Архитектура операционной системы

Обзор некоторых файловых систем

BTRFS

Btrfs	
Разработчик	Oracle, Fujitsu, Red Hat[1]
Файловая система	Btrfs
Дата представления	4.18: август 2018 года[2] (Linux)
Структура	
Содержимое папок	B-tree
Размещение файлов	экстент
Ограничения	
Максимальный размер файла	16 ЭиБ
Максимальная длина имени файла	255 байт[3]
Максимальный размер тома	16 ЭиБ
Допустимые символы в названиях	Все байты кроме NUL и '/'
Возможности	
Атрибуты	POSIX
Права доступа	POSIX, ACL
Фоновая компрессия	Да (LZO, zlib, начиная с ядра 4.14: — zstd)
Фоновое шифрование	нет
Поддерживается ОС	Linux

Основные возможности

- Multi-volumes
- Copy-on-Write Style Update
- Data/Metadata Checksum
- Subvolume
- Snapshot
- Transparent Compression
- Поддержка SSD (TRIM)

Понятия

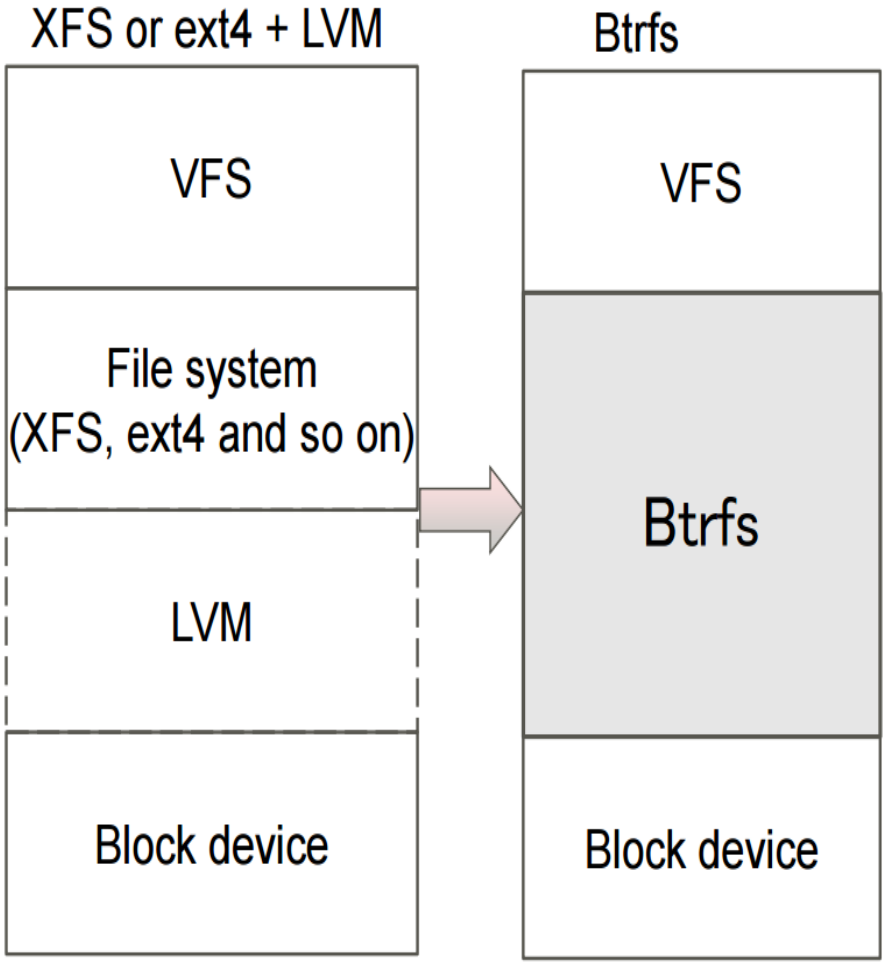
- Subvolume (том) – файловая система внутри файловой системы
  - монтируемая
  - с отдельными квотами
- Snapshot – копия тома (возможно RO)
  - btrfs subvolume snapshot [-r] ./sub ./snap

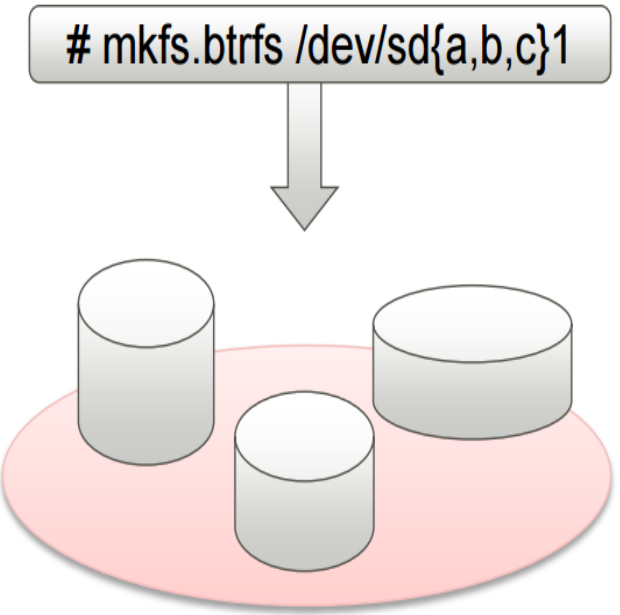
История и разработчики

- 2007 – начало разработки
- 2011 – поддержка компрессии
- 2013 – поддержка RAID
- 2014+ стабилизация, производительность
- Разработчики:
  - Fujitsu
  - Fusion-IO
  - Intel
  - Oracle
  - RedHat
  - SUSE

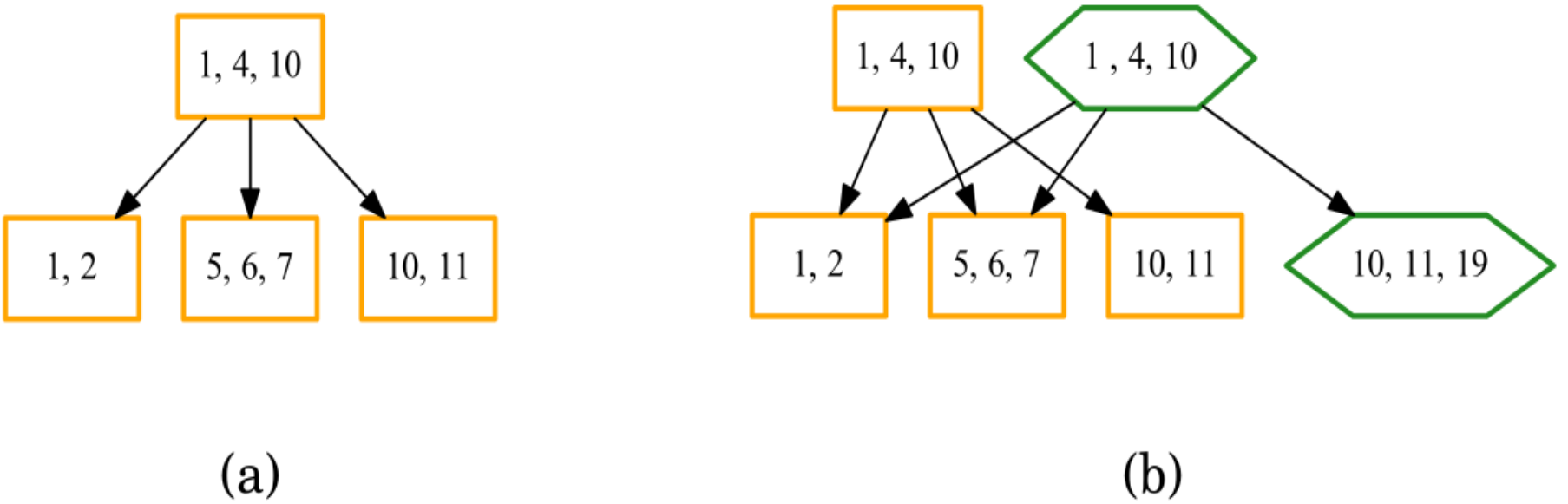
Архитектура

- Page block
- Extent
- COW

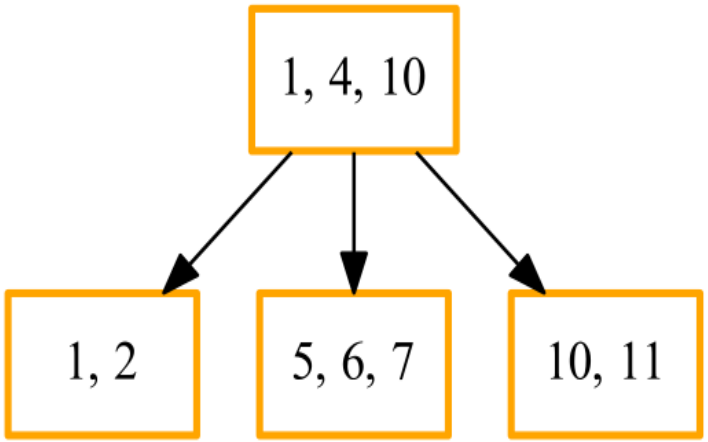


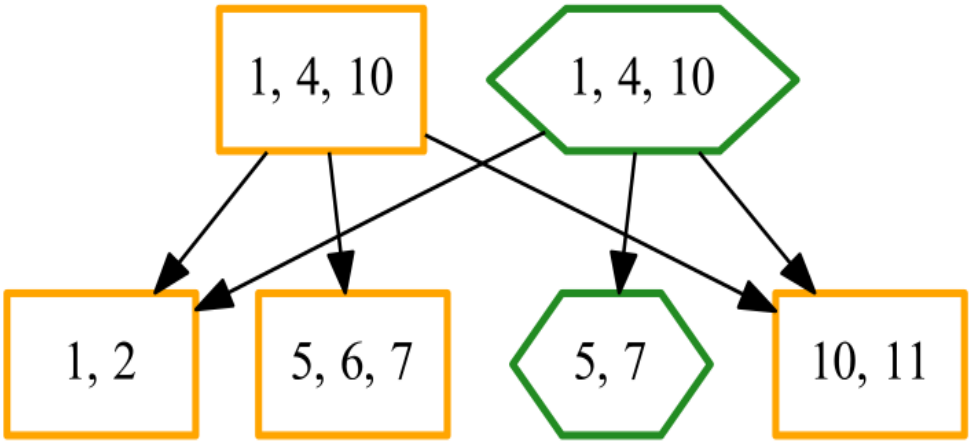


Вставка (19)

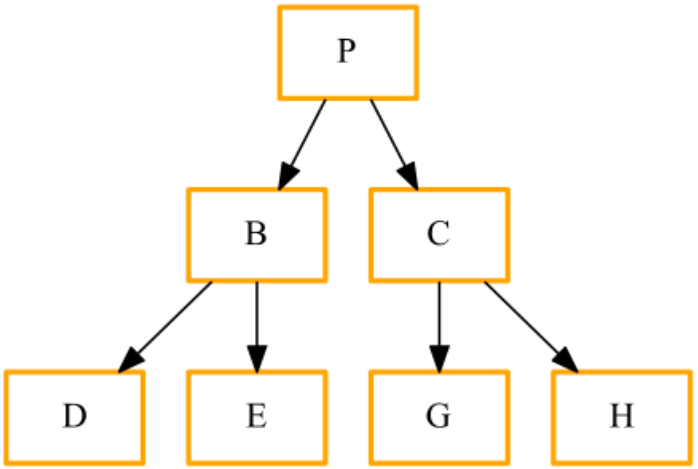


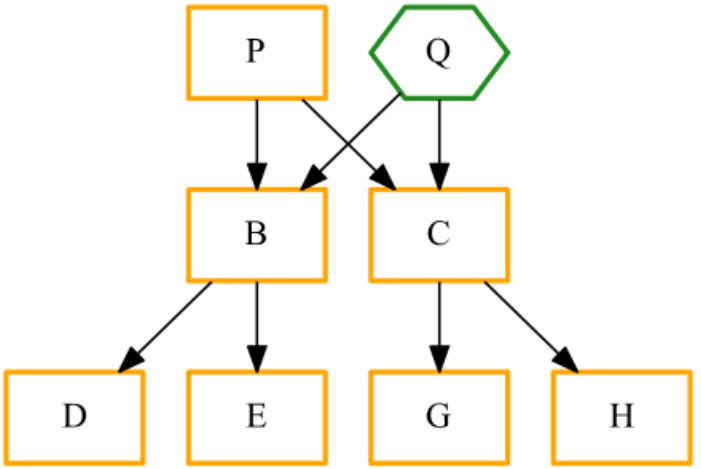
Удаление (6)



