UCLOUD

UCloud分布式文件系统架构解析

邓瑾@UCloud

UCLOUD

目录

UCLOUD

01

分布式文件系统简介

分布式文件系统

UCLOUD

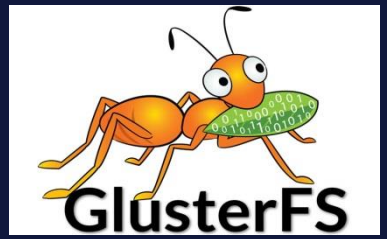
分布式文件系统是传统文件系统的延伸,通过分布式技术手段和公有云规模效应获取传统文件系统所没有的存储能力.

- scale out: 容量和性能的线性/近线性提升
- fault tolerant: 屏蔽硬件故障,提升数据可靠性,系统可用性
- lower TCO & pay-as-you-go: 公有云红利

几种经典系统

UCLOUD

- Google File System/HDFS: 高吞吐场景
- Taobao File System/Haystack: 小文件场景
- GlusterFS/CephFS
- Azure Blob Storage: 通用型存储系统
- WAFL/PolarFS



其他存储类型

- 对象存储: immutable storage(WORM)/web-oriented/lake of native file api(FUSE)/unlimited storage space
- KV存储: lower latency/NoSQL flexibility/smaller data scale
- 文件存储: standard file api/tools/PB+

UCLOUD

02

UFS 发展历程

UFS 是什么?

- 完全自主研发
- 面向公有云业务设计,支持多类型主机
- 支持标准协议NFSv3/v4(SMB is coming)
- 高可用/高可靠的文件存储服务

使用场景

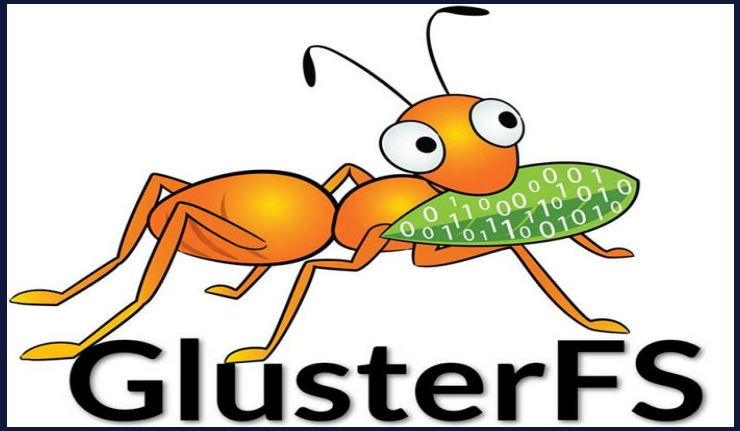
UCLOUD

- 备份: 数据库备份/日志存储
- 数据分析/操作: Hadoop,广电非编
- 数据共享: web站点/企业办公/代码托管
- Share Nothing 架构: 更易于业务实现 Layer Storage

原型阶段

UCLOUD

- 原型验证: 利用开源软件快速在公有云环境中进行产品原型验证
- 运营积累: 从运营角度积累分布式文件产品在多租户的公有云环境中的痛点和难点,进行自研产品的设计改进.
- 吸收社区经验.



What We Learned

UCLOUD

- 规模拓展性具有瓶颈(peering开销大),节点数量受限
- 无法进行多集群的管理,无法做灰度管理
- 索引和数据耦合,索引操作容易引起高IO从而影响数据操作性能,小文件访问和大目录操作性能极差
- 文件直接全文存储,数据安全隐患大
- 数据修复和迁移不透明
- 社区推进乏力

