

[www.qconferences.com](http://www.qconferences.com)  
[www.qconbeijing.com](http://www.qconbeijing.com)



伦敦 | 北京 | 东京 | 纽约 | 圣保罗 | 上海 | 旧金山

London . Beijing . Tokyo . New York . Sao Paulo . Shanghai . San Francisco

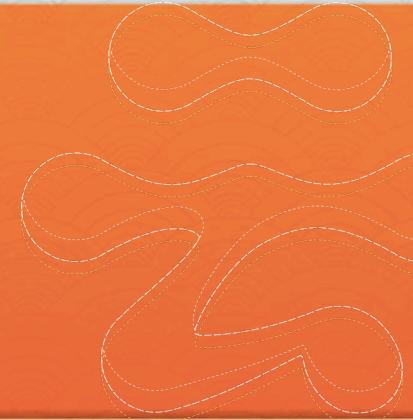


QCon全球软件开发大会

International Software Development Conference

# 阿里云计算的实践

倪浩

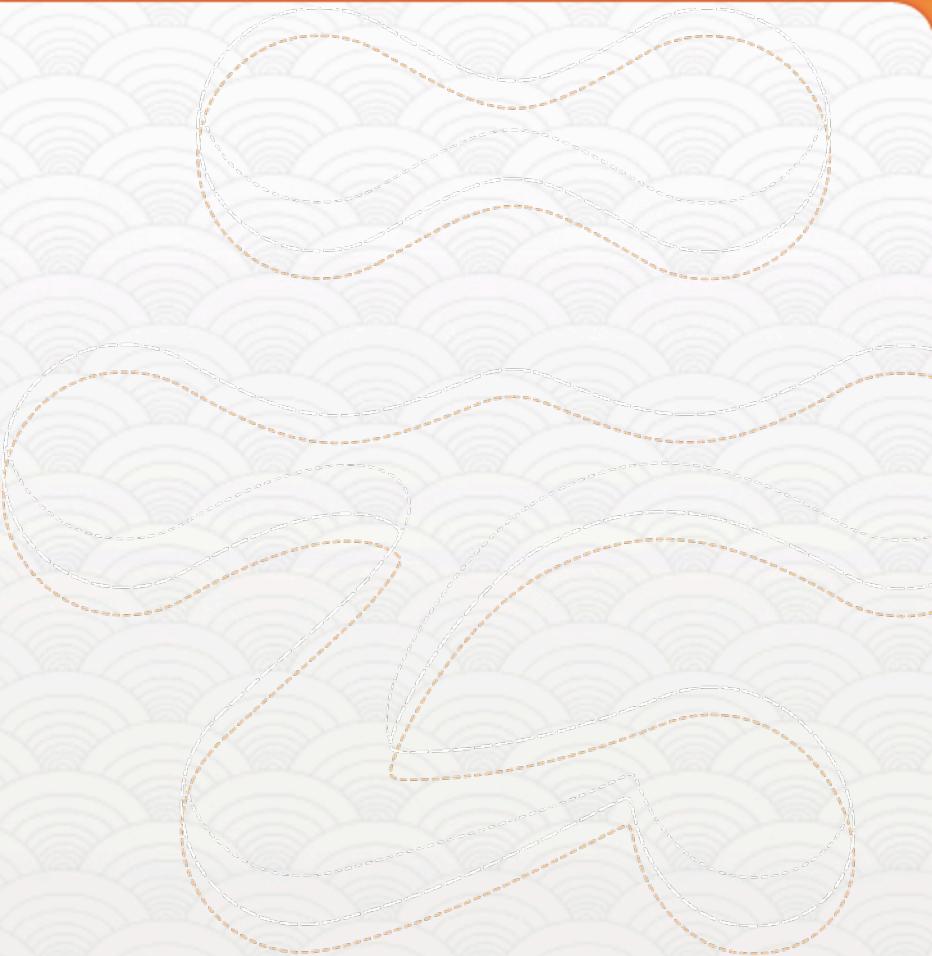


# 大纲

阿里云服务体系介绍

弹性计算和存储技术演进分享

最佳实践



## 第一部分：阿里云服务体系介绍，技术路线选择



## 弹性计算

1. 云服务器(ECS)
2. 负载均衡(SLB)
3. 云盾 & 云监控



## 存储和数据库服务

1. 开发存储服务(OSS)
2. 开放结构化数据服务(OTS)
3. 关系型数据库服务(RDS)
4. 未来 : Cache/Queue/CDN