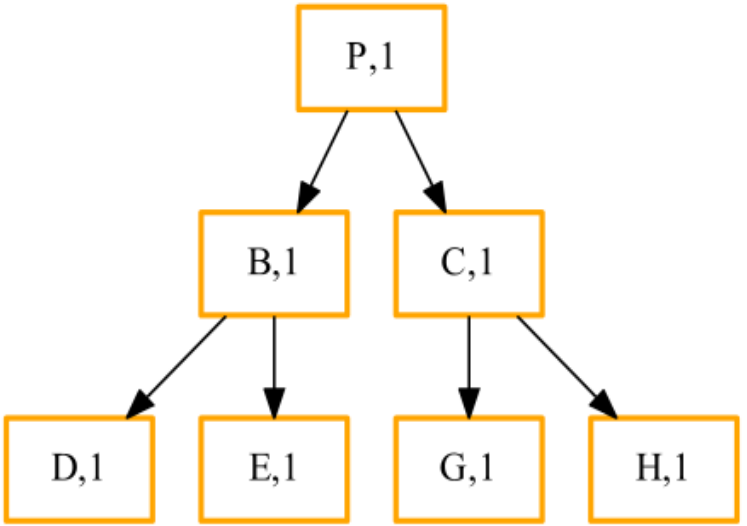


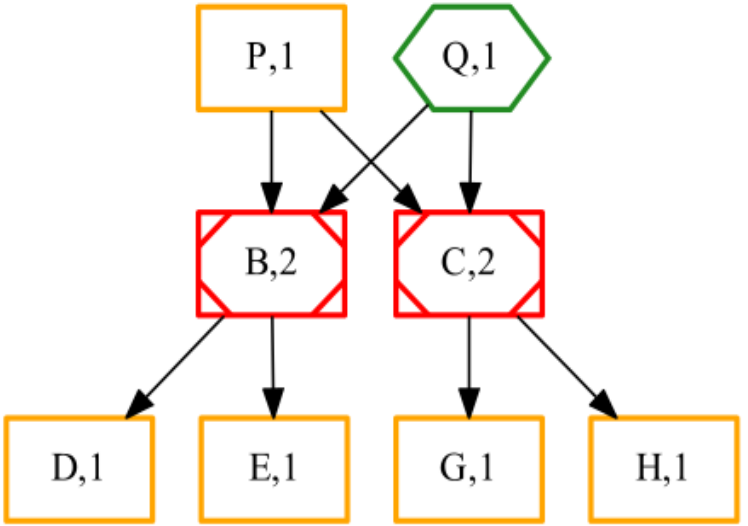
Клонирование (P->Q)



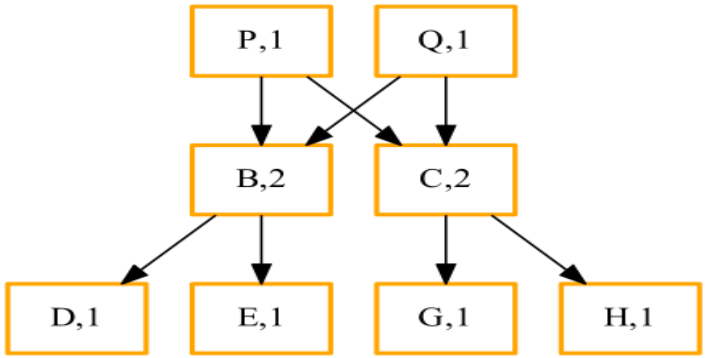


Подсчет ссылок (клонирование)

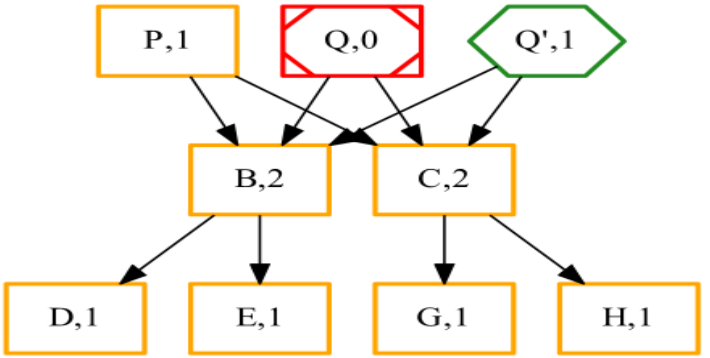




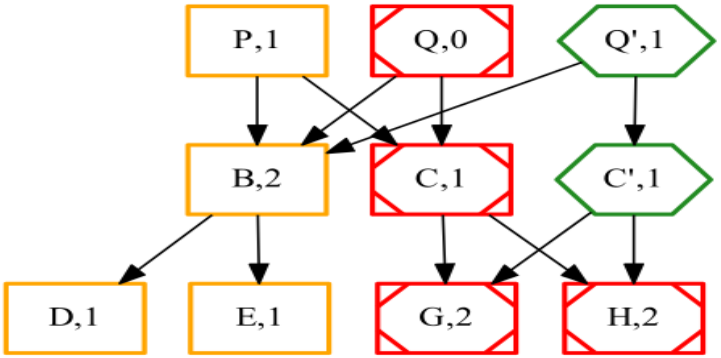
Вставка ключа (H=>Q)



