UCLOUD

OStorage
奥思数据

分布式存储中的数据分布算法

李明宇 @ 奥思数据

OStorage 奥思数据 UCLOUD

目录

OStorage 奥思数据 UCLOUD

01

一致性哈希算法及其在实际应用中遇到的挑战

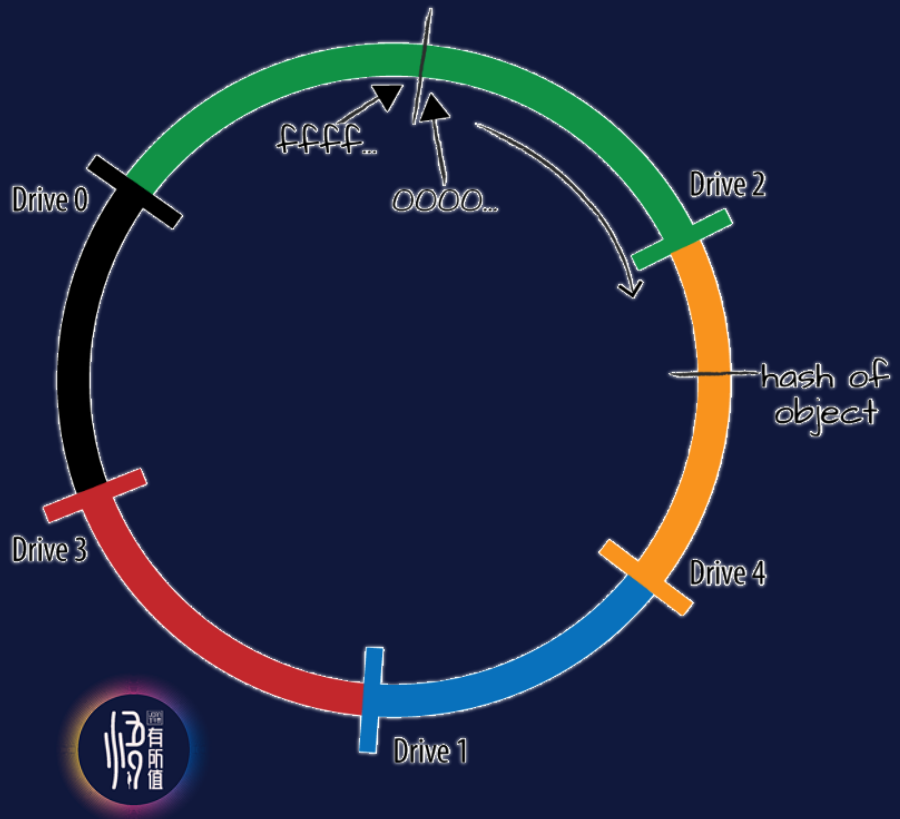
哈希表及其在分布式系统中的问题

MD5 (“/account/container/object”) =
f9db0f833f1545be2e40f387d6c271de

- 节点的频繁退出（故障）和加入（扩容）

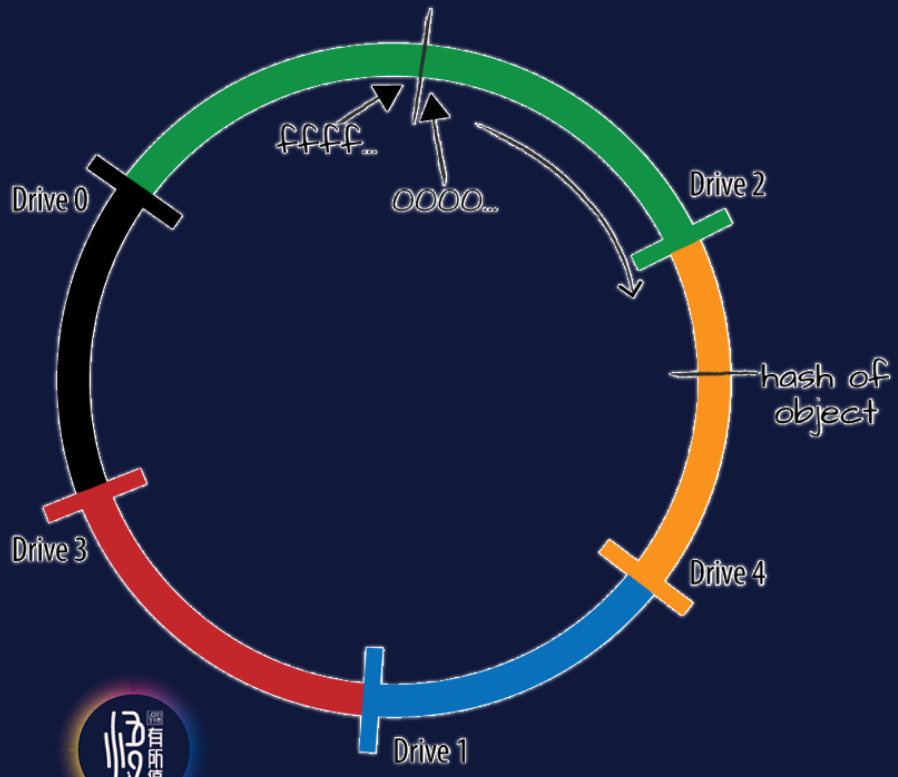
Total drive count	Remainder	Maps to
6	(hash) % 6 = 4	Drive 4
7	(hash) % 7 = 6	Drive 6
8	(hash) % 8 = 6	Drive 6
9	(hash) % 9 = 1	Drive 1
10	(hash) % 10 = 8	Drive 8