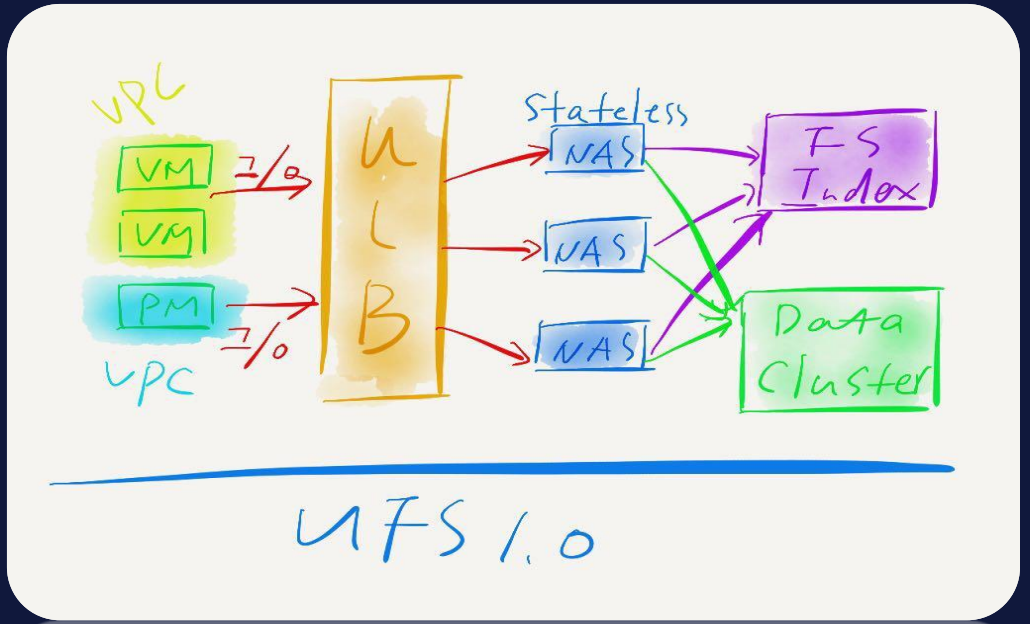
UFS 1.0 的自研改造

UCLOUD

- 索引和数据分离
- 自定义的索引结构和语义,便于后续拓展非 NFS 协议
- 独立设计的存储服务.支持 set 管理,灰度等策略
- 支持百万级大目录和TB+文件大小
- 支持QoS
- 数据安全性高

