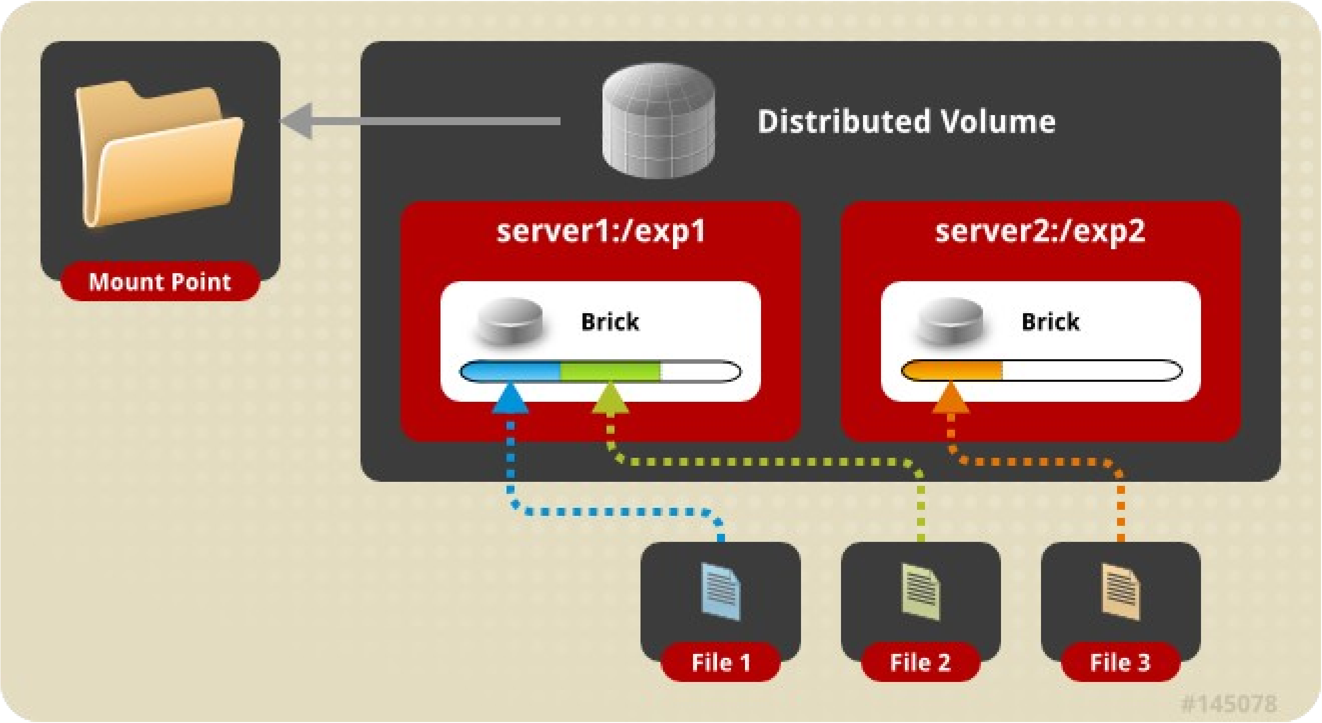
The default configuration

Files “evenly” spread across bricks

Similar to **file-level** RAID 0

Server/Disk failure could be catastrophic

Distributed



Replicated

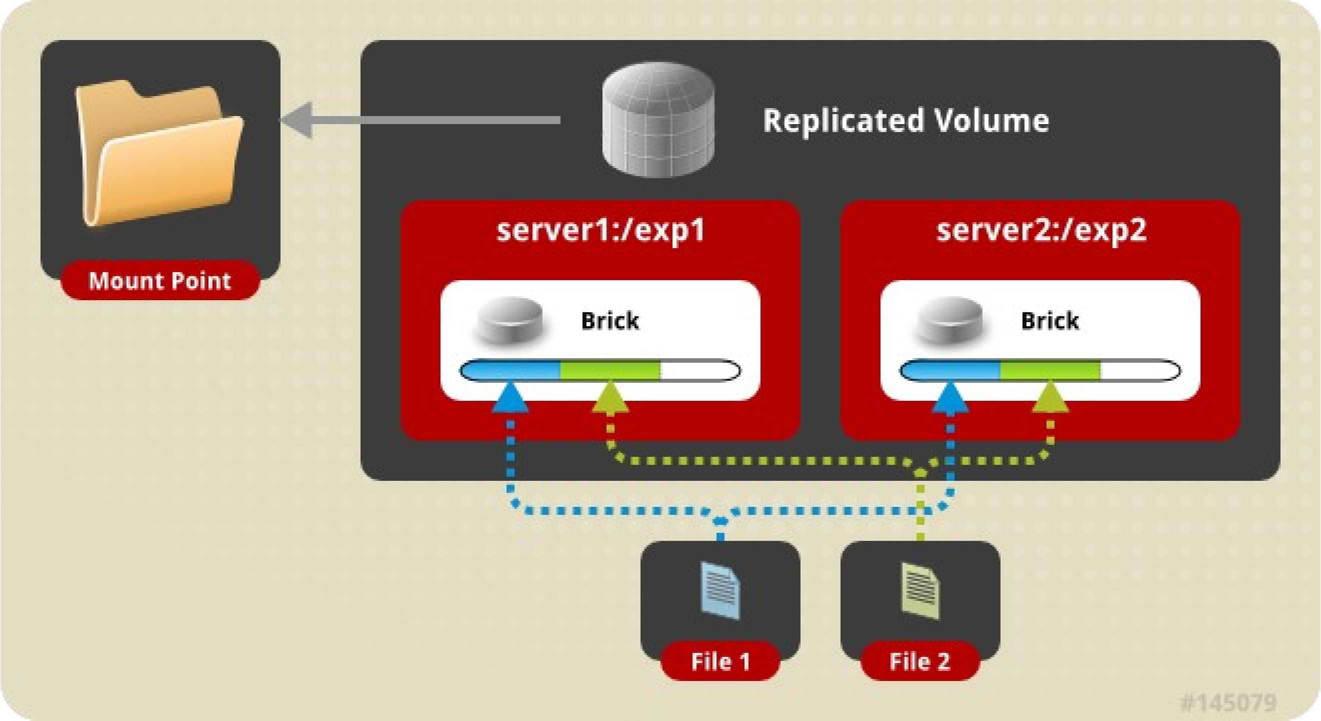
Files written synchronously to replica peers

Files read synchronously,

but ultimately serviced by the first responder

Similar to **file-level** RAID 1

Replicated


Replicated Volume
Mount Point
server1:/exp1
Brick
server2:/exp2
Brick
File 1
File 2
#145079