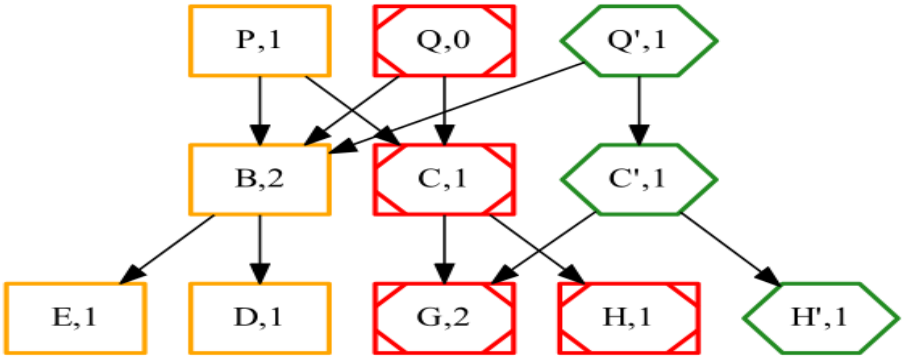
Исходные деревья P,Q



Затенение Q

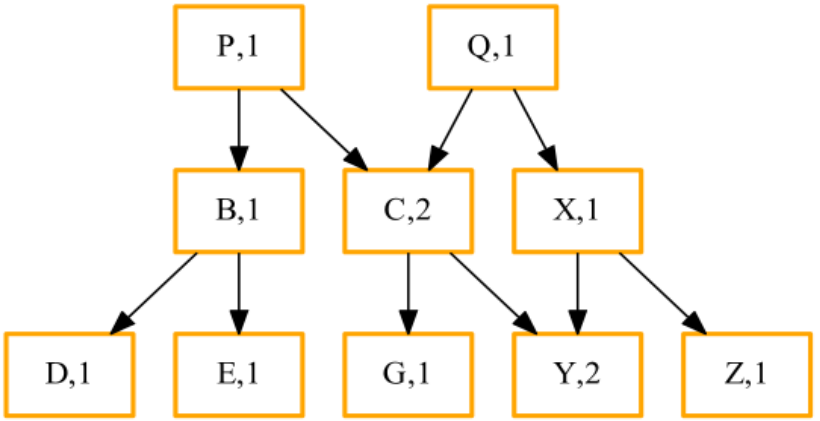


Затенение C

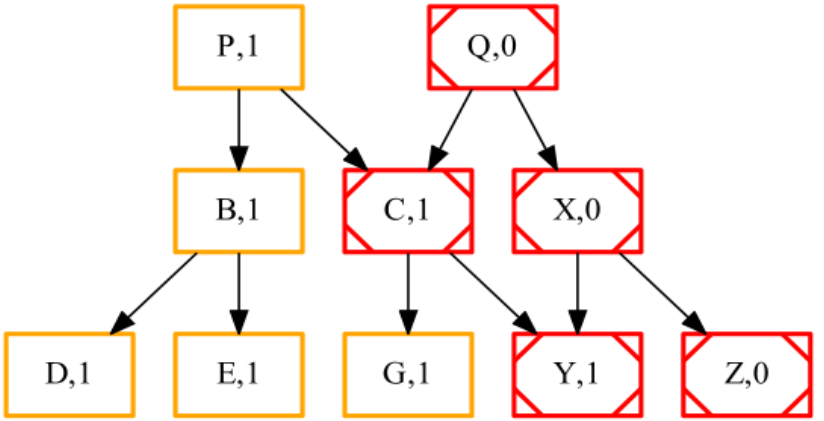


Затенение H

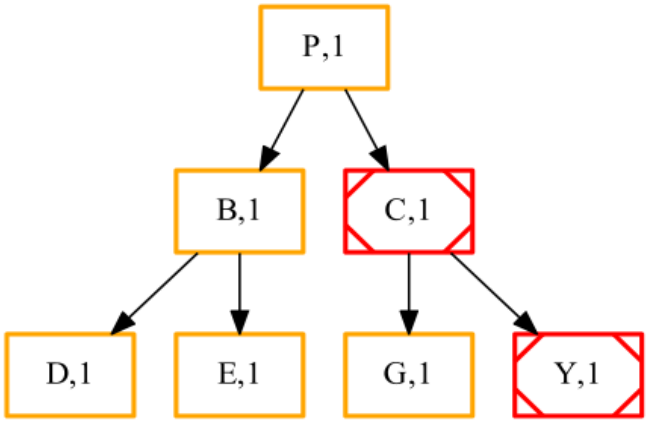
Удаление



Исходные деревья P, Q

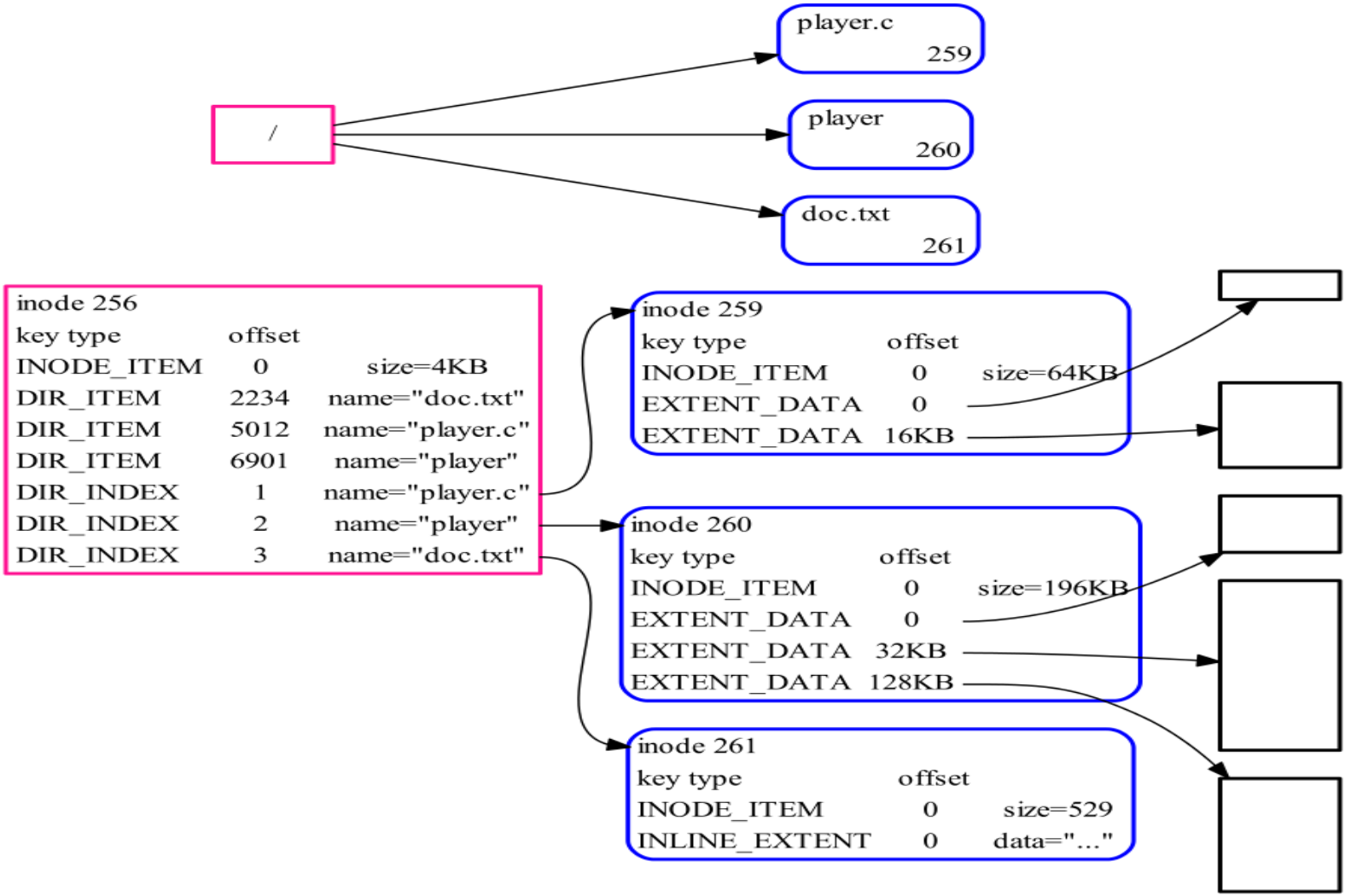


Удаление Q

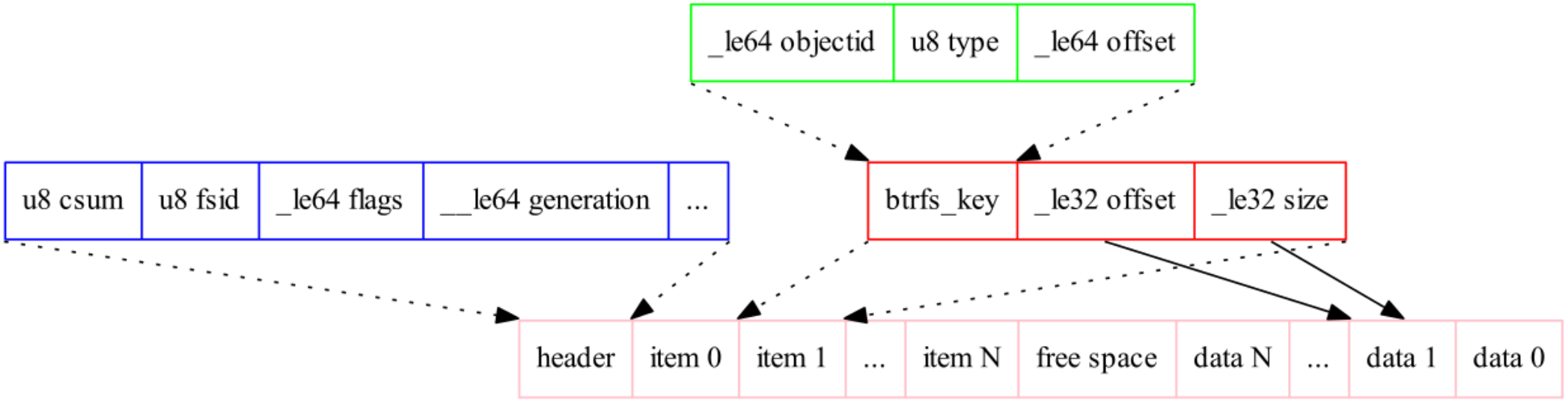


Сборка мусора

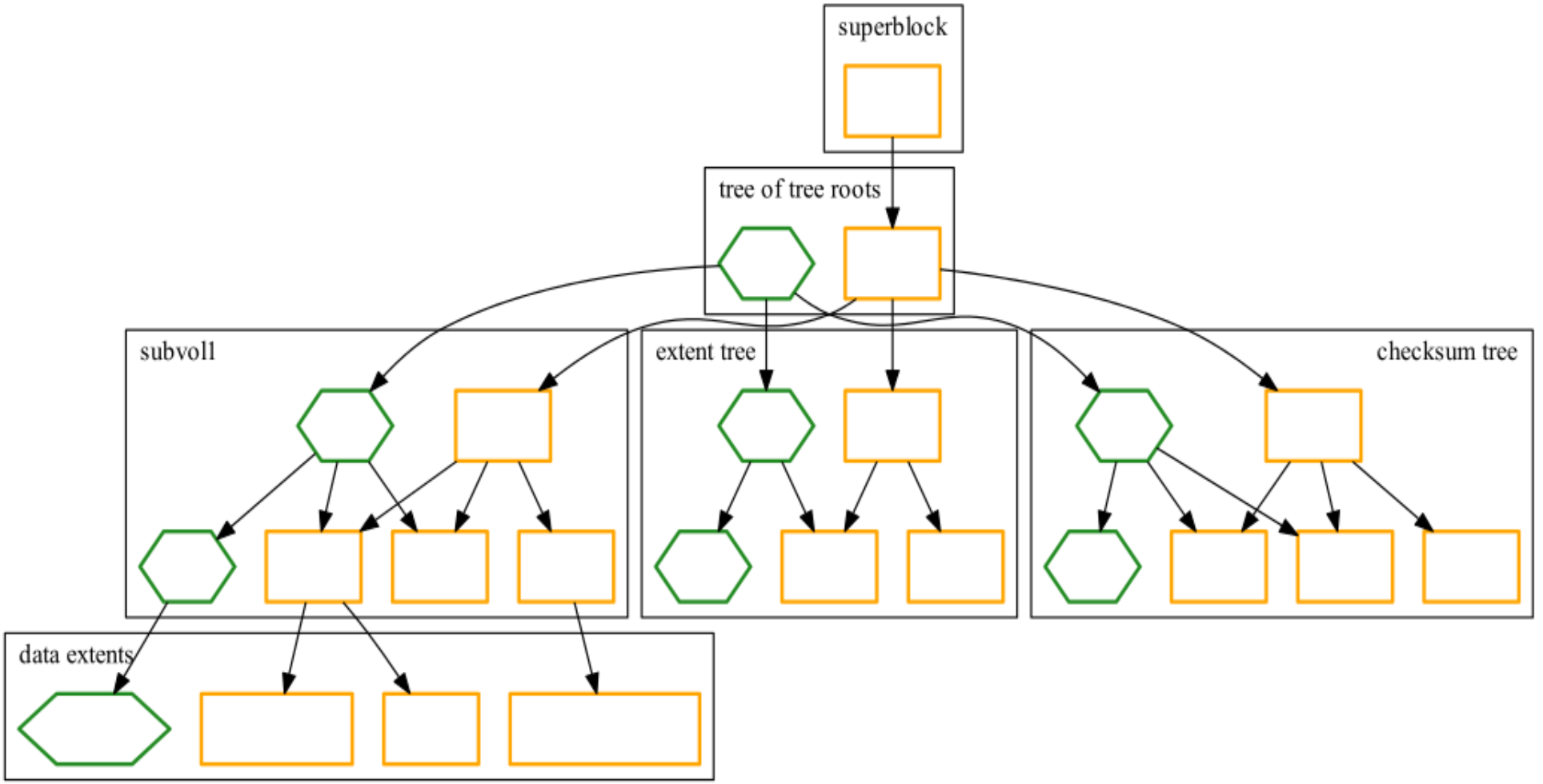
Устройство каталога



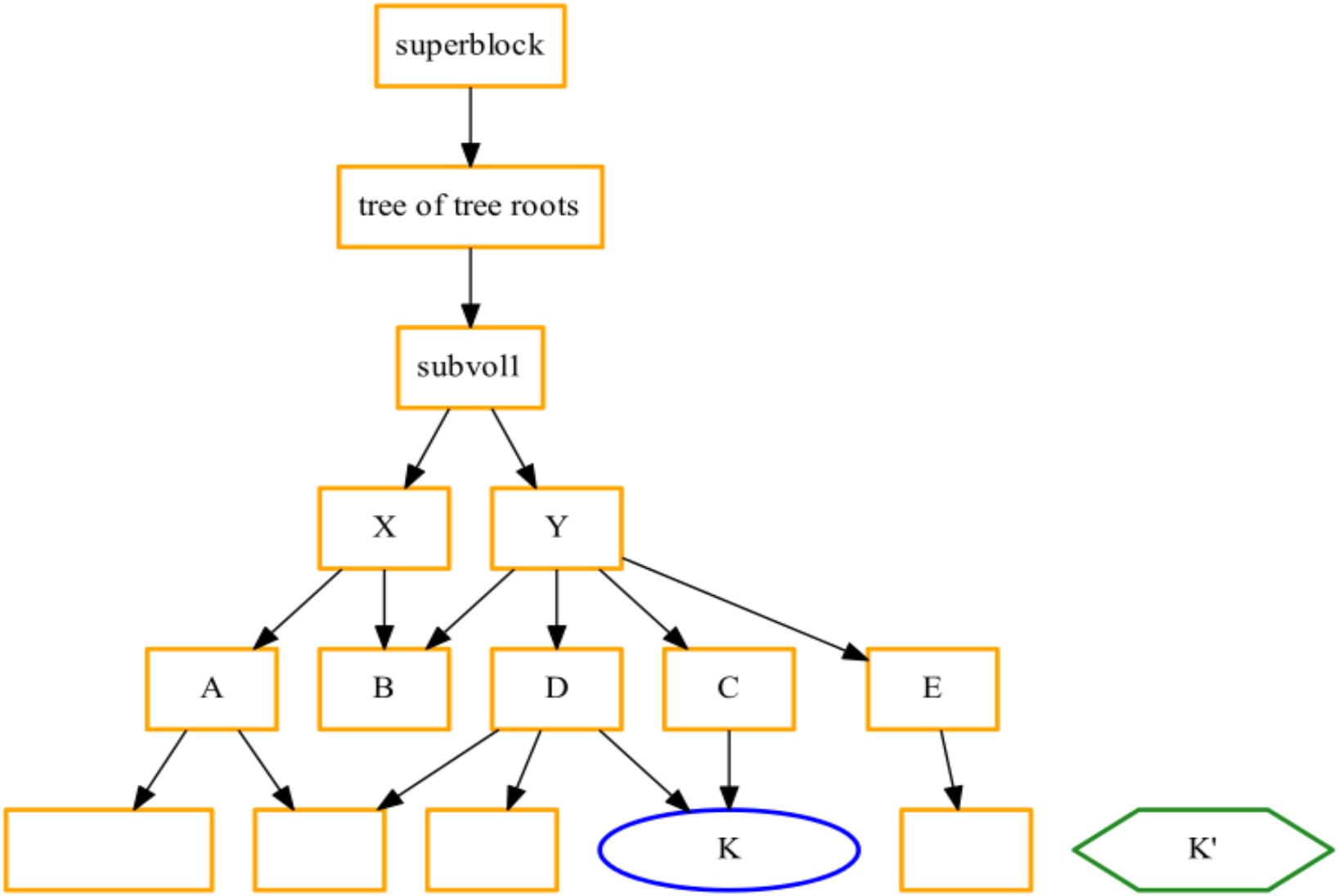
Устройство листа (leaf node)