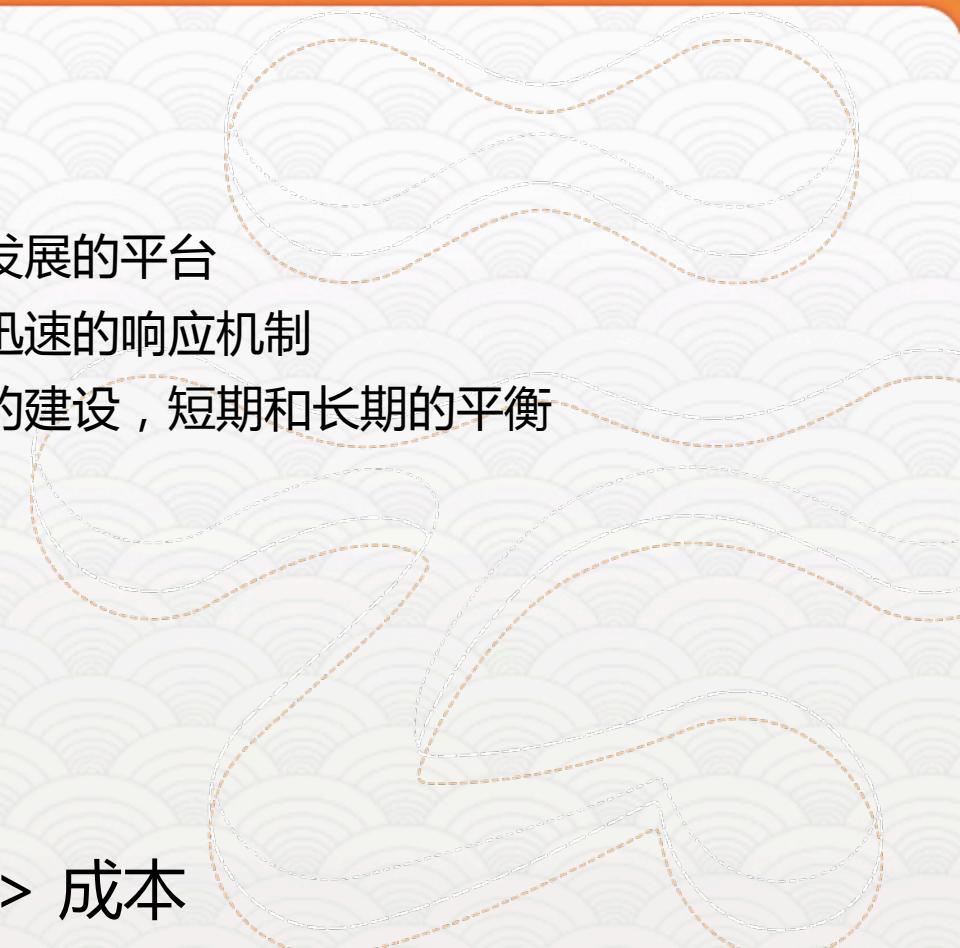


## 大规模计算

1. 开放数据处理服务(ODPS)

# 一个开放的服务体系

- 公共的基础云计算服务
  - 技术上
    - 系统内核：一个健壮可以持续发展的平台
    - 运维体系：完备的运维机制、迅速的响应机制
    - 实施：机房的地域规划、网络的建设，短期和长期的平衡
  - 非技术问题
    - 在线的使用、支付方式
    - 响应迅速的客服、运维体系
    - 备案等相关服务
- 对用户的价值：灵活 > 简单 > 成本



# 为什么我们选择从头做？

- CloudStack, OpenStack, Eucalyptus, Hadoop, Mongo
  - 非常优秀的开源软件，但是.....
- 缺点
  - 当你的业务发展大了，缺乏主线的控制力
  - 做一个完整的云计算产品体系，用开源软件拼凑起来很困难
  - 各种软件几乎不可能共享集群的资源
- 从头走的难点
  - 从头做，非常辛苦，很多坑要踩

# 飞天：阿里云计算的内核

## Aliyun API

开放存储服务  
( OSS)

开放结构化数  
据服务(OTS)

开放数据处理  
服务  
(ODPS)

弹性计算  
服务  
( ECS)

关系型数  
据库服务  
( RDS)