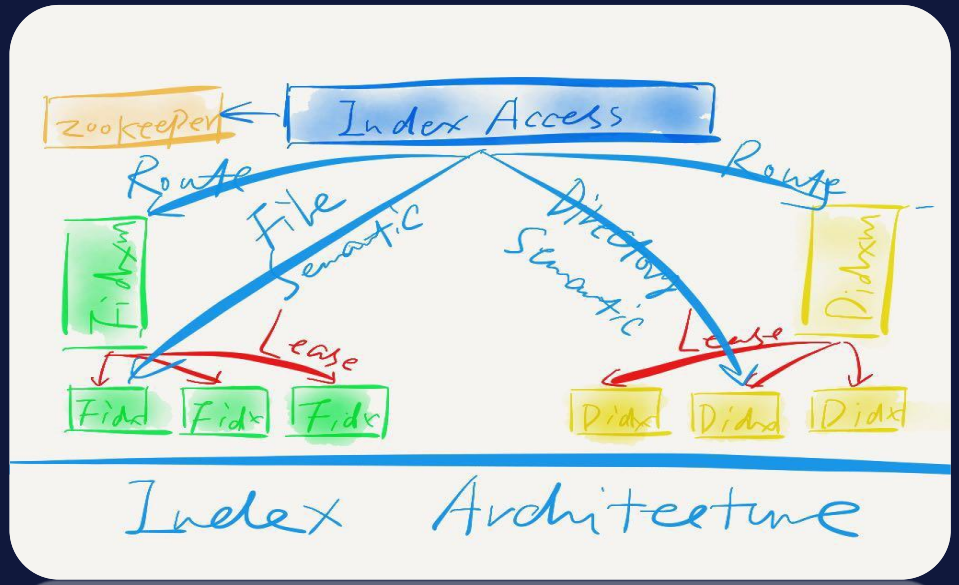
1.0 整体架构

- 索引/数据分离
- 无状态接入设计
- 处理能力/数据容量可线性扩展
- NO SPOF



索引层

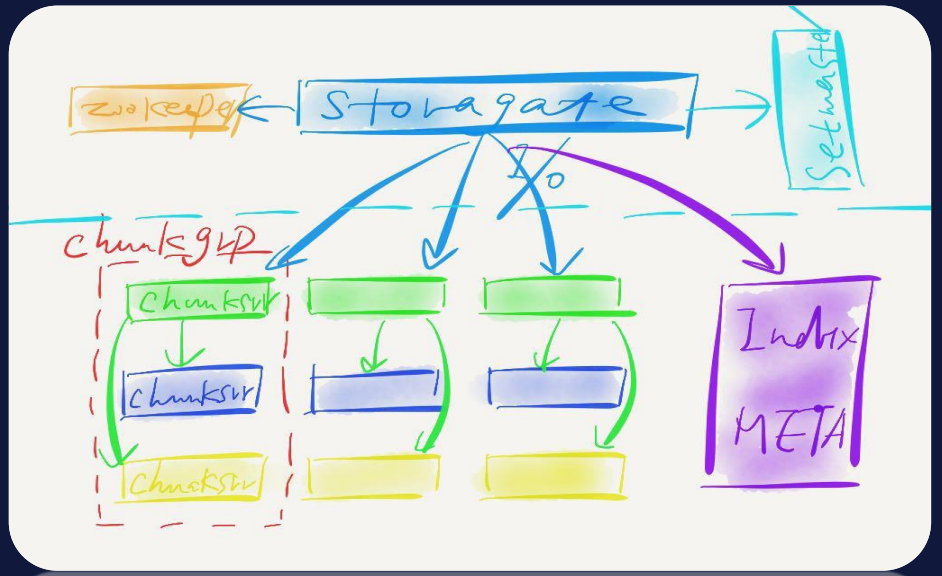
- 平台无关的索引格式
- 处理能力可线性扩容
- 保证元数据一致性
- 灵活实现业务特性(大文件/大目录支持)