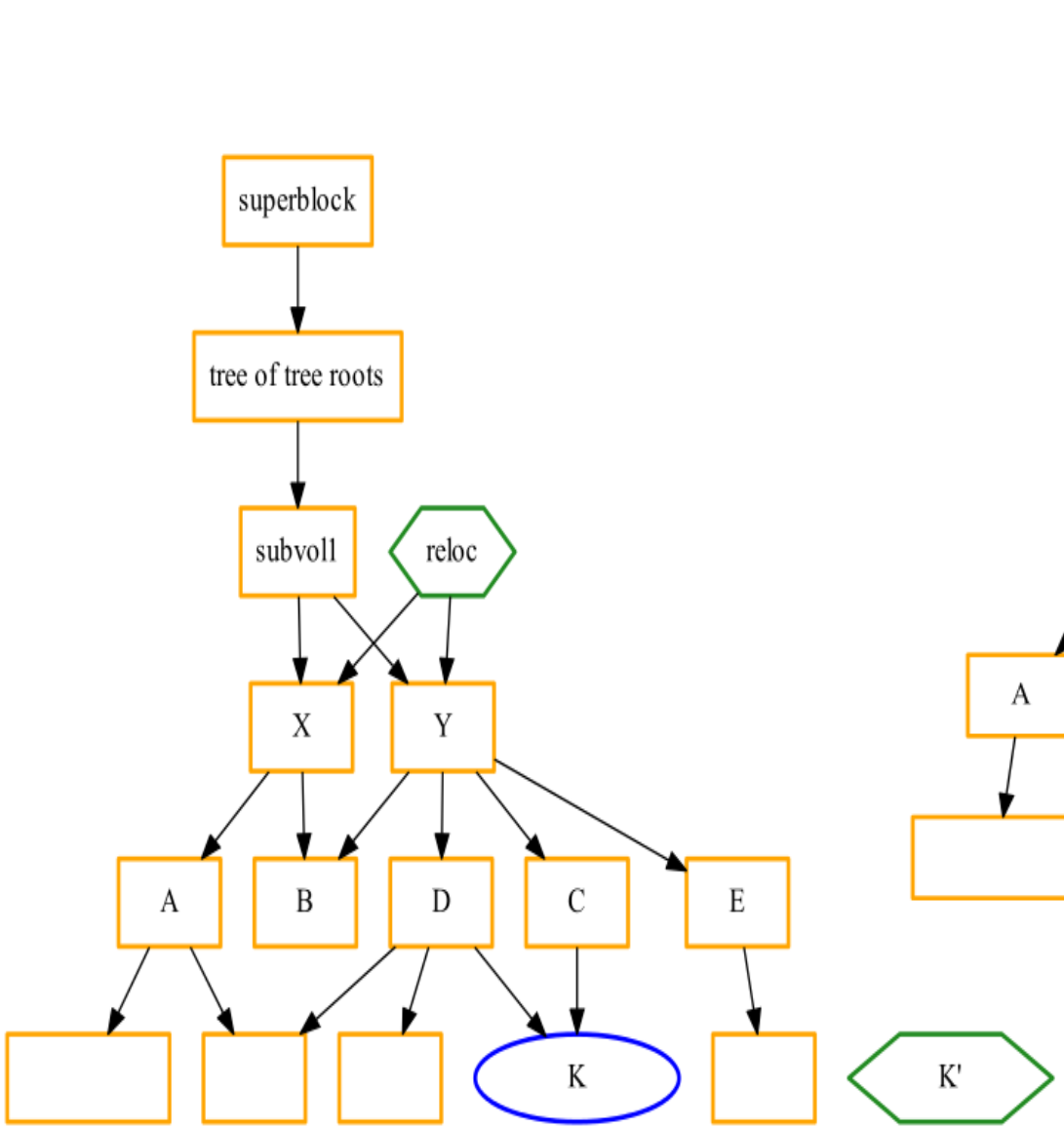
Деревья btrfs

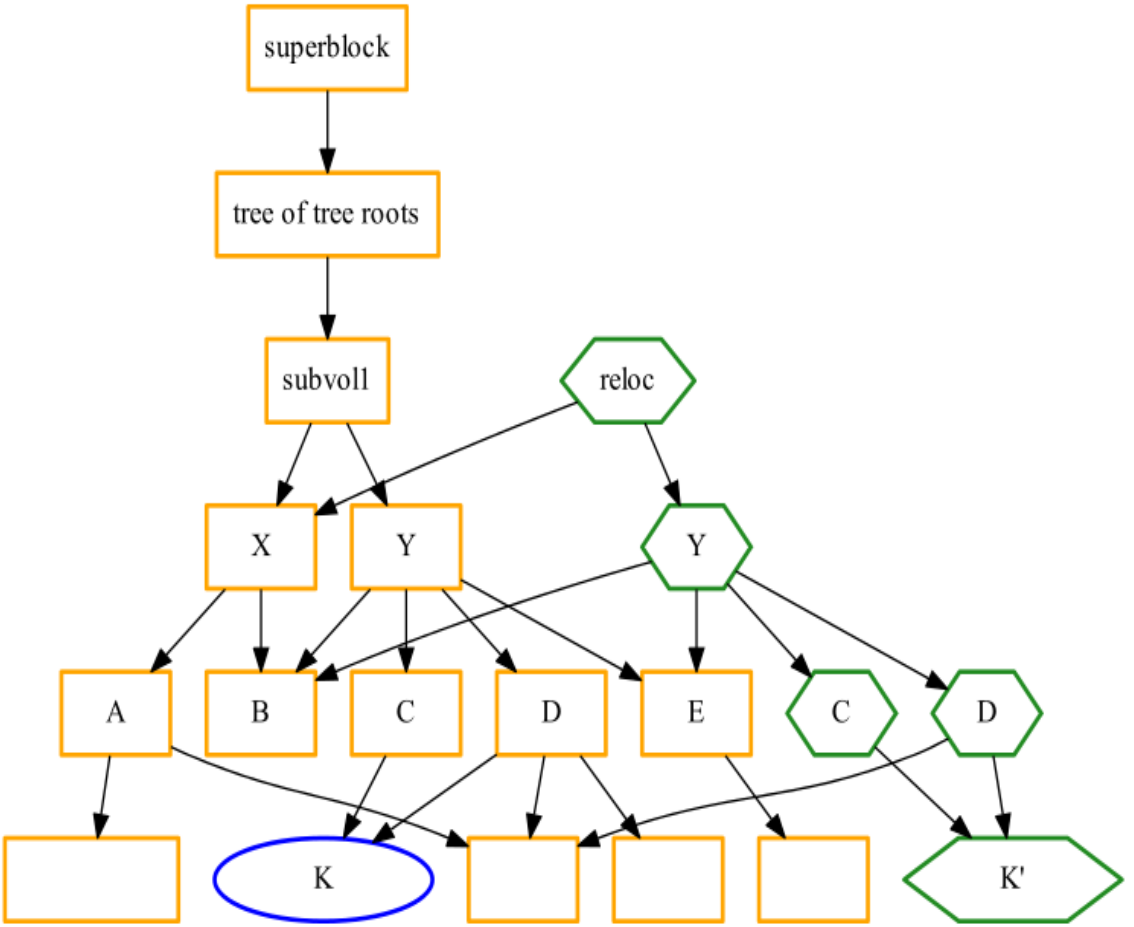


Клонирование (1/3)

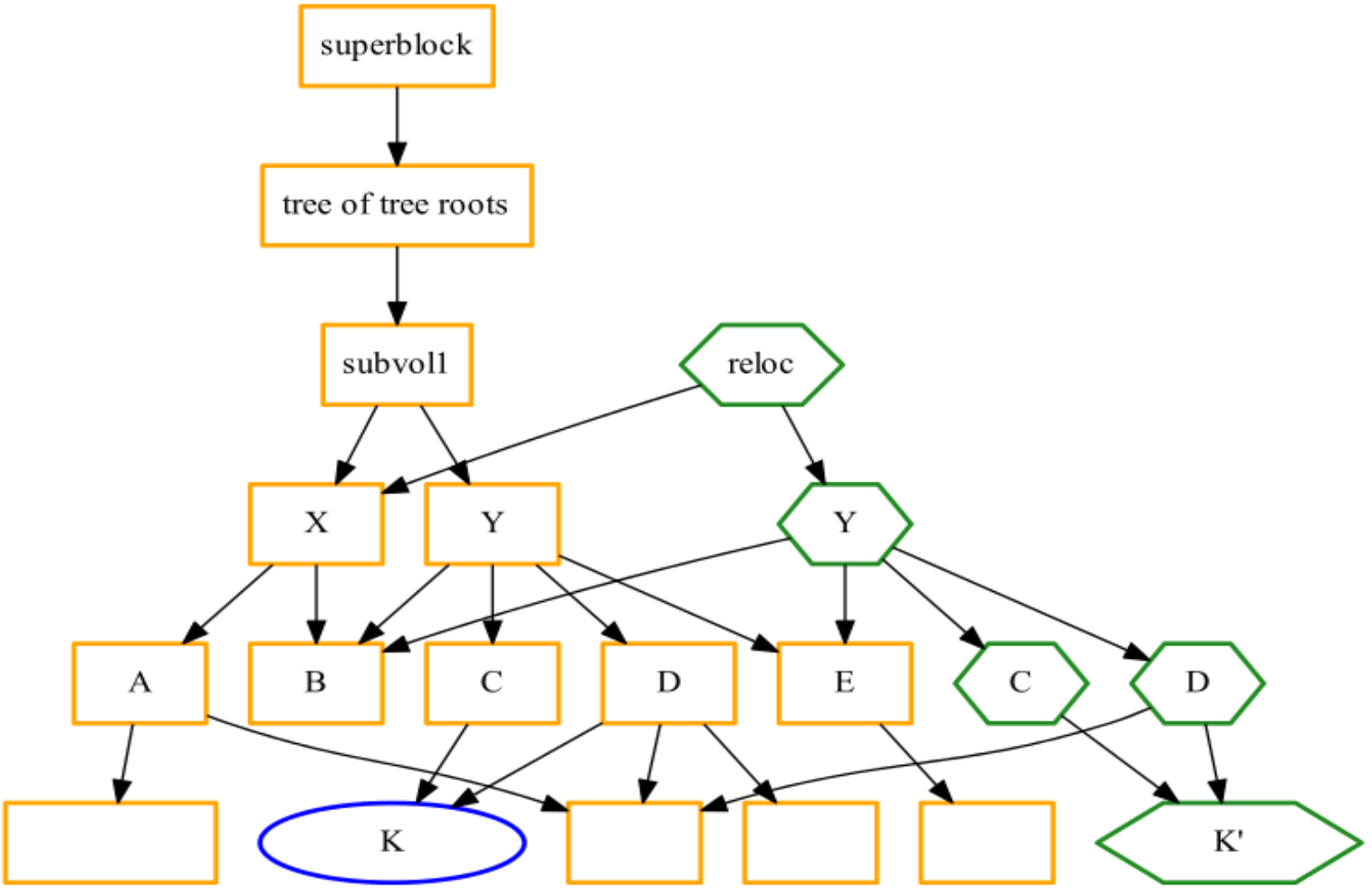


Клонирование (2/3)





Клонирование (3/3)



Источники и ссылки

- Roden, Bacik, Mason BTRFS: The Linux B-Tree Filesystem – 2012
- btrfs.wiki.kernel.org
- Satoru Takeuchi. Btrfs Current Status and Future Prospects
- https://www.howtoforge.com/a-beginners-guide-to-btrfs

Ceph

Темы

- Черты Software Defined Storage
- История Ceph
- Архитектура и технические решения

NB: ссылки на источники, использованные в презентации, приведены на последнем слайде

Software Defined Storage

- SDS – «Интеллектуальная» часть сети хранения данных, не привязанная к оборудованию;
- SDS способна самостоятельно принимать решения относительно места хранения, методов защиты и перемещения данных
- Имеет линейно масштабируемую архитектуру
- Control-path отделен от data-path

CAP-теорема, или 2 из 3

- **C**onsistency – в лобой момент времени данный не противоречат друг другу на узлах
- **A**vailability – любой запрос завершается корректным откликом
- ***P**artition tolerance – расщепление системы на независимые компоненты не приводит к некорректности отклика от каждой*

RAID в прошлом?

- Объем 1 диска растет, время восстановления увеличивается → занимает часы
- RAID требует идентичных резервных дисков
- RAID не защищает от сбоев сети, ОС,...
- Деградация производительности при сбое нескольких дисков

Вместо RAID → Erasure Codes

- стратегия прямой коррекции ошибок:
  - исходное сообщение длинной K трансформируется в сообщение длиной N (N>K)
  - по любым K символам сообщение восстановимо
- примеры реализации:
  - Parity в RAID
  - Коды Рида-Соломона

История и развитие Ceph

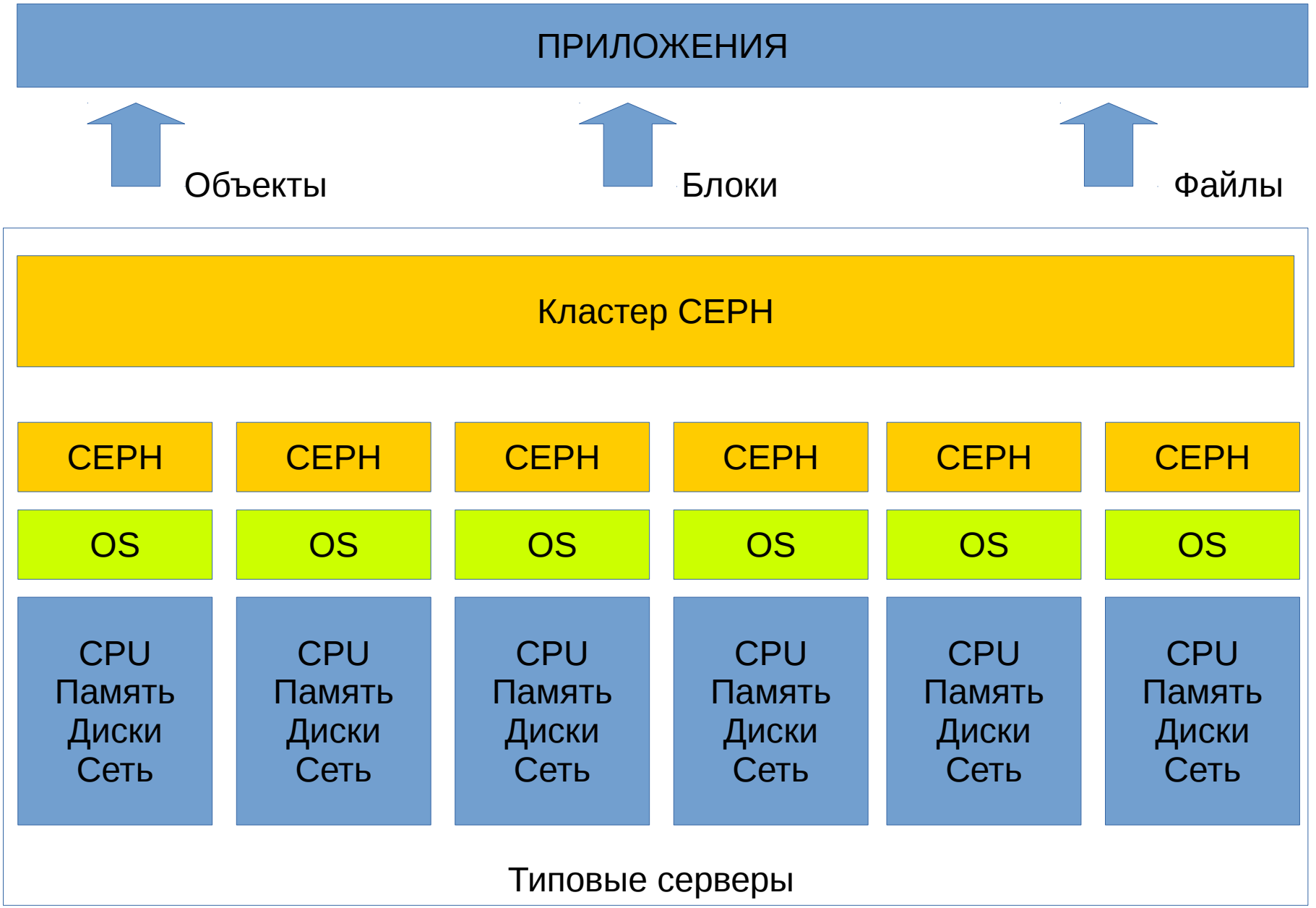
- 2003, Сейдж Вейл (Sage Weil) часть проекта докторской диссертации – ФС
- 2003—2007, Исследовательский проект, развивался сообществом
- 2007—2011, DreamHost, начало промышленного применения
- 2012 – Inktank, корпоративная подписка, саппорт
- 2014 – Red Hat Inc. (Cisco, CERN и Deutsche Telekom, Dell, Alcatel-Lucent, …)