集群布署 Deployment

分布式文件系统  
Distributed File System

任务调度  
Job Scheduling

分布协同服务  
Distributed  
Coordination  
Service

安全管理  
Security  
Management

远程过程调用  
Remote  
Procedure Call

资源管理  
Resource  
Management

集群监控 Monitoring

Linux

数据中心 Data Center

## 第二部分：弹性计算和存储技术演进分享

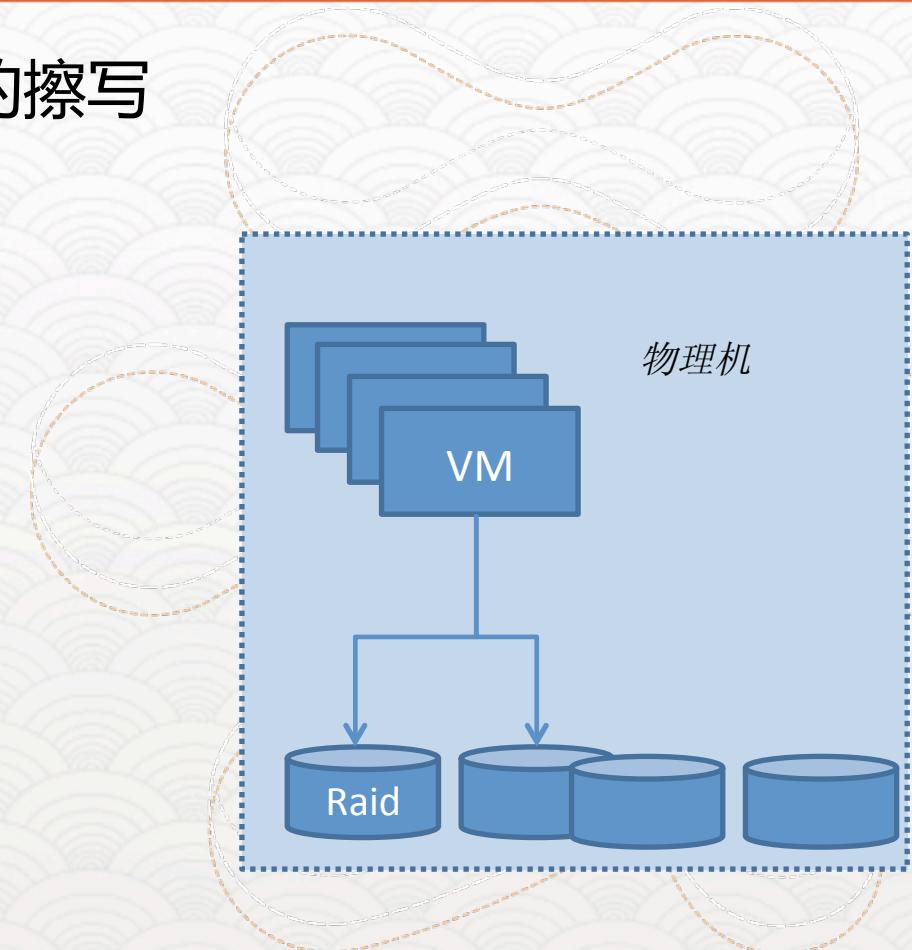
# 开放服务的技术架构：云服务器为例

- 云服务器
  - 稳定性：数据不能丢（或出错）
  - 一致性：地址不能变
  - 安全性：防御攻击
- 数据不能丢：Raid , SAN , KeyValue存储 , 分布式**存储** ?
- 地址不能变：**NAT** , 大二层**网络** ?
- 防御攻击：交给用户 , 自动**安全**防御 ?

# 云服务器的存储 : Step 1



- 特点 : 大量的随机IO , 频繁的擦写
- 使用Raid : 无法满足需求
  - 数据在本地 , 容机无法迁移



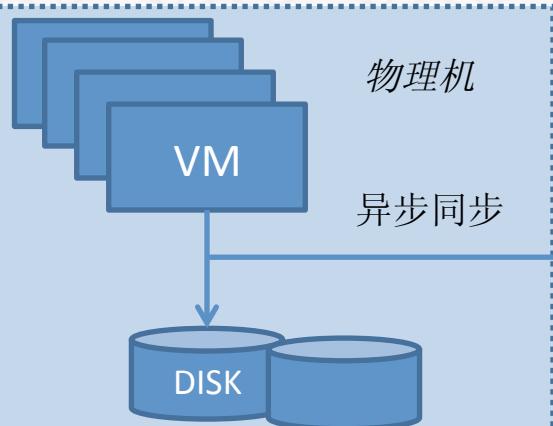
# 云服务器的存储 : Step 2



- 本地存储 + KVEngine (Based on 飞天盘古 )
  - Runtime的读写发生在本地，同时异步向KeyValue同步(以扇区为单位)
  - 本地宕机时，通过KVEngine中的数据在另外一台机器上恢复

