

# GlusterFS

Подкопаев Антон, podkoav239@gmail.com  
Алексеев Антон, anton.m.alexeyev@gmail.com

Computer Science Center

14 марта 2013

- Доступ с многих хостов
- Инкапсуляция расположения файлов
- Реплики и отказоустойчивость
- Параллельный доступ
- Масштабируемость

# Распределенные файловые системы. Основные компоненты

- Клиент
- Сервер данных
- Сервер метаданных

# Распределенные файловые системы. Основные компоненты

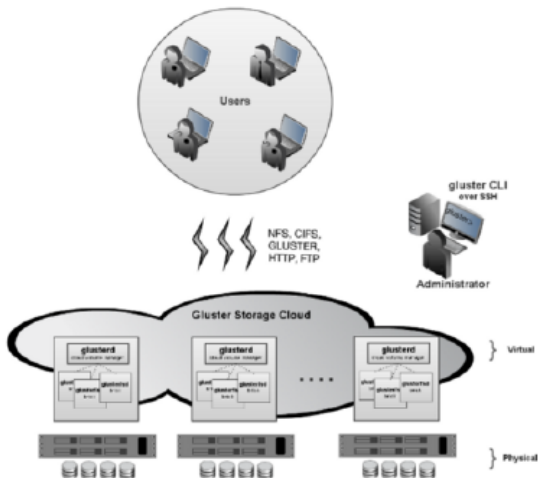
- Клиент
- Сервер данных
- Сервер метаданных

GlusterFS:

Сервер данных в том числе выполняет и функции сервера метаданных

# GlusterFS

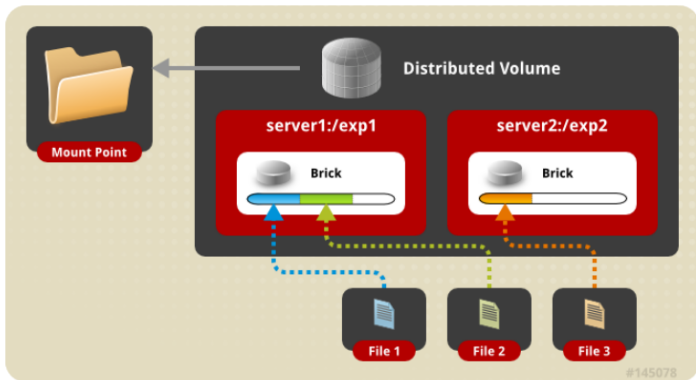
- Сетевая файловая система, работающая в user space
- До нескольких петабайт данных в одной точке монтирования



- Brick
- Логический диск
- Доверенные хранилища

- Распределенные
  - Реплицируемые
  - Разделяющие
- 
- Распределенные разделяющие
  - Распределенные реплицируемые
  - Разделяющие реплицируемые

# Распределенные логические диски (1)





# Распределенные логические диски (2)

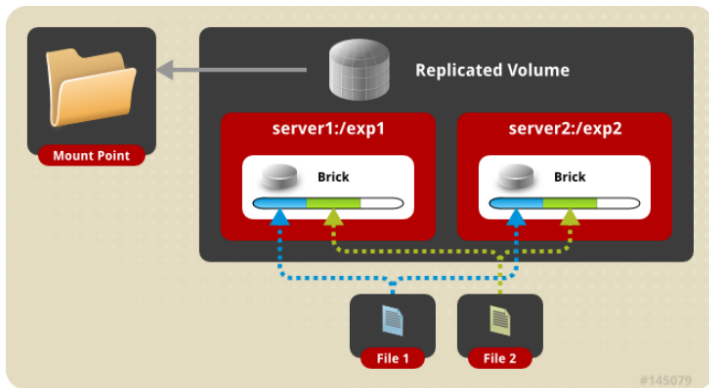
## Плюсы

- Больше серверов => выше производительность при параллельном доступе
- Увеличение диска = добавление сервера (можно во время работы)

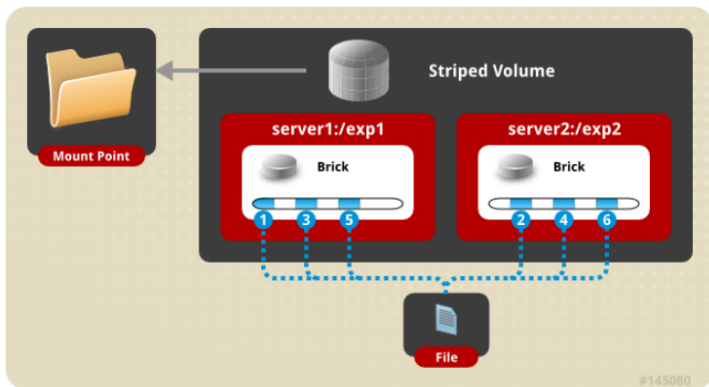
## Минусы

- Потеря сервера = потеря данных на нем
- Файл не может быть больше размера узла
- Смена имени файла => дополнительное время на lookup