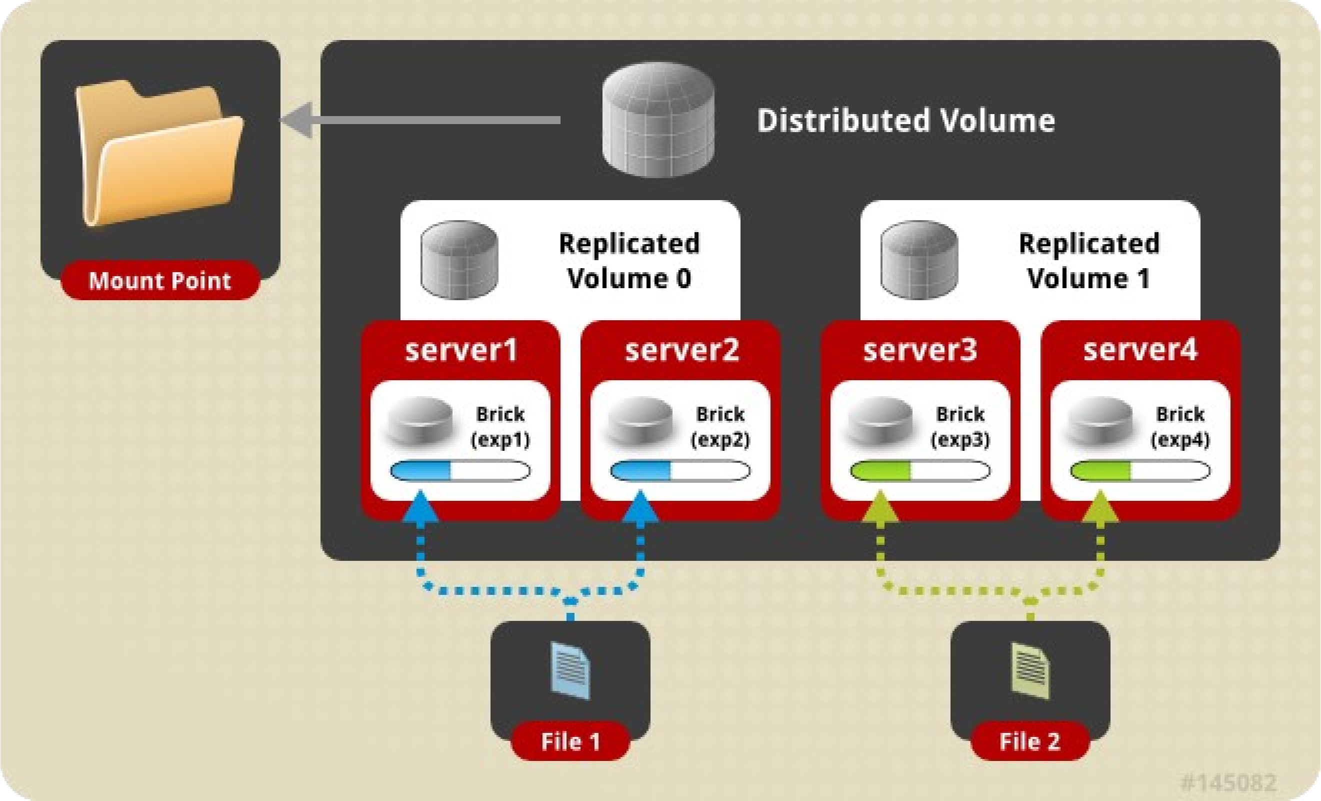
Distributed + replicated

Distribued + replicated

Similar to **file-level** RAID 10

Most used layout

Distributed replicated



Striped

Individual files split among bricks (sparse files)