## • Why KVEngine

- 本身是开源的
- KVEngine是阿里云自研的引擎
- KVEngine是飞天盘古的一个子模块



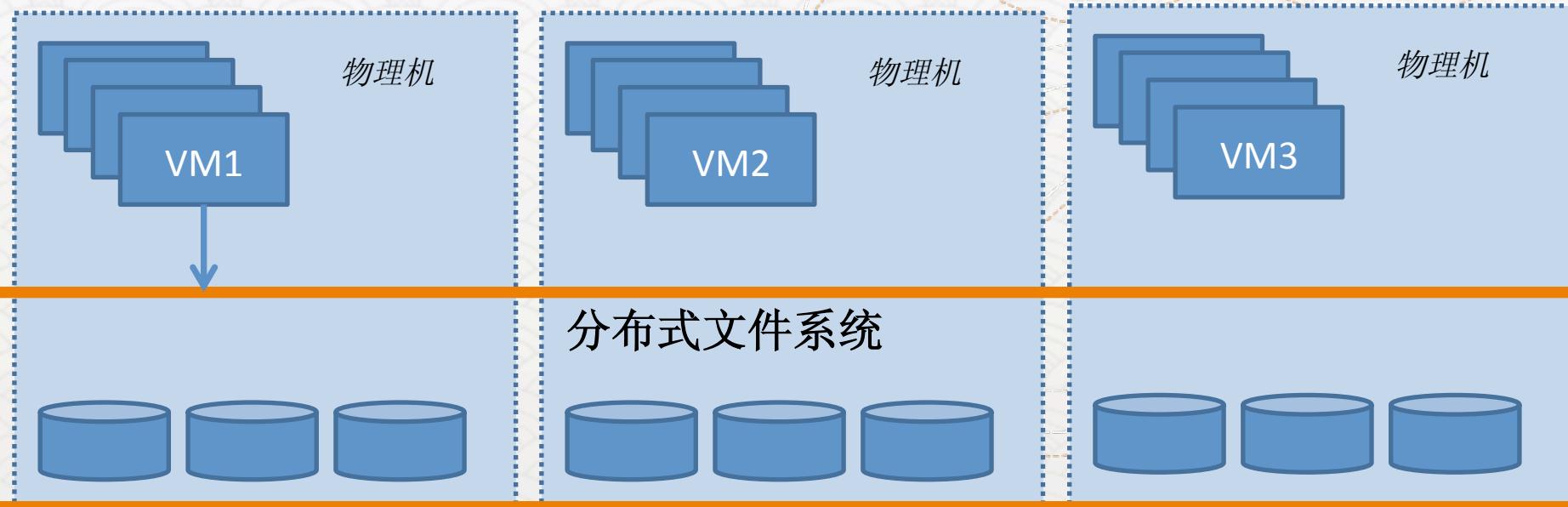
## • 带来的问题

- KVEngine读写的性能
- 异步同步数据仍然可能丢失
- 成本太高

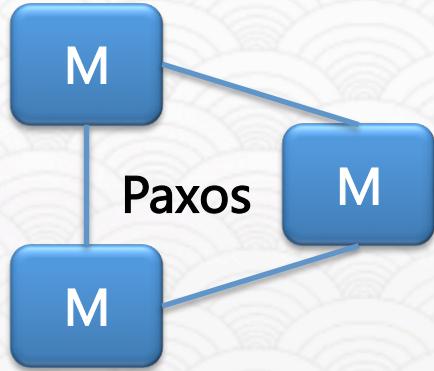
解决了随机写的问题  
冗余存储



- 再次考察云服务器存储的特性：
  - 大量的随机IO，频繁的擦写
