# Swift架构与实践

@SinaAppEngine

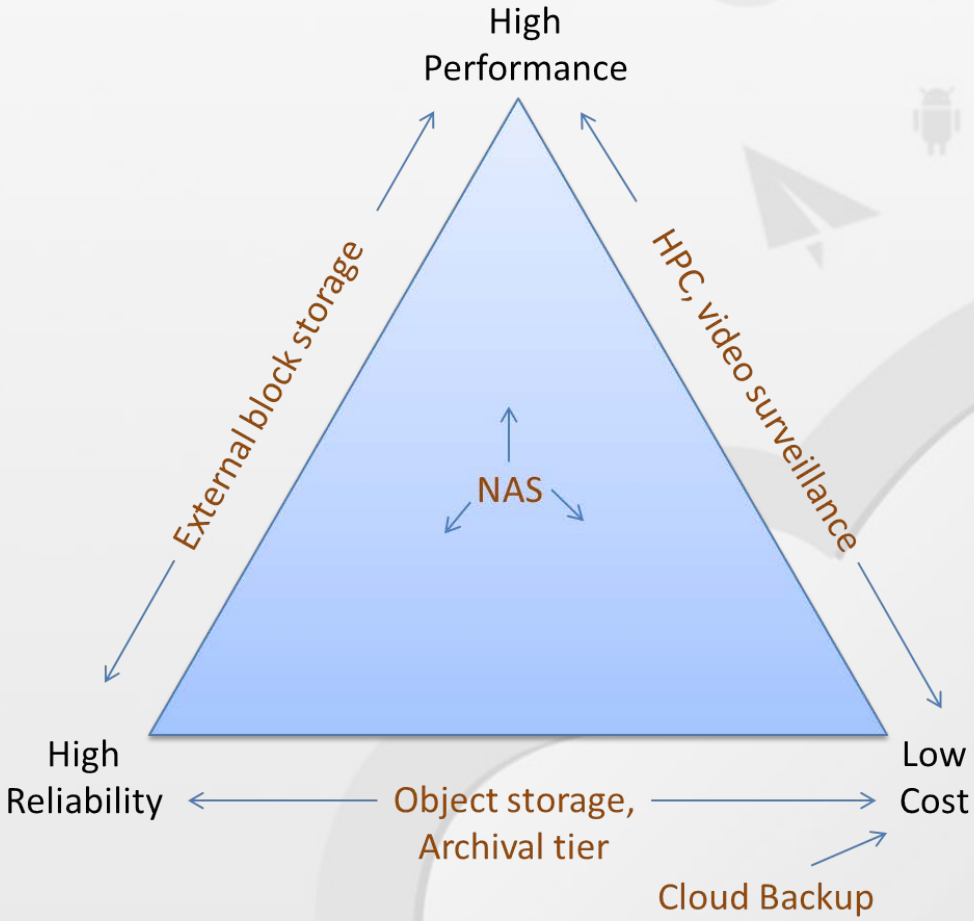
杨雨/2012-08-11

# Content

# Storage Types

| *Types* | *Protocol* | *Application* |
| --- | --- | --- |
| ***Block Storage*** | ***SATA, SCSI, iSCSI*** | ***SAN, NAS, EBS*** |
| ***File Storage*** | ***Ext3/4, XFS, NTFS*** | ***PC, Servers, NFS*** |
| ***Object Storage*** | ***HTTP, REST*** | ***Amazon S3, Google Cloud Storage, Rackspace Cloud Files*** |
| ***Specific Storage*** | ***Specific protocol based on tcp*** | ***MySQL, MongoDB, HDFS*** |

# Storage Types



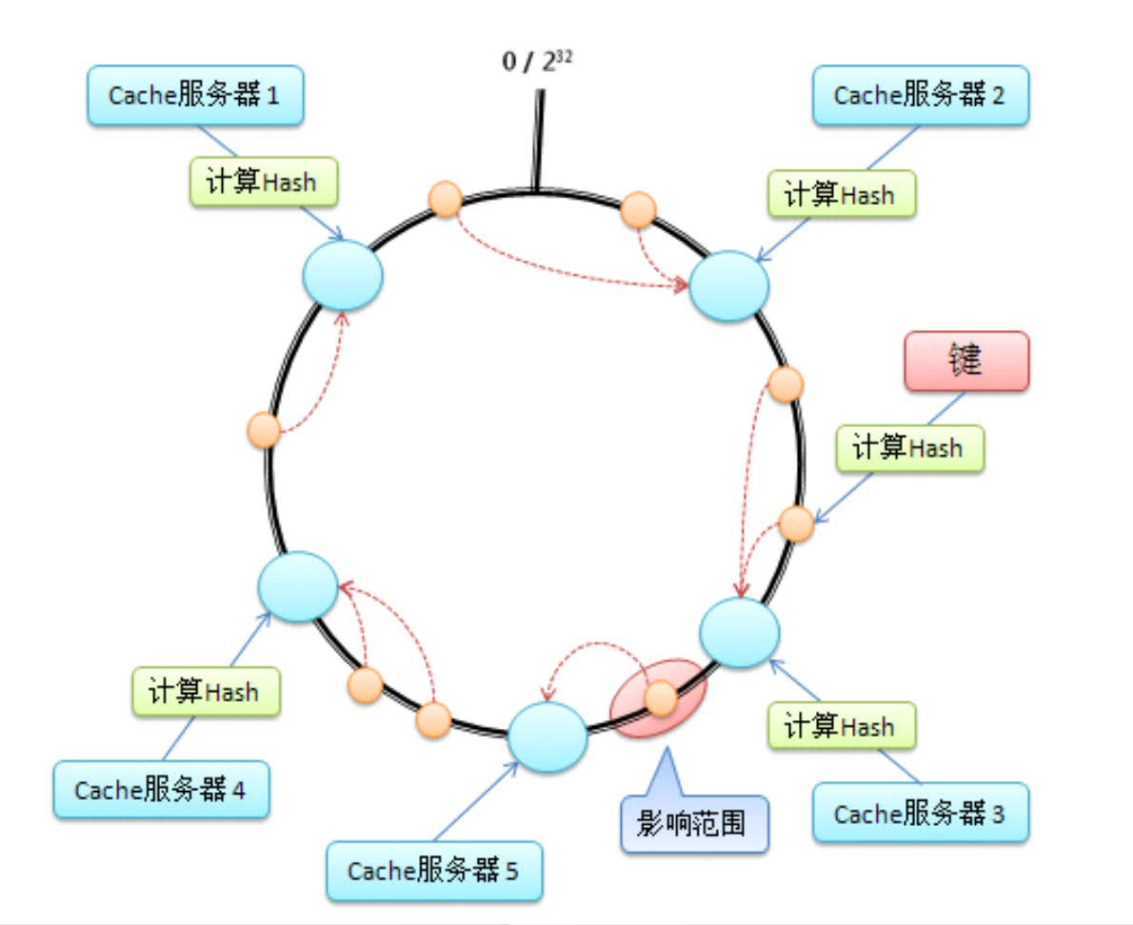

From: http://www.buildcloudstorage.com/2012/08/is-openstack-swift-reliable-enough-for.html

# Targets

- High Reliability = High Durability + High Availability
- High Durability - Replicas and Recovery
- High Availability - Replicas and Partition
- Low Cost - Commodity Hardware
- Scale Out - No Single Bottleneck, Share Nothing