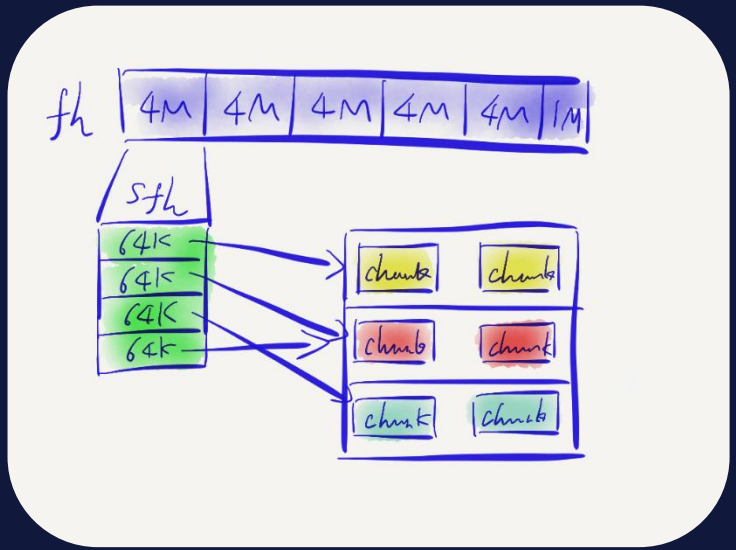


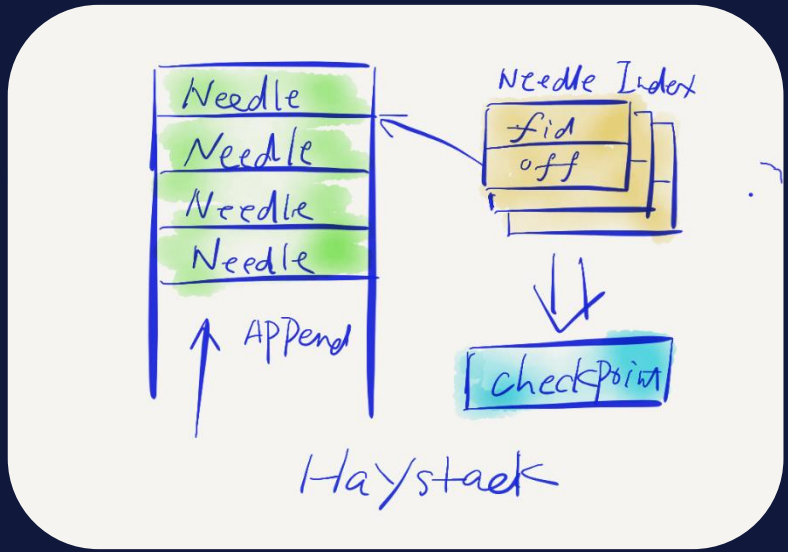
数据层

- 多集群,灰度可控
- 多副本冗余

  Haystack数据模型
- Immutable torage




fh
4M
4M
4M
4M
4M
1M
sfh
64K
64K
64K
64K
chunk
chunk
chunk
chunk
chunk
chunk


Needle
Needle
Needle
Needle
APPend
Needle Index
fid
off
checkpoint
Haystack

1.0 的局限

- 存储模型比较适合小分片场景,不够通用
- 固定的底层存储分片造成了一定的空间浪费
- 存储层支持的文件尺度较小
- 对随机写的支持不够

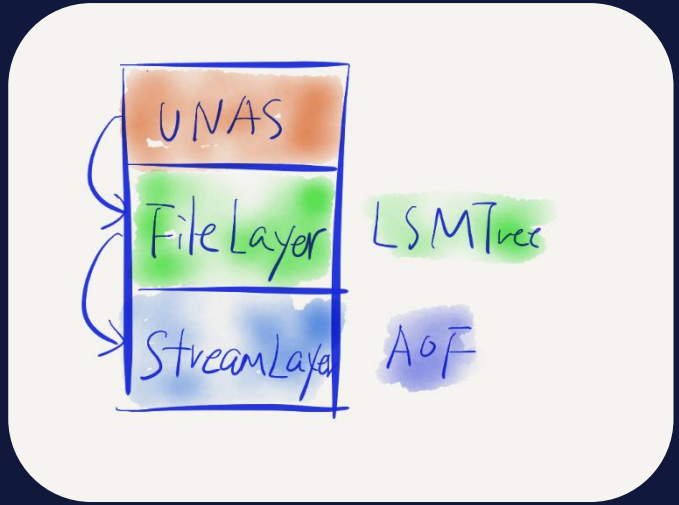