  - 任何时候不丢失数据
- 结论：一个分布式（分片+冗余）支持随机读写的存储系统



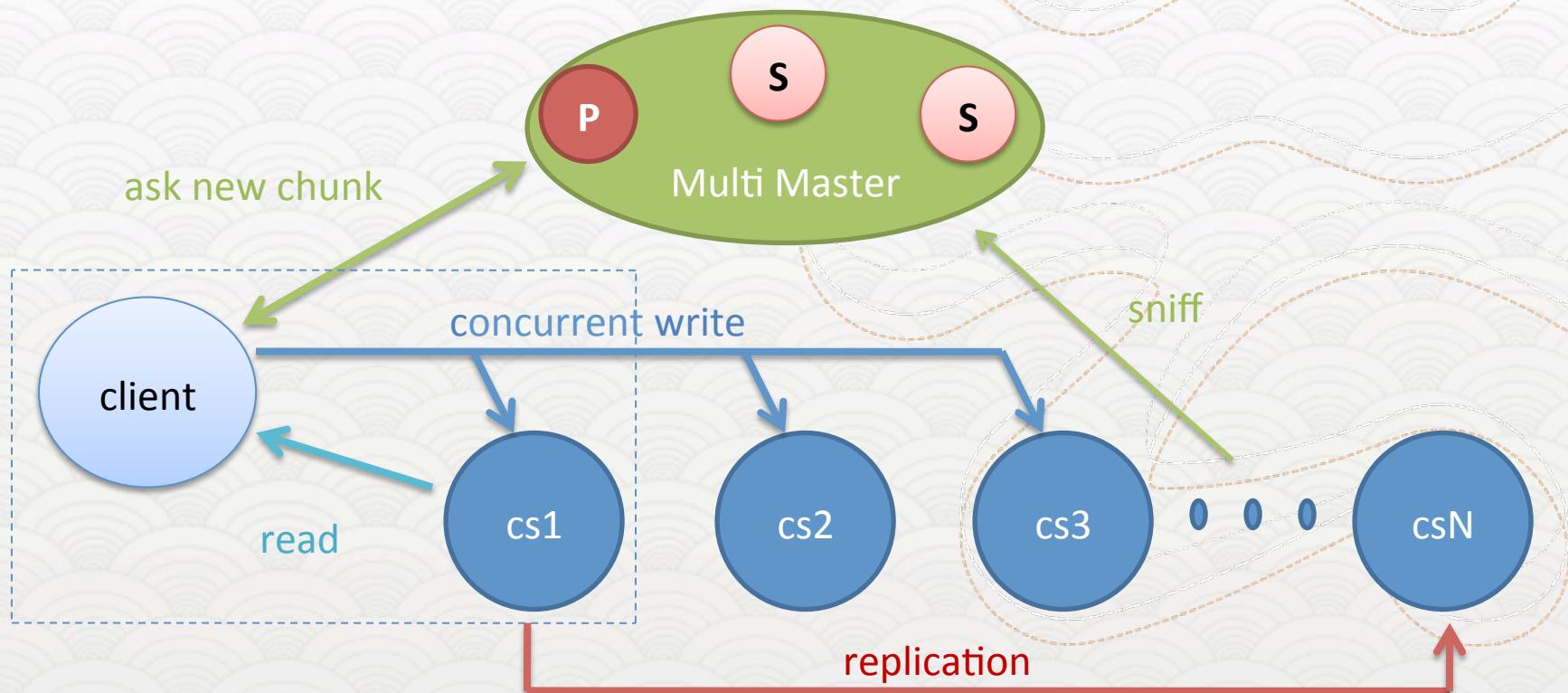
# | 盘古：分布式文件系统 (Append-Only)



- Master-Slave 主从架构
  - Master负责元数据管理，Slave(Chunk Server)负责读写
- 基于Paxos的多Master架构，故障恢复小于一分钟
- 文件分片(chunk)，每个chunk存三份副本，分布于不同机架
- 端到端的数据校验