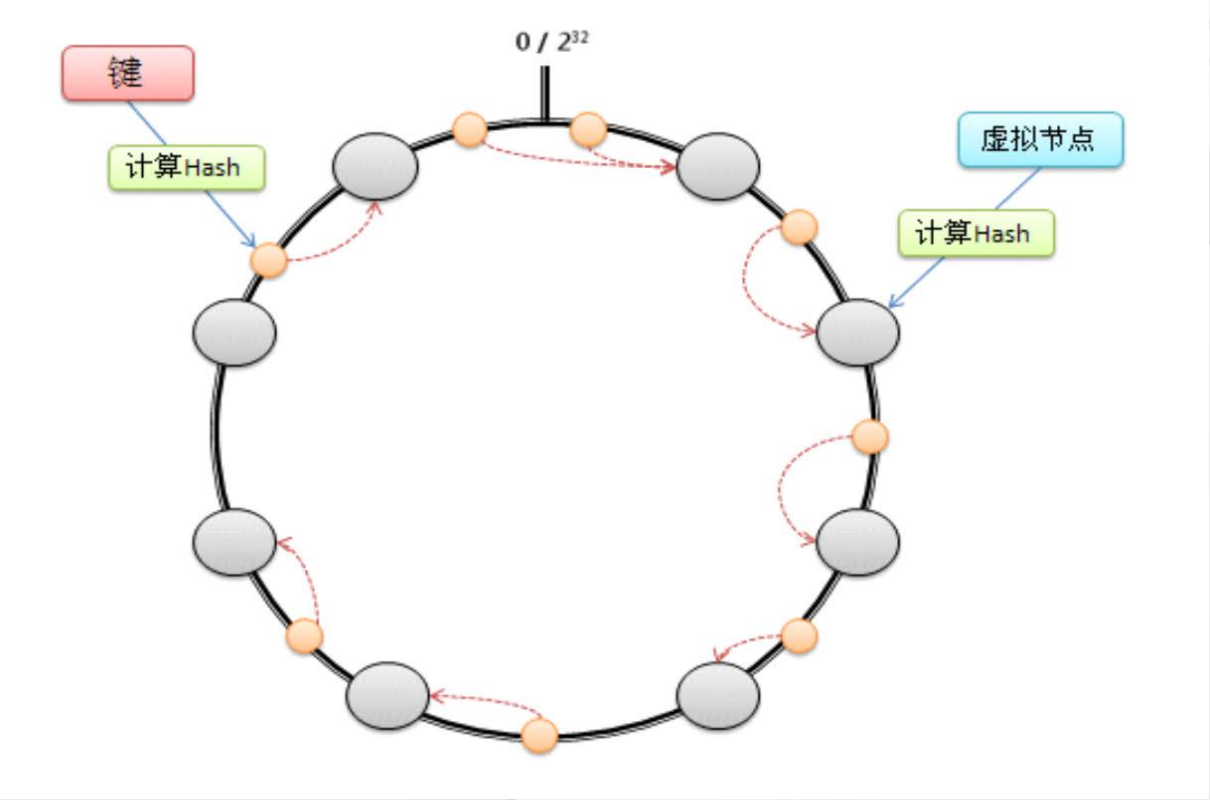
# Reliability

## Consistent Hashing



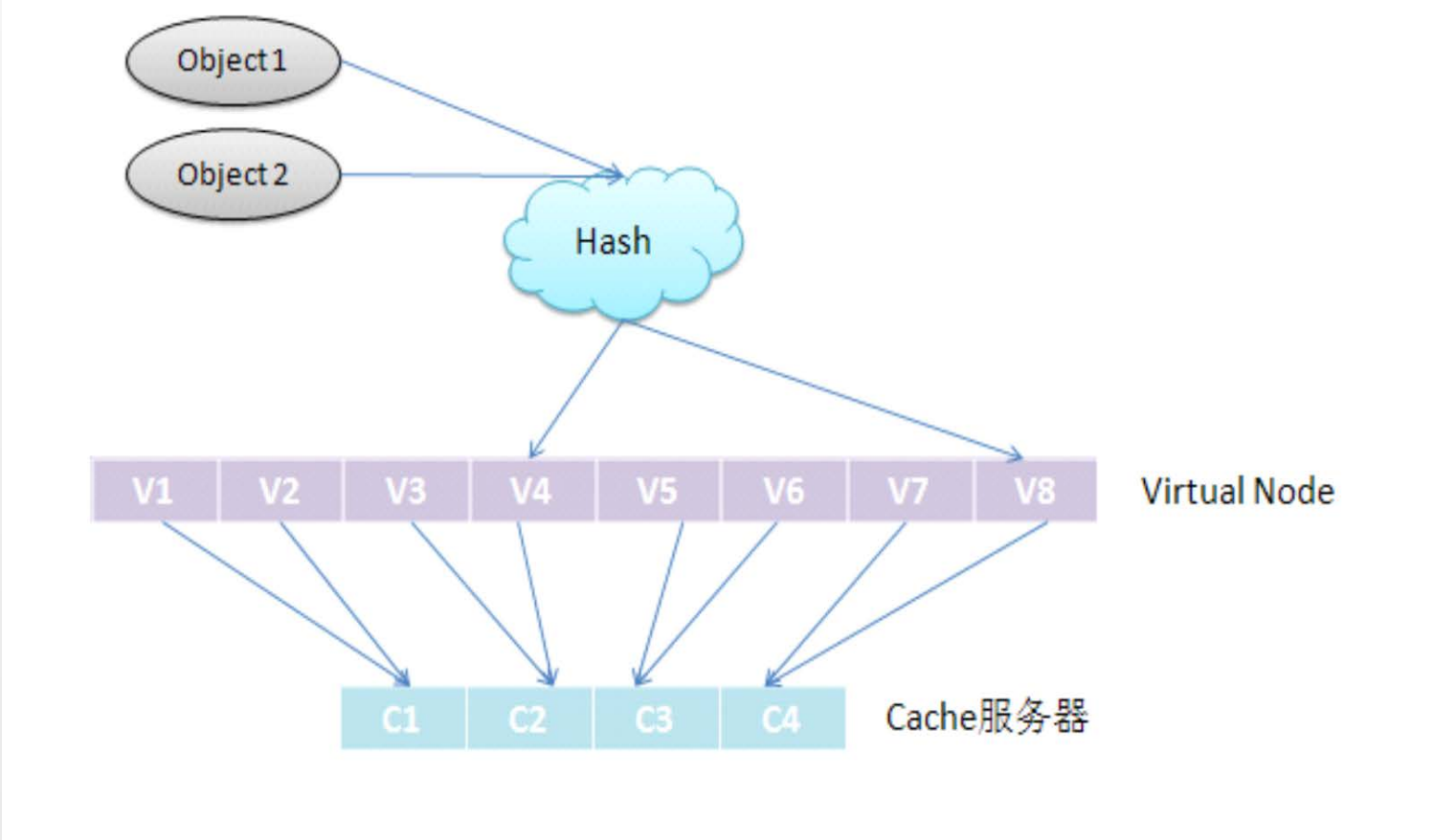
# Reliability

## Consistent Hashing with Virtual Node



# Reliability

## Consistent Hashing with Virtual Node



# Reliability

The advantages of consistent hashing:

1.Metadata is small;(AWS S3 has 762 billion objects)
2.Distribution unformity;
3.Peer to perr comunication;
4.Load blance.

