Как распределённо хранить триллионы файлов

mayflower.work

Лебедев Константин - DevOps Engineer

О себе

Konstantin Lebedev
kmlebedev

- **Beeline** - Lead engineer VoIP and Telematics - 9 лет
- **Yandex** - DevOps Engineer streaming - 5 лет
- **Tochka** - DevOps Engineer infrastructure - 4 лет
- **Mayflower** - DevOps Core Engineer - 2 год

СОДЕРЖАНИЕ

Характеристики
Распределённых Систем Хранения Данных

Распределённость

Данные хранятся
на нескольких узлах
(серверах) сети

01

Характеристики
Распределённых Систем Хранения Данных

Распределённость

Данные хранятся
на нескольких узлах
(серверах) сети

01

Масштабируемость

Система может легко
масштабироваться

02

Характеристики
Распределённых Систем Хранения Данных

Распределённость

Данные хранятся на нескольких узлах (серверах) сети

01

Масштабируемость

Система может легко масштабироваться

02

Высокая доступность

Система непрерывно работает даже в случае выхода из строя одного или нескольких узлов

03

Характеристики
Распределённых Систем Хранения Данных

Распределённость

Данные хранятся на нескольких узлах (серверах) сети

01

Масштабируемость

Система может легко масштабироваться

02

Высокая доступность

Система непрерывно работает даже в случае выхода из строя одного или нескольких узлов

03

Отказоустойчивость

Снижены риски потери данных в случае отказа отдельных компонентов

04

Характеристики
Распределённых Систем Хранения Данных

Снижение затрат

За счёт
использования
меньших
по мощности
серверов

05

Характеристики
Распределённых Систем Хранения Данных

Снижение затрат

За счёт использования меньших по мощности серверов

05

Безопасность

Использование шифрования данных и аутентификация

06

Характеристики
Распределённых Систем Хранения Данных

Снижение затрат

За счёт использования меньших по мощности серверов

05

Безопасность

Использование шифрования данных и аутентификация

06

Прозрачность

Приложениям не нужно знать о внутреннем устройстве системы хранения

07

Прозрачность

простота использования через (S3) API

- Package cache (Nexus)
- Docker registry (Harbor, Gitlab)
- Artifacts (Gitlab runner)
- Backups (Restic)
- Logs (Elastic snapshots)
- NextCloud

Характеристики
Распределённых Систем Хранения Данных

Снижение затрат

За счёт использования меньших по мощности серверов

05

Безопасность

Использование шифрования данных и аутентификация

06

Прозрачность

Приложениям не нужно знать о внутреннем устройстве системы хранения

07

Асинхронность

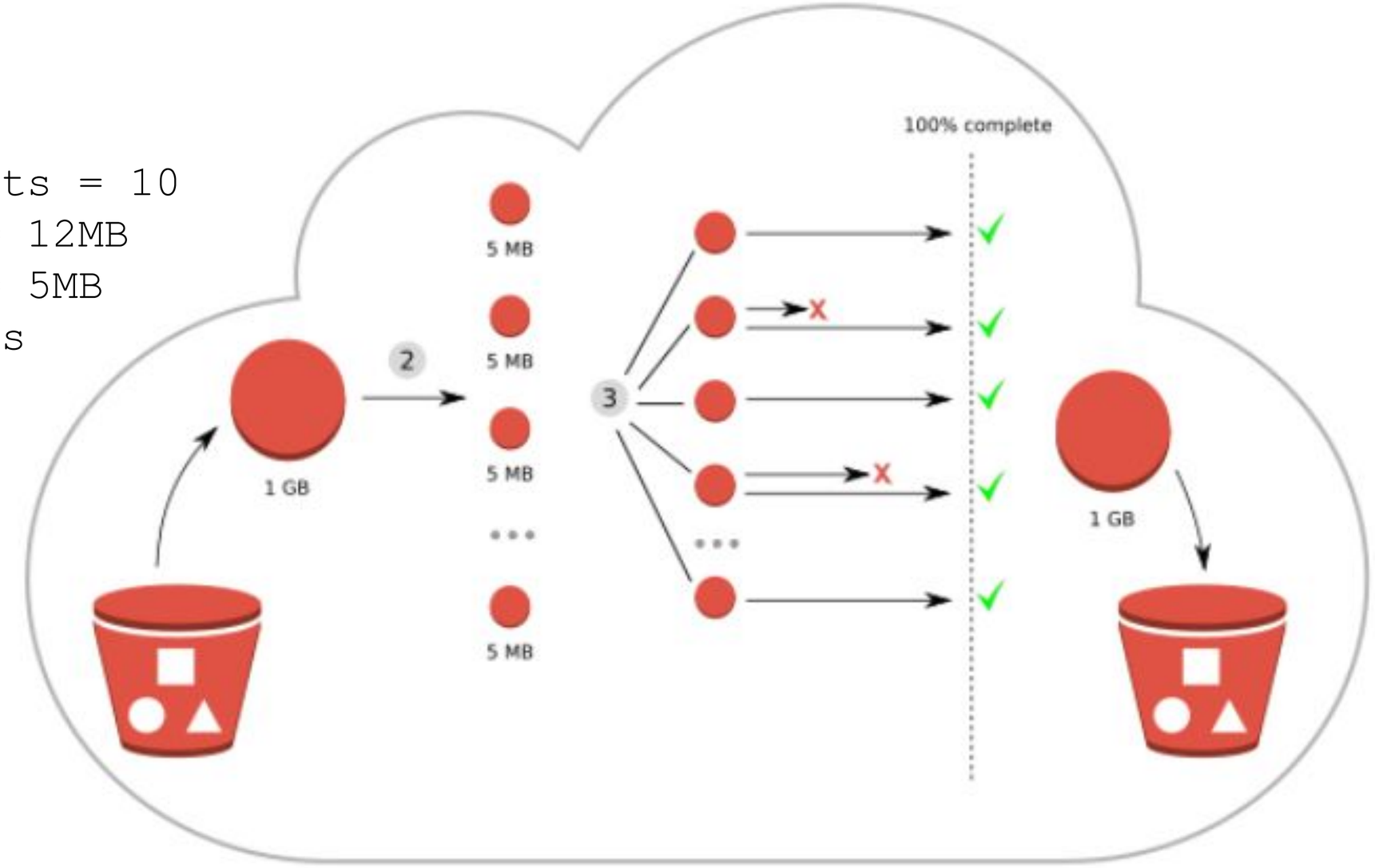
Разделение работы между несколькими узлами позволяет обрабатывать запросы параллельно

08

Асинхронность

Amazon S3 transfer manager with AWS SDK

```
~/.aws/config:
s3 =
  max_concurrent_requests = 10
  multipart_threshold = 12MB
  multipart_chunksize = 5MB
  max_bandwidth = 50MB/s
```