RH: A next-generation platform for petabyte-scale storage

Архитектурные принципы

- Все компоненты должны быть масштабируемы
- Нет единой точки отказа
- Решение должно опираться на открытое программное обеспечение
- ПО должно работать на обычном железе (commodity hardware)
- Максимальная самоуправляемость, везде, где возможно

Унифицированный стек Ceph

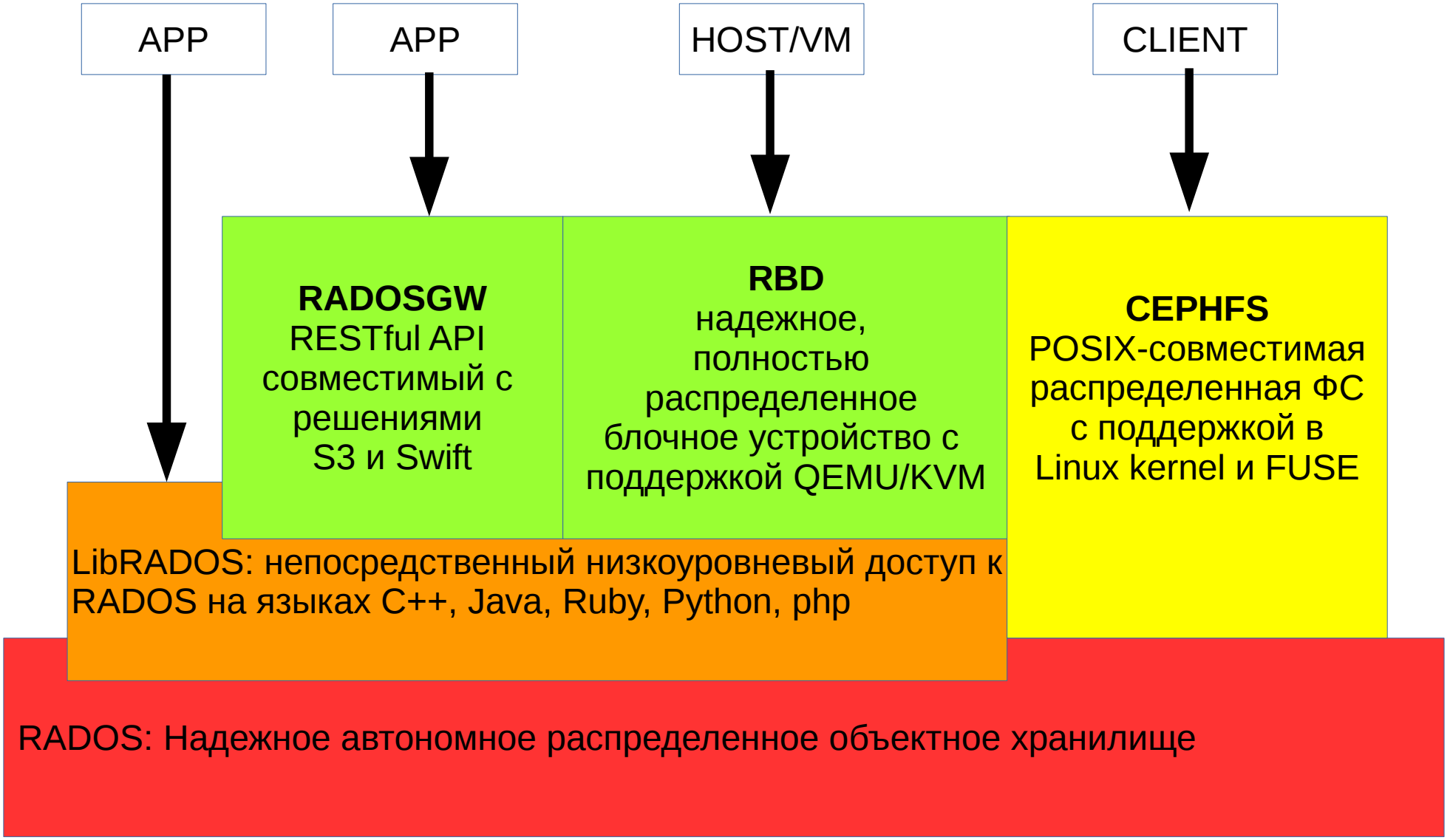


[Предвзятое] сравнение с открытыми аналогами

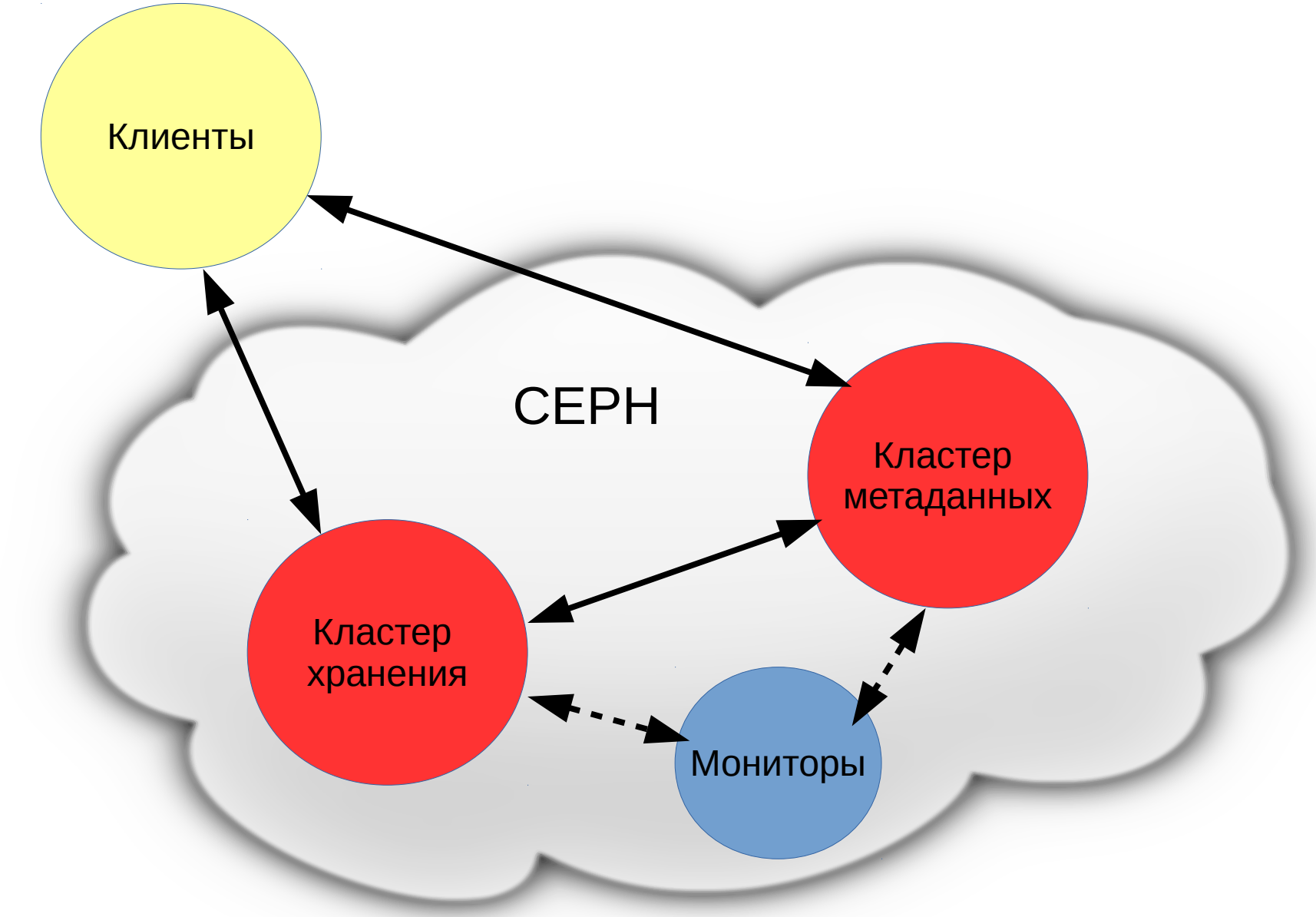
Наименование	Отличия
Integrated Rule-Oriented Data System (iRODS)	Сервер метаданных iCAT, является единой точкой отказа, нет блочного хранения, нет RESTful
HDFS	Нет блочного хранения, Сервер метаданных NameNode – потенциальная точка отказа
Lustre	Сервер метаданных – bottle neck, отсутствие встроенного механизма обнаружения и исправления сбоев узлов
GlusterFS	Блочный доступ и удаленная репликация не являются встроенными, а доступны как расширения

Облачные решения использующие Ceph (на 2016)

Архитектура



Концептуальный взгляд



Reliable Autonomic Distributed Object Store → RADOS

- Репликация CRUSH (Controlled Replication Under Scalable Hashing)
- Автоматическое восстановление объектов из копий, при разрушении
- Автоматическая миграция данных

Файловая система OSD Ceph

- Файловые системы:
  - **btrfs**
  - **xfs**
  - ext4
- Поддержка (xATTRs):
  - xattr_name → xattr_value,

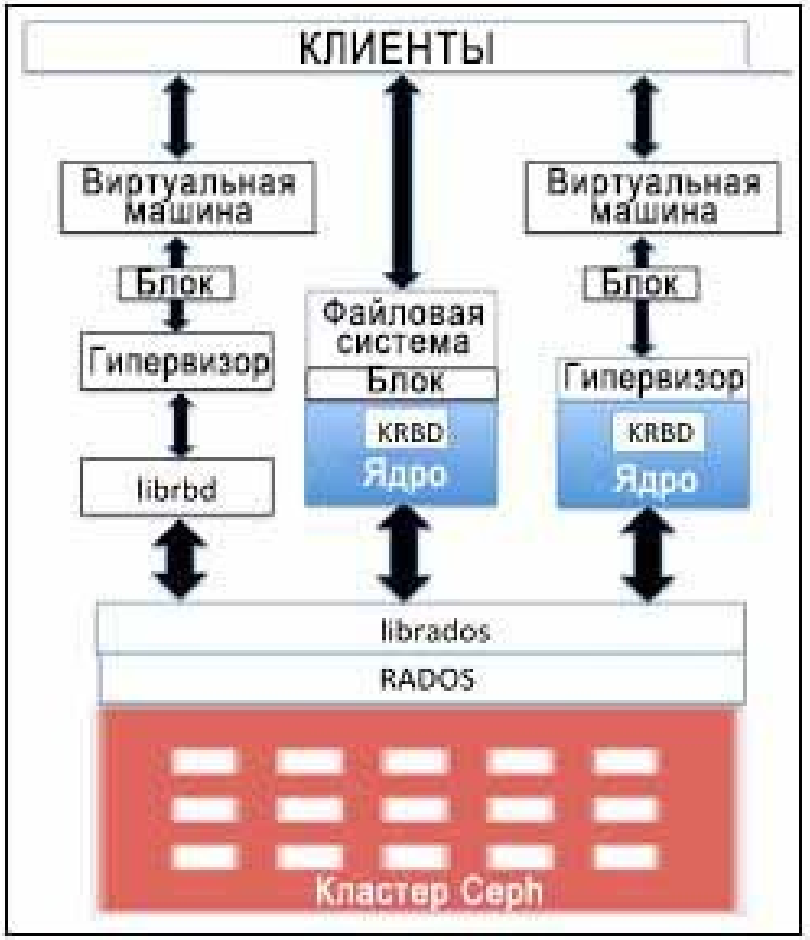
Мониторы Ceph

- Монитор – демон обеспечивающий поддержание режима членства в кластере, хранение настроек и состояния.
- Карты:
  - монитора
  - OSD
  - PG
  - CRUSH
  - MDS
- Согласованность принятия решений обеспечивает paxos → число мониторов нечетно, >=3

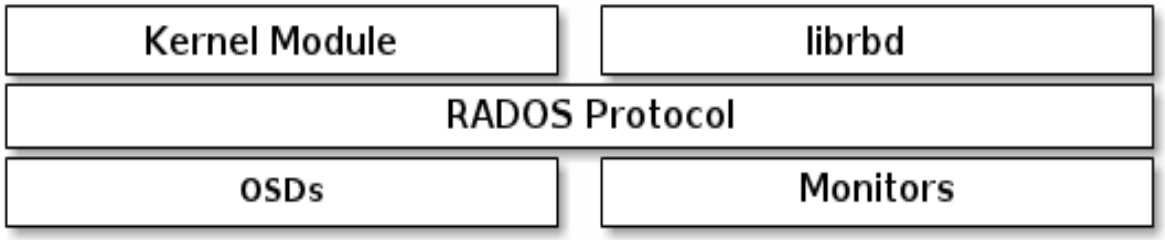
Алгоритм PAXOS

- Обеспечивает консенсус
- Соответствует показателям:
  - Согласованность → решение принимается только единогласно
  - Нетривиальность → количество вариантов решения известно заранее и больше 1
  - Живучесть → если предлагается принять решение, то решение (не обязательно предложенное) рано или поздно будет принято.