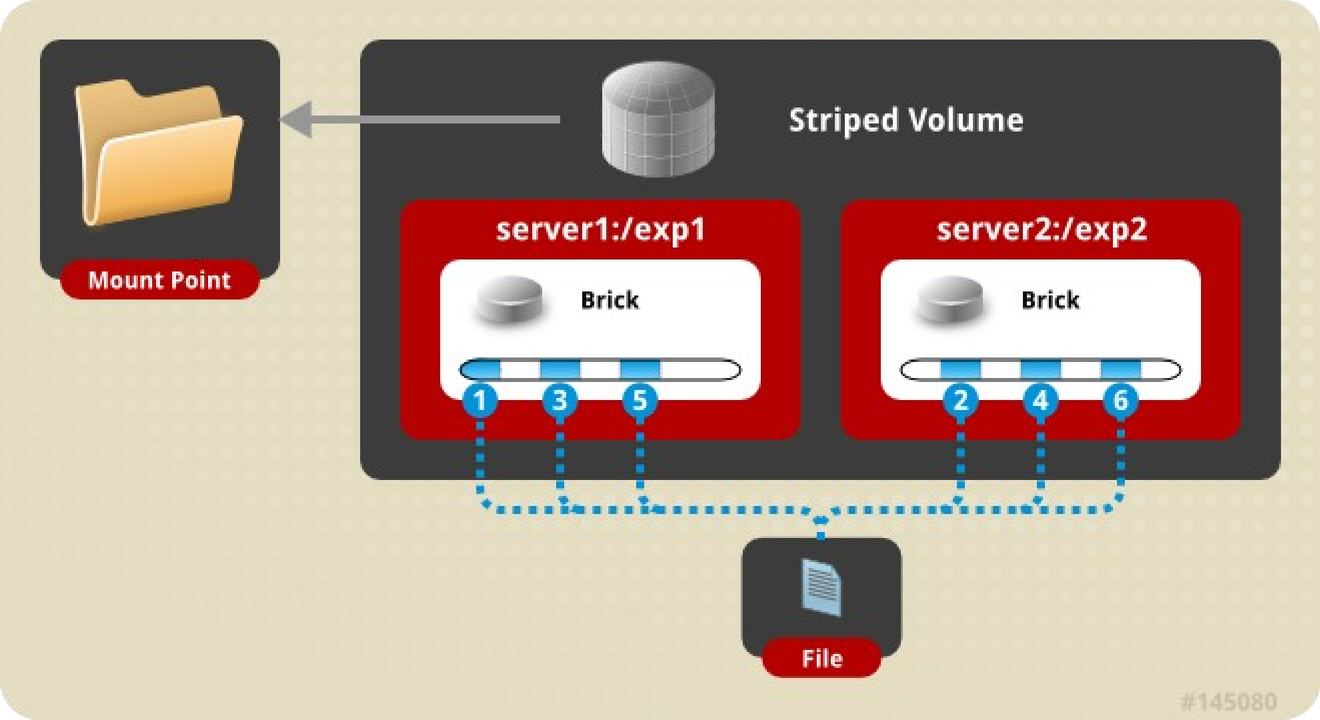
Similar to **block-level** RAID 0

Limited Use Cases

HPC Pre/Post Processing

File size exceeds brick size

Striped


Striped Volume
Mount Point
server1:/exp1
Brick
1
3
5
server2:/exp2
Brick
2
4
6
File
#145080