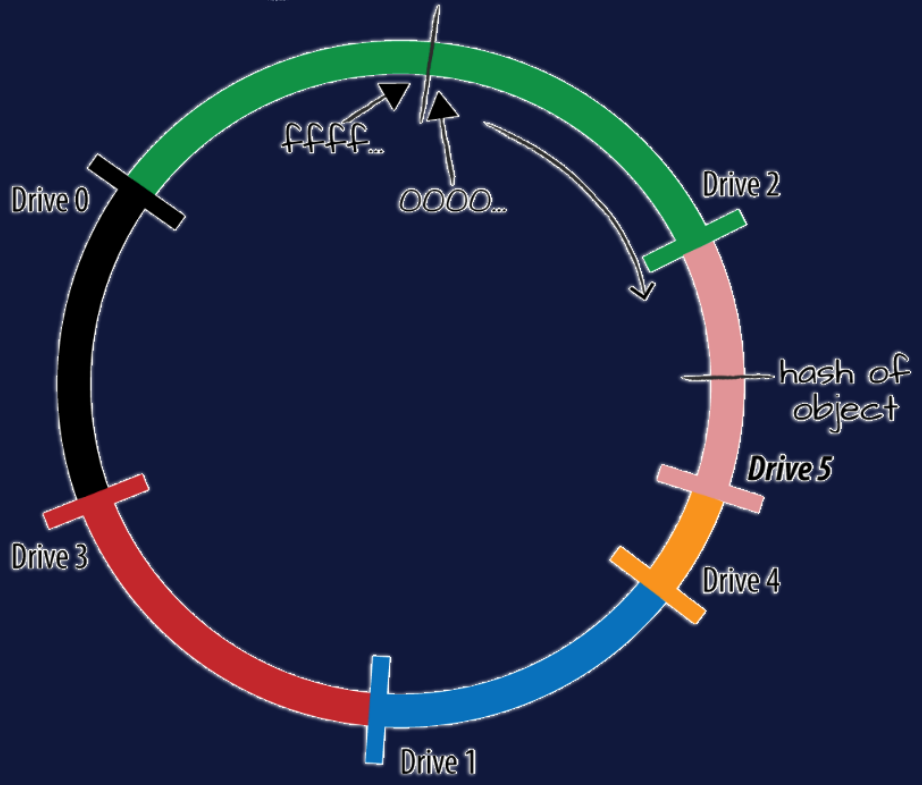
一致性哈希算法



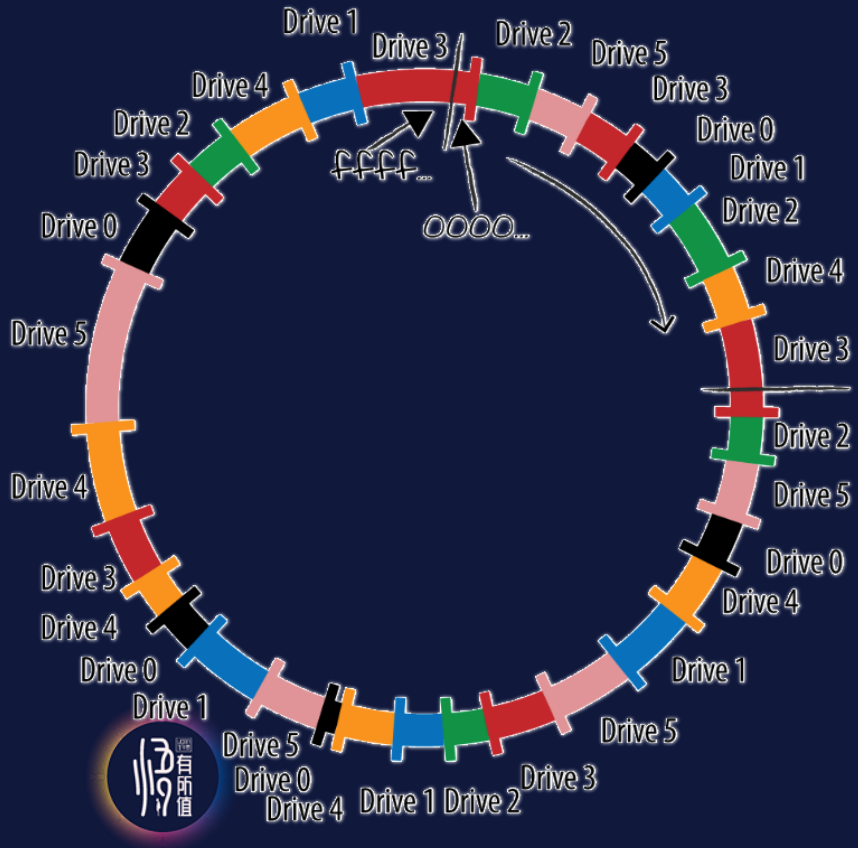
- 不仅对数据算hash
  也对设备/节点算hash
- 模除（求余）运算改为
  移位运算（顺时针寻找最近的设备/节点）

