
UCan下午茶
让块数据更有保障
彭晶鑫
有所值
UCLOUD

目录

- 块数据可用性/数据保护的场景和案例
- 从块存储角度谈高可用
- 块数据保护-让数据失而复得
- 为了更好的高可用

块数据可用性/

数据保护的场景和案例

可用性场景

磁盘故障

机器宕机

网络故障

机房断电

- 机房内高可用
- 多机房高可用
- 跨地域高可用

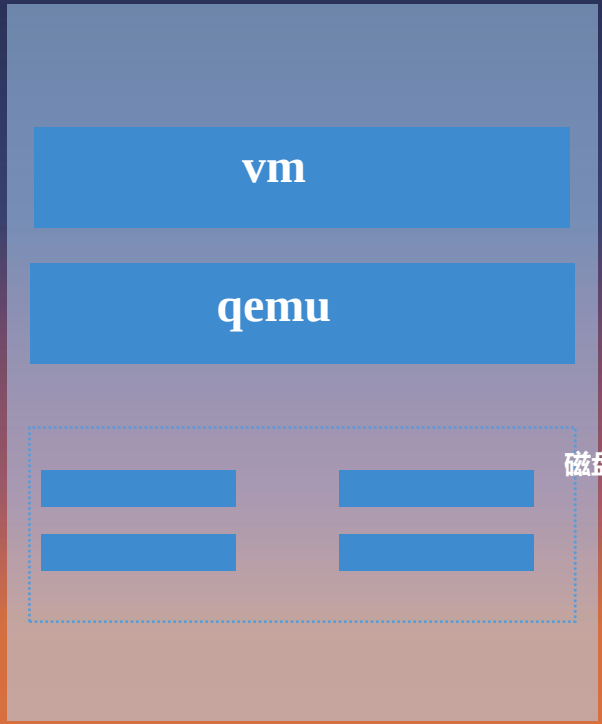
数据保护案例

- GitLab 300G 数据误删
- 黑客攻击删除重要数据
- 某互联网公司游戏数据库主库从库因故障无法及时回档
- 比特币勒索病毒

怎么让数据失而复得？

块存储角度谈高可用

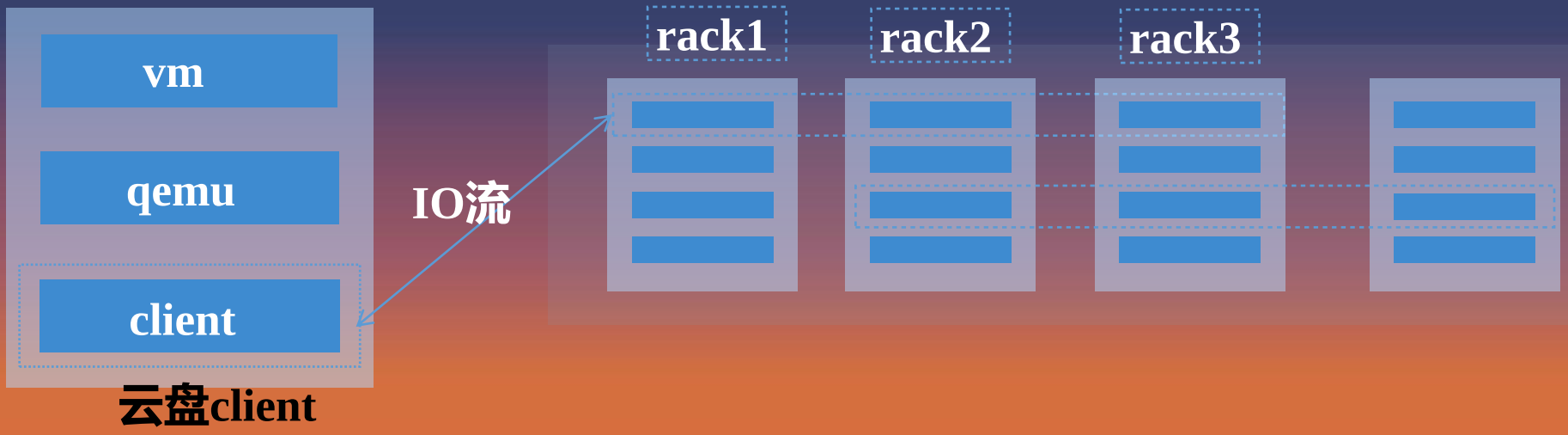
从块存储角度谈高可用



传统虚机块存储-本地盘

- 数据冗余需要靠raid —- 目前NVME还只能做软raid
- 宿主机宕机 - 恢复时间依赖宿主机拉起时间

分布式存储-云盘架构