Moving parts

Components

- `glusterd`
  - Management daemon
  - One instance on each GlusterFS server
  - Interfaced through `gluster` CLI
- `glusterfsd`
  - GlusterFS brick daemon
  - One process for each brick on each server
  - Managed by `glusterd`

Components

- `glusterfs`
  - Volume service daemon
  - One process for each volume service
    - NFS server, FUSE client, Self-Heal, Quota, ...
- `mount.glusterfs`
  - FUSE native client mount extension
- `gluster`
  - Gluster Console Manager (CLI)

CELI
Adaptive Language Technology
Enabling
Speech Applications
Semantic Search
Text Analytics
Opinion Mining
Social Media Intelligence

Clients

Clients: native

FUSE kernel module allows the filesystem to be built and operated entirely in userspace

Specify mount to any GlusterFS server

Native Client fetches volfile from mount server, then communicates directly with **all nodes** to access data

Recommended for high concurrency and high write performance

Load is inherently balanced across distributed volumes

Clients:NFS

Standard NFS v3 clients

Standard automounter is supported

Mount to any server, or use a load balancer

GlusterFS NFS server includes Network Lock Manager (NLM) to synchronize locks across clients

Better performance for reading many small files from a single client

Load balancing must be managed externally

Clients: libgfapi

Introduced with GlusterFS 3.4

User-space library for accessing data in GlusterFS

Filesystem-like API

Runs in application process

no FUSE, no copies, no context switches

...but same volfiles, translators, etc.

Clients: SMB/CIFS

- In GlusterFS 3.4 – Samba + libgfapi
  - No need for local native client mount & re-export
  - Significant performance improvements with FUSE removed from the equation
- Must be setup on each server you wish to connect to via CIFS
- CTDB is required for Samba clustering