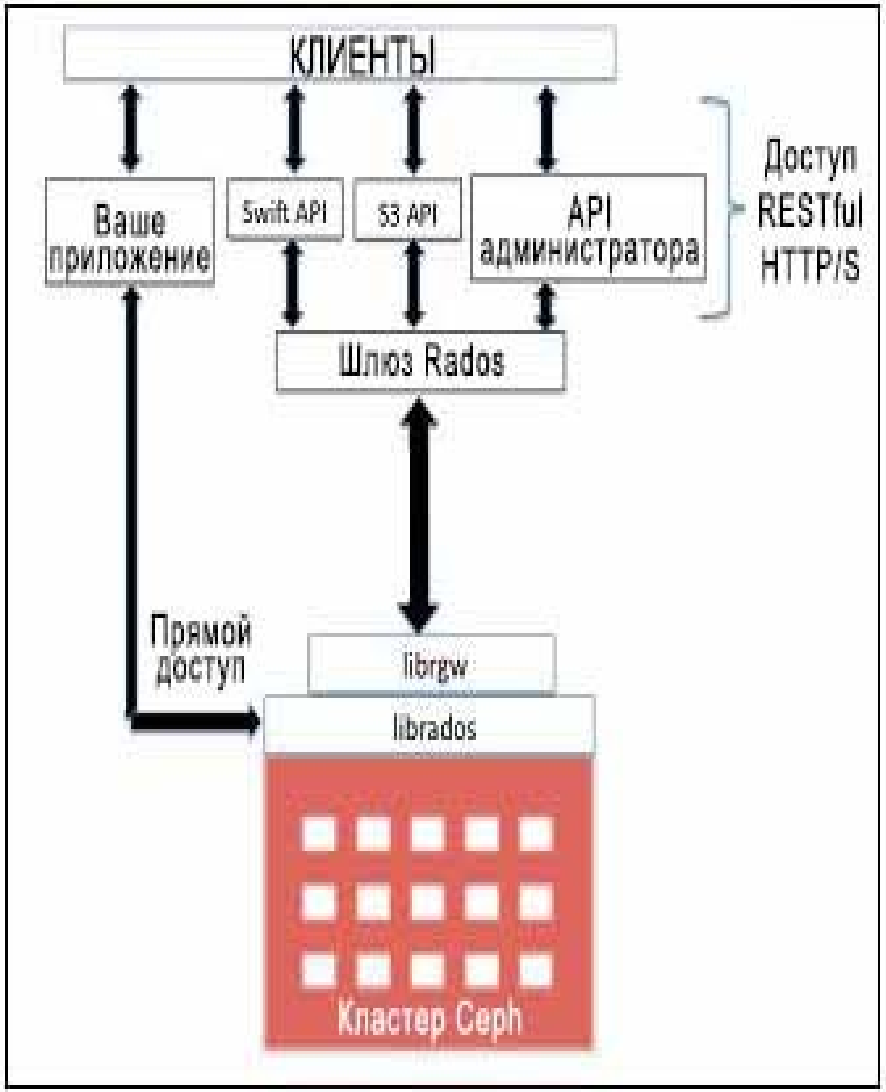
KRBD



- предоставление блочного хранения гипервизорам и виртуальным машинам
- реализация thin provisioning
- Поддержка:
  - XEN
  - KVM
  - QEMU



Шлюз объектов Ceph (RADOS)

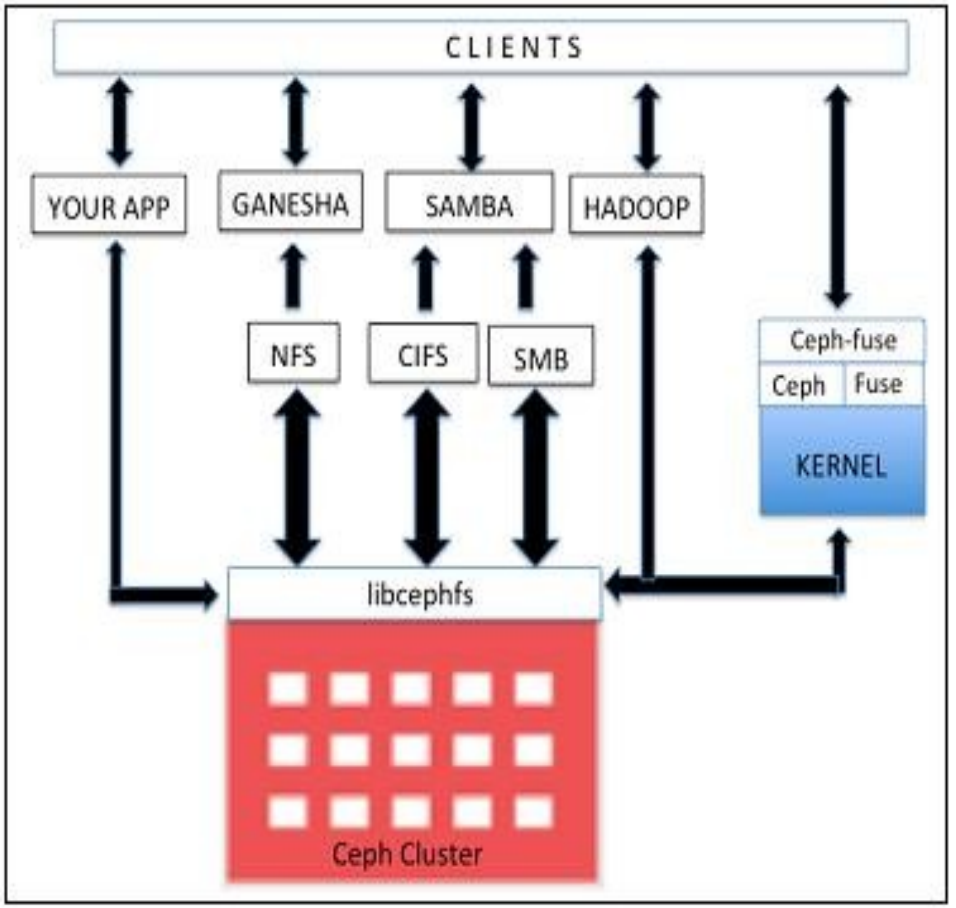


- Amazon S3 RESTful API
- OpenStack Swift API
- HTTP RESTful API (Admin)

Сервер метаданных MDS Ceph

- Ceph MDS – демон, обеспечивающий
  - возможность монтировать на клиентах POSIX ФС произвольного размера
  - управление filesystem namespace
  - координация доступа к OSD кластеру

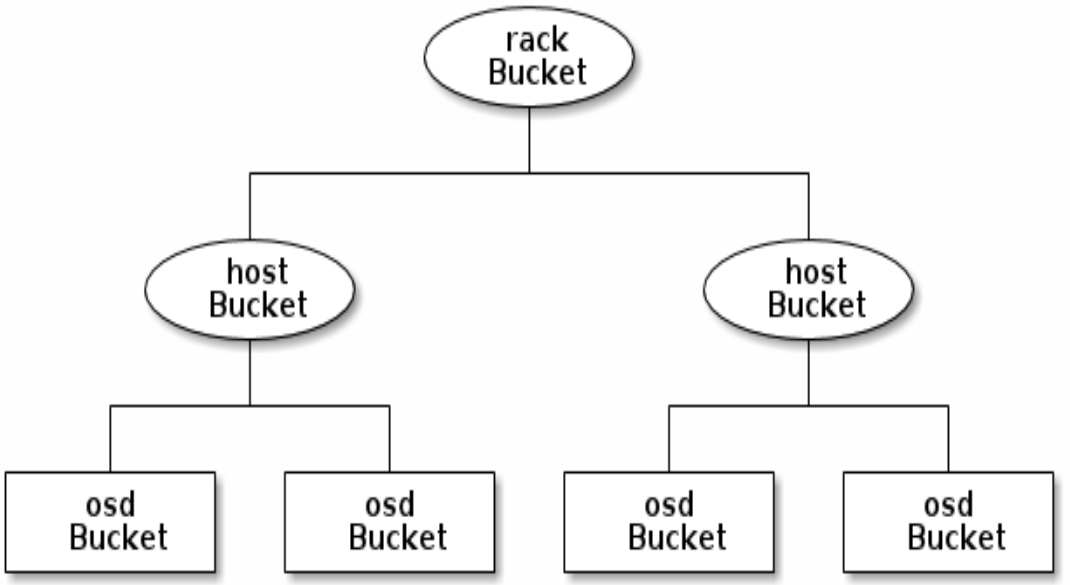
CephFS



- Реализована в libcephfs
- поддержка:
  - NFS
  - CIFS
  - SMB
- Альтернатива Hadoop HDFS

CRUSH

- CRUSH: Controlled Replication Under Scalable Hashing
- CRUSH map (см. пример)
- Типы корзинок (размен между производительностью и организационной
  - Uniform
  - List
  - Tree
  - Straw

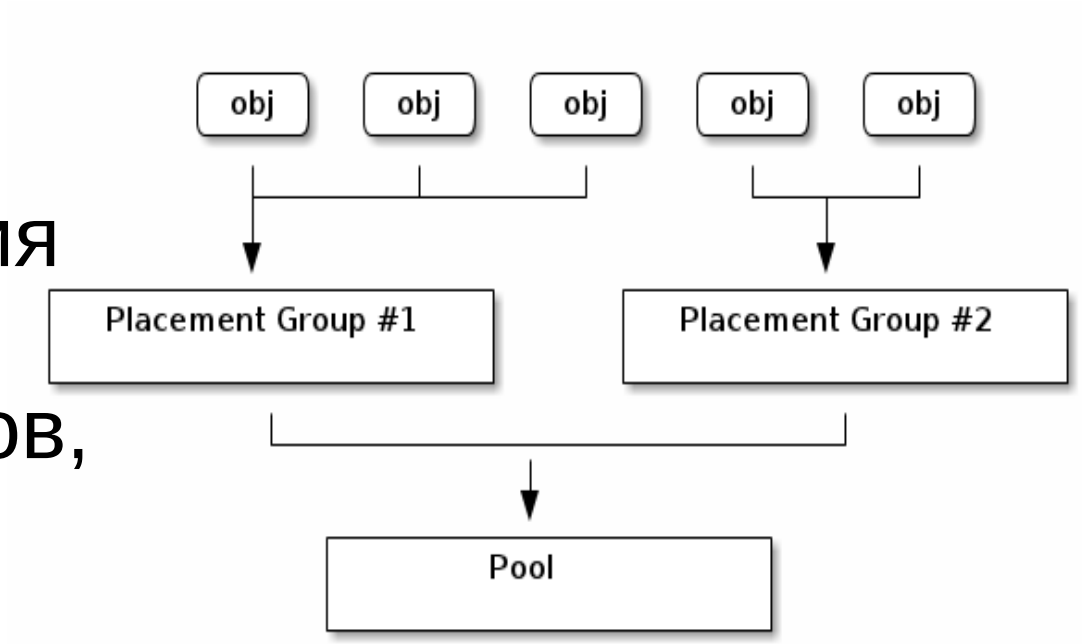


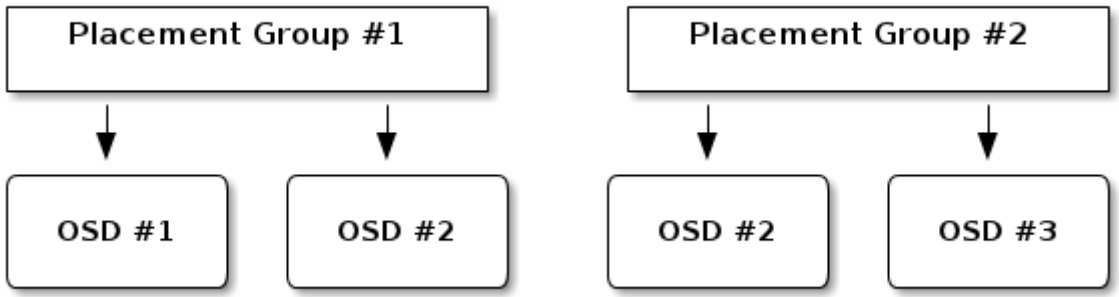
Алгоримтмы распределения

- Uniform – все веса строго одинаковы. Подходит, когда кластер состоит из совершенно одинаковых машин и дисков
- List – перемещаемые данные с некоторой вероятностью попадают в новое или старое хранилище. Expanding cluster.
- Tree – бинарные деревья, оптимизация скорости помещения объектов в хранилище.
- Straw – комбинация стратегий List и Tree для реализации принципа «разделяй и властвуй». Обеспечивает быстрое размещение, но иногда создает проблемы для реорганизации

Placement groups (PG)

- Группа размещения – логическая коллекция объектов, внутри пула.
- Группа реплицируется на несколько OSD