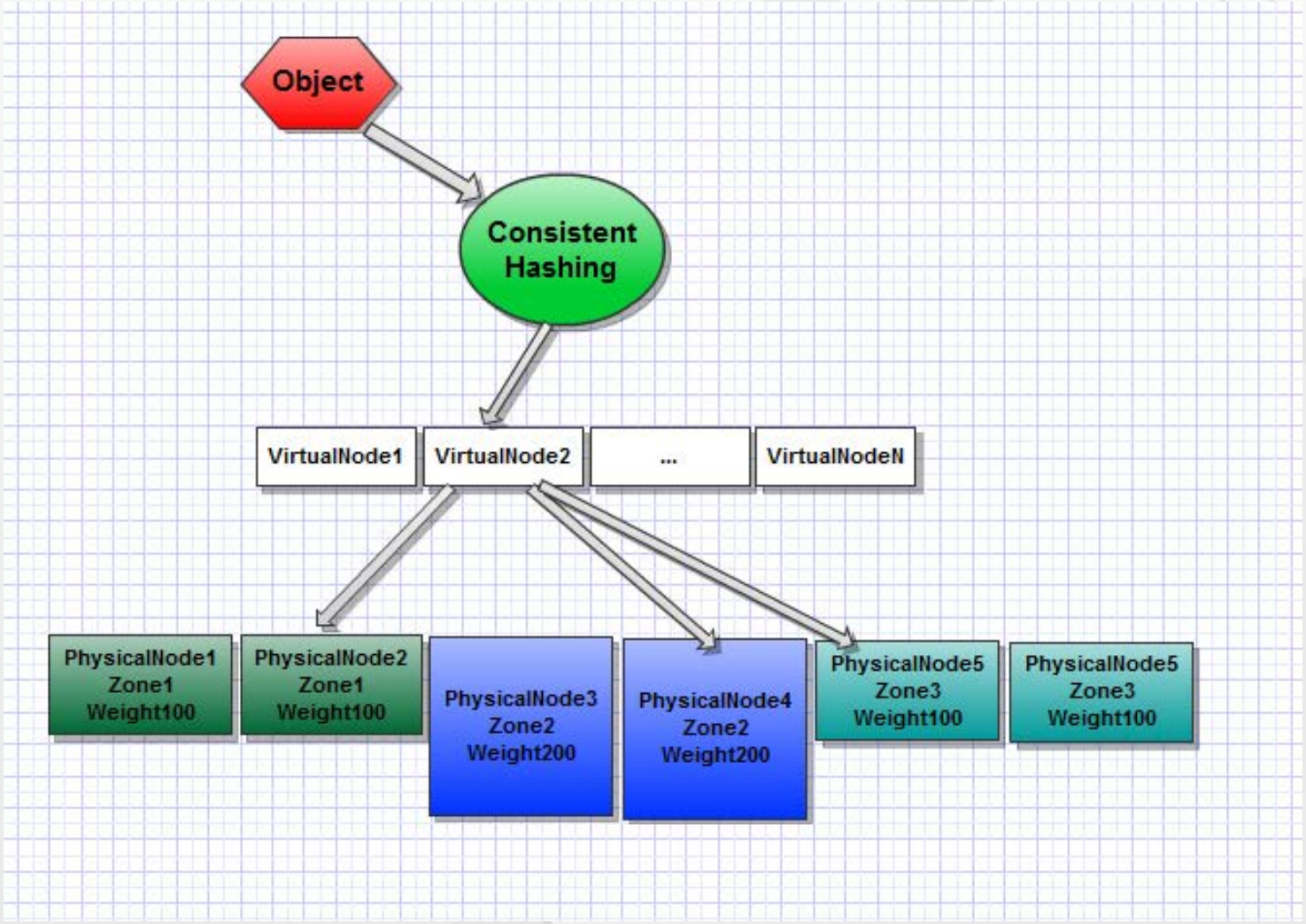
# Reliability

- Virtual node(partition) - Object distribution uniformity
- Weight - Allocate partitions dynamically
- Zone - Partition Tolerance

Zones can be used to group devices based on physical locations, power separations, network separations, or any other attribute that would lessen multiple replicas being unavailable at the same time.

Swift :
Ring - The index file for locating object in the cluster.

# Reliability



# Reliability

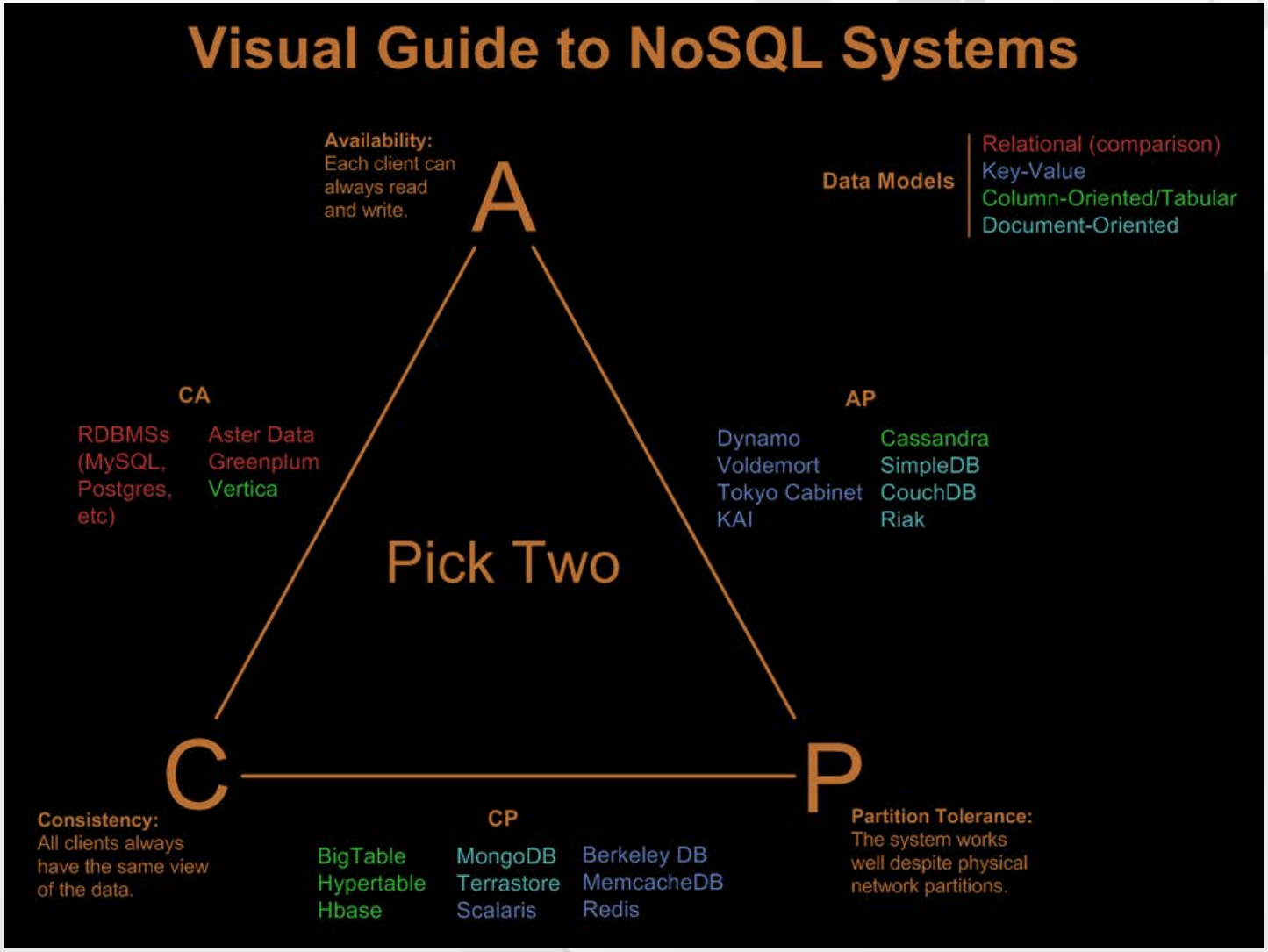
## Durability: 99.999..9%? AWS S3 with 99.999999999%

- Annualized failure rate of devices.(disk, network...)
- Bit rot rate and the time to detect bit rot.
- Mean time to recovery.
- More about durability :

http://www.buildcloudstorage.com/2012/08/is-openstack-swift-reliable-enough-for.html

Swift :
Auditor - To detect bit rot;
Replicator-To keep the consistency of object;

# Consistency Model



# Consistency Model

Quroum + Object Version + Async Replicatiton

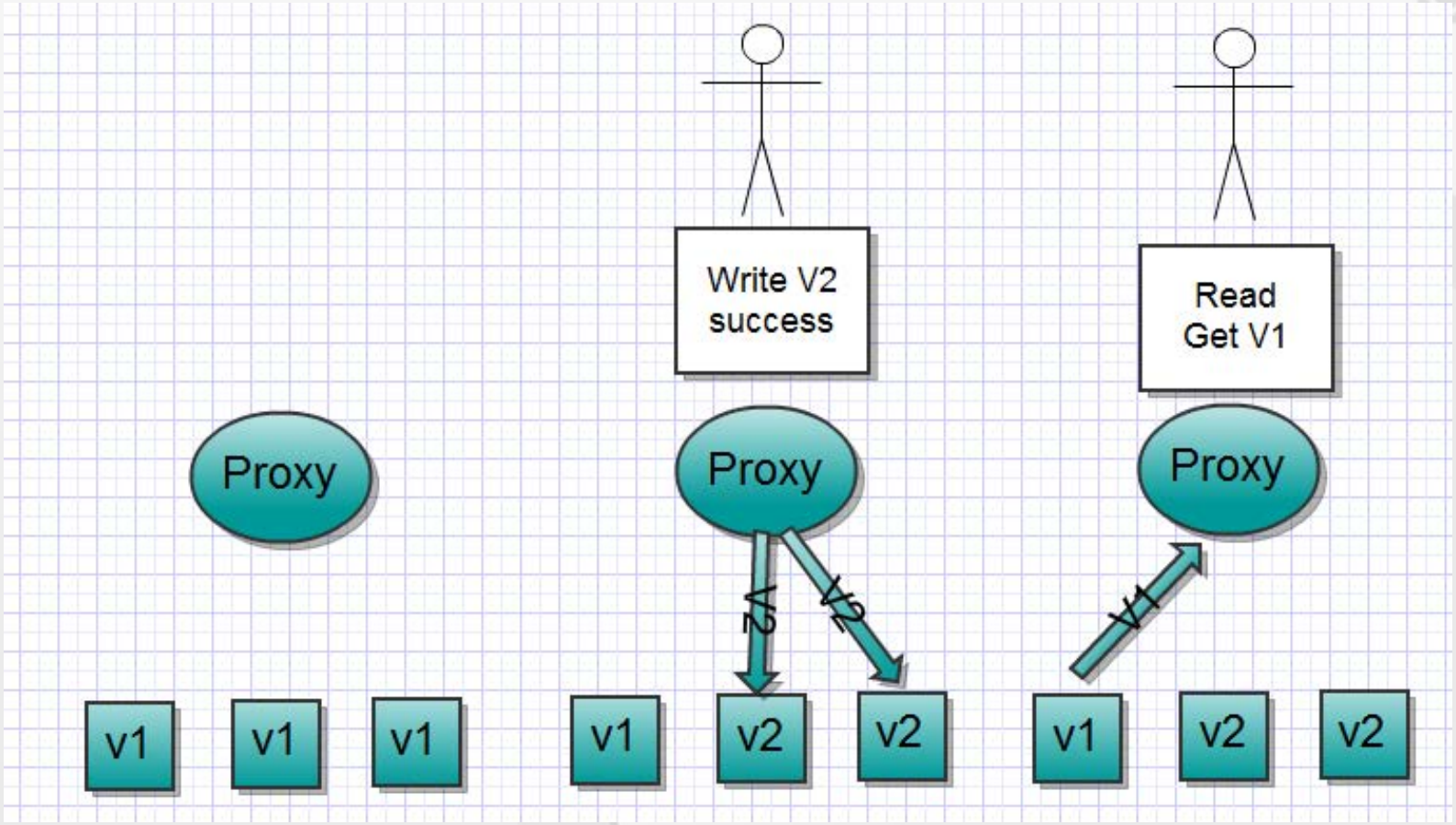
Quroum Protocol:
N, the number of nodes that store replicas of the data;
W, the number of replicas that need to acknowledge the receipt of the update before the update completes;
R, the number of replicas that are contacted when a data object is accessed through a read operation;