Clients: HDFS

Access data **within** and **outside** of Hadoop

No HDFS name node single point of failure / bottleneck

Seamless replacement for HDFS

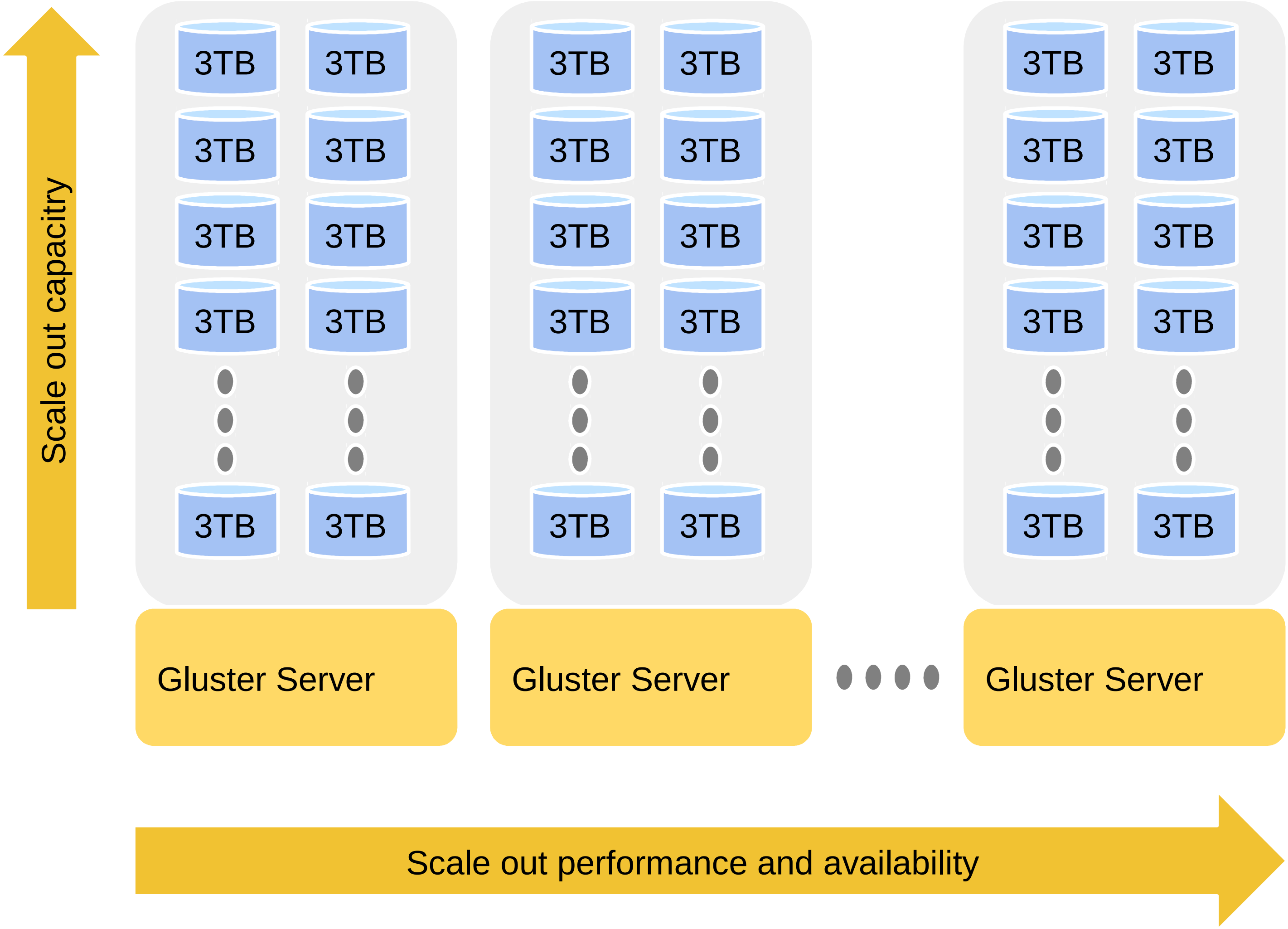
Scales with the massive growth of big data

Scalability

Under the hood

- Elastic Hash Algorithm
- No central metadata
  - No Performance Bottleneck
  - Eliminates risk scenarios
- Location hashed intelligently on filename
  - Unique identifiers (GFID), similar to md5sum

Scalability



Scalability

Add disks to servers to increase **storage size**

Add servers to increase **bandwidth** and **storage size**

Add servers to increase **availability** (replica factor)

What we do with glusterFS

What we do with GFS

Daily production of more than 10GB of Lucene inverted indexes stored on glusterFS

more than 200GB/month

Search stored indexes to extract different sets of documents for every customers

YES: we open indexes directly on storage

(it's POSIX!!!)

2010: first installation

Version 3.0.x

8 (not dedicated) servers

Distributed replicated

No bound on brick size (!!!!)

Ca 4TB avaliable

NOTE: stuck to 3.0.x until 2012 due to problems on 3.1 and 3.2 series, **then RH acquired gluster** (RH Storage)

2012: (little) cluster

New installation, version 3.3.2

4TB available on 8 servers (DELL c5000)

still not dedicated

1 brick per server **limited** to 1TB

2TB-raid 1 on each server

Still in production

2012: enlarge

New installation, upgrade to 3.3.x

6TB available on 12 servers (still not dedicated)

Enlarged to 9TB on 18 servers

Bricks size bounded **AND** unbounded

2013: fail

18 not dedicated servers: too much

18 bricks of different sizes

2 big down due to bricks out of space

Didn’t restart after a move

but…

All data were recovered

(files are scattered on bricks, read from them!)