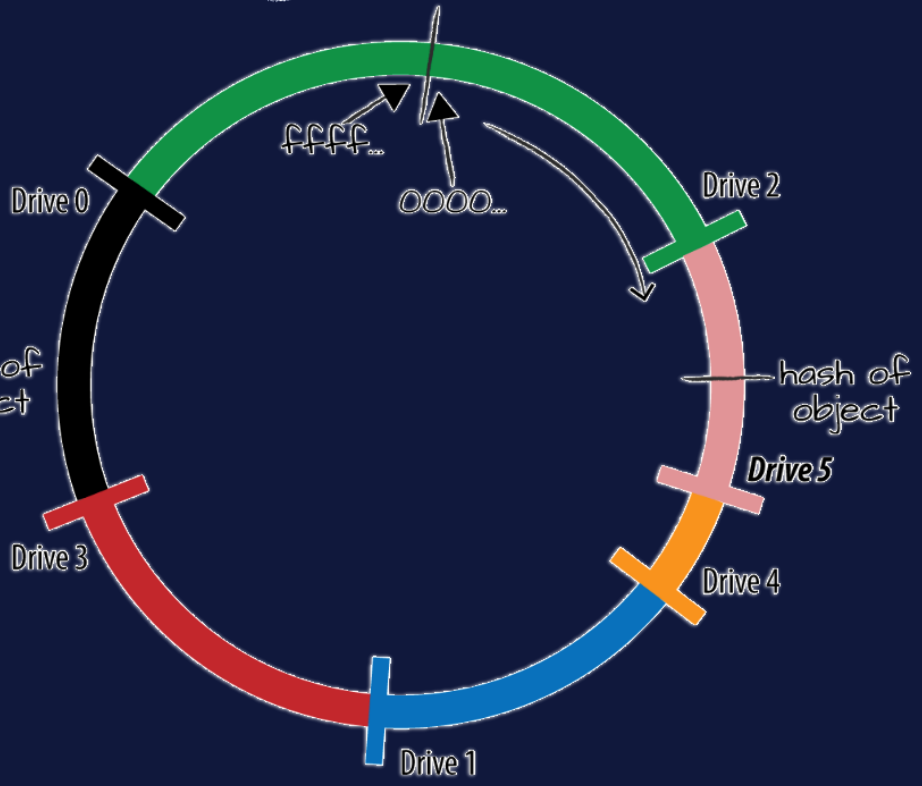
一致性哈希算法


ffff...
0000...
Drive 0
Drive 2
hash of object
Drive 3
Drive 4
Drive 1

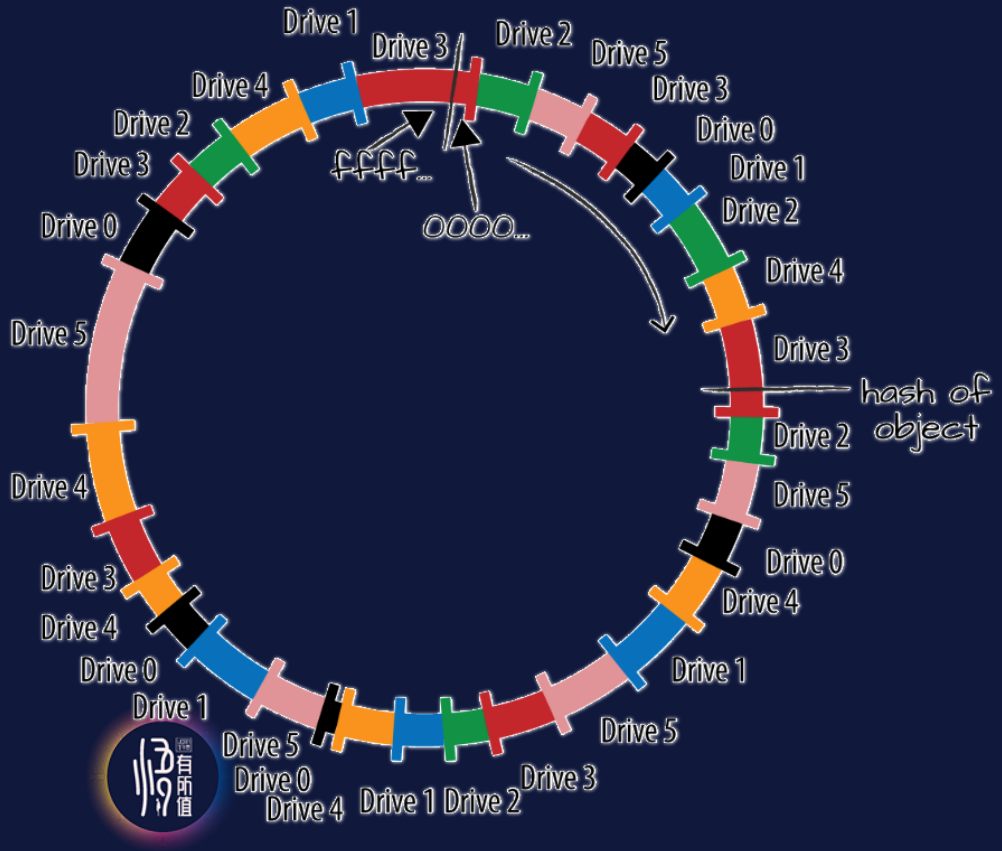

ffff...
0000...
Drive 0
Drive 2
hash of object
Drive 5
Drive 3
Drive 4
Drive 1

一致性哈希算法


Drive 1
Drive 3
Drive 2
Drive 5
Drive 4
Drive 3
Drive 2
Drive 0
Drive 3
Drive 1
Drive 0
Drive 2
Drive 4
Drive 5
Drive 3
hash of object
Drive 2
Drive 4
Drive 5
Drive 3
Drive 0
Drive 4
Drive 4
Drive 1
Drive 0
Drive 5
Drive 1
Drive 0
Drive 4
Drive 1
Drive 2
Drive 3
ffff...
oooo...


Drive 0
Drive 2
hash of object
Drive 5
Drive 3
Drive 4
Drive 1
ffff...
oooo...

一致性哈希算法

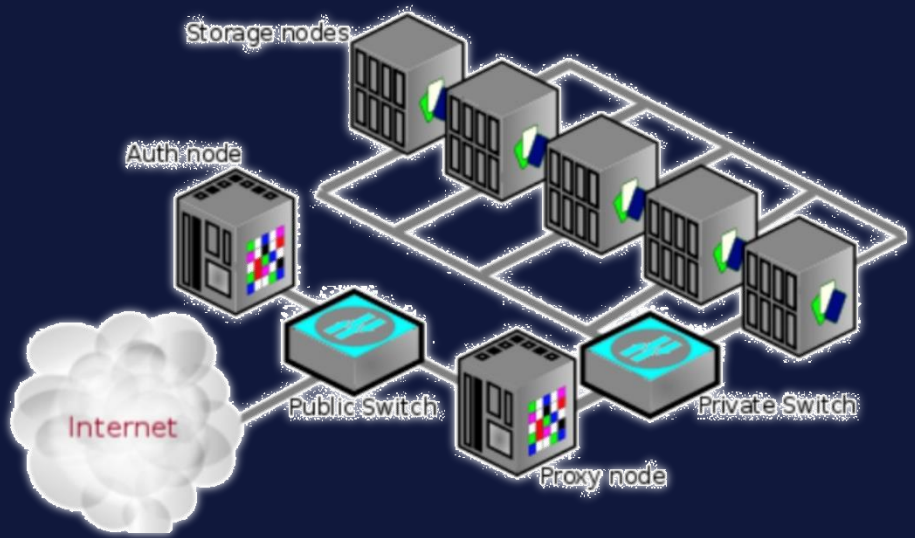


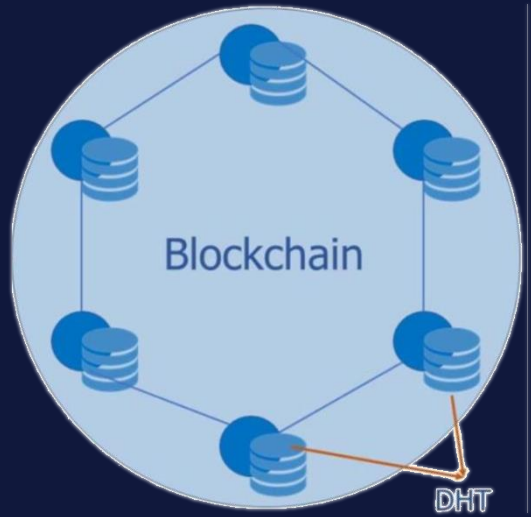
- 一个实际设备对应到n个虚拟设备
- 一致性哈希的优点：
  - 计算复杂度
  - 均匀性
  - 数据迁移开销