UCLOUD

03

UFS 2.0 存储架构

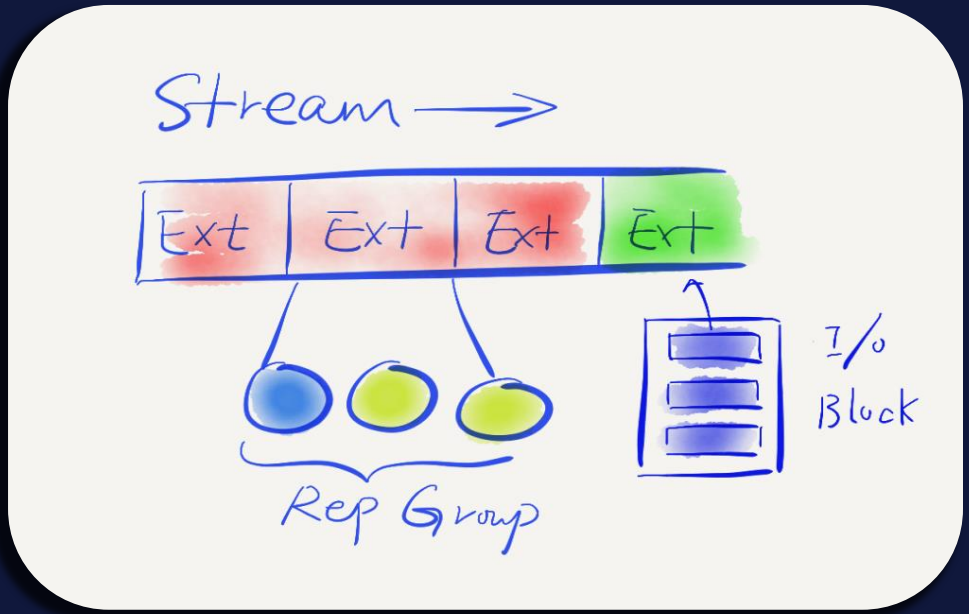
Nebula

- append-only 模型
- 超大文件支持
- 灵活冗余机制
- 低延迟/高性能场景



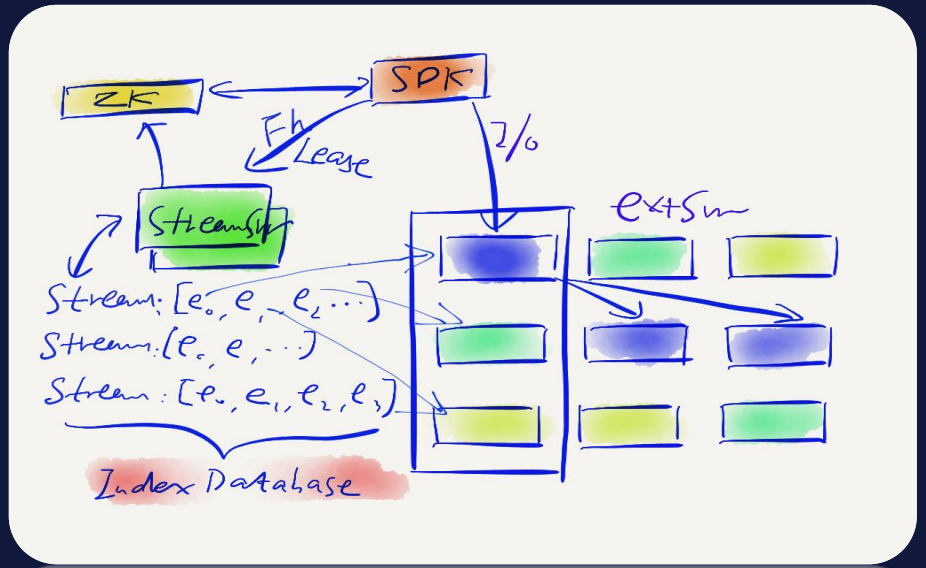
Append-only Model

- Stream:代表一个文件流,可在尾部追加
- Extent: stream 中的数据分片,分片大小不固定,每个extent 以文件的形式落地

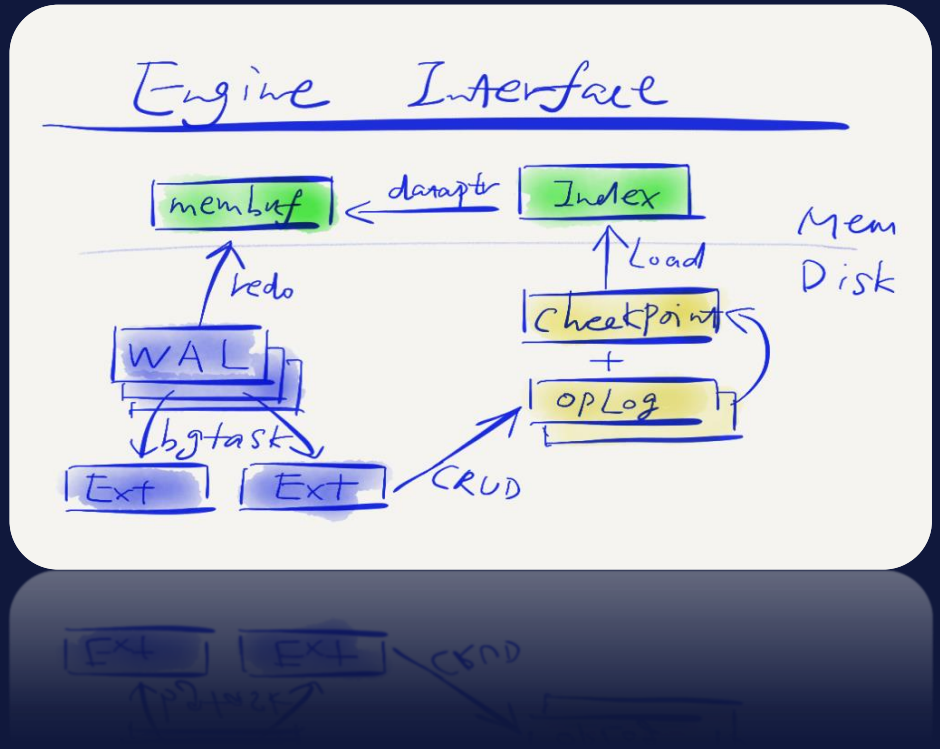


数据层

- streamsvr: 负责维护stream和extent的索引/路由信息
- extentsvr: 持久化数据,维护extent内的block索引信息.提供直连客户端的读写请求.