- 宿主机宕机-可通过云盘挂载在另一台宿主机上快速拉起虚机
- 后端故障 - 可以切到副本上

块数据保护

——让数据失而复得

浅谈数据保护的方式

- 发生问题时需要回滚，发现很久没有怎么使用这套备份机制，回滚异常
- 回滚时时间消耗较大，7-8小时，甚至超过一天
- 回滚成功，但回滚的时间点已经是1天甚至几天前

数据保护的目标

RPO
(复原点目标）

RTO
（复原时间目标）

- RPO 数据可以恢复到哪个时间点
- RTO 数据恢复要消耗多长时间

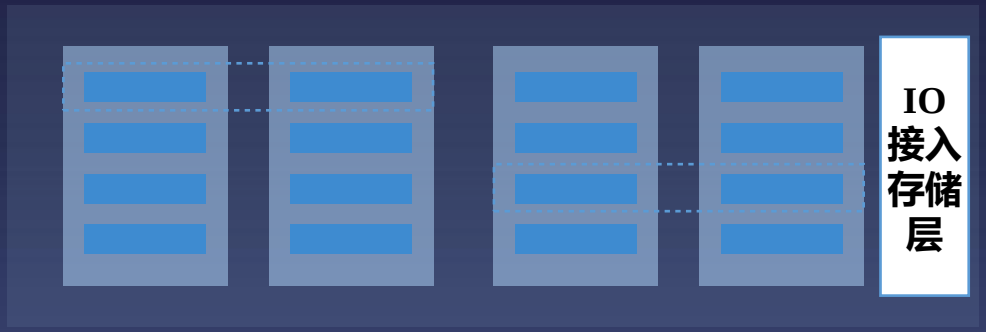
架构设计的目标

- 恢复时保留原盘
- 实时IO的接入能应对云环境下的IOPS
- 成本上要有一定的衡量
- 利用分布式存储和计算加快恢复的速度

总体架构

分层混合存储

- SSD层 可以满足云环境下海量IO写的需求
- HDD 层 通过程序上的优化，顺序读写发挥HDD的效率
- SSD层容量小，HDD层容量大，借助中间处理层传输数据