一致性哈希算法在实际应用中遇到的挑战

OStorage 奥思数据 UCLOUD

- 企业级IT场景
  - 多副本可靠存储问题
  - 成本要可接受，数据千万不能丢
- “存储区块链”场景
  - 几乎不可能获取全局视图
  - 甚至没有一刻是稳定的





OStorage 奥思数据 UCLOUD

02

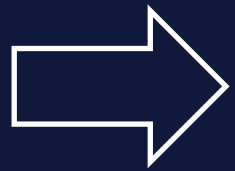
典型“存储区块链”中的数据分布算法

当我们在谈论“存储区块链”
我们在谈论什么？

OStorage 奥思数据 UCLOUD

- 存储区块链 / 区块链存储
  - 分布式存储（p2p存储） + 区块链
  - 通过token激励，鼓励大家贡献存储资源，参与构建一个全世界范围的分布式存储系统*（今天我们只讨论技术，只谈存储技术）*
- 代表项目
  - Sia
  - Storj
  - IPFS + filecoin
  - ……

激励全球用户自发参与



大型P2P存储网络中的寻址/路由问题

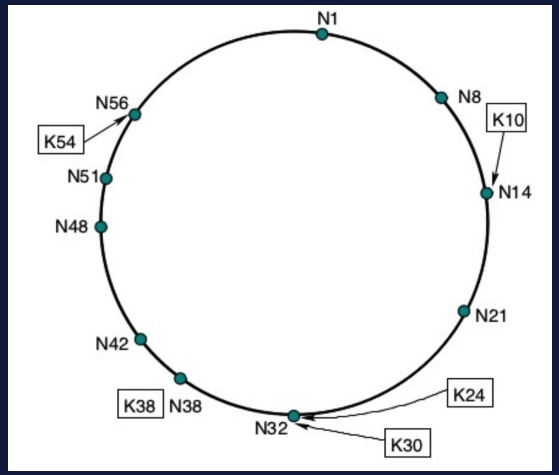
当我们在谈论“存储区块链”我们在谈论什么？

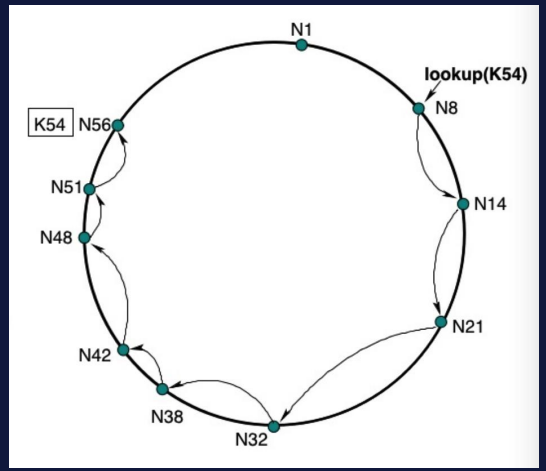
- 存储区块链 / 区块链存储
  - 分布式存储（p2p存储） + 区块链
  - 通过token激励，鼓励大家贡献存储资源，参与构建一个全世界范围的分布式存储系统*（今天我们只讨论技术，只谈存储技术）*