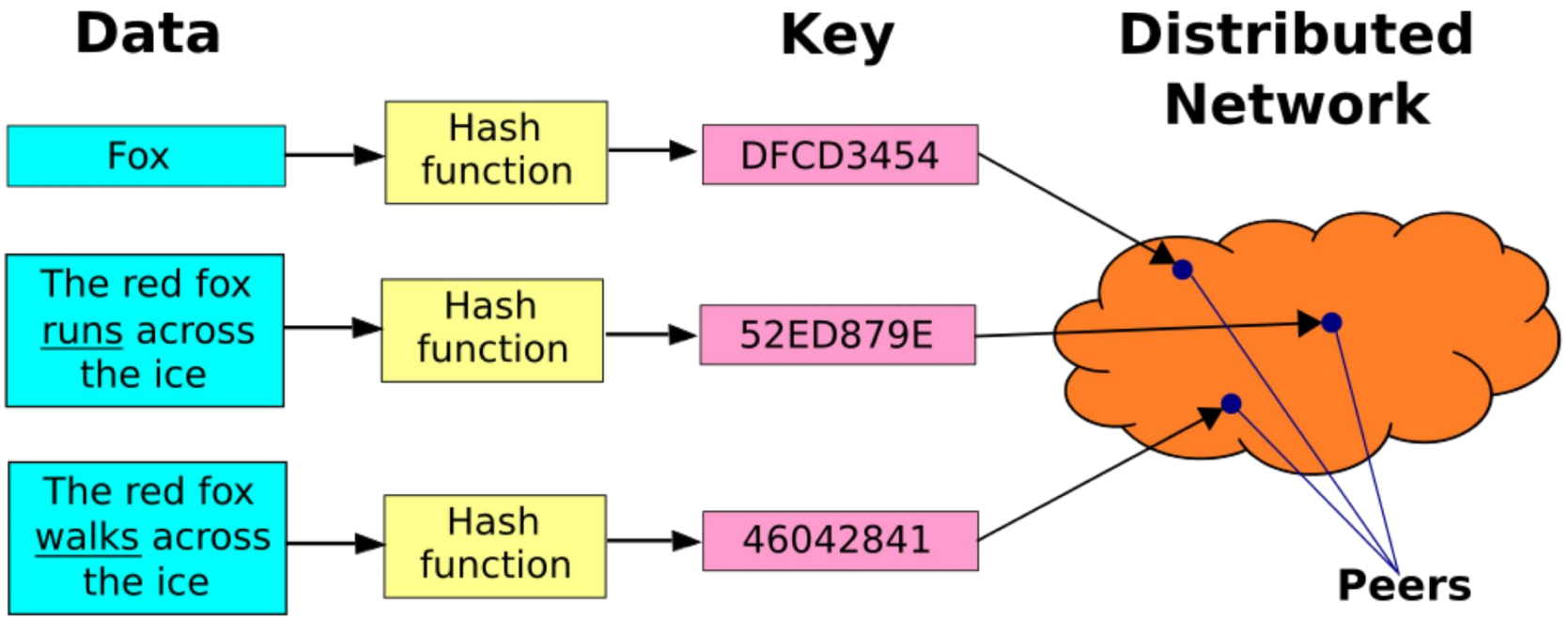
Сравнение распределённых СХД

Система	Модель	Язык	Первый релиз	Метаданные файла	Ребаланс	Число файлов	Общий размер	Слабые стороны
GlusterFs	DFS	C	2005	hashing	более 30 дней	50 млн	4 PB	Прозрачность Масштабируемость

Distributed Hash Table GlusterFS

Consistent Hashing

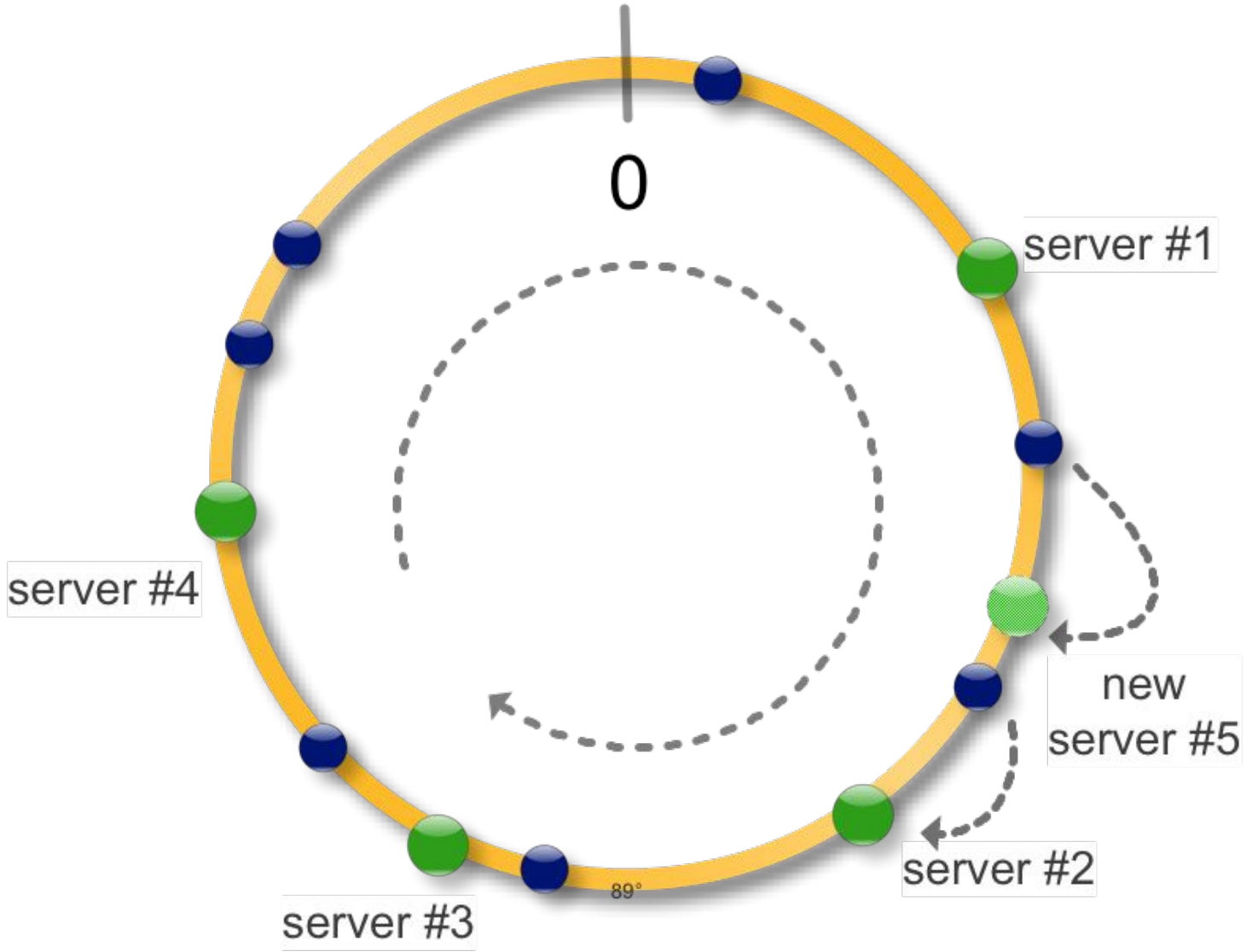
- Все операции управляются клиентами, которые все равны.
- Каталоги существуют на всех подтомах.
- Файлы распределяются по подтомам на основе consistent hashing



Distributed Hash Table GlusterFS

Consistent Hashing

- Все операции управляются клиентами, которые все равны.
- Каталоги существуют на всех подтомах.
- Файлы распределяются по подтомам на основе consistent hashing
- При удалении/добавление brick требуется полная дорогостоящая перебалансировка кластера



Red Hat Bugzilla – Bug 1523258

[GSS] Rebalance slow on large GlusterFS cluster

Keywords:

Status: CLOSED WONTFIX

Alias: None

Product: Red Hat Gluster Storage

Component: distribute

Version: rhgs-3.3

Hardware: x86_64

OS: Linux

Priority: medium

Severity: medium

Reported: 2017-12-07 14:35 UTC by Cal Calhoun

Modified: 2021-06-10 13:51 UTC (History)

CC List: 12 users (show)

Fixed In Version:

Doc Type: If docs needed, set a value

Doc Text:

Clone Of:

Environment:

Last Closed: 2018-04-22 06:32:19 UTC

Embargoed:

Dependent Products:

Bug 1523258 - [GSS] Rebalance slow on large GlusterFS cluster

Description of problem:
4.7 PB volume (16 node x 5 Bricks each one x 60 Disks in Raid 6) running a rebalance volume from 744 hours, the rebalance command status output show this:

```
[root@gs1 ~]# gluster volume rebalance <volume> status
Node Rebalanced-files          size       scanned      failures       skipped               status  run time in h:m:s
---------      -----------   -----------   -----------   -----------   -----------         ------------  --------------
localhost            40893        25.3TB         89076             1         28945        in progress    774:54:21
node15                4772         4.0TB         47724             1         10244        in progress    774:54:20
node6                45146        22.7TB        100728             1         24980        in progress    774:54:20
node4                57755        35.4TB         97283             1         15143        in progress    774:54:20
node12               54851        34.0TB         96159             1         14335        in progress    774:54:20
node3                36124        21.6TB         88480            12         31814        in progress    774:54:20
node11               57502        31.6TB         97654             1         13150        in progress    774:54:20
node10               44509        22.9TB        102635             1         26957        in progress    774:54:20
node9                33127        20.7TB        126724             1         59430        in progress    774:54:20
node8                58226        36.1TB        103005             1         11446        in progress    774:54:20
node2                35714        19.7TB         85995             1         24227        in progress    774:54:20
node5                41277        26.3TB        104782             1         35168        in progress    774:54:20
node14               12916        10.2TB         72550             1         24339        in progress    774:54:20
node13                8940         6.5TB         89816             1         40115        in progress    774:54:20
node16                1816         1.5TB         39147             1          9720        in progress    774:54:20
node7                58457        33.9TB        100447             1         11820        in progress    774:54:20
Estimated time left for rebalance to complete :     6595:53:14
```

```
Version-Release number of selected component (if applicable):
  SERVER VERSIONS:
    OS:      RHEL 6.9
    Kernel:  kernel-2.6.32-696.13.2.el6.x86_64
    Gluster: glusterfs-server-3.8.4-44.el6rhs.x86_64
```