CPU

IOPS

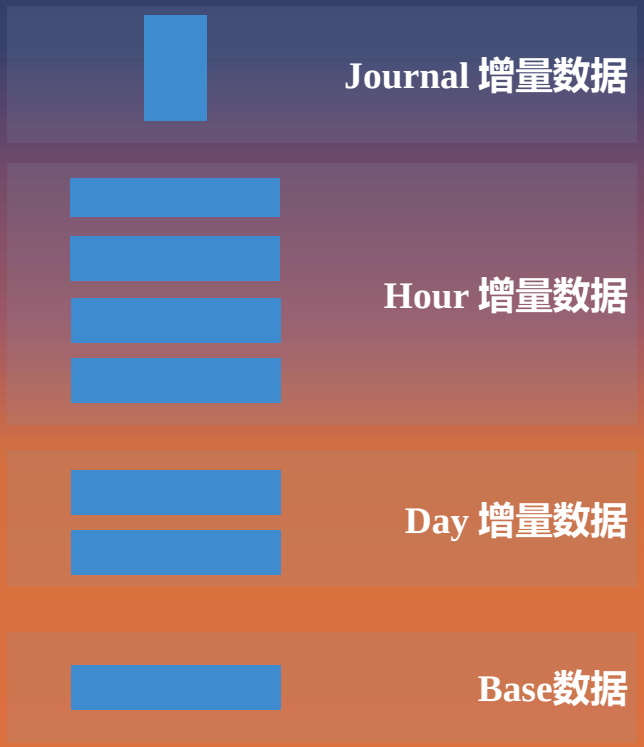
磁盘容量

IO 带宽

网卡带宽

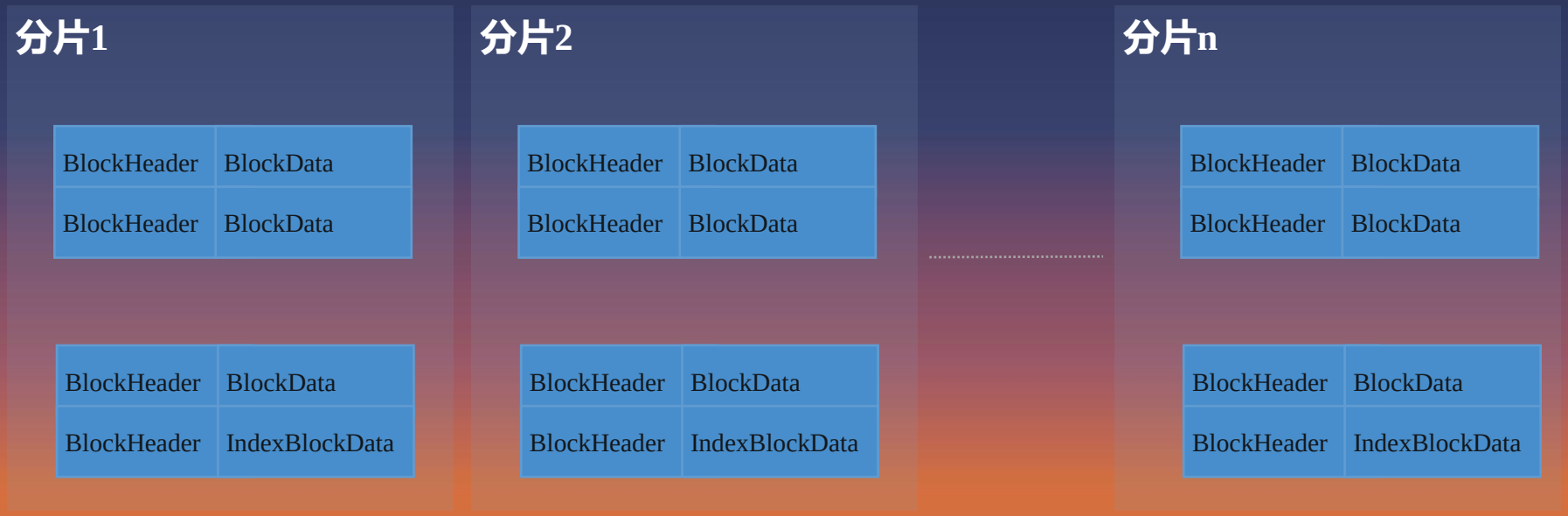
- io接入层动态漂移的能力
- 中间处理层要适应io接入层动态迁移

增量的存储方式



- 增量数据存储方式
- 按一定时间粒度或者大小去做单位切割
- 每个单位都进行分布式存储

一段时间粒度的数据组成

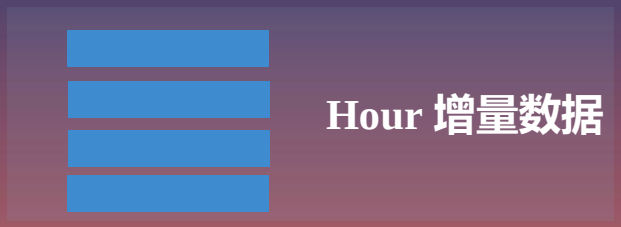


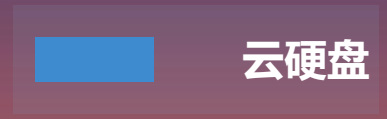
- 每种时间粒度的数据都按统一格式存储，统一解析
- 利用压缩优化成本