If W+R>N, then the write set and the read set always overlap and one can guarantee strong consistency.

In Swift, NWR is configurable.
General configuration: N=3, W=2, R=1 or 2, So the swift can providetwo models of consistency, strong and eventual.

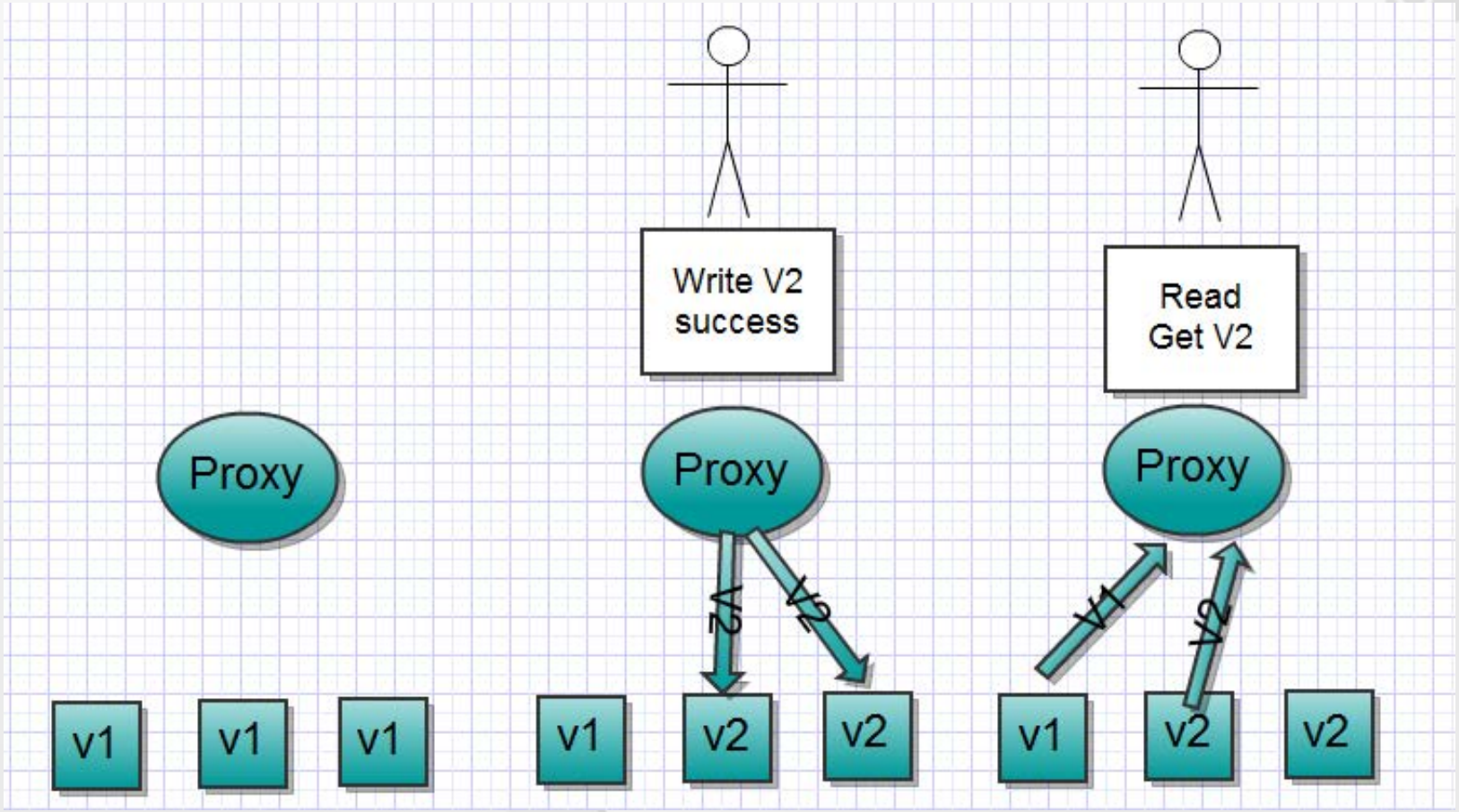
# Consistency Model

Weak Consistency ( N=3,W=2,R=1 )