1,5Y+

rebalance to comlpete

Сравнение распределённых СХД

Система	Модель	Язык	Первый релиз	Метаданные файла	Ребаланс	Число файлов	Общий размер	Слабые стороны
GlusterFs	DFS	C	2005	hashing	более 30 дней	50 млн	4 PB	Прозрачность Масштабируемость
Ceph	Object Store	C++	2010	mds + file per object	более 30 дней	50 млн	4 PB	Масштабируемость

Тестирование Ceph

- Высокая стоимость операций по вводу/выводу OSD для соответствия алгоритму CRUSH;
- 2,5 млн файлов по 1кб заняли 310 Гб на 2,5 Гб самих данных;
- Низкая скорость доступа на чтение/запись;

Сравнение распределённых СХД

Система	Модель	Язык	Первый релиз	Метаданные файла	Ребаланс	Число файлов	Общий размер	Слабые стороны
GlusterFs	DFS	C	2005	hashing	более 30 дней	50 млн	4 PB	Прозрачность Масштабируемость
Ceph	Object Store	C++	2010	mds + file per object	более 30 дней	50 млн	4 PB	Масштабируемость
MinIO	Object store	Go	2014	In memory + file per object	-	10 млн	1 PB	Масштабируемость

Сравнение распределённых СХД

Система	Модель	Язык	Первый релиз	Метаданные файла	Ребаланс	Число файлов	Общий размер	Слабые стороны
GlusterFs	DFS	C	2005	hashing	более 30 дней	50 млн	4 PB	Прозрачность Масштабируемость
Ceph	Object Store	C++	2010	mds + file per object	более 30 дней	50 млн	4 PB	Масштабируемость
MinIO	Object store	Go	2014	In memory + file per object	-	10 млн	1 PB	Масштабируемость
Cloud S3	Object Store	-	2006	DB	-	-	-	Невозможность управлять затратами

JBOD60

Clouds S3

цены per month

- Storage 4PB - $60K
- GET - $700
- PUT - $500

x10

от стоимости владения
с учетом амортизации

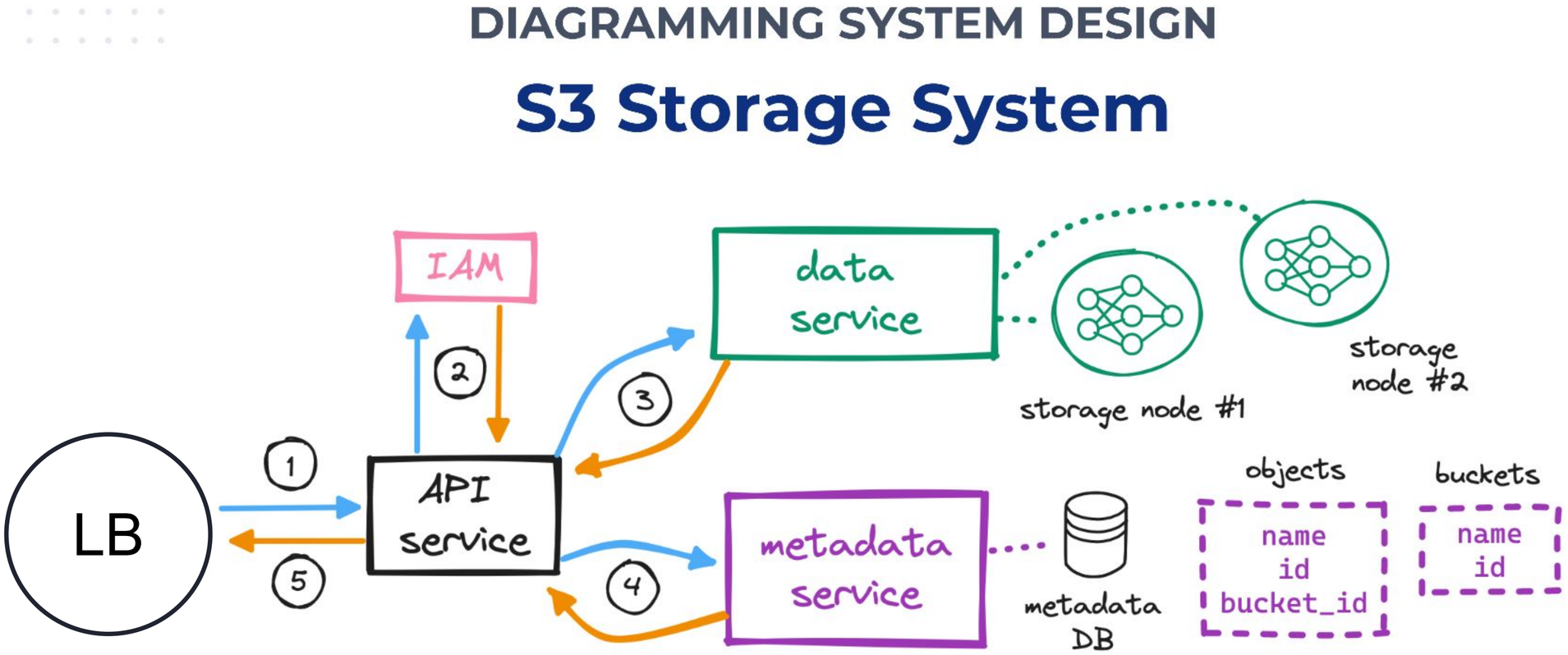
Comparison Table

	Cloudflare R2	Amazon S3
Storage / Month	$0.015 per GB	Starts at $0.023 per GB
Class A Operations (POST)	$0.0045 per 1,000	$0.005 per 1,000
Class B Operations (GET)	$0.00036 per 1,000	$0.0004 per 1,000
Egress Fees	FREE	Starts at $0.09 per GB

Buy Bare metal

0,0015$ per GB

Clouds S3
as web service



Сравнение распределённых СХД

Система	Модель	Язык	Первый релиз	Метаданные файла	Ребаланс	Число файлов	Общий размер	Слабые стороны
GlusterFS	DFS	C	2005	hashing	более 30 дней	50 млн	4 PB	Прозрачность Масштабируемость
Ceph	Object Store	C++	2010	hashing + rules	более 30 дней	50 млн	4 PB	Масштабируемость
MinIO	Object store	Go	2014	In memory + file per object	-	10 млн	1 PB	Масштабируемость
Cloud S3	Object Store	-	2006	-	-	-	-	Невозможность управлять затратами
SeaweedFS	Object Store	Go	2015	any DB	менее 3 дней	на 1 Тб дика 2,5 трлн	100PB	Автоматизация управления

Решение проблемы хранения большого числа файлов в SeaweedFS

Все метаданные файлов и информация о каталогах храним через компонент Filer Store - в распределённой базе данных на диске

Решение проблемы хранения большого числа файлов в SeaweedFS

Все метаданные файлов и информация о каталогах храним через компонент Filer Store - в распределённой базе данных на диске

Уменьшить кол-во объектов за счёт "упаковывания" множества файлов в один том/файл по дизайну Haystack Object Store от FB

USENIX 2009
Finding a needle in Haystack: Facebook's photo storage

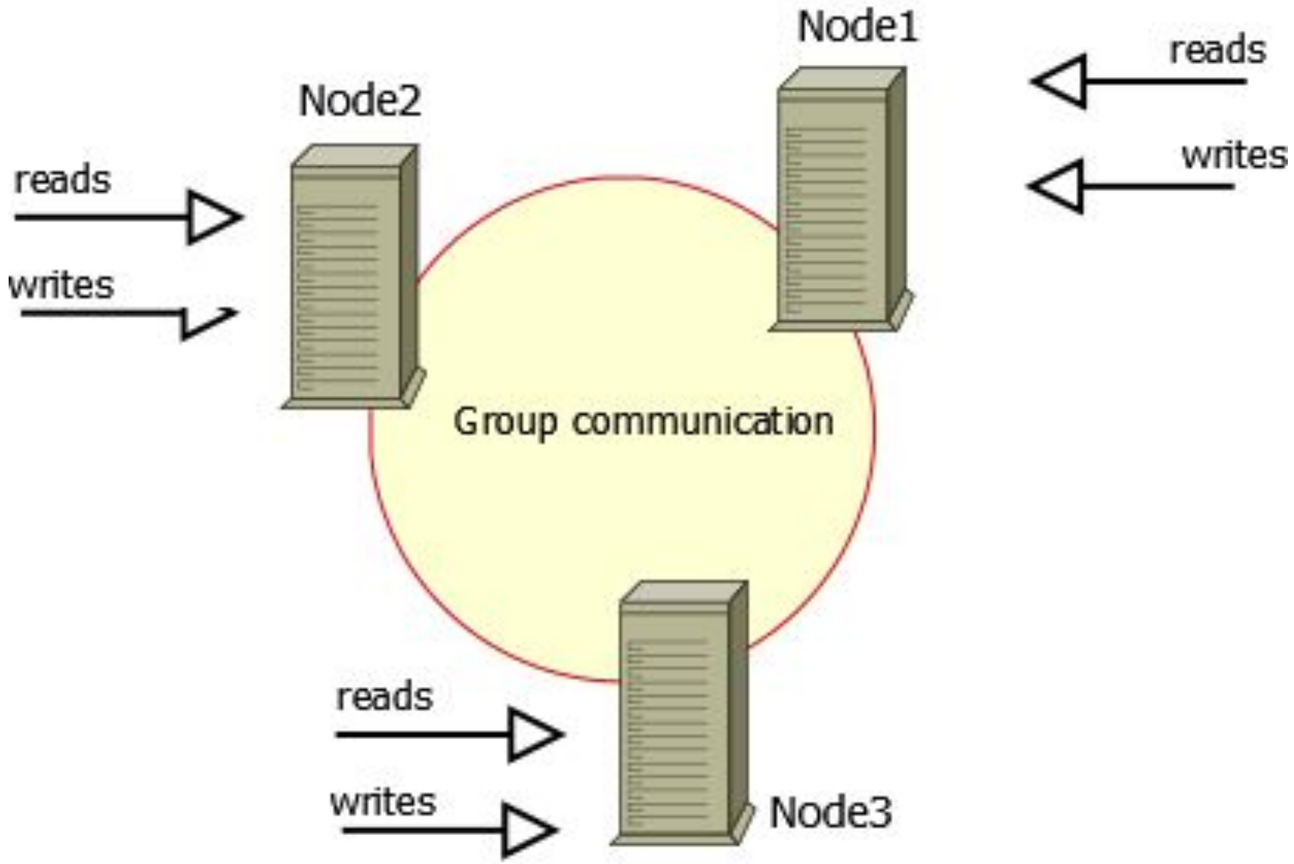
Filer Store (MetaData Service)