Пулы Ceph

- Устойчивость
  - установка количества копий объекта
  - для EC количество кодированных блоков (chunks)
- Группы размещения
- CRUSH правила
  - для пула можно определить правила избыточности
- Снапшоты
- Установка владельца

Источники и ссылки

- http://ceph.com
- Karan Singh «Learning Ceph» Packt Publishing
- Sage A. Weil et al. RADOS: A Scalable, Reliable Storage Service for Petabyte-scale Storage Clusters
- Sage A. Weil et al. CRUSH: Controlled, Scalable, Decentralized Placement of Replicated Data
- http://rus-linux.net/nlib.php?name=/MyLDP/file-sys/ceph/ceph.html
- https://github.com/carmstrong/multinode-ceph-vagrant

GlusterFS

Определение

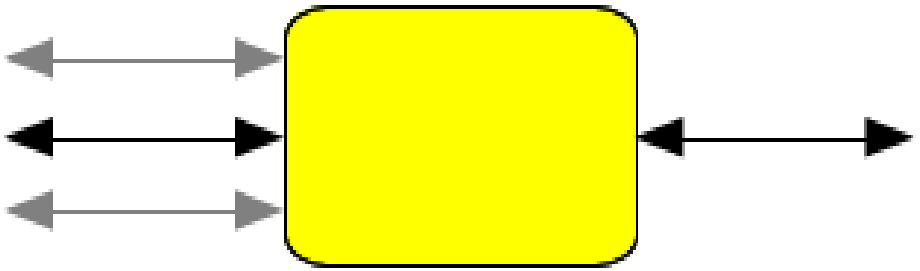
- **GlusterFS = GNU + Cluster**
  - масштабируемая
  - сетевая файловая система
  - ориентированная на интенсивный обмен данными типа:
    - облачное хранилище
    - потоковое мультимедиа,
  - использующая типовое [commodity] оборудование

Терминология

- Brick
- Том
- FUSE
- Транслятор
- Cluster
- Namespace

Транслятор

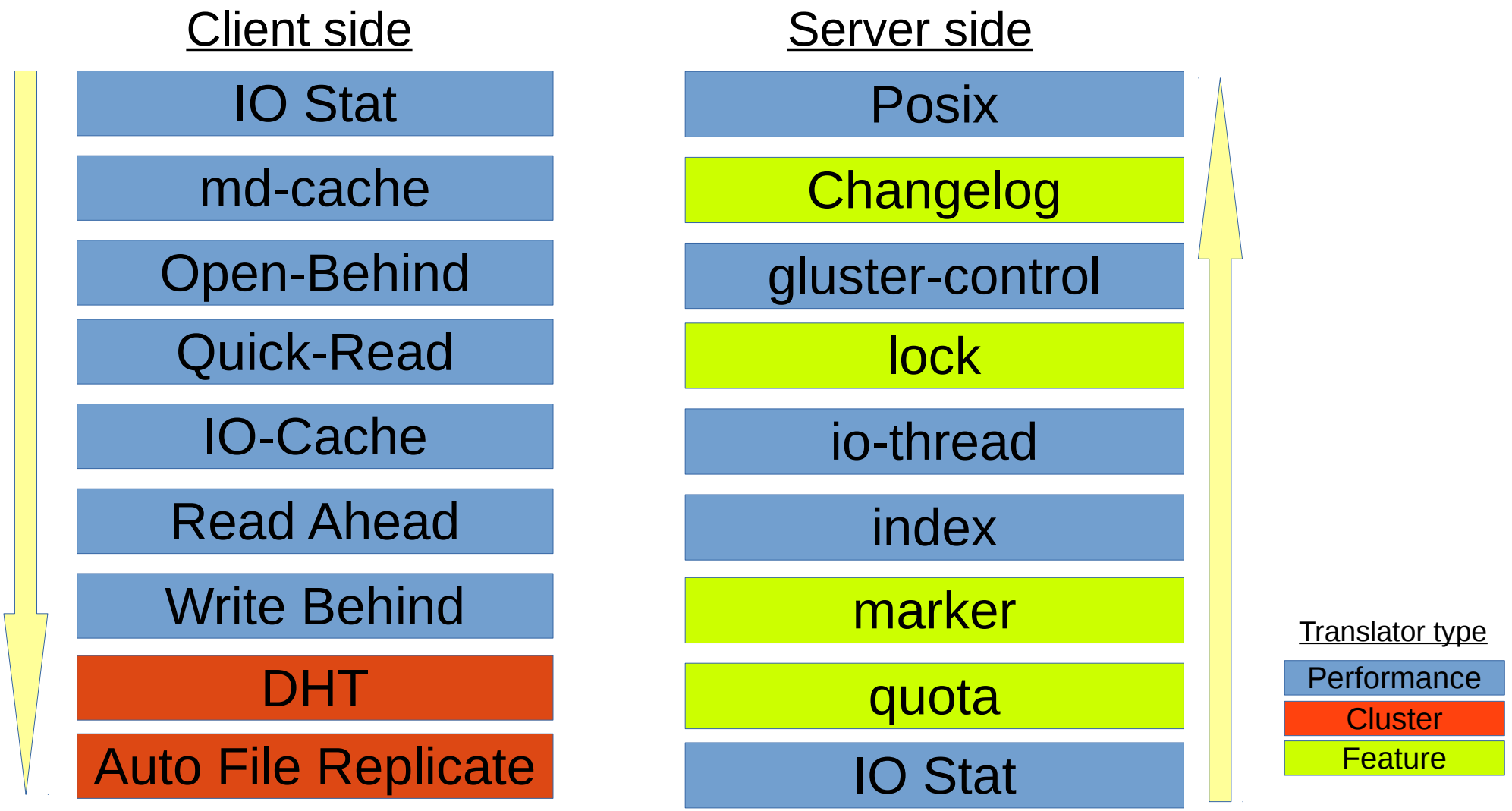
- Транслятор – модуль, конвертирующий запросы
  - от пользователей к хранилищу
  - от запроса к запросу
  - реализация возможностей
  - построение стека



Типы [уровни] трансляторов

- Storage
- Debug
- Cluster
- Encryption
- Protocol
- Performance
- Binding
- System
- Scheduler
- Реализация дополнительных возможностей [квоты, фильтры, блокировки]

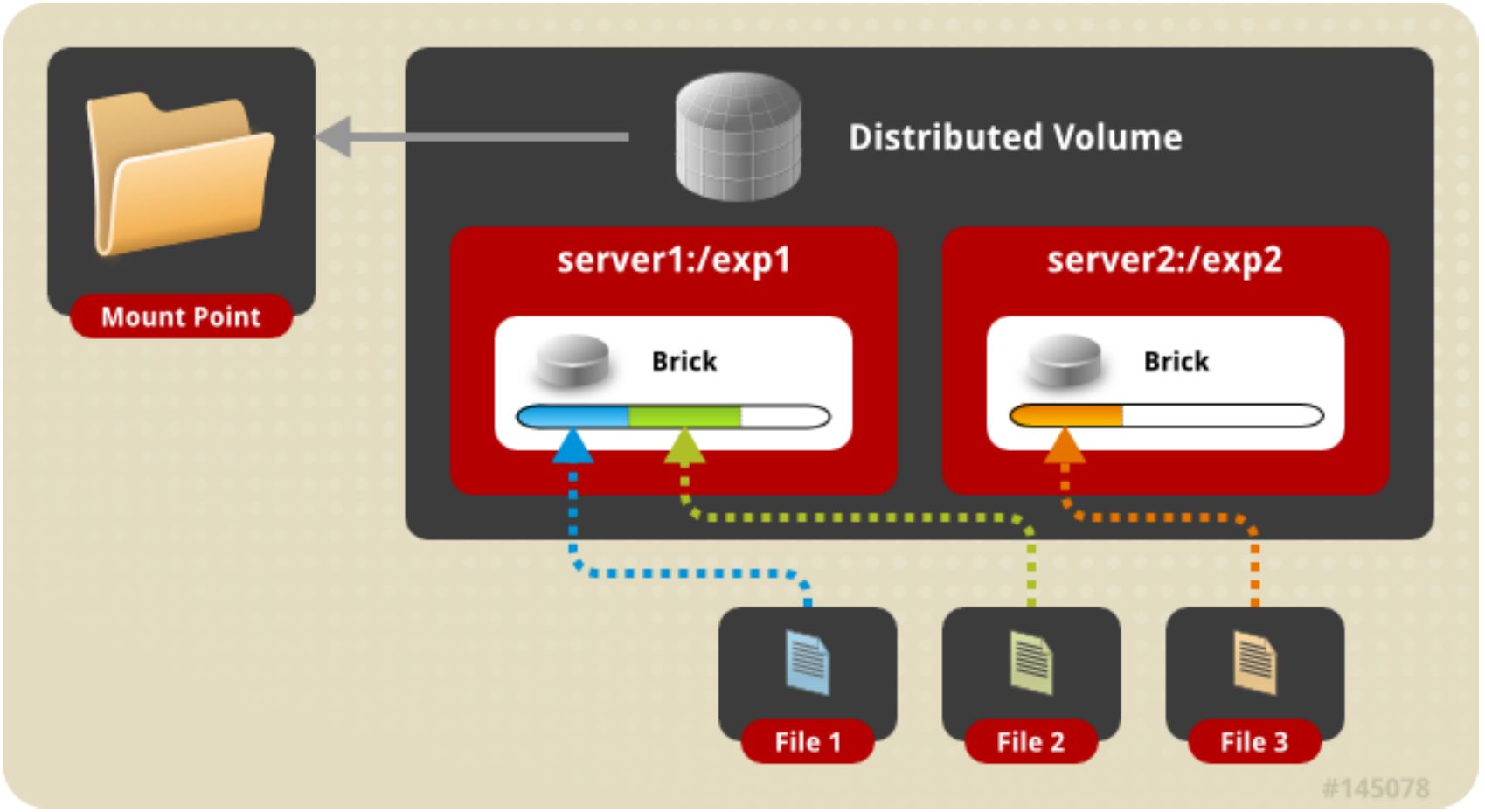
Типовой стек трансляторов



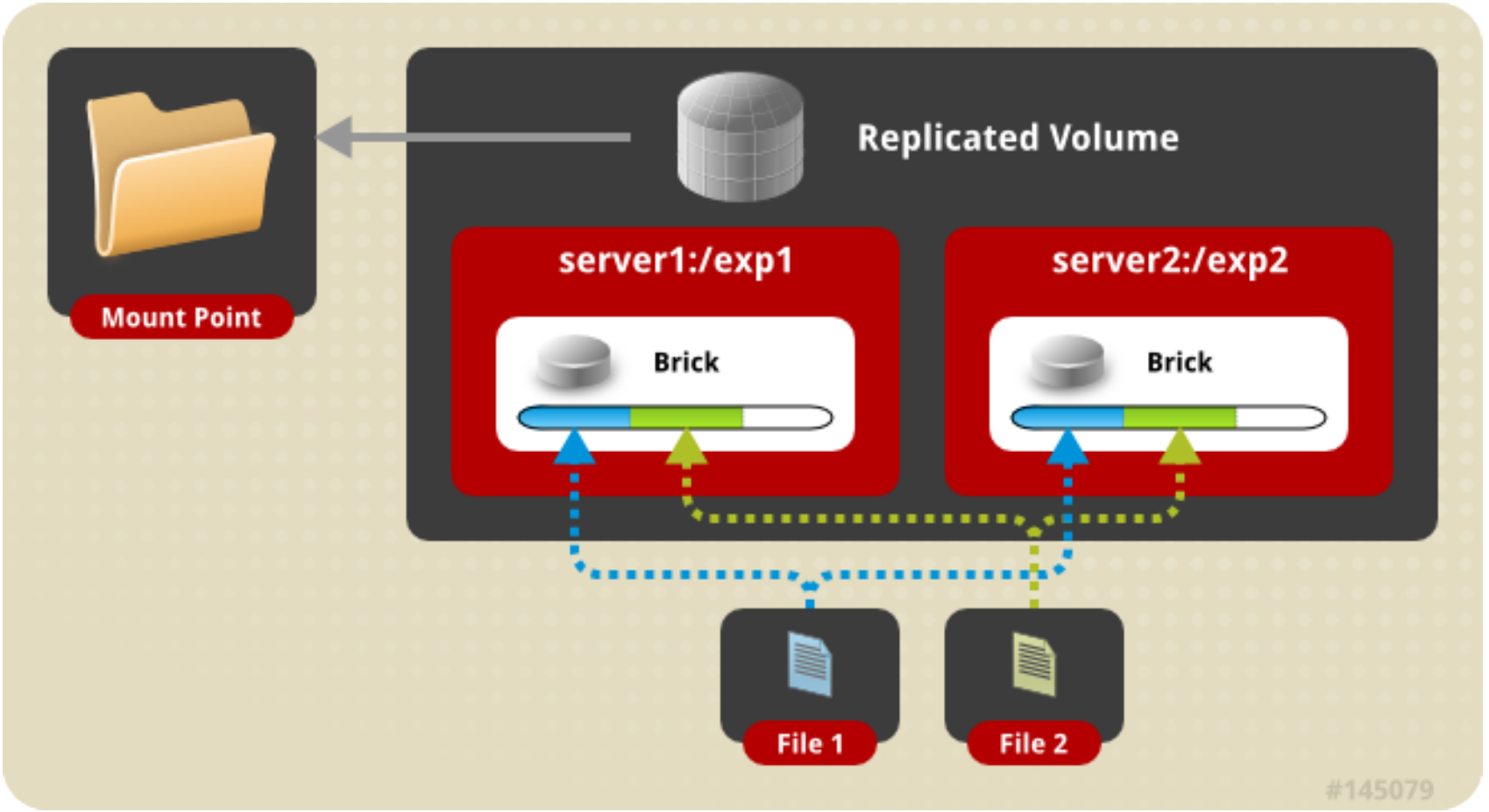
Типы томов

- Распределенный том (по-умолчанию)
- Реплицированный том
- Распределенный реплицированный
- Страйп
- Распределенный страйп