2014: consolidate

2 **dedicated** servers

12 x 3TB SAS raid6

4 bricks per server

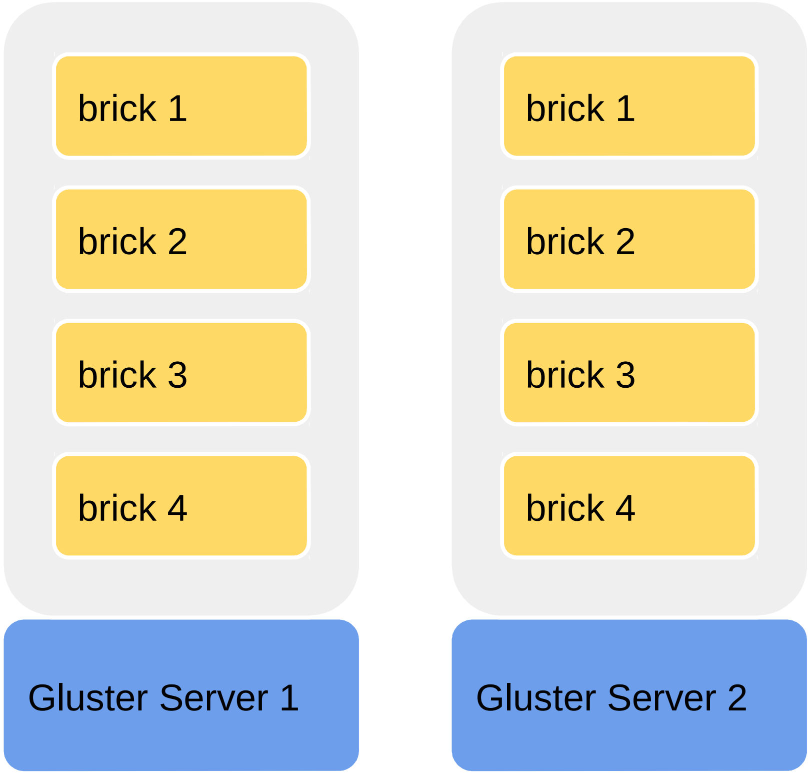
28 TB available

distributed replicated

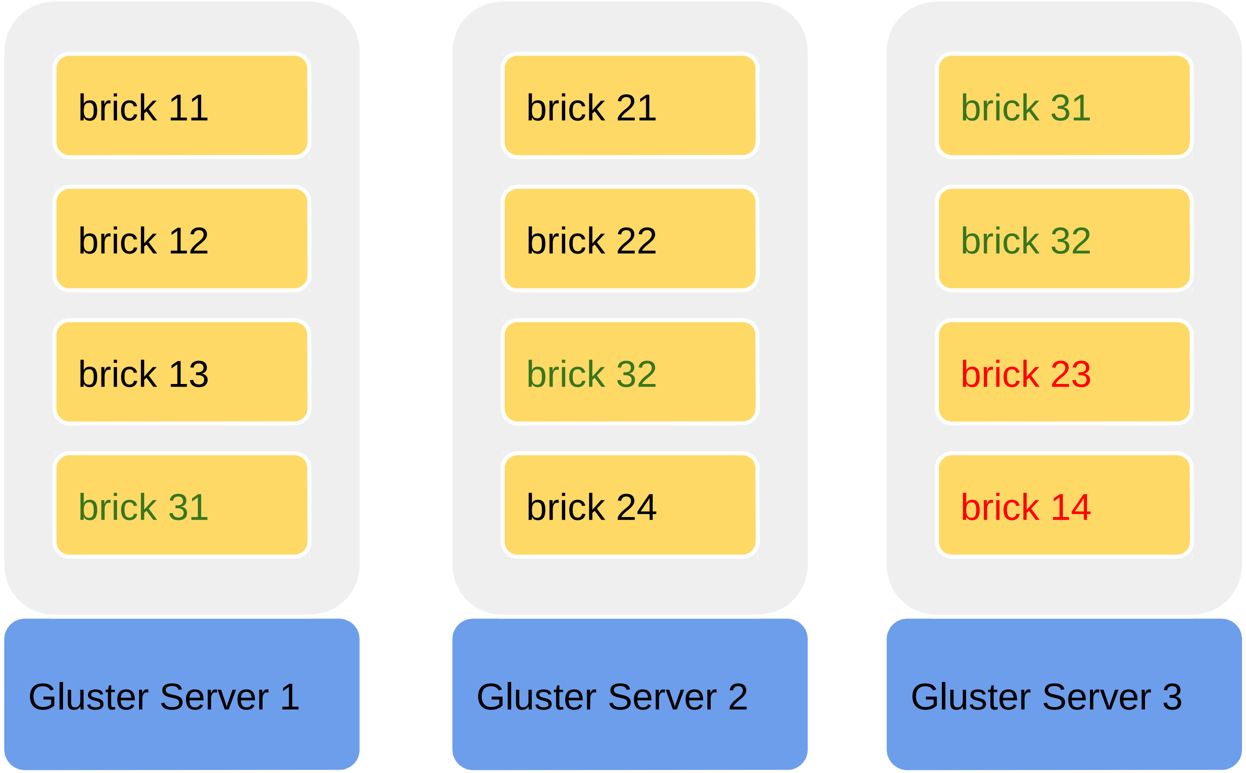
4x1Gb bonded NIC

ca **40** clients (FUSE) (other servers)