Percona Xtradb Cluster

Name	Lookup	Entries in a folder	Directory Renaming	TTL	Note
Mongodb	O(logN)	unlimited		Yes	Easy to manage
Arangodb	O(logN)	unlimited		Native	Easy to manage; Scalable
YDB	O(logN)	unlimited	Atomic	Native	Easy to manage; True elastic Scalability; High Availability.
Redis2	O(1)	limited		Native	one directory's children are stored in one key~value entry
Cassandra	O(logN)	unlimited		Native	
MySql	O(logN)	unlimited	Atomic	Yes	Easy to manage
Postgres	O(logN)	unlimited	Atomic	Yes	Easy to manage
MemSql	O(logN)	unlimited	Atomic	Yes	Scalable
TiDB	O(logN)	unlimited	Atomic	Yes	Scalable
CockroachDB	O(logN)	unlimited	Atomic	Yes	Scalable
YugabyteDB	O(logN)	unlimited	Atomic	Yes	Scalable
Etcd	O(logN)	unlimited		Yes	No SPOF. High Availability. Limited Capacity.
ElasticSearch	O(logN)	unlimited		Yes	Scalable, Searchable. Need to manually build.
HBase	O(logN)	unlimited		Native	Scalable
TiKV	O(logN)	unlimited	Atomic	Yes	Scalable. High Availability. Need to manually build.

Filer Store (MetaData Service)

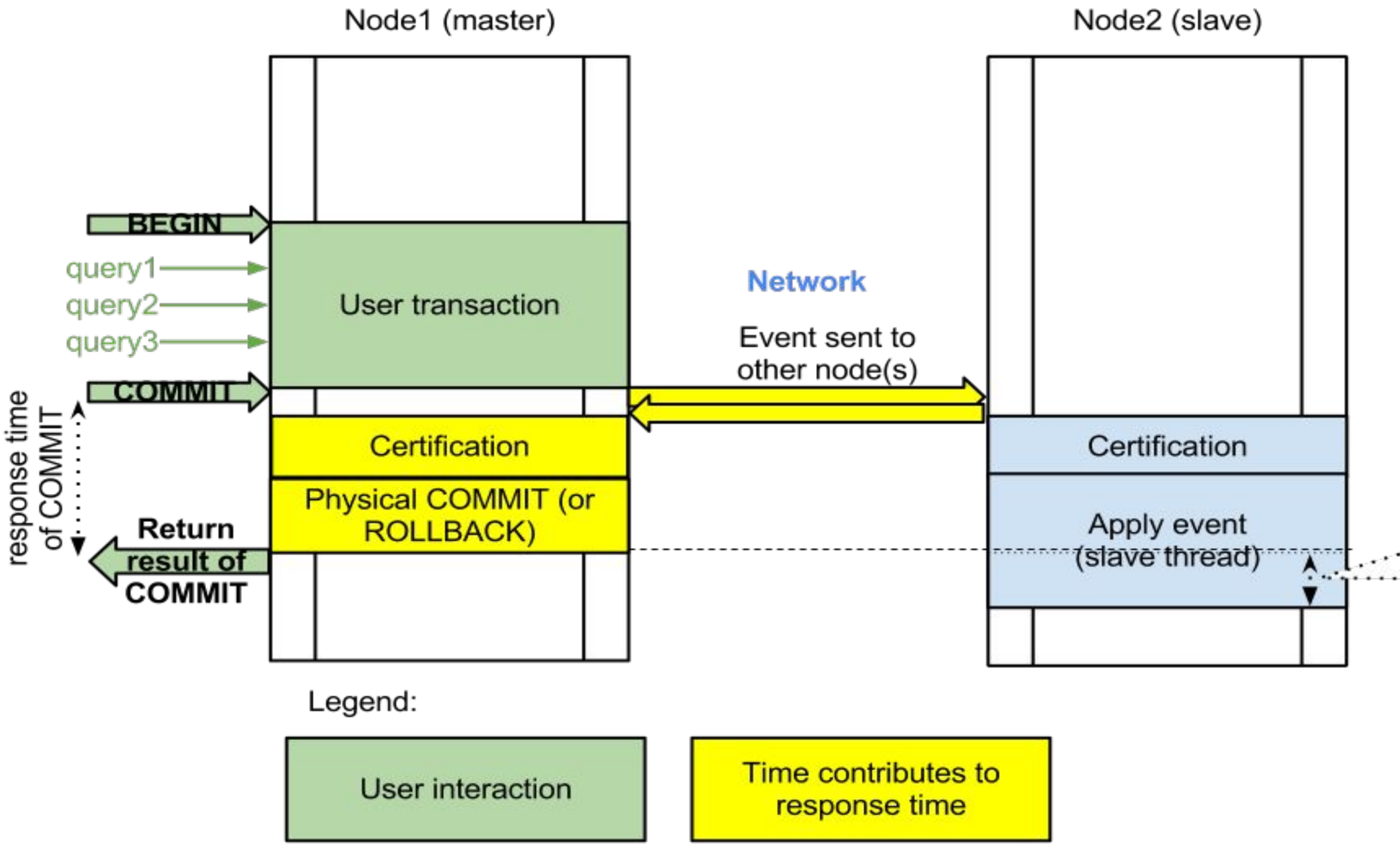
Percona Xtradb Cluster



Высокая доступность, где каждый узел содержит полную копию консистентных данных

Filer Store (MetaData Service)
Percona Xtradb Cluster

Multi-Master даёт возможность писать на любой узел в вашем кластере и не беспокоиться о том, что в конечном итоге вы получите ситуацию рассинхронизации



Filer Store (MetaData Service)
Percona Xtradb Cluster

- Под каждый бакет может создается отдельная табличка с именем бакета

```
CREATE TABLE IF NOT EXISTS `%s` (
  `dirhash`   BIGINT NOT NULL,
  `name`      VARCHAR(766) NOT NULL,
  `directory` TEXT NOT NULL,
  `meta`      LONGBLOB,
  PRIMARY KEY (`dirhash`, `name`)
) DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin;
```

Filer Store (MetaData Service)
Percona Xtradb Cluster

- Под каждый бакет может создается отдельная табличка с именем бакета
- Оптимизация размера индекса за счёт преобразования имени директории big int из md5 хэша

```
CREATE TABLE IF NOT EXISTS `%s` (
  `dirhash`   BIGINT NOT NULL,
  `name`      VARCHAR(766) NOT NULL,
  `directory` TEXT NOT NULL,
  `meta`      LONGBLOB,
  PRIMARY KEY (`dirhash`, `name`)
) DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin;
```

Filer Store (MetaData Service)

Percona Xtradb Cluster

- Под каждый бакет может создается отдельная табличка с именем бакета
- Оптимизация размера индекса за счет преобразования имени директории big int из md5 хэша
- Метаданные хранятся в бинарном protobuf формате

```
CREATE TABLE IF NOT EXISTS `%s` (
  `dirhash`   BIGINT NOT NULL,
  `name`      VARCHAR(766) NOT NULL,
  `directory` TEXT NOT NULL,
  `meta`      LONGBLOB,
  PRIMARY KEY (`dirhash`, `name`)
) DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin;
```