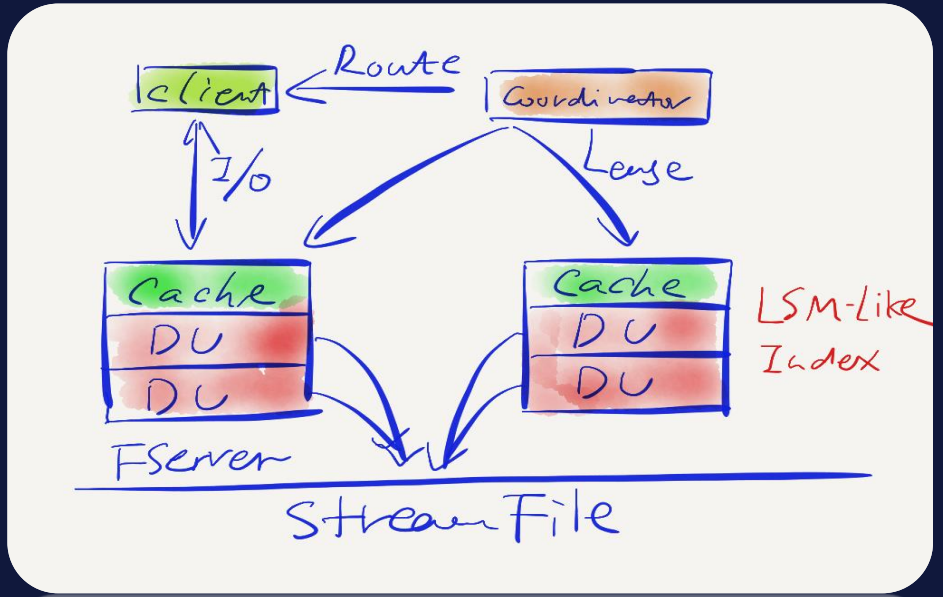
- 插件式引擎设计
- WAL: 降低写入毛刺
- Index: extent内的block索引信息
- 充分利用内存buffer降低读毛刺

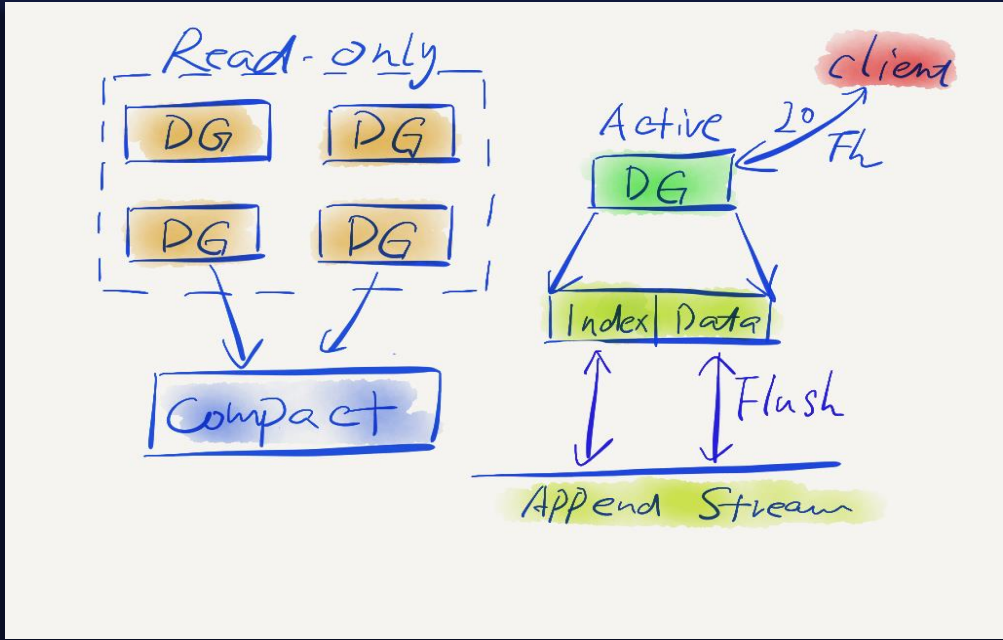


FileLayer

- 解决底层存储引擎随机写不友好的问题
- 对热点数据进行缓存,降低存储压力
- 数据分区
- Inspired By LSMTree

- DU: 数据集被切分为众多的DU,数据调度的最小粒度(100GB).每个DU只有一个fserver进行处理
- fserver: 承载DU请求的模块,实现随机写逻辑




Read-only
DG
DG
DG
DG
Compact
client
Active
20
Fh
DG
Index
Data
Flush
Append Stream

UCLOUD

04

Summary & QA

UCLOUD

THANKS

2018.11.10 武汉