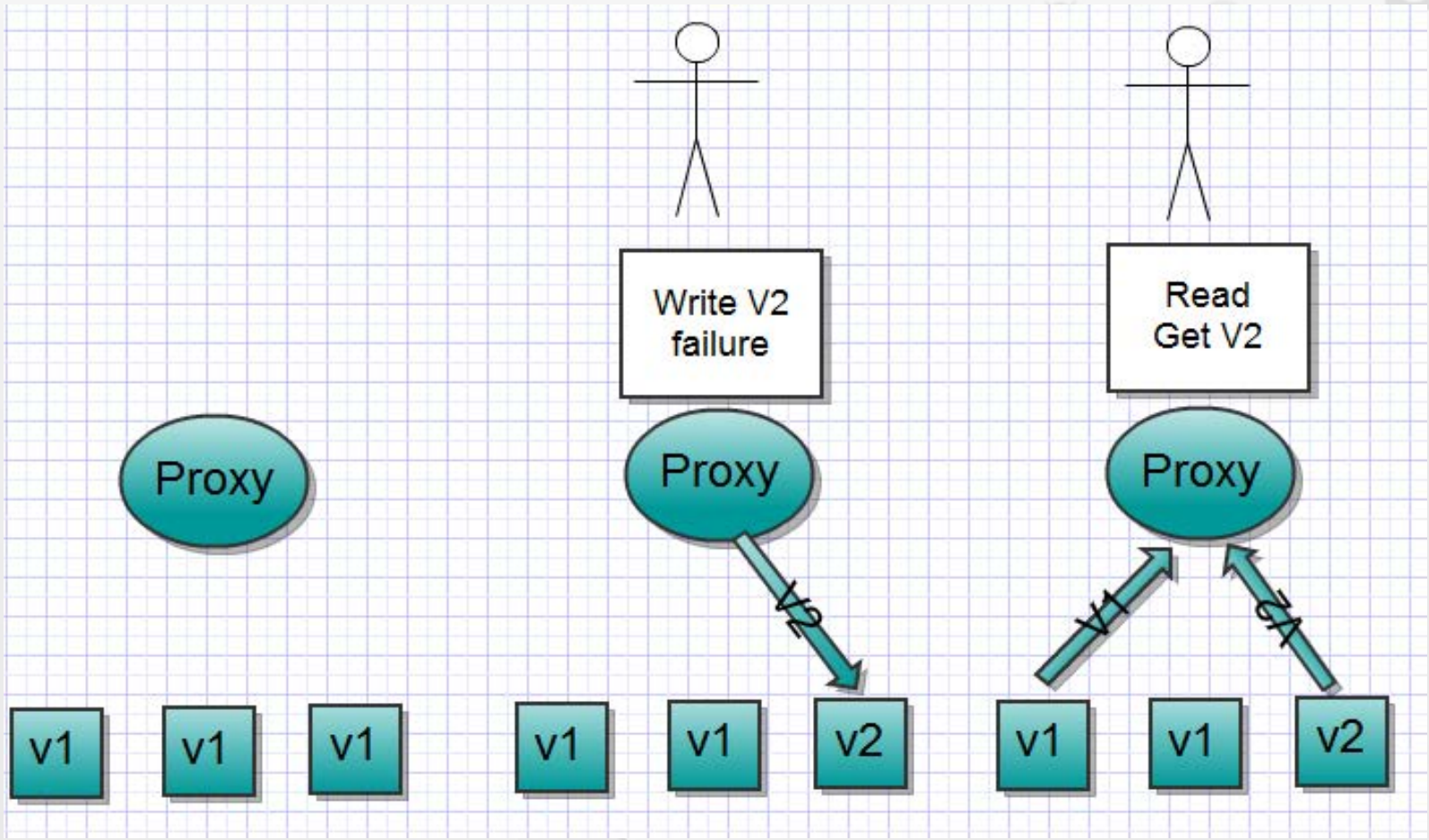
# Consistency Model

Strong Consistency（N=3,W=2,R=2）

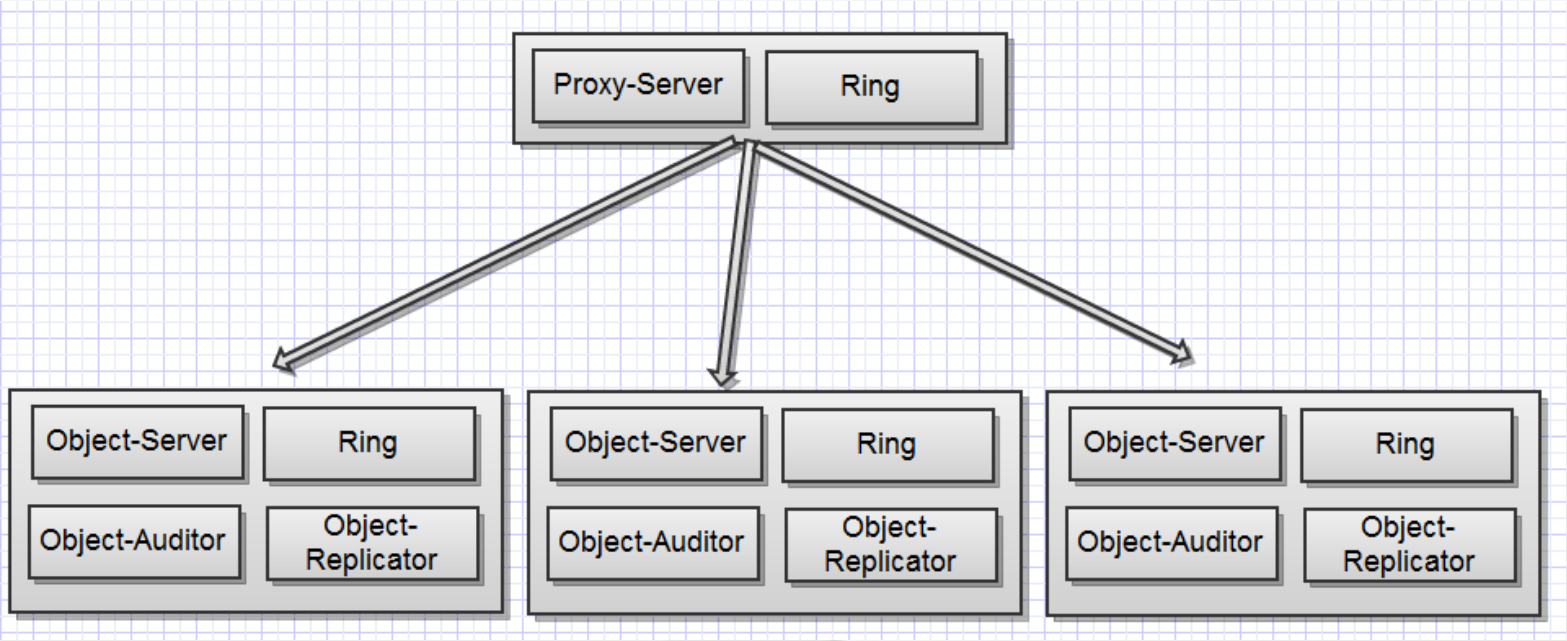


# Consistency Model

Special Scene： dirty read



# Architecture Prototype



# Metadata

account
|-- container1
|------obj1
|------obj2
|-- container2
|------objN

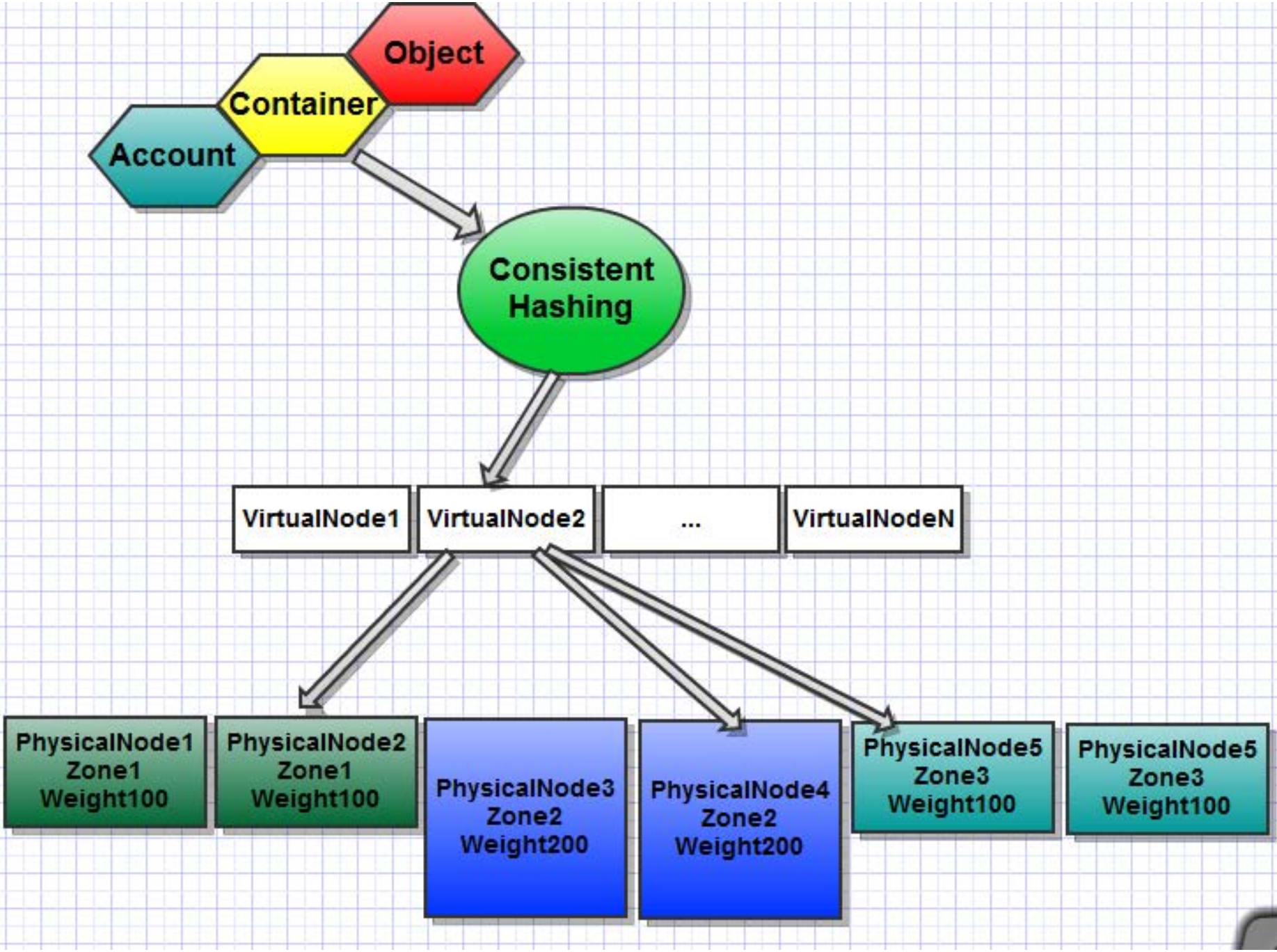
How to store the relationship of account, container and object?