# RAF : 让分布式文件系统支持随机写

- RAF : Random Access File
- 最复杂的问题 : 数据一致性

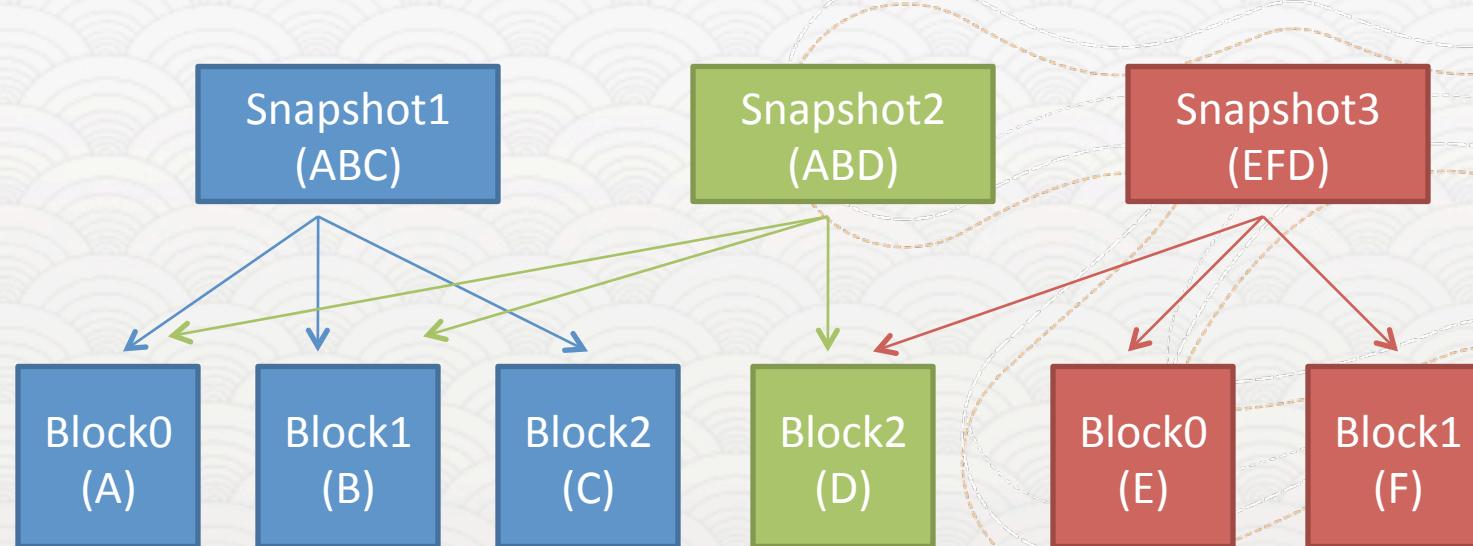


# RAF一些工程上的经验

1. 一致性是最重要的问题，值得花费最多的精力，首先要在理论模型上成立
2. 仔细的去设计数据的分布(Placement)：例如虚拟机的本地读数据机制
3. 做好流控，绝不失控，避免出现复制风暴
4. 在有的数据拷贝丢失（如磁盘故障）开始数据复制时，平衡好新写入数据和复制速度之间的关系