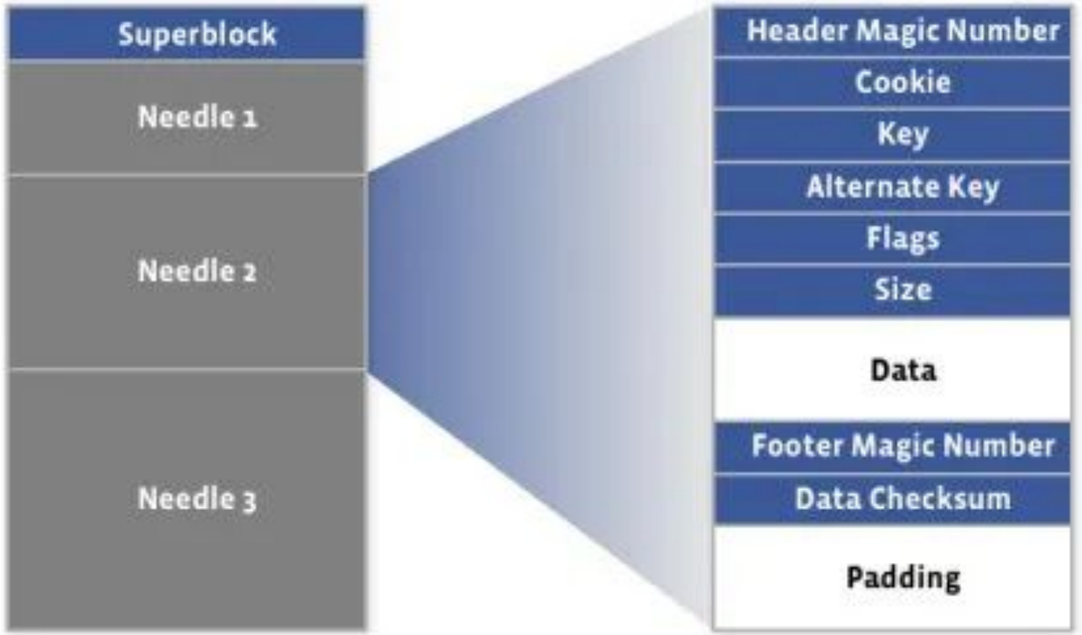
META

```
{
  "name": "filer.conf",
  "isDirectory": false,
  "chunks": [
    {
      "fileId": "1837,036fa85b4db0a357",
      "offset": "0",
      "size": "1166",
      "modifiedTsNs": "1727962916792850551",
      "eTag": "txs72UpjBugBCjyOEAHlng==",
      "sourceFileId": "",
      "fid": {
        "volumeId": 1837,
        "fileKey": "57649243",
        "cookie": 1303421783
      },
      "sourceFid": null,
      "cipherKey": "",
      "isCompressed": false,
      "isChunkManifest": false
    }
  ],
  "attributes": {
    "fileSize": "1166",
    "mtime": "1727962916",
    "fileMode": 432,
     …..
  },
}
```

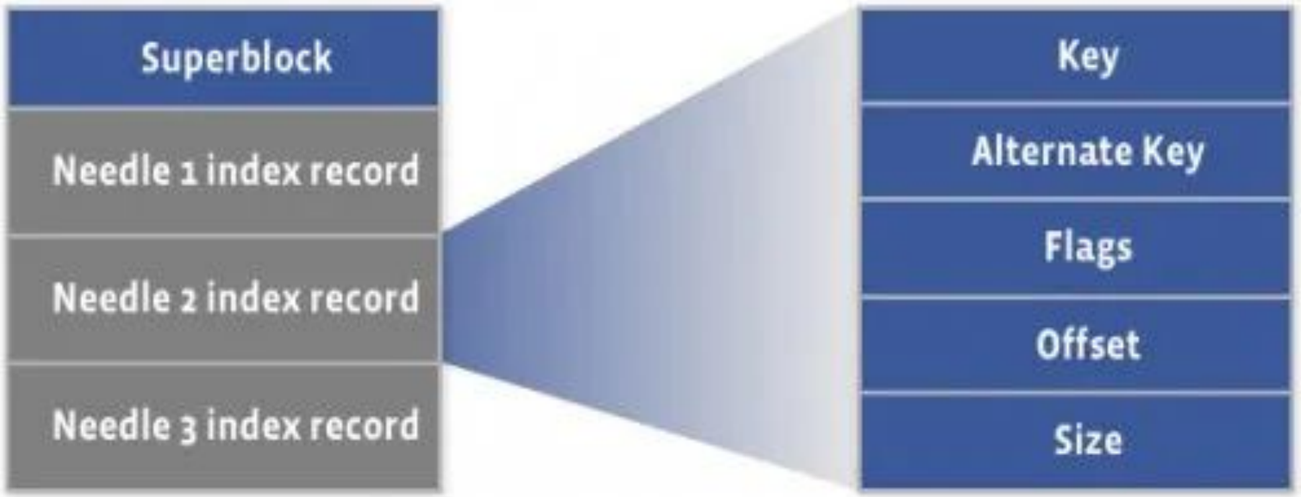
Blob Store (Data Service)

Дизайн от Haystack Object Store

Первые 8 КБ хранилища haystack заняты суперблоком.
Сразу за суперблоком идут иглы,
каждая из которых состоит из заголовка,
данных и нижнего колонтитула



Игла однозначно идентифицируется своим кортежем <Offset, Key, Alternate Key, Cookie>,
где смещение — это смещение иглы
в хранилище haystack.



Blob Store (Data Service)

volume(store) file

```
type Needle struct {  Chris Lu +2
    Cookie Cookie   `comment:"random number to mitigate brute force lookups"`
    Id     NeedleId `comment:"needle id"`
    Size   Size     `comment:"sum of DataSize,Data,NameSize,Name,MimeSize,Mime"`

    DataSize     uint32 `comment:"Data size"` //version2
    Data         []byte `comment:"The actual file data"`
    Flags        byte   `comment:"boolean flags"` //version2
    NameSize     uint8  //version2
    Name         []byte `comment:"maximum 255 characters"` //version2
    MimeSize     uint8  //version2
    Mime         []byte `comment:"maximum 255 characters"` //version2
    PairsSize    uint16 //version2
    Pairs        []byte `comment:"additional name value pairs, json format, maximum 64kB"`
    LastModified uint64 //only store LastModifiedBytesLength bytes, which is 5 bytes to disk
    Ttl          *TTL

    Checksum   CRC    `comment:"CRC32 to check integrity"`
    AppendAtNs uint64 `comment:"append timestamp in nano seconds"` //version3
    Padding    []byte `comment:"Aligned to 8 bytes"`
}
```

Blob Store (Data Service)

volume(store) index

```
type NeedleValue struct {   Chris Lu
    Key    NeedleId
    Offset Offset `comment:"Volume offset"` //since aligned to 8 bytes, range is 4G*8=32G
    Size   Size   `comment:"Size of the data portion"`
}
```

```
./see_idx -volumeId 1837 -collection video

key:22f5488 offset:1 size:927508(905.77 KiB)
key:22f71ac offset:115944 size:735165(717.93 KiB)
key:22f7d47 offset:207848 size:807124(788.21 KiB)
```

```
                offset
    1*8 = 8
115944*8 = 927552
207848*8 = 1662784
```

```
./see_dat -volumeId 1837 -collection video

I1108 00:28:37.078209 volume_loading.go:96 readSuperBlock volume 1436 version 3
I1108 00:28:37.086168 see_dat.go:37 1436,22f548818f06abe offset 8 size 927508(905.77 KiB) cookie 18f06abe
appendedAt 2024-11-07 23:19:11.052636935 +0500 +05 name vo_mu_en_wm_70.mp4
I1108 00:28:37.088818 see_dat.go:37 1436,22f71acf8b3c037 offset 927552 size 735165(717.93 KiB) cookie
f8b3c037 appendedAt 2024-11-07 23:32:30.727045133 +0500 +05 name ol_d15e.mp4
I1108 00:28:37.089157 see_dat.go:37 1436,22f7d47e1a697d6 offset 1662784 size 807124(788.21 KiB) cookie
e1a697d6 appendedAt 2024-11-07 23:37:45.381550513 +0500 +05 nameending_f08b9.mp4
```