- Relation Database
- NoSQL, Cassandra, MongoDB
- Relation Datatbase with Sharding

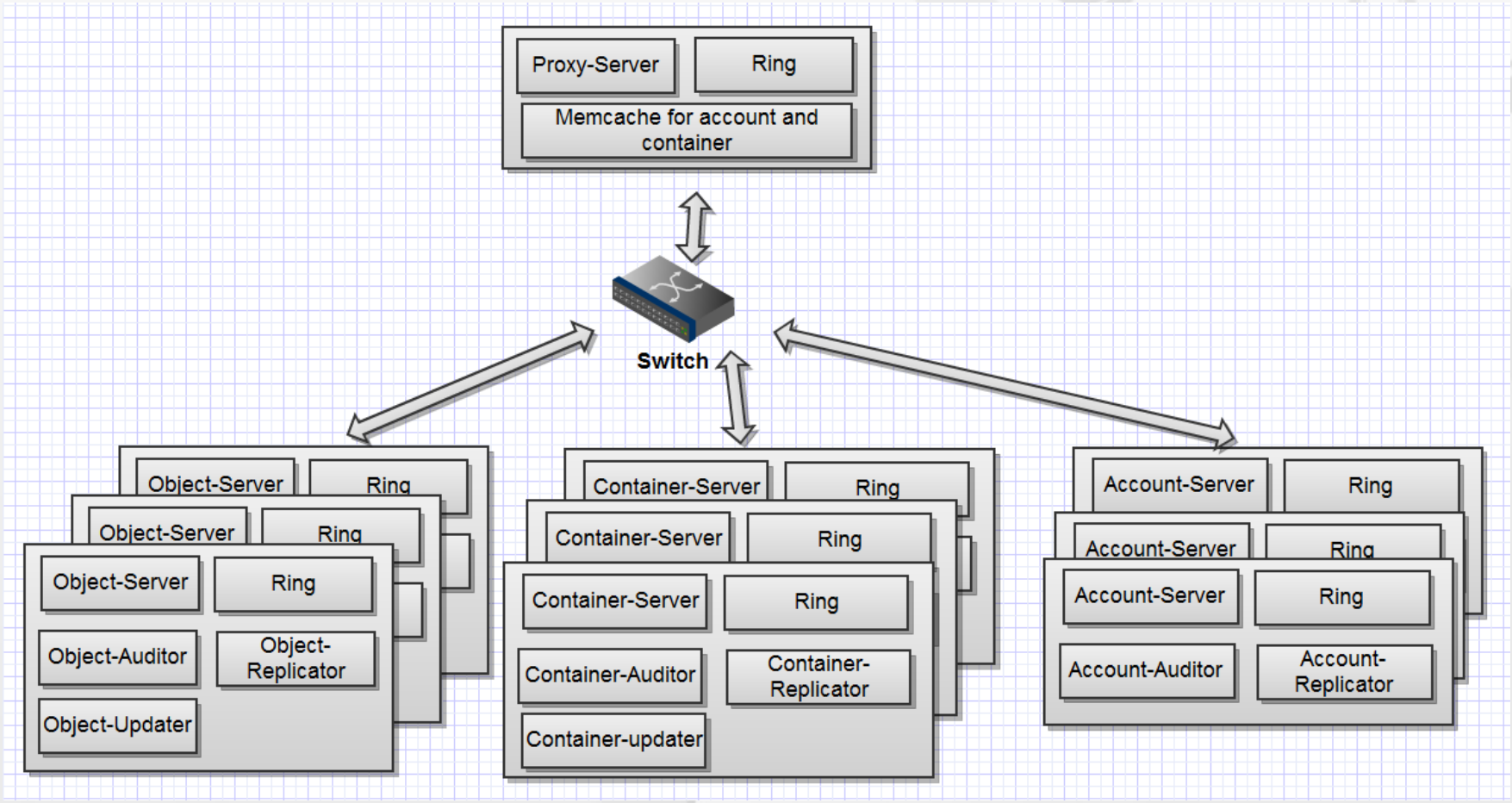
# Metadata

The target of swift: no single failure, no bottletneck, scale out

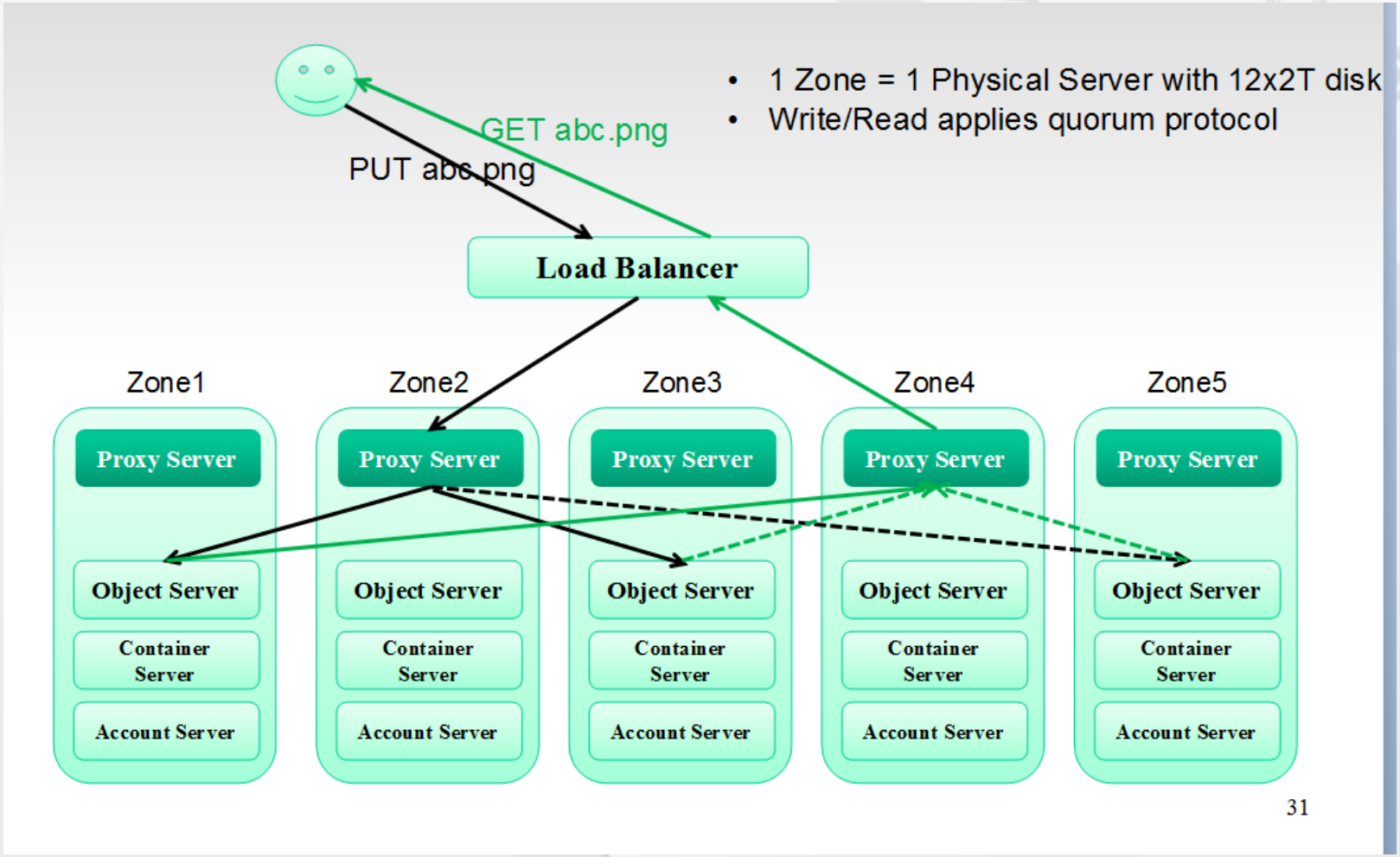
The Swift way: sqlite + consistent hashing + quroum
A sqlite db file is an object.
So the database is HA, durable and with eventual consistency.


Object
Container
Account
Consistent Hashing
VirtualNode1
VirtualNode2
...
VirtualNodeN
PhysicalNode1 Zone1 Weight100
PhysicalNode2 Zone1 Weight100
PhysicalNode3 Zone2 Weight200
PhysicalNode4 Zone2 Weight200
PhysicalNode5 Zone3 Weight100
PhysicalNode5 Zone3 Weight100