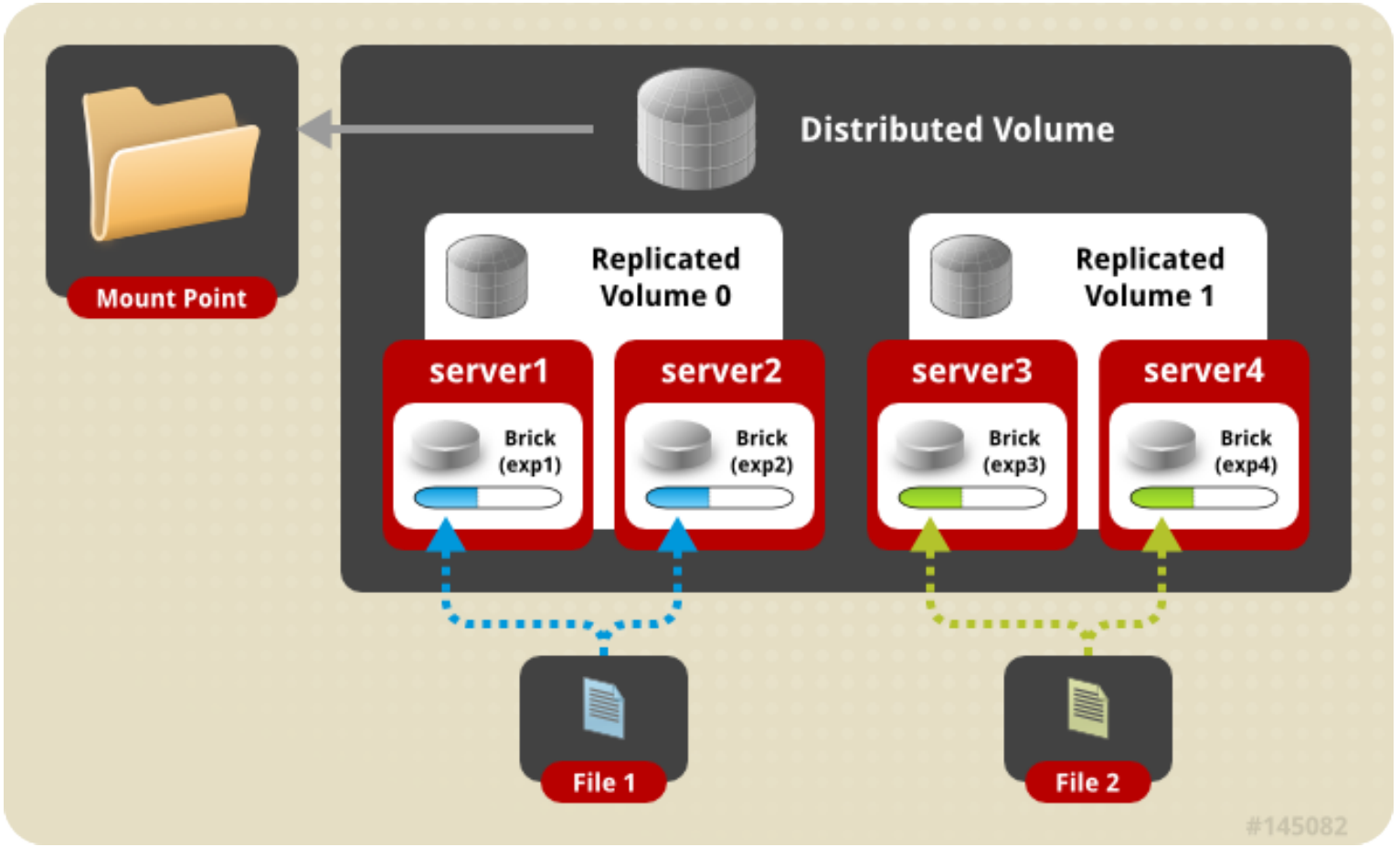
Распределенный том



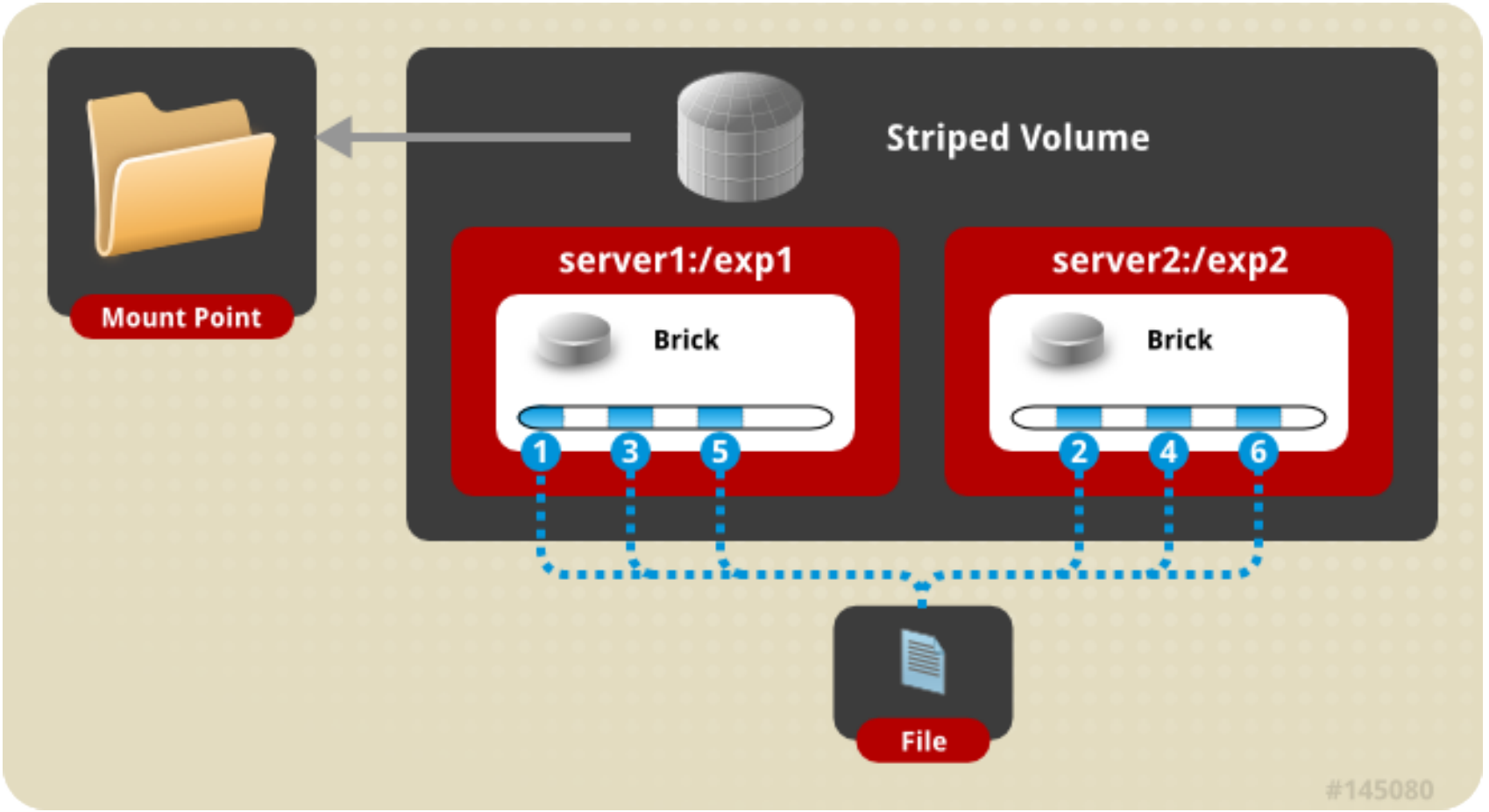
Реплицированный том



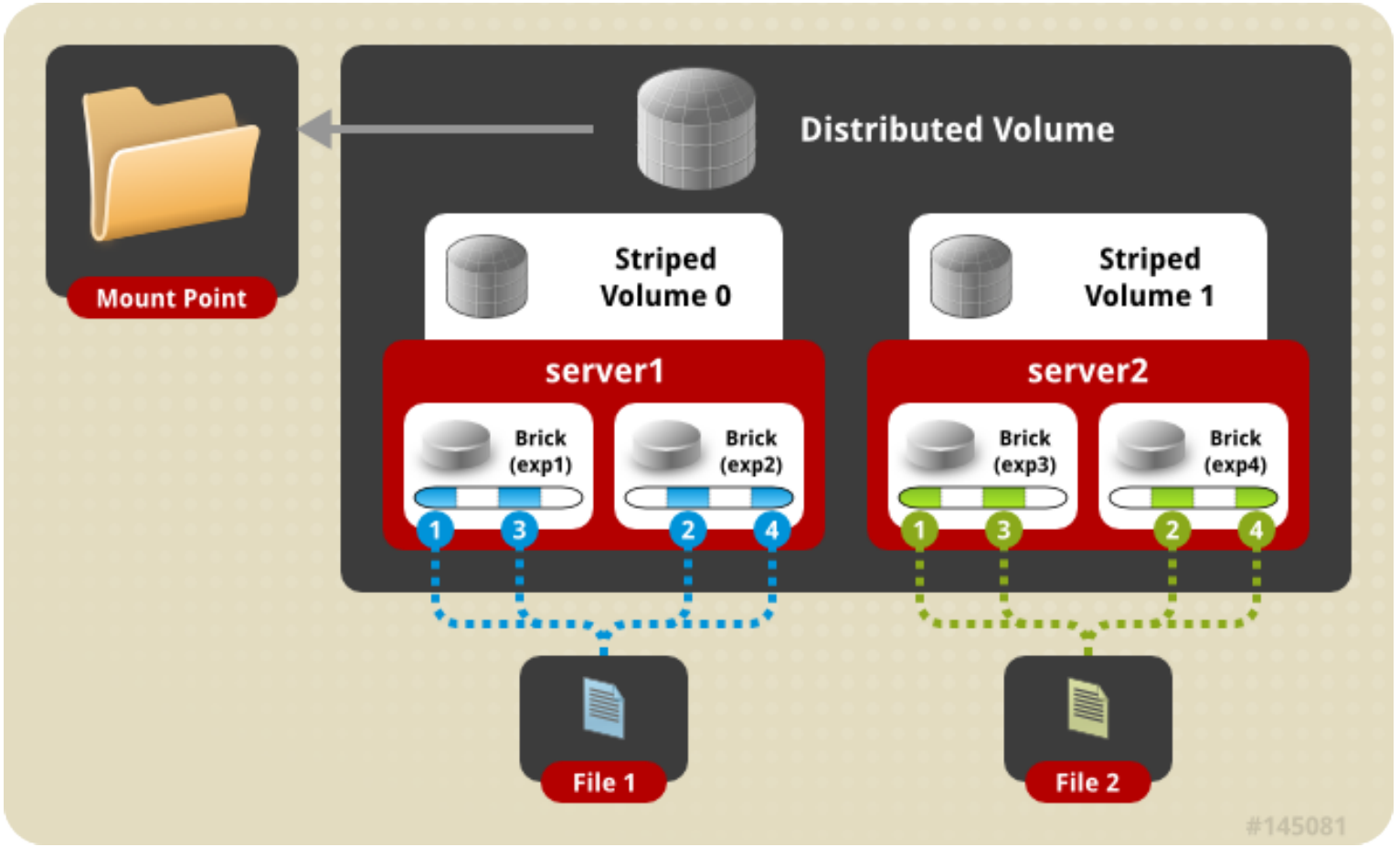
Распределенный реплицированный том



Страйп


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

Распределенный страйп



Возможности

- Распределенные Хэш таблицы
- Split Brain Resolution
- libgfapi
- Non Uniform File Access
- Export via pNFS
  - Ganesha
- Интеграция с oVirt
- Интеграция с qemu
- Rebalance
- WORM (Write Once Read Many)
- Распрелеленная гео-репликация
- Шардинг транслятор
- Tiering
- Automatic File Replication
- Файловые снапшоты
- Brick Failure Detection

Источники и ссылки

- http://www.gluster.org/
- http://gluster.readthedocs.org/en/latest/
- https://github.com/gluster/glusterfs