Blob Store (Data Service)
с динамической топологией томов

- При увеличение Capacity просто и дёшево добавлять volume сервера

```
Topology volumeSizeLimit:1024 MB hdd(volume:177/1542 active:175 free:1365 remote:0)
  DataCenter dc1 hdd(volume:52/514 active:51 free:462 remote:0)
    Rack DefaultRack hdd(volume:52/514 active:51 free:462 remote:0)
      DataNode 172.22.0.27:8080 hdd(volume:52/514 active:51 free:462 remote:0)
        Disk hdd(volume:52/514 active:51 free:462 remote:0)
          volume id:4024627 size:16960 collection:"webdav" file_count:13 rp:100 version:3 ttl:259 msec:1727955628
          volume id:4024626 size:14640 collection:"webdav" file_count:14 rp:100 version:3 ttl:259 msec:1727955648
          volume id:4024629 size:19656 collection:"webdav" file_count:16 rp:100 version:3 ttl:259 msec:1727955638
        Disk hdd total size:51256 file_count:43
      DataNode 172.22.0.27:8080 total size:51256 file_count:43
    Rack DefaultRack total size:51256 file_count:43
  DataCenter dc1 total size:51256 file_count:43
  DataCenter dc3 hdd(volume:60/514 active:59 free:454 remote:0)
    Rack DefaultRack hdd(volume:60/514 active:59 free:454 remote:0)
      DataNode 172.22.0.28:8080 hdd(volume:60/514 active:59 free:454 remote:0)
        Disk hdd(volume:60/514 active:59 free:454 remote:0)
          volume id:4024627 size:16960 collection:"webdav" file_count:13 rp:100 version:3 ttl:259 msec:1727955628
          volume id:4024628 size:8 collection:"webdav" rp:100 version:3 ttl:259 msec:1727914230
        Disk hdd total size:16968 file_count:13
      DataNode 172.22.0.28:8080 total size:16968 file_count:13
    Rack DefaultRack total size:16968 file_count:13
  DataCenter dc3 total size:16968 file_count:13
total size:102528 file_count:86
```

Blob Store (Data Service)
с динамической топологией томов

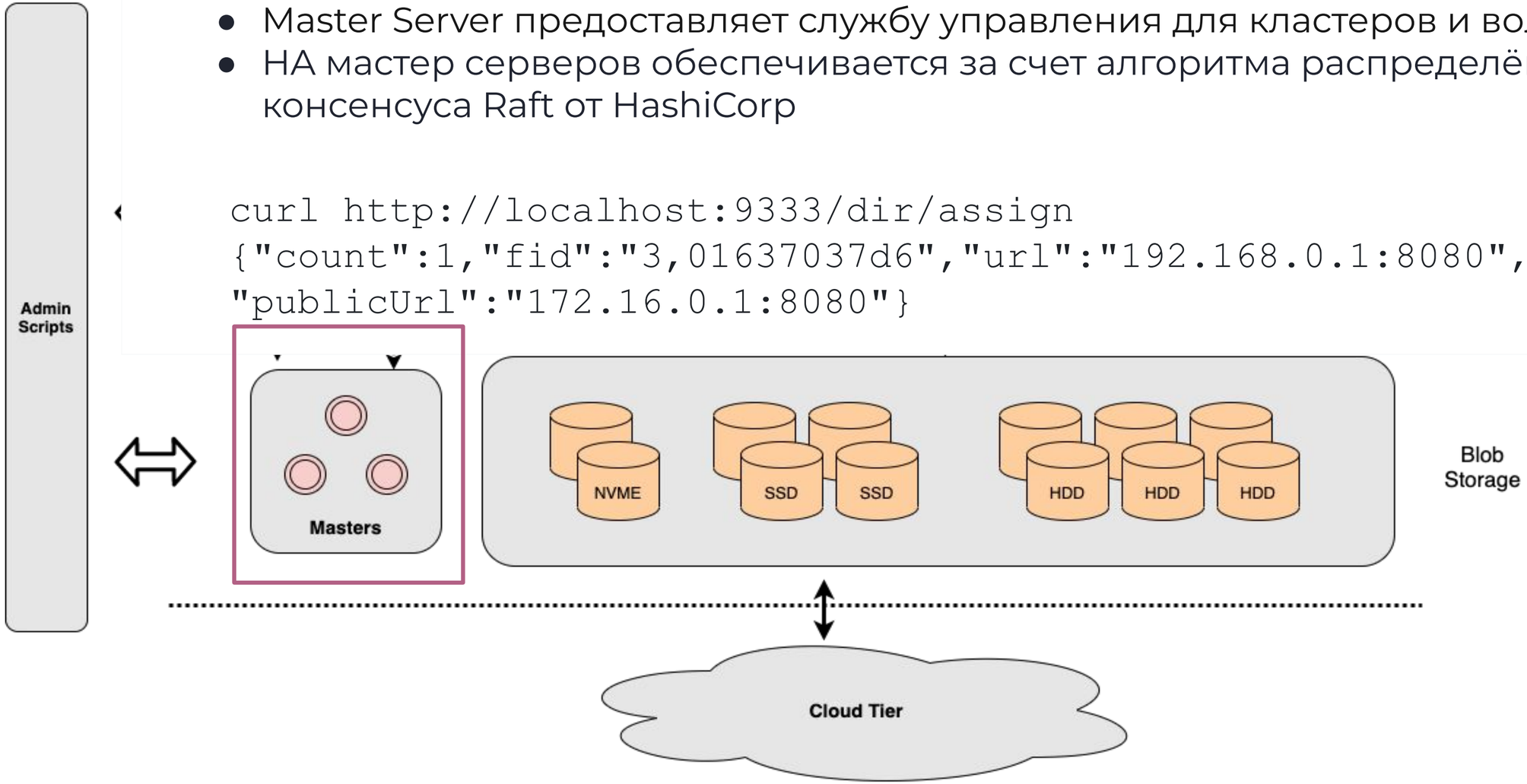
- При увеличение Capacity просто и дёшево добавлять volume сервера
- При ребалансировке тома перемещаются между volume серверами

```
Topology volumeSizeLimit:1024 MB hdd(volume:177/1542 active:175 free:1365 remote:0)
  DataCenter dc1 hdd(volume:52/514 active:51 free:462 remote:0)
    Rack DefaultRack hdd(volume:52/514 active:51 free:462 remote:0)
      DataNode 172.22.0.27:8080 hdd(volume:52/514 active:51 free:462 remote:0)
        Disk hdd(volume:52/514 active:51 free:462 remote:0)
          volume id:4024627 size:16960 collection:"webdav" file_count:13 rp:100 version:3 ttl:259 msec:1727955628
          volume id:4024629 size:19656 collection:"webdav" file_count:16 rp:100 version:3 ttl:259 msec:1727955638
        Disk hdd total size:51256 file_count:43
      DataNode 172.22.0.27:8080 total size:51256 file_count:43
    Rack DefaultRack total size:51256 file_count:43
  DataCenter dc1 total size:51256 file_count:43
  DataCenter dc3 hdd(volume:60/514 active:59 free:454 remote:0)
    Rack DefaultRack hdd(volume:60/514 active:59 free:454 remote:0)
      DataNode 172.22.0.28:8080 hdd(volume:60/514 active:59 free:454 remote:0)
        Disk hdd(volume:60/514 active:59 free:454 remote:0)
          volume id:4024627 size:16960 collection:"webdav" file_count:13 rp:100 version:3 ttl:259 msec:1727955628
          volume id:4024626 size:14640 collection:"webdav" file_count:14 rp:100 version:3 ttl:259 msec:1727955648
          volume id:4024628 size:8 collection:"webdav" rp:100 version:3 ttl:259 msec:1727914230
        Disk hdd total size:16968 file_count:13
      DataNode 172.22.0.28:8080 total size:16968 file_count:13
    Rack DefaultRack total size:16968 file_count:13
  DataCenter dc3 total size:16968 file_count:13
total size:102528 file_count:86
```

Архитектура SeaweedFS
dataservice / blob storage (OSS+MON)