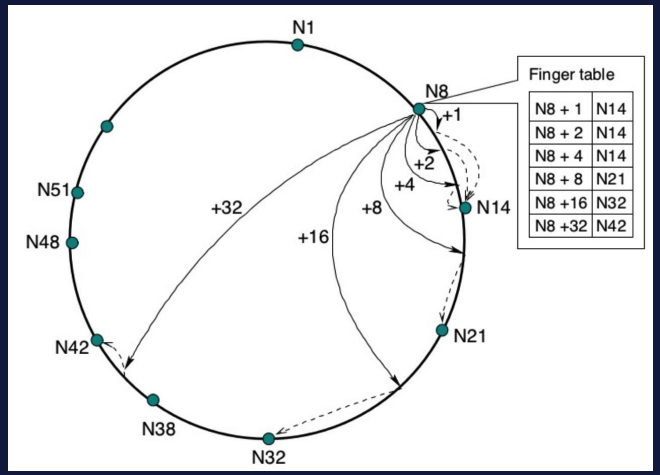
激励全球用户自发参与 → 大型P2P存储网络中的寻址/路由问题

- 代表算法
  - Chord
  - Kademlia
  - Tapestry
  - S/Kademlia
  - ……

Chord（2001）

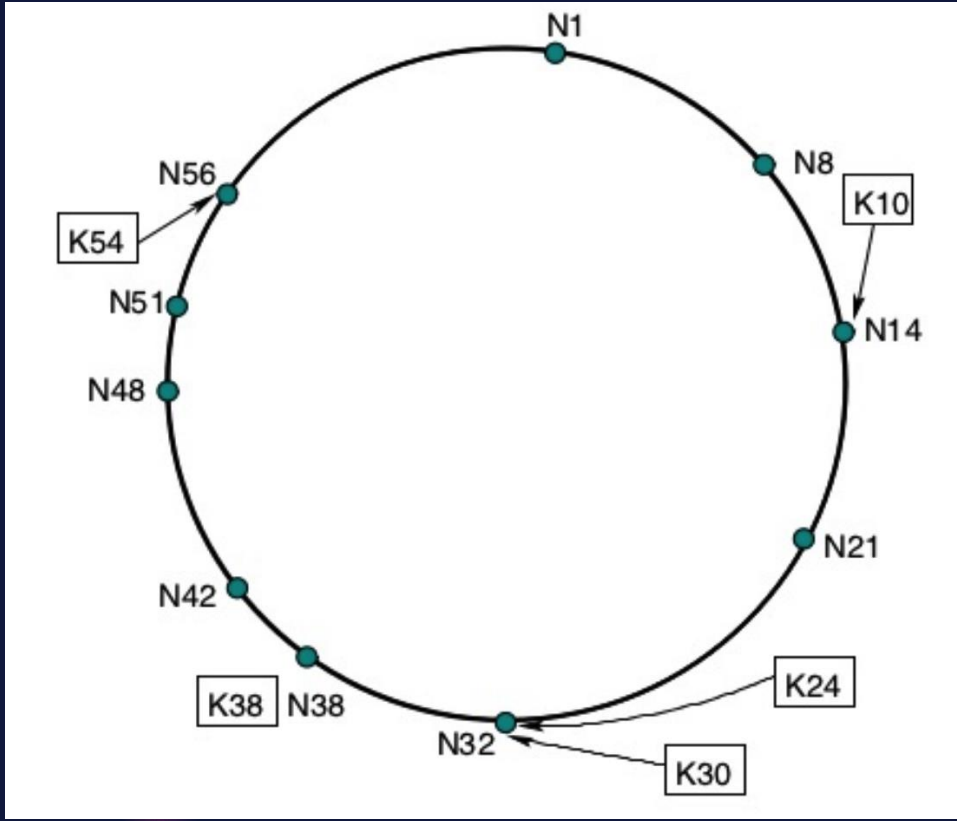

N1
N8
K10
N14
N21
K24
K30
N32
K38
N38
N42
N48
N51
K54
N56

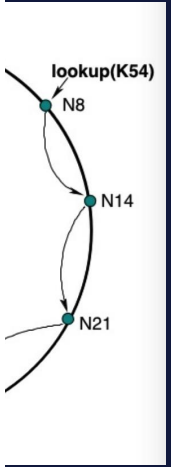

N1
lookup(K54)
N8
N14
N21
N32
N38
N42
N48
N51
K54
N56

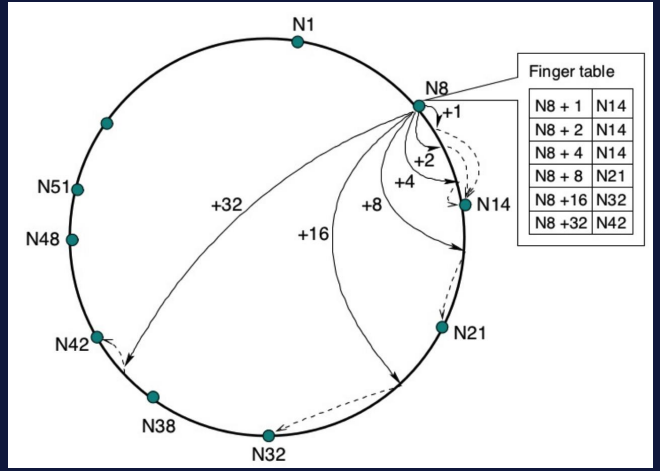

N1
N8
+1
+2
+4
+8
+16
+32
N14
N21
N32
N38
N42
N48
N51
Finger table
N8 + 1 N14
N8 + 2 N14
N8 + 4 N14
N8 + 8 N21
N8 +16 N32
N8 +32 N42