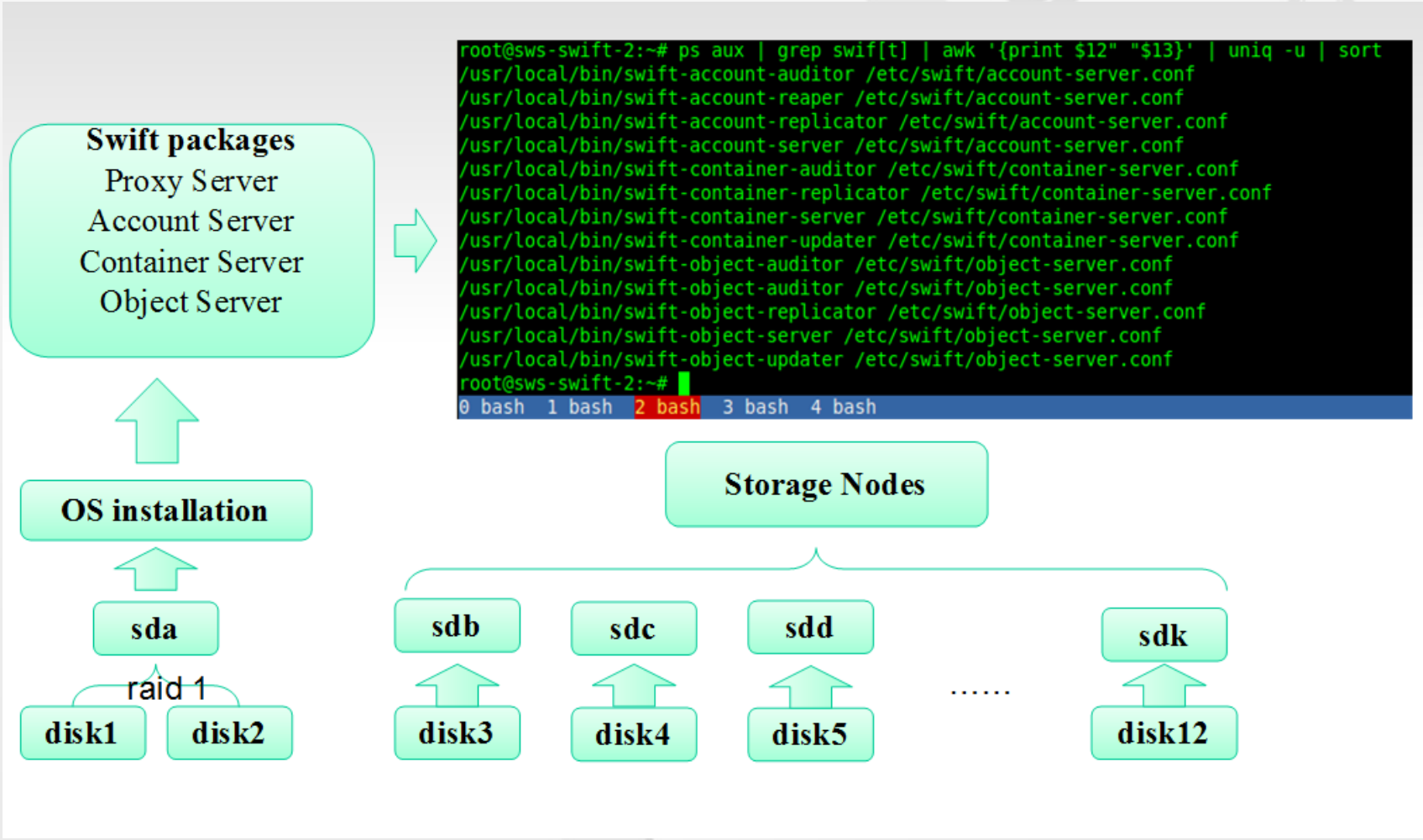
# Architecture



# Swift Practice@SinaAppEngine



# Swift Practice@SinaAppEngine



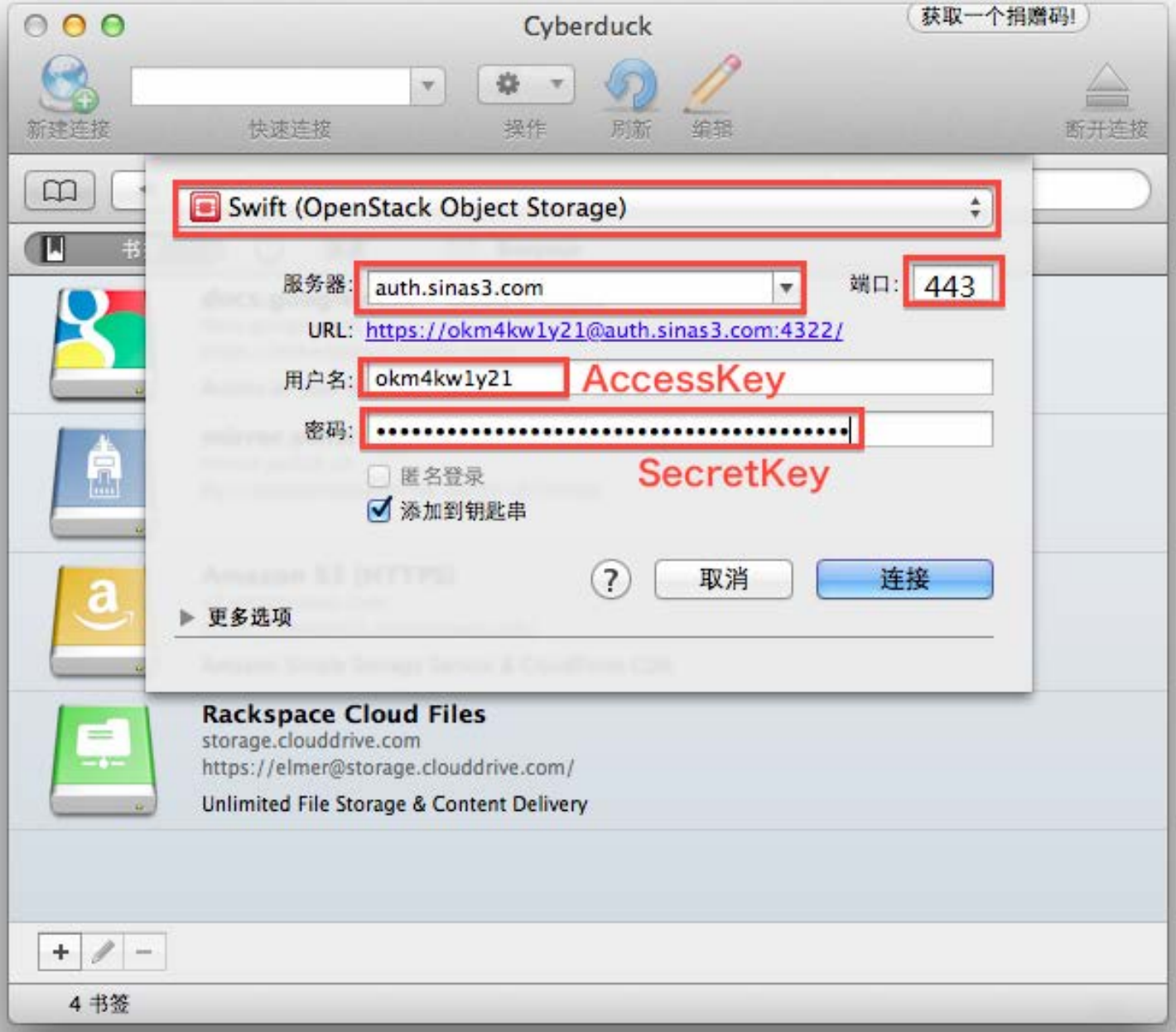
# Our Works

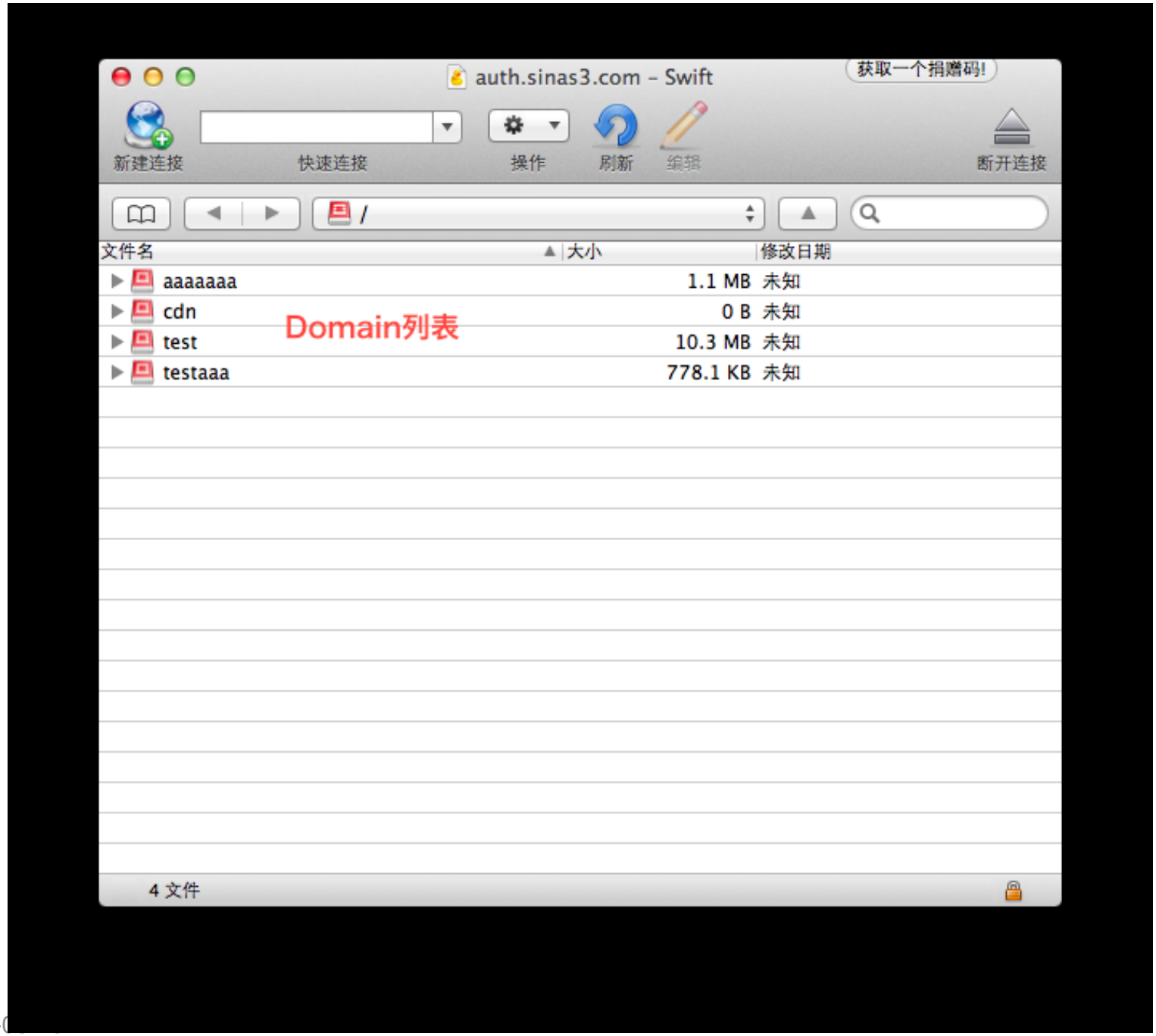
•**Swift as the SAE Storage**
-Auth module for SAE(Key-Pair)
-Keystone for SAE(Token)
-HTTP Cache-Control module
-Quota(limit the number of containers and objects, limit the storage usage)
-Domain remap
app-domain.stor.sinaapp.com/obj To
sinas3.com/v1/SAE_app/domain
-Rsync with bwlimit
-Billing
-Storage Firewall

•**Swift as the SWS Simple Storage Service**
-Container unique module
container.sinas3.com/object
-Keystone middleware for auth protocol converting

# Our Steps for Switching SAE Storage

- Bypass test one month online(旁路读写测试)
- Switch step by step one month(灰度切换)
- Ops
  - Monitoring: I/O, CPU, Memery, Disk
  - LogCenter: syslog-ng, statistics and analytics


Cyberduck
获取一个捐赠码!
新建连接
快速连接
操作
刷新
编辑
断开连接
Swift (OpenStack Object Storage)
服务器:
auth.sinas3.com
端口:
443
URL:
https://okm4kw1y21@auth.sinas3.com:4322/
用户名:
okm4kw1y21
AccessKey
密码:
SecretKey
匿名登录
添加到钥匙串
取消
连接
更多选项
Rackspace Cloud Files
storage.clouddrive.com
https://elmer@storage.clouddrive.com/
Unlimited File Storage & Content Delivery
4 书签


auth.sinas3.com – Swift
获取一个捐赠码!
新建连接
快速连接
操作
刷新
编辑
断开连接
/
文件名
大小
修改日期
aaaaaaa 1.1 MB 未知
cdn 0 B 未知
test 10.3 MB 未知
testaaa 778.1 KB 未知