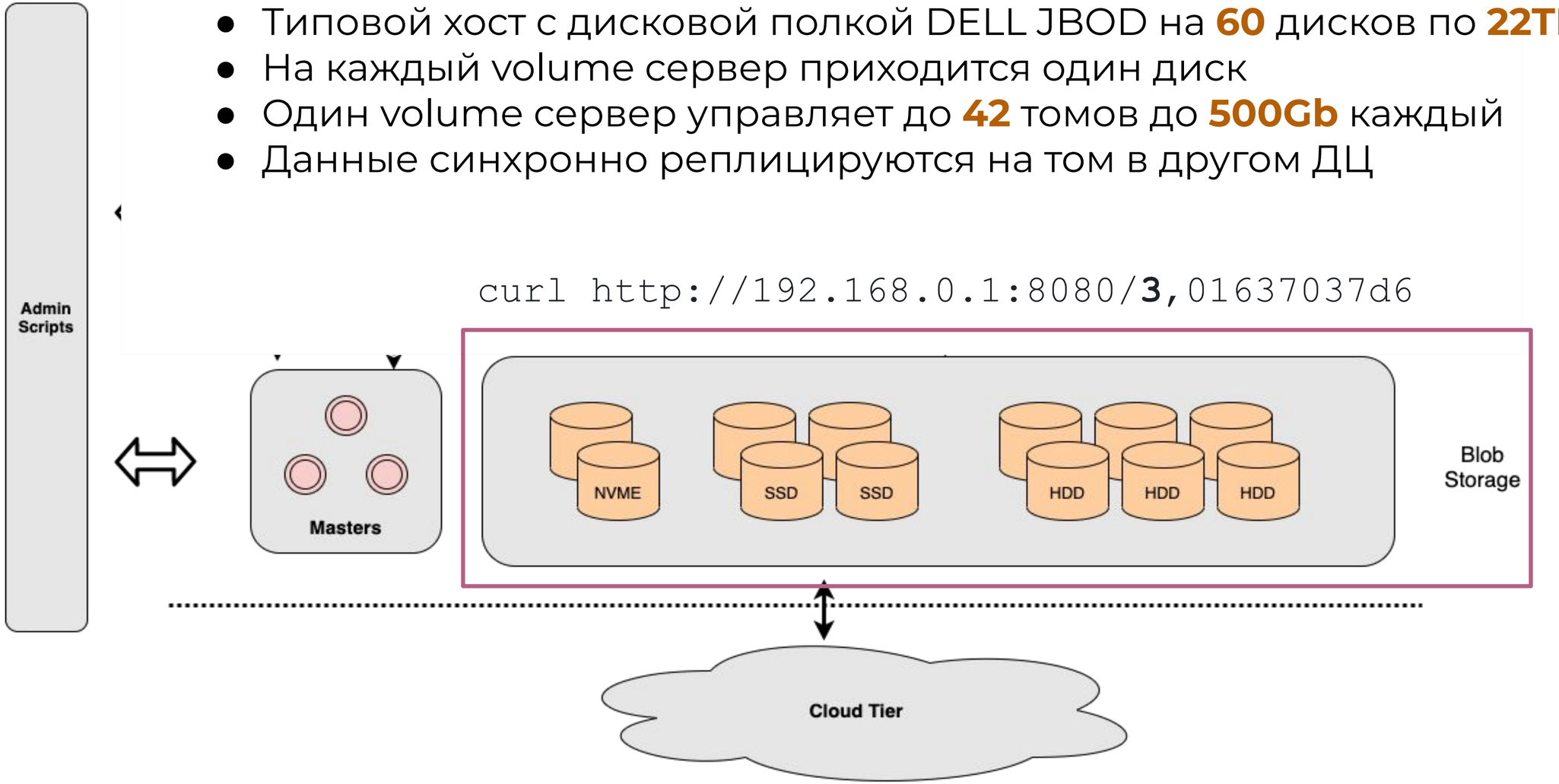
- Master Server предоставляет службу управления для кластеров и волюмов
- HA мастер серверов обеспечивается за счет алгоритма распределённого консенсуса Raft от HashiCorp

```
curl http://localhost:9333/dir/assign
{"count":1,"fid":"3,01637037d6","url":"192.168.0.1:8080",
"publicUrl":"172.16.0.1:8080"}
```



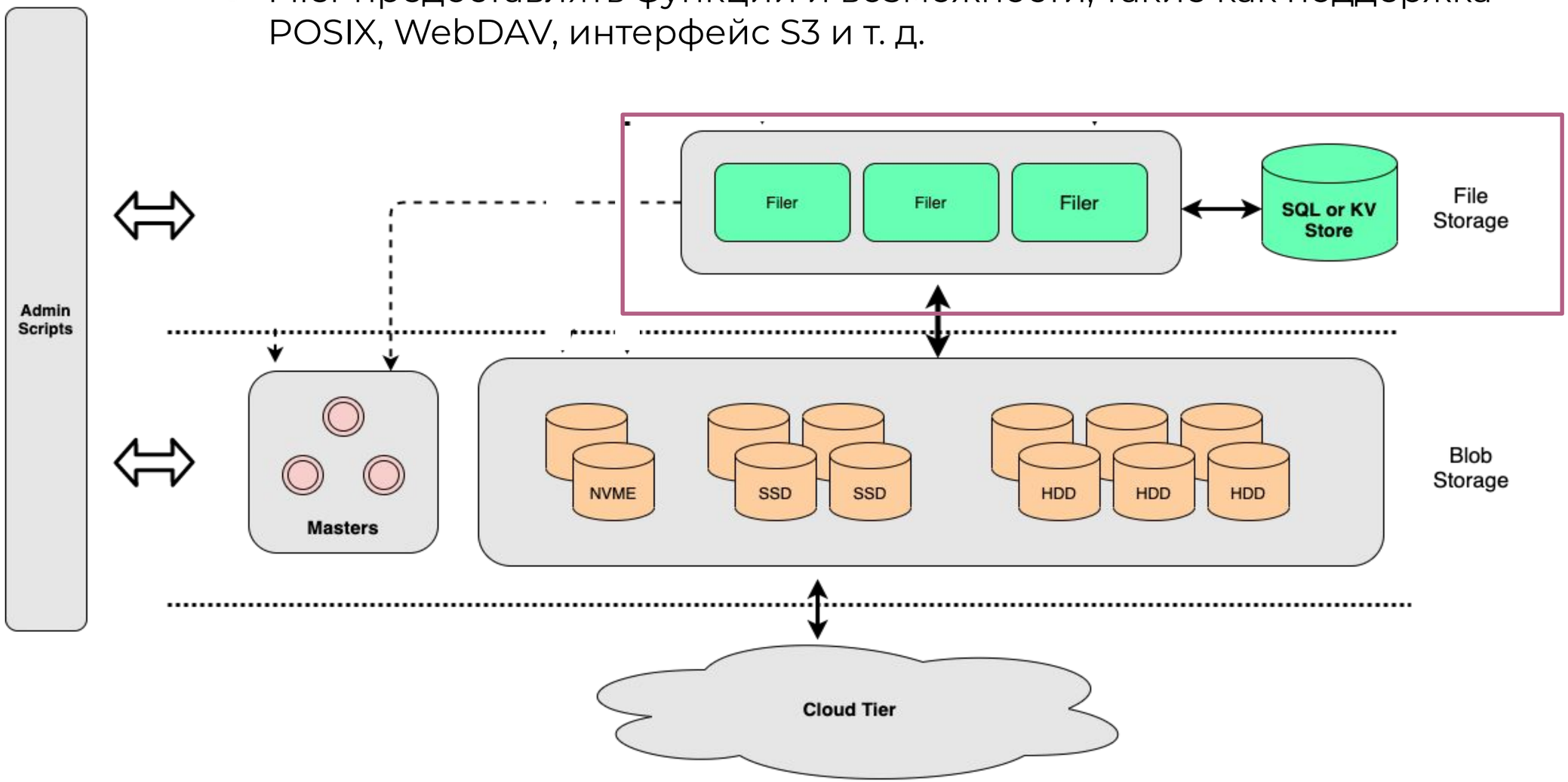
Архитектура SeaweedFS
dataservice / blob storage (OSS+MON)

- Типовой хост с дисковой полкой DELL JBOD на **60** дисков по **22TB**
- На каждый volume сервер приходится один диск
- Один volume сервер управляет до **42** томов до **500Gb** каждый
- Данные синхронно реплицируются на том в другом ДЦ