Chord（2001）

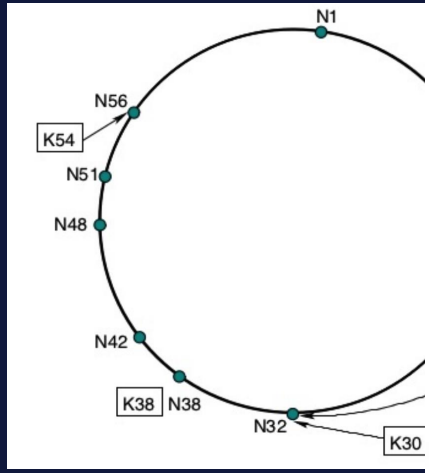
OStorage 奥思数据 UCLOUD

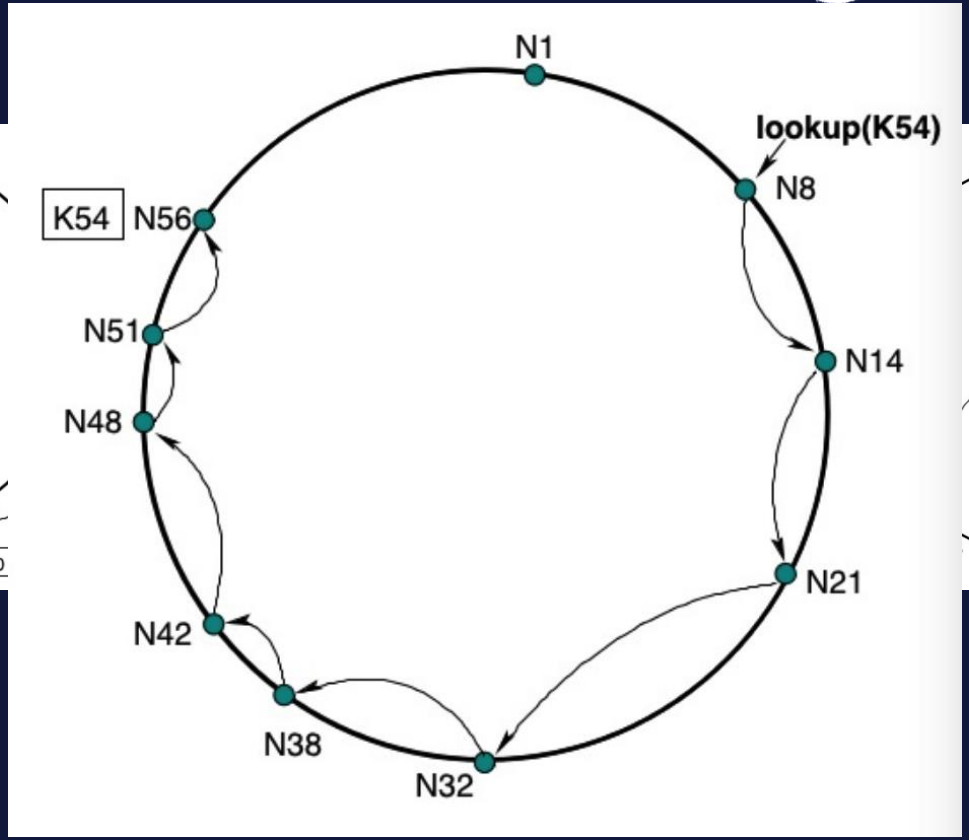


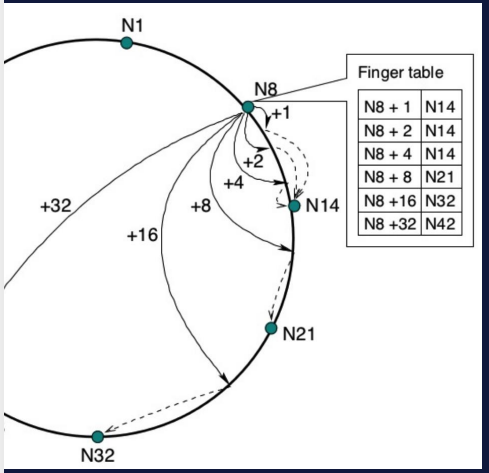




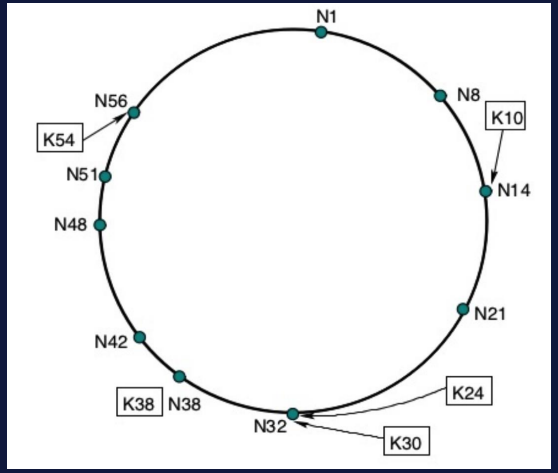
Chord （2001）

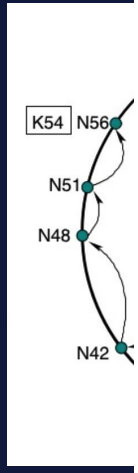

N1
N56
K54
N51
N48
N42
K38
N38
N32
K30

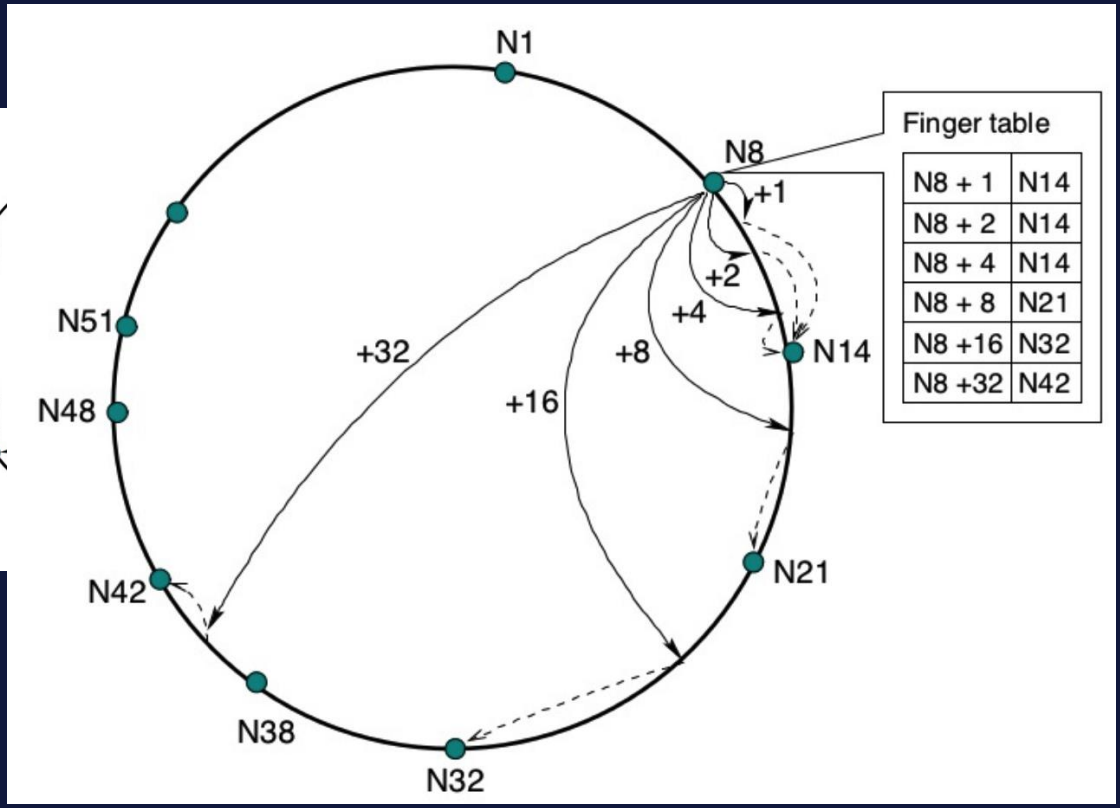

N1
lookup(K54)
N8
K54
N56
N51
N14
N48
N21
N42
N38
N32


N1
N8
Finger table
N8 + 1 N14
N8 + 2 N14
N8 + 4 N14
N8 + 8 N21
N8 +16 N32
N8 +32 N42
+1
+2
+4
+8
+16
+32
N14
N21
N32

Chord（2001）


N1
N8
K10
N14
N21
K24
K30
N32
K38
N38
N42
N48
N51
N56
K54


K54
N56
N51
N48
N42


N1
N8
N14
N21
N32
N38
N42
N48
N51
+1
+2
+4
+8
+16
+32
Finger table
N8 + 1 N14
N8 + 2 N14
N8 + 4 N14
N8 + 8 N21
N8 +16 N32
N8 +32 N42