调度模块触发分布式计算

调度模块

- 分布式存储
- 分布式计算



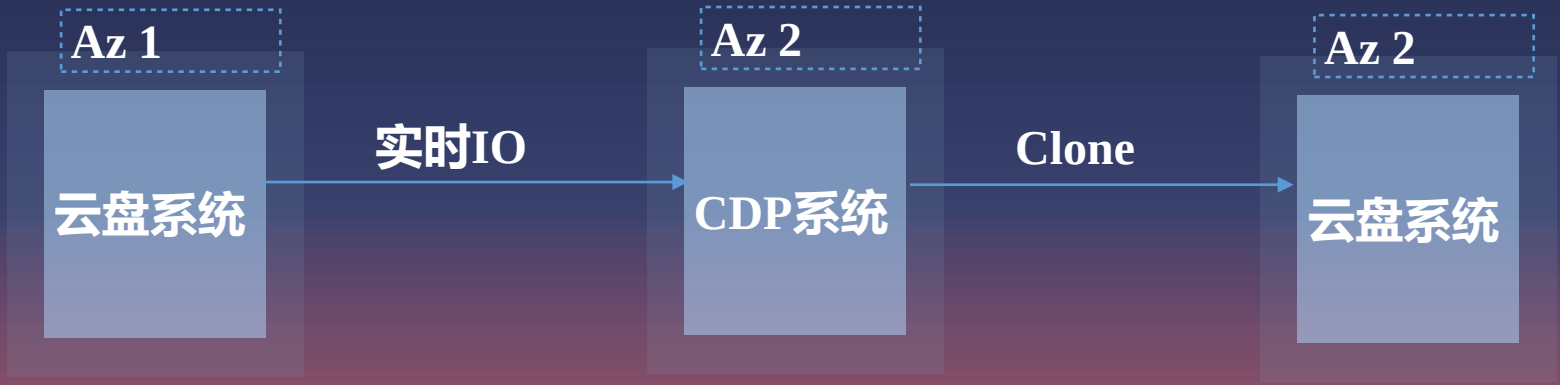


- 从公有云运营的角度，数据恢复在后端都是恢复到新盘，让客户更放心回滚，也能更好的保护数据

UCan 下午茶 悟 有所悟

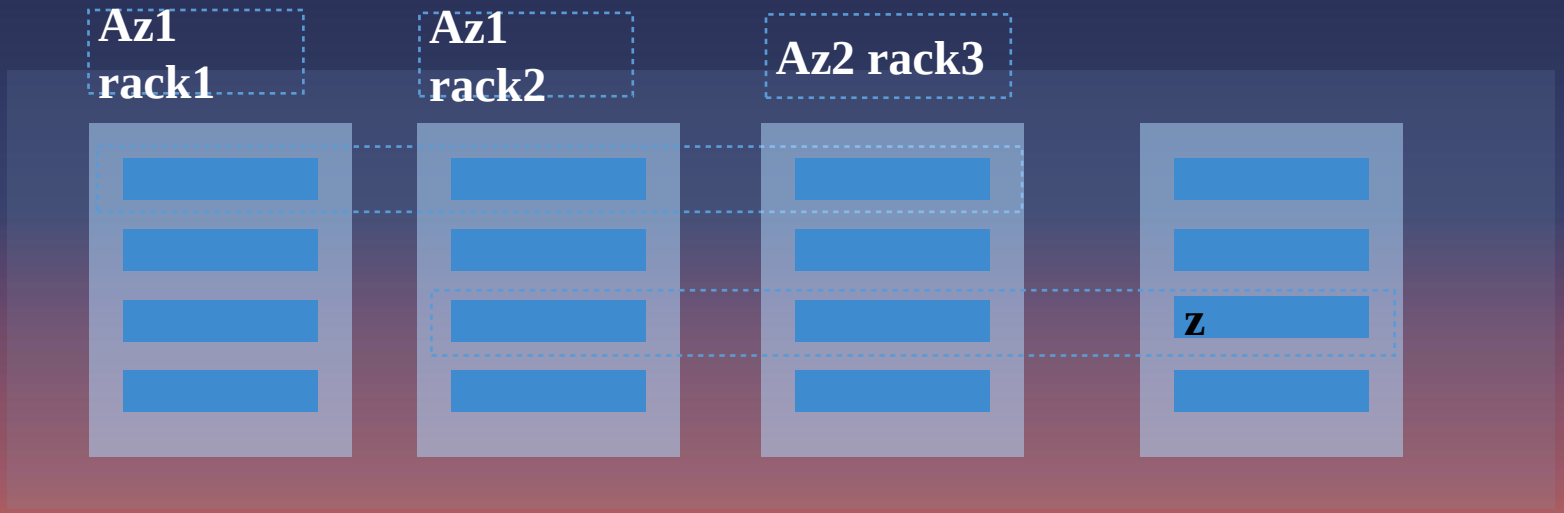
为了 更好的高可用

UCLOUD



- AZ1 - AZ2 实时IO异步推送
- 机房级别故障时，拉起业务需要等待clone
- 恢复业务时会有秒粒度数据丢失

- 不影响云盘性能
- 成本更低



- 云盘自身跨机房容灾
- 故障时直接切换
- 延迟增加1MS
- 成本较高

Thanks

Q&A