Consolidate



Scale up



brick 11
brick 12
brick 13
brick 31
Gluster Server 1

brick 21
brick 22
brick 32
brick 24
Gluster Server 2

brick 31
brick 32
brick 23
brick 14
Gluster Server 3

Do

Dedicated server (phisical or virtual)

RAID 6 or RAID 10 (with small files)

Multiple bricks of same size

Plan to scale

Do not

Multi purpose server

Bricks of different size

Very small files

Write to bricks

Some raw tests

read

Total transferred file size: 23.10G bytes

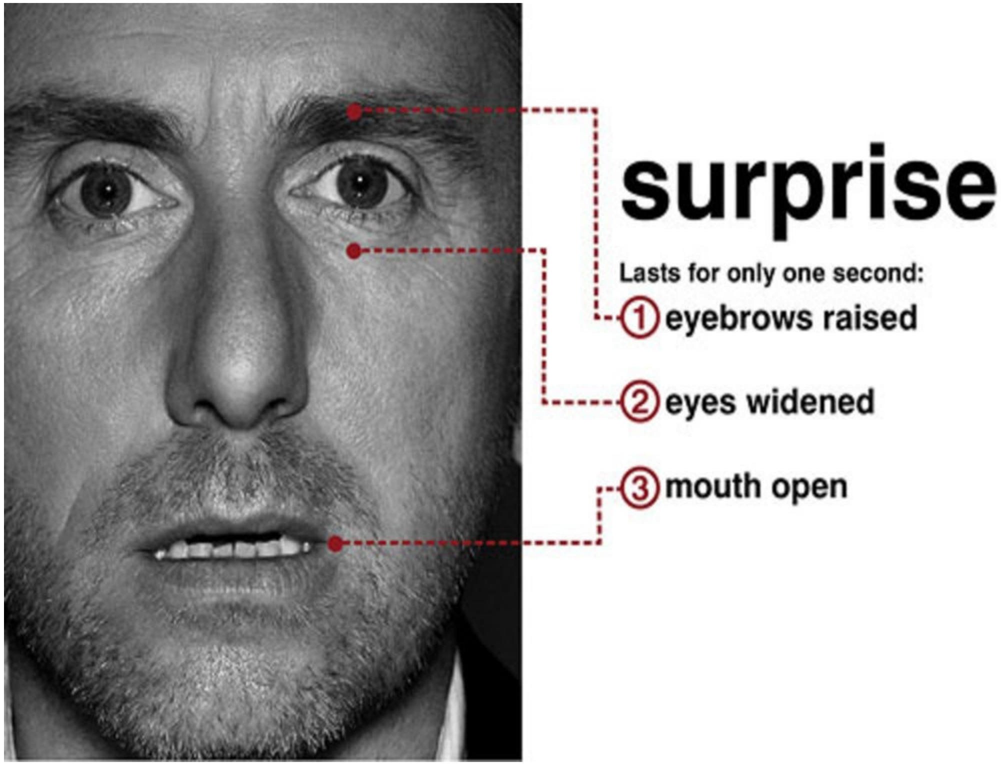
43.46M bytes/sec

write

Total transferred file size: 23.10G bytes

38.53M bytes/sec

Raw tests



NOTE: ran in production under heavy load, no clean test room

Resources

http://www.gluster.org/

https://access.redhat.com/documentation/en-US/Red_Hat_Storage/

https://github.com/gluster

http://www.redhat.com/products/storage-server/

http://joejulian.name/blog/category/glusterfs/

http://jread.us/2013/06/one-petabyte-red-hat-storage-and-glusterfs-project-overview/

CELI
Adaptive Language Technology
Enabling
Speech Applications
Semantic Search
Text Analytics
Opinion Mining
Social Media Intelligence

Thank you!

Natural Language Processing

www.celi.it/en - info@celi.it

Enabling

Speech Applications

Semantic Search

Social Media Intelligence

Text Analytics

Opinion Mining

CELI

LANGUAGE & INNOVATION

Roberto Franchini
franchini@celi.it

Language and
Information Technology

Via San Quintino 31 - 10121 Torino
Tel. +39 011.562.71.15
Fax +39 011.506.40.86
info@celi.it - www.celi.it