# 云服务器的快照和镜像

- 存储在开放存储服务（OSS）中，便于跨集群、跨地域访问
- 增量快照的机制：计数应用



- 网络：大二层网络设计，避开VLAN的限制

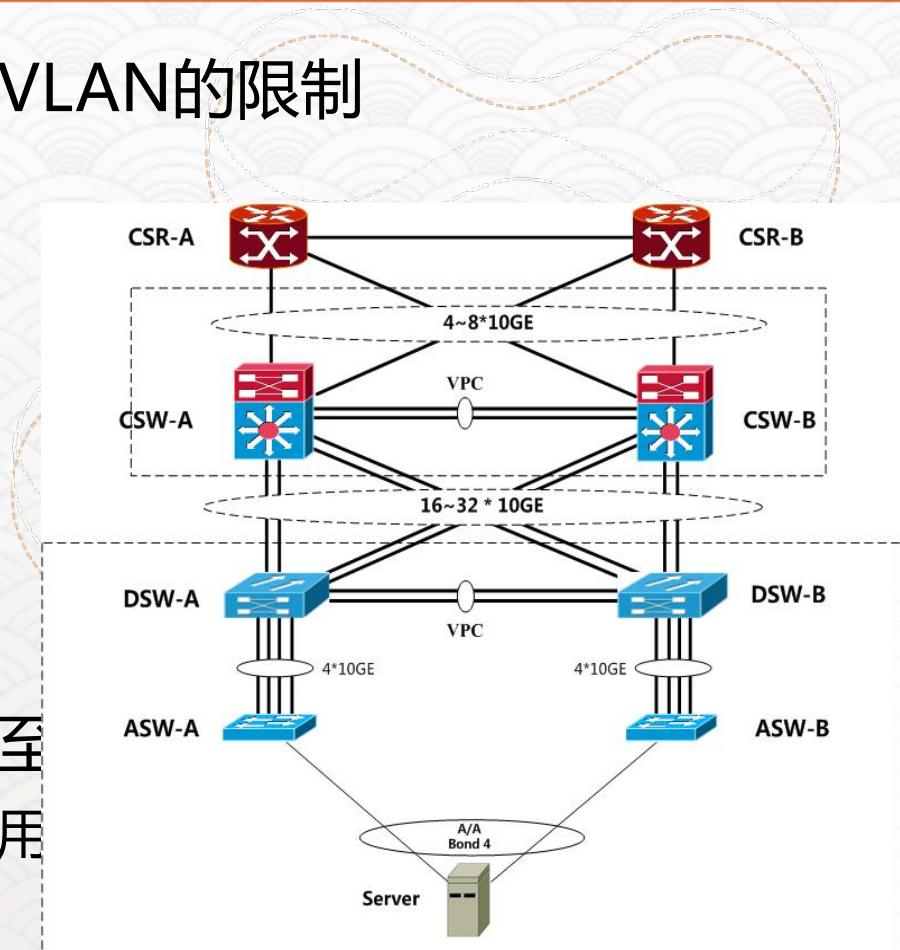
- 网络双活设计
  - 对外的连接全部为多线BGP接入

- 安全是用户无法去解决的问题

- 暴力攻击
  - 系统漏洞

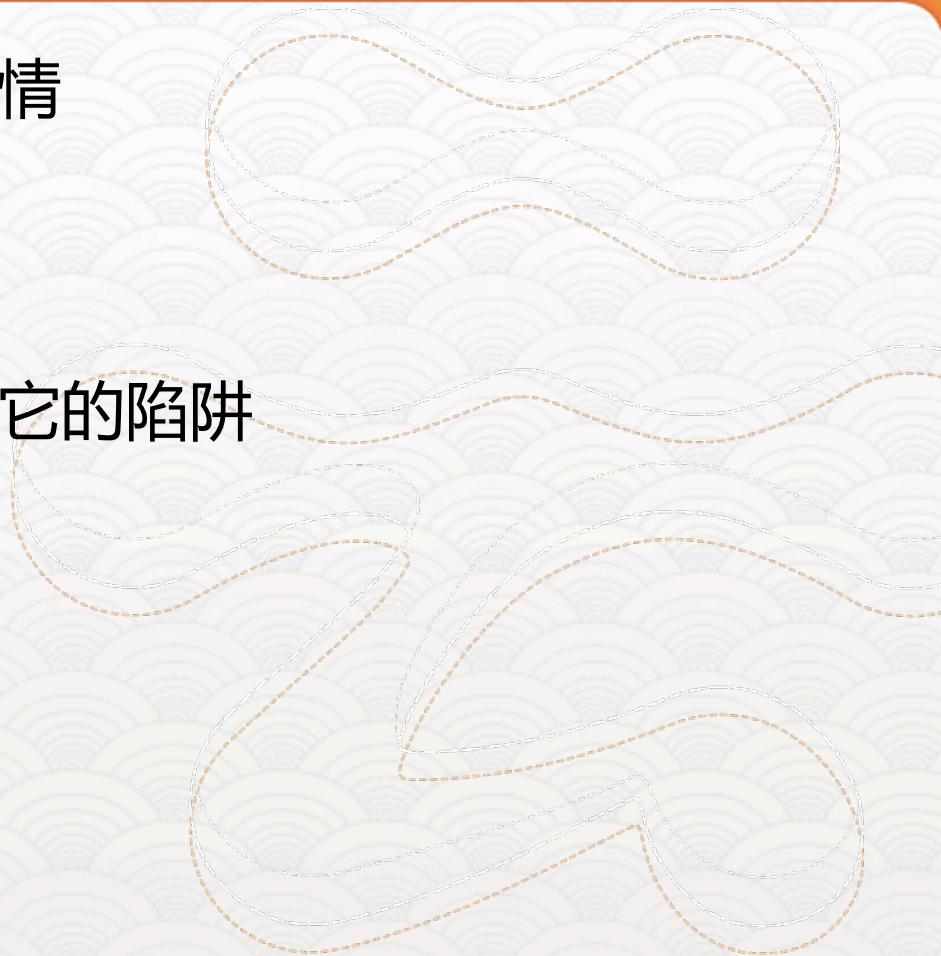
- 提供一个全面的安全防御体系至

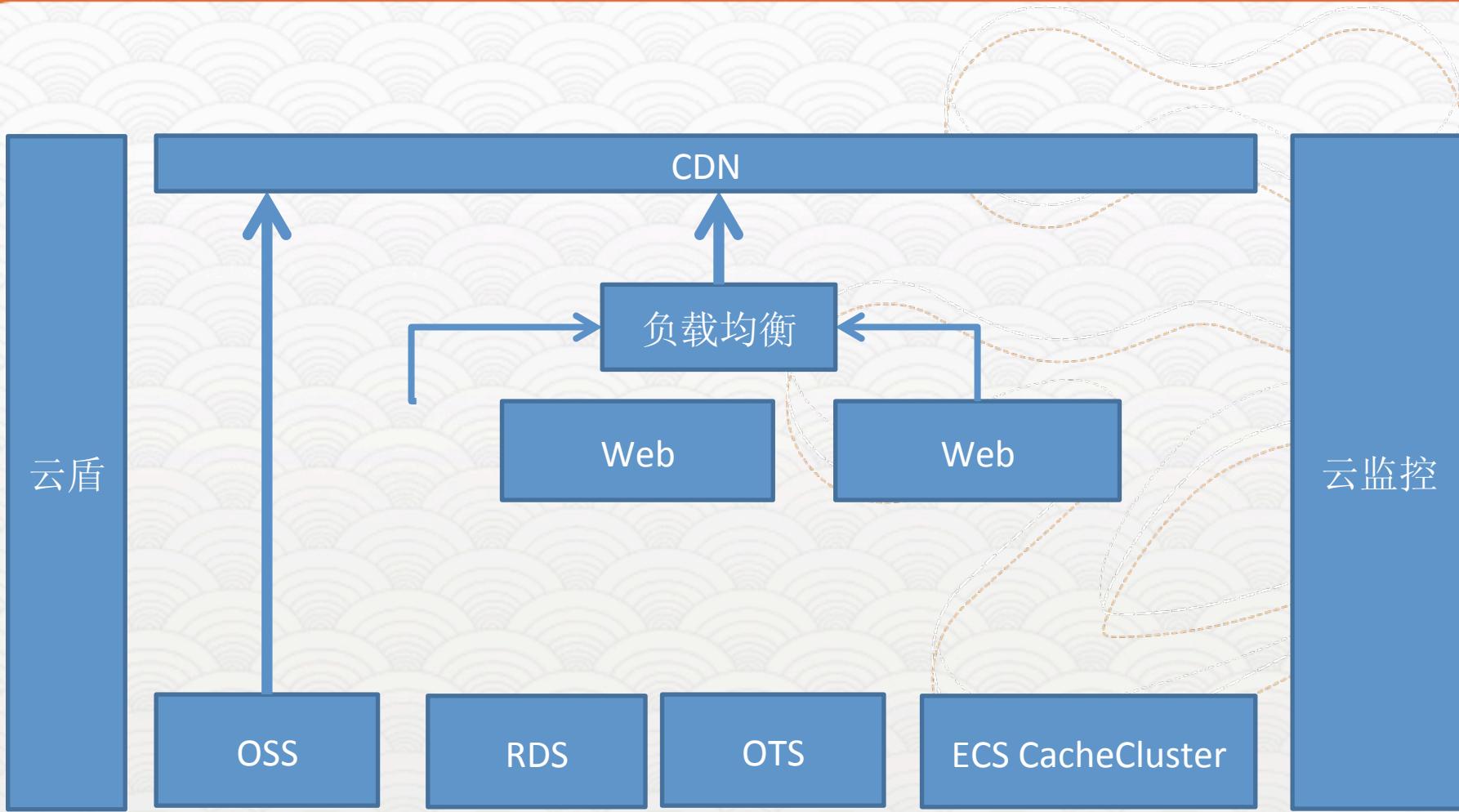
- 不仅仅是防御攻击，还需要告知用



## 第三部分：最佳实践

- 每个服务被设计为做好一件事情
- 每个服务都有一个SLA
- 利用好每个服务的优势，避开它的陷阱





# 需要知道的事情

- 永远使用云监控来监控你的应用及服务
- 云服务器
  - 宿主物理机宕机，故障迁移需要几分钟的时间
  - 按月购买的服务器的带宽是受限的，但上行带宽很大
  - 如果对磁盘的读写很重，考虑把压力转移到OSS、OTS、RDS
  - 如果能做到应用无状态，就能够结合SLB做完全水平(session保持)
- OSS
  - 把它作为一个带宽不受限，空间不受限，并发不受限的在线存储
- RDS
  - 具备优越的读写性能（FusionIO），但它的总数据量要小于1TB
- SLB
  - 考虑好是用HTTP还是TCP，配置非常简单
- OTS
  - 如果你能将不需要关系型操作的结构化数据放到OTS，一定能大大减轻数据库的负担

# 阿里云的用户



谢谢！

[www.infoq.com/cn](http://www.infoq.com/cn)

# InfoQ Queue



@InfoQ



infoqchina

软件  
正在改变世界！