OStorage 奥思数据 UCLOUD

03

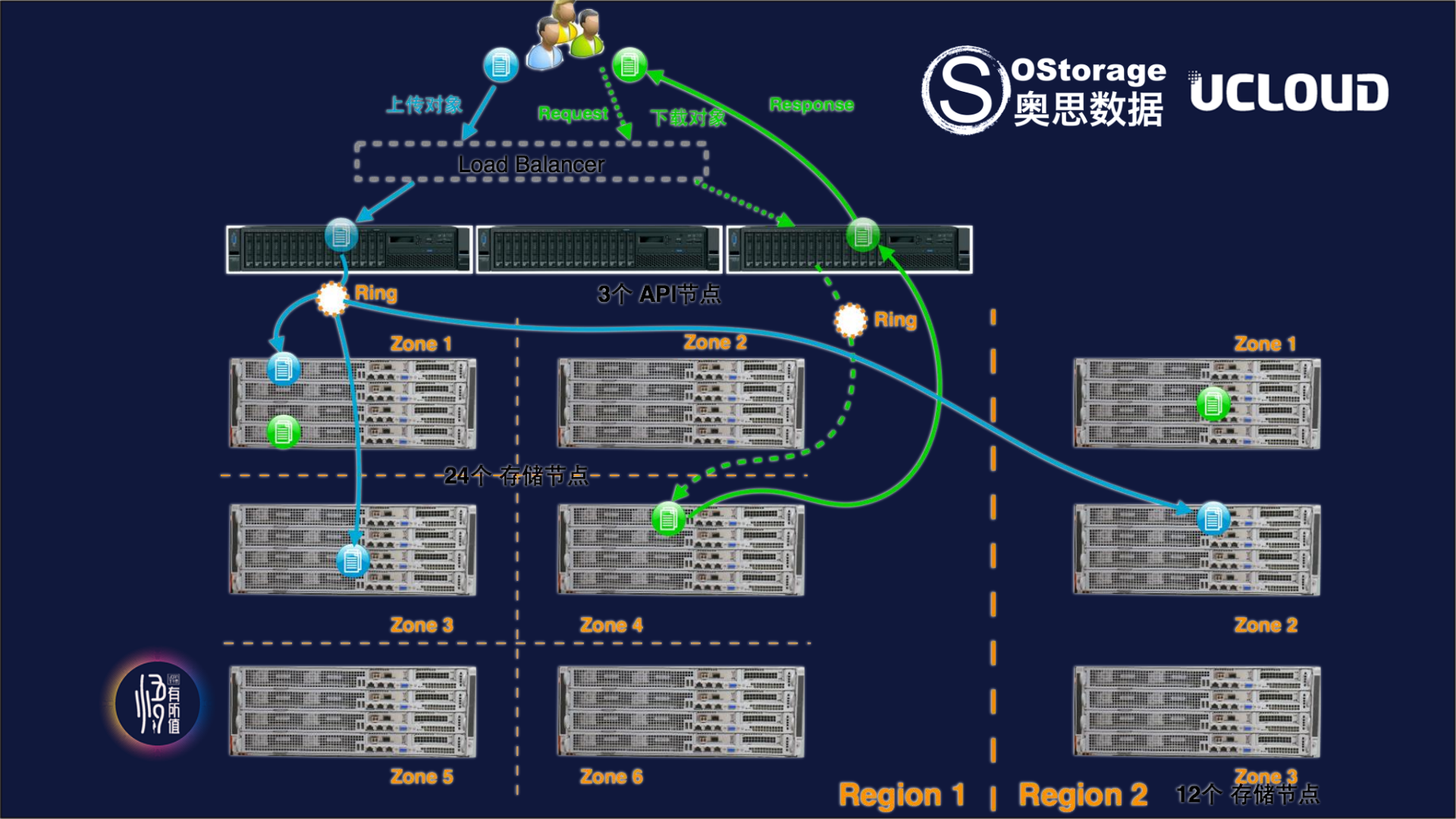
企业级存储中典型的数据分布算法

企业级存储中典型的数据分布算法

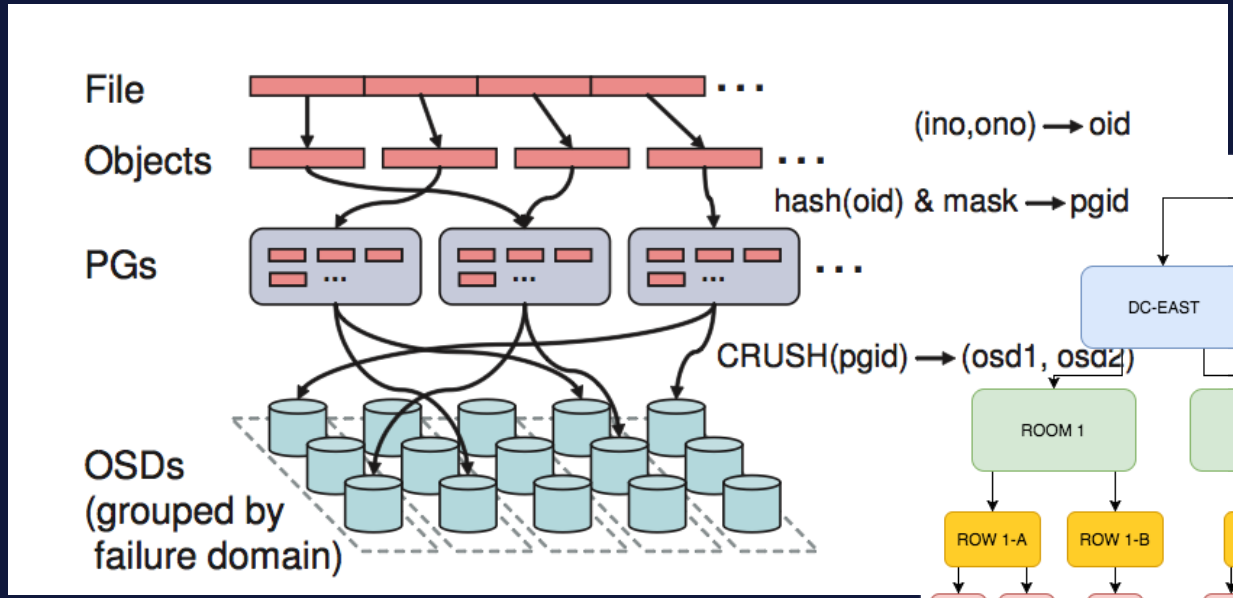
OStorage 奥思数据 UCLOUD

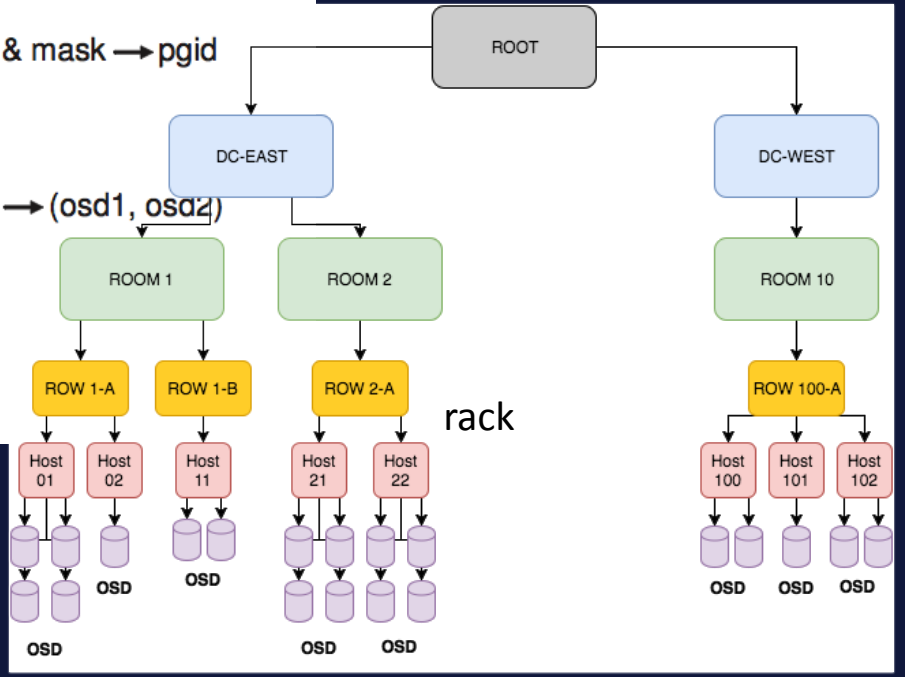
- Dynamo: Amazon’s Highly Available Key-value Store
- Ceph - CRUSH · Gluster - Elastic Hashing · Swift - Ring

- 这些算法都有相似的特点
  - 基于/借鉴一致性哈希
  - 引入对数据中心物理拓扑的建模（Cluster Map），数据多副本 / EC分片跨故障域 / 可用区分布
  - 可以对节点 / 设备划分权重
  - 多种存储策略选择（这条更多是一种存储系统的工程实现，跟分布算法本身关系不大）


OStorage
奥思数据
UCLOUD