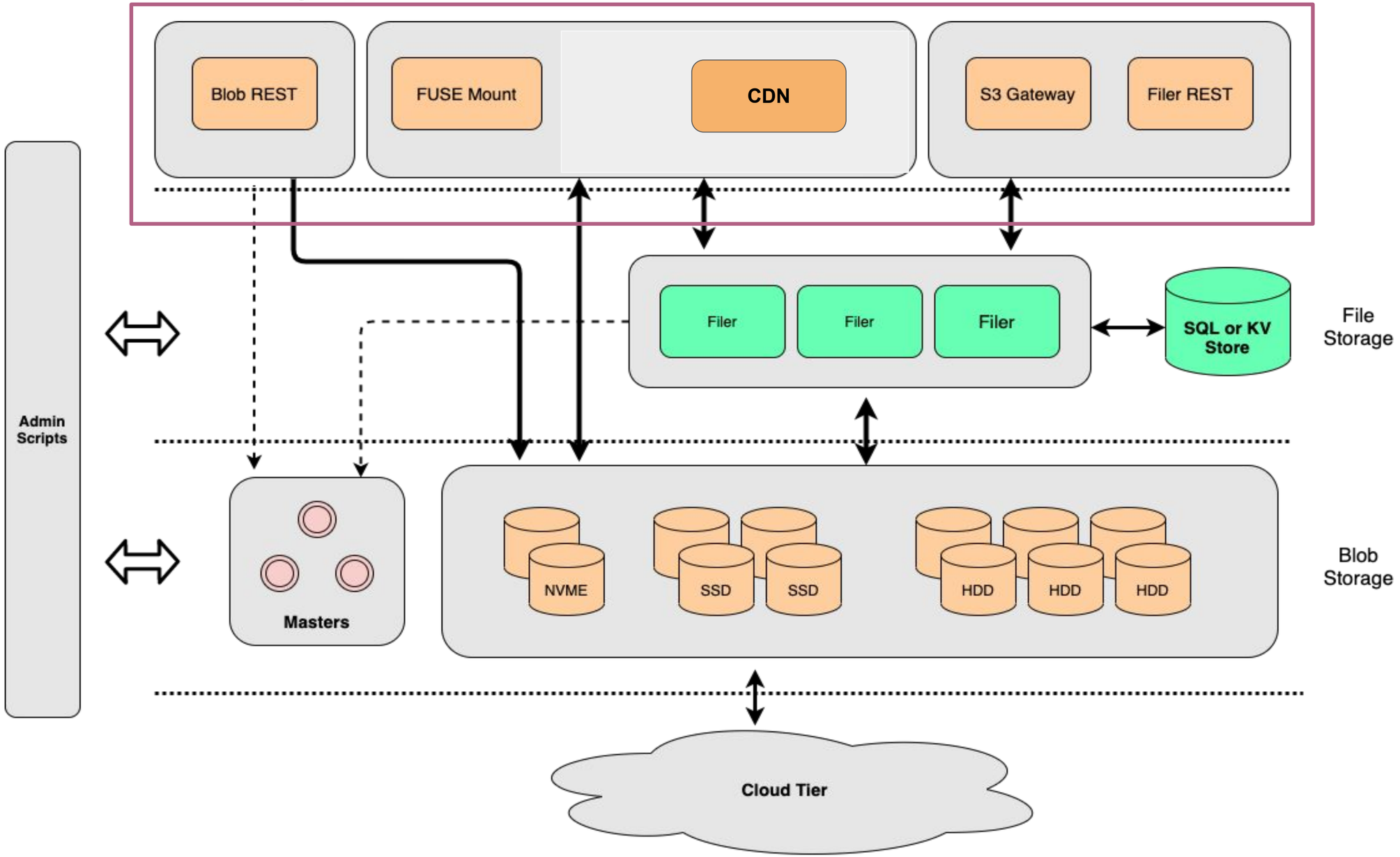


Архитектура SeaweedFS
metadataservice / file storage (MDS)

- Filer предоставлять функции и возможности, такие как поддержка POSIX, WebDAV, интерфейс S3 и т. д.



Архитектура SeaweedFS



Сравнение распределенных СХД

Система	Модель	Язык	Первый релиз	Метаданные файла	Ребаланс	Число файлов	Общий размер	Maintainers
GlusterFS	DFS	C	2005	hashing	более 30 дней	50 млн	4 PB	46 из redhat
Ceph	Object Store	C++	2010	hashing + rules	более 30 дней	50 млн	4 PB	redhat
MinIO	Object store	Go	2014	In memory + file per object	-	10 млн	1 PB	ex-redhat
Cloud S3	Object Store	-	2006	-	-	-	-	-
SeaweedFS	Object Store	Go	2015	any DB	менее 3 дней	на 1 Тб дика 2,5 трлн	100PB	Chris Lu

Community SeaweedFS

github/seaweedfs/seaweedfs

- 22.4K github stars,
- 2.3K forks, 536 watches.
- 314 Github contributor
- 1.6K closed PR
- 2.4 closed issues
- 5M+ Docker pull

@SEAWEEDFS

Maintainer

Chris Lu (Fremont, California, USA) — инженер-программист в Roblox, ранее Uber, Facebook, Salesforce, Oracle и нескольких стартапах. Крис работал над системами хранения данных, реляционными базами данных, графовыми базами данных, федеративными запросами, поиском и хранилищем данных.

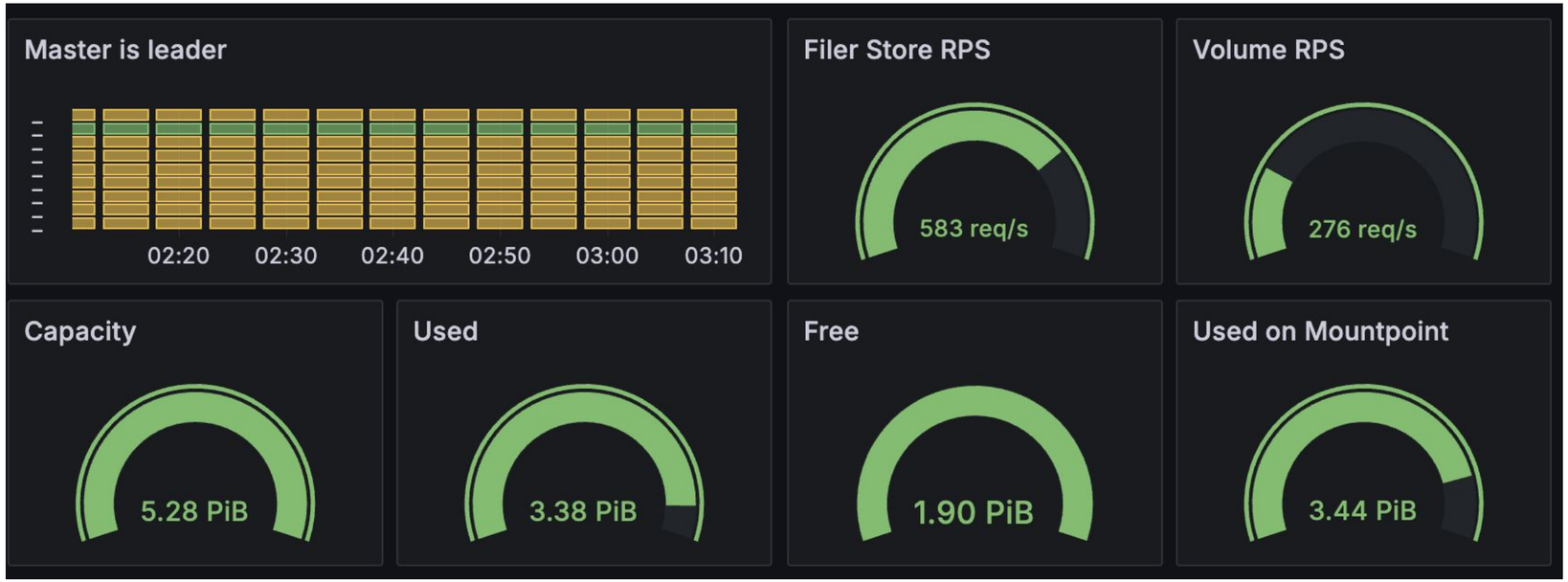
Текущие параметры
одного из наших кластеров

1B+
храним файлов

350G+
размер базы данных

3PB+
составляют данные

0.2TB
загружают в неделю



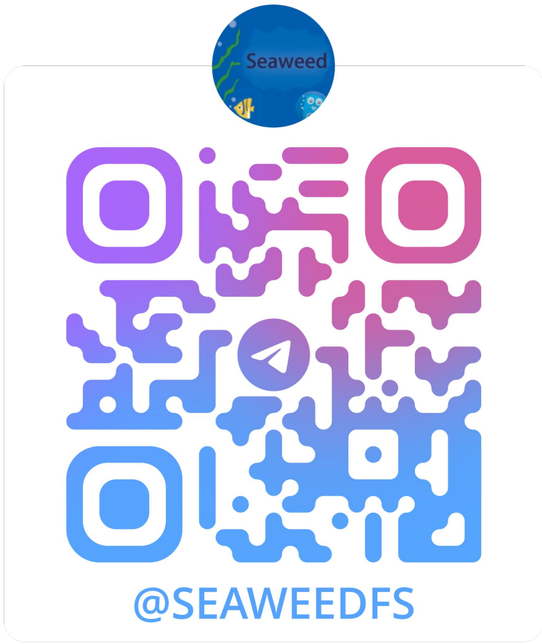
ИТОГ

Как распределённо хранить

триллионы файлов в SeaweedFS

1. Нет ограничений при масштабировании (True elastic Scalability)
2. Архитектура приближена к Cloud S3
3. Оптимизирован под хранение большого числа файлов
4. SeaweedFs вобрала в себя все лучшие решения
5. Открытый код с более чем 300 контрибьюторами

https://github.com/seaweedfs/seaweedfs



Ваши ВОПРОСЫ

mayflower.work

Лебедев Константин - DevOps Engineer