Domain列表
4 文件


alibuybuy-img11.stor.sinaapp.com/2012/08/4b51_shoplocket.png
shoplocket
BETA
Sell anything from anywhere, in minutes.
Connect with Facebook
or
Enter your email address
Choose a password
Try it for free

# Problems&Imporvements

The async processor for keeping eventual consistency is inefficient.
Replicator, auditor, container-updater
**Logic:** loop all objects or dbs on disk, query replica's server to determin whether to sync.
**Results:** High I/O; High CPU usage; Stress on account/container/object-servers, impact the availability; Long time for eventual consistency, impact the durability; The list operation is not consistent.

How to improvements?
1.Runing replicator, auditor and container updater during idle time;
2.An appropriate deployment;
3.A new protocol for keeping relplica's consistency;
(based on log and message queue)
4.Adding new nodes, scale out.

# Problems&Imporvements

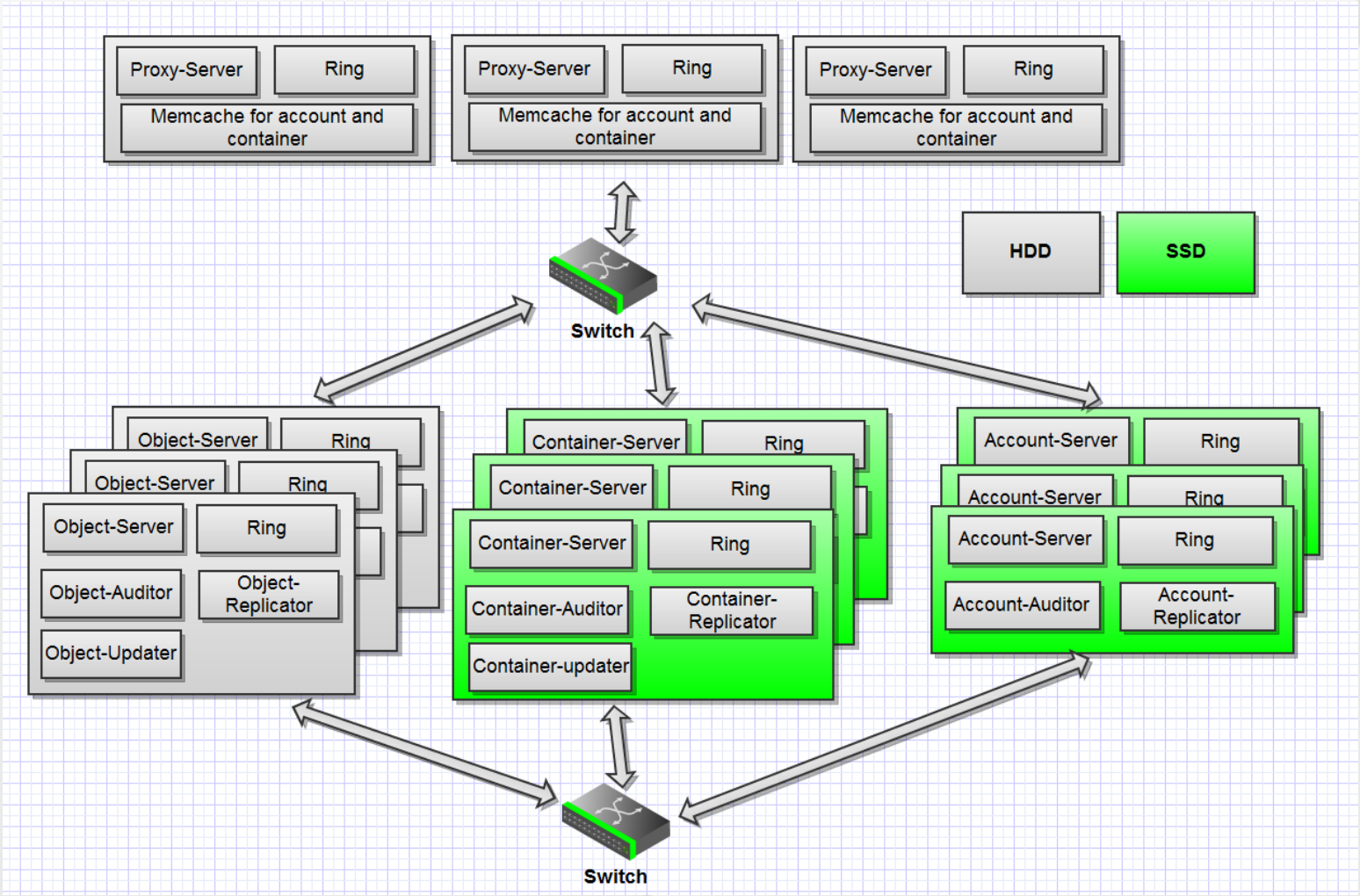
The performance of sqlite.
Quota for objects and containers
Running sqlite on the high performance I/O devices

The bandwidth of rsync is not  under control.
Out-of-band management
Add bandwidth limitations for rsync

Database centralized or distributed?

# Problems&Imporvements

An appropriate deployment



# Q&A

Weibo: @AlexYang_Coder
Email: yangyu4@staff.sina.com.cn
GTalk: alex890714@gmail.com
Blog: http://alexyang.sinaapp.com