上传对象
Request
下载对象
Response
Load Balancer
Ring
3个 API节点
Ring
Zone 1
Zone 2
Zone 1
24个 存储节点
Zone 3
Zone 4
Zone 2
Zone 5
Zone 6
Zone 3
Region 1
Region 2
12个 存储节点

企业级存储中典型的数据分布算法

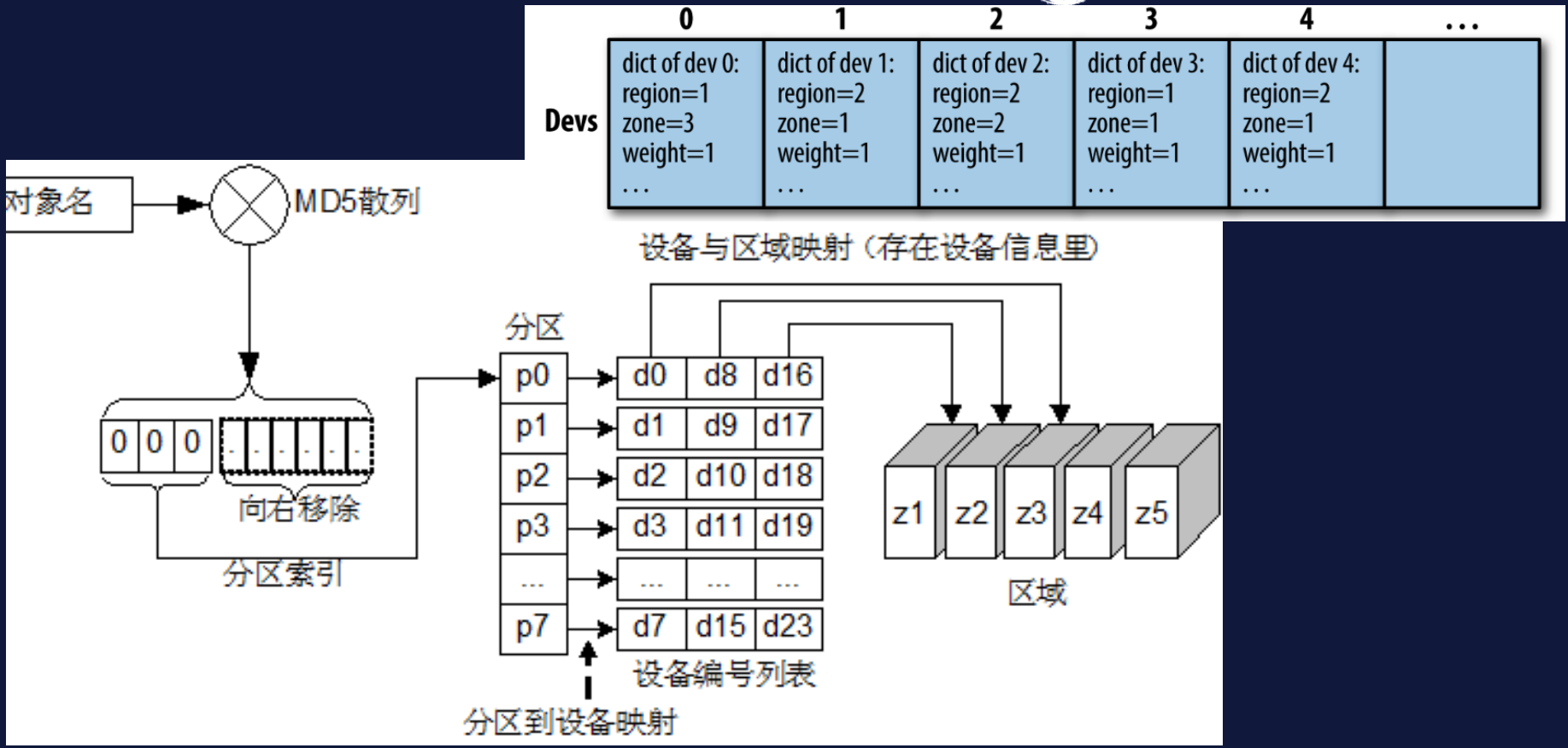




权重（Weight）

企业级存储中典型的数据分布算法

OStorage 奥思数据 UCLOUD



OStorage 奥思数据 UCLOUD

04

比较与总结

比较与总结

- 一致性哈希
- 企业级存储
  - 实际不依赖一致性哈希的特性扩容
  - 设备数量有限，且全局可控
  - 故障域、权重等因素
- 区块链存储
  - 设备数量可能巨大
  - 收敛的DHT算法
- 都是分布式存储，但是存在一些有意思的差别
  - 例如节点下线的概率

UCLOUD

更多分享与交流

2018.12.16 北京