# Реплицируемые логические диски



# Разделяющие логические диски



# Запуск GlusterFS (1)

```
bashsc2
```

# Запуск GlusterFS (1)

```
bashsc2
```

```
bashsc1
```

# Запуск GlusterFS (1)

```
bashsc2
```

```
bashsc1
```

```
bashsc3
```

# Запуск GlusterFS (2)

## Распределенные

```
$gluster volume create test-volume server1:/exp1 server2:/exp2
```

## Реплицируемые

```
$gluster volume create test-volume replica 2 server1:/exp1 server2:/exp2
```

## Разделяющие

```
$gluster volume create test-volume stripe 2 server1:/exp1 server2:/exp2
```

# Добавление серверов

```
$gluster peer probe new_server
```



# Добавление серверов

```
$gluster peer probe new_server  
$gluster volume add-brick vol_name new_brick
```

# Добавление серверов

```
$gluster peer probe new_server  
$gluster volume add-brick vol_name new_brick  
$gluster volume info
```

# Добавление серверов

```
$gluster peer probe new_server  
$gluster volume add-brick vol_name new_brick  
$gluster volume info
```

Для расширения реплицируемых (разделяющих) дисков надо добавлять количество серверов, кратное фактору реплики (разбиения)

# Удаление серверов

```
$gluster volume remove-brick vol__name new__brick start
```

```
$gluster volume remove-brick vol_name new_brick start  
$gluster volume remove-brick vol_name new_brick status
```

# Удаление серверов

```
$gluster volume remove-brick vol_name new_brick start  
$gluster volume remove-brick vol_name new_brick status  
$gluster volume remove-brick vol_name new_brick commit
```

Все настройки осуществляются так

```
$gluster volume set vol_name option param
```

Пример

```
$gluster volume set myvolume performance.cache-size 256MB
```

<code>auth.allow</code>	IP-адреса клиентов, с джокерами: 192.168.1.
<code>auth.reject</code>	IP-адреса заблокированных клиентов
<code>client.grace-timeout</code>	время жизни замков, поставленных клиентом, после разрыва соединения, в секундах (10–1800)
<code>cluster.self-heal-window-size</code>	максимальное число блоков в файле, для которых self-heal происходит одновременно (0-1025)



`cluster.data-self-heal-algorithm`

тип self-heal: full (копирование всего файла), diff (копирование только несинхронизированных блоков), reset (если файла нет — полное копирование, если размер файла близок к размеру страницы диска — чтение несколькими операциями)

<code>cluster.min-free-disk</code>	сколько процентов диска должны оставаться незанятыми
<code>cluster.stripe-block-size</code>	размер страйпа (в байтах)
<code>cluster.self-heal-daemon</code>	включение / выключение фонового self-heal на репликах (on/off)
<code>diagnostics.brick-log-level</code>	уровень логов bricks: INFO, DEBUG, WARNING, ERROR, CRITICAL, NONE, TRACE
<code>diagnostics.client-log-level</code>	уровень логов клиентов
<code>diagnostic.dump-fd-stats</code>	статистика по операциям над файлами (on/off)

`feature.read-only`

монтирование диска в режиме read-only для всех клиентов (on/off)

`features.lock-heal`

self-healing замков при разрыве соединения (on/off)

`features.quota-timeout`

объём памяти, для каждой директории кэшируется; можно указать макс. допустимое время пребывания любой директории в кэше (0-3600 сек.)

`geo-replication.indexing`

автоматическая синхронизация изменений от master к slave

`network.frame-timeout`

время, через которое операция объявляется «мёртвой», если сервер не отвечает

`network.ping-timeout`

время ожидания клиентом ответа от сервера (default: 42 сек.); после разрыва все данные, связанные с клиентом, уничтожаются; при восстановлении соединения — восстановление всех ресурсов и замков. Очень, очень дорого. Не надо так делать.

<code>nfs.volume-access</code>	тип доступа для sub-volume (read-write/read-only)
<code>nfs.trusted-write</code>	запрещать клиенту коммитить, если произошёл UNSTABLE write; STABLE writes — синхронизируются (on/off)
<code>nfs.trusted-sync</code>	все записи и запросы на коммит считаются асинхронными (на момент ответа не гарантируется, что запись произведена на все диски)
<code>nfs.rpc-auth-unix</code>	включение аутентификации AUTH_UNIX (default: On)
<code>nfs.rpc-auth-null</code>	включение аутентификации AUTH_NULL (default: On)

nfs.rpc-auth-allow

IP-адреса и hostnames, которым разрешается устанавливать соединение с сервером те, кому нельзя

nfs.rpc-auth-reject

nfs.ports-insecure

разрешать соединения непривилегированным хостам (default: off)

nfs.port

для связывания Gluster NFS с недефолтным номером порта (38465–38467)

nfs.disable

запретить экспорт диска NFS (default: off)

<code>performance.io-thread-count</code>	число потоков транслятора управляющего IO
<code>performance.cache-max-file-size</code>	max размер кэша транслятора (до $2^{64} - 1$ байт)
<code>performance.cache-min-file-size</code>	min размер кэша транслятора (от 0 байт)
<code>performance.cache-refresh-timeout</code>	интервал, с которым кэш будет обновляться (0-61 сек.)
<code>performance.cache-size</code>	размер кэша чтения (default: 32 MB)

<code>server.allow-insecure</code>	разрешать соединения непривилегированным хостам (default: on)
<code>server.grace-timeout</code>	время жизни замка после разрыва связи (10–1800 сек.)
<code>server.statedump-path</code>	путь к дампу состояния системы (default: /tmp brick'a)



- <http://www.gluster.org/>
- Взрослеем с GlusterFS <http://habrahabr.ru/post/140031/>
- GlusterFS, опыт новой версии <http://habrahabr.ru/post/157029/>