云计算中的服务质量保障与资源隔离

汪源
网易.杭州研究院.副院长

网易私有云介绍

- 目标：为公司主流的大量WEB类产品提供统一的云计算平台，以：
  - 提高硬件资源利用率，促进资源共享，从而降低硬件成本
  - 提高资源管理与系统运维的自动化水平，从而降低运维人员成本
  - 提高资源使用弹性，从而增强对业务波动的适应能力
  - 促进公共技术平台的研发与应用，从而使业务获得更好的基础技术服务
- 功能
  - 以提供虚拟硬件资源的IaaS服务为核心
  - 7大服务：云主机(NVS)、云硬盘(NBS)、对象存储(NOS)、关系型数据库(RDS)、分布式数据库(DDB)、云搜索(NCS)、云监控
- 历程
  - 2012.Q2开始研发
  - 2012.Q4，网易相册与网易云课堂正式上线
  - 2013.Q2，网易博客正式上线
  - 13个产品，500+云主机

网易云 beta
cloud.163.com

上午好，wangyuan　安全退出 | 帮助

云主机 | 云硬盘 | 对象存储 | 关系型数据库 | 分布式数据库 | 云监控 | 云搜索

- 云主机首页
- 云主机管理
- 镜像管理
- 安全组管理
- 快照管理
- 网络资源管理
- 密钥管理

云主机首页

创建云主机

云主机是网易提供的云端计算服务，用户可以根据自己的需求创建云主机，系统负责对云主机进行全生命周期管理。点击下面按钮来创建云主机。

创建云主机

我的配额

当前云主机服务中包含的配额如下：

刷新

云主机配额（台）：5	云主机CPU配额（个）：10
ECU配额（个）：80	内网浮动IP配额（个）：10
外网浮动IP配额（个）：2	安全组配额（个）：10
实例存储配额（GB）：200	内存配额（MB）：20480
外网带宽配额（Mb/s）：无	

我的资源

当前云主机服务中包含的资源如下：

刷新

云主机（个）：0	云主机CPU（个）：0
ECU（个）：0	内网浮动IP（个）：0
外网浮动IP（个）：1	安全组（个）：1
实例存储（GB）：0	内存（MB）：0
外网带宽（Mb/s）：0	

相关链接

- 特性介绍
- 用户手册
- 产品价格
- 配额申请

应用环境与需求

- 需求：三类用途
  - 产品生产环境

    数据规模中等到较大，负载较高

    重视性能、可靠性、可用性等服务质量

    易受攻击
  - 研发测试

    规模小，负载低

    重视成本经济性
  - 性能测试

    规模较大，负载高

    重视性能稳定性

    避免影响产品生产环境服务质量
- 硬件
  - 充分利用各种现有硬件资源，规格不一（CPU Intel/AMD、网络千兆/万兆、机型 刀片/机架等）
- 软件
  - 类型复杂，上层软件架构的可伸缩性、可靠/可用性一般
- 用户：公司运维团队
  - 对云计算没有不切实际的要求
  - 能理解较复杂的概念，掌握较复杂功能的使用
  - 不会恶意搞破坏

服务质量保障

- 用户视角
  - 提供高性价比的质量恰到好处的服务（而不是最高质量的服务）
  - 给予用户充分的选择权与控制权
  - 要有明确的SLA（SLA不一定达到很高的水平）
  - QoS需求点

    性能：云主机计算性能、网络带宽(内外网)、存储IOPS、存储IO带宽、稳定性

    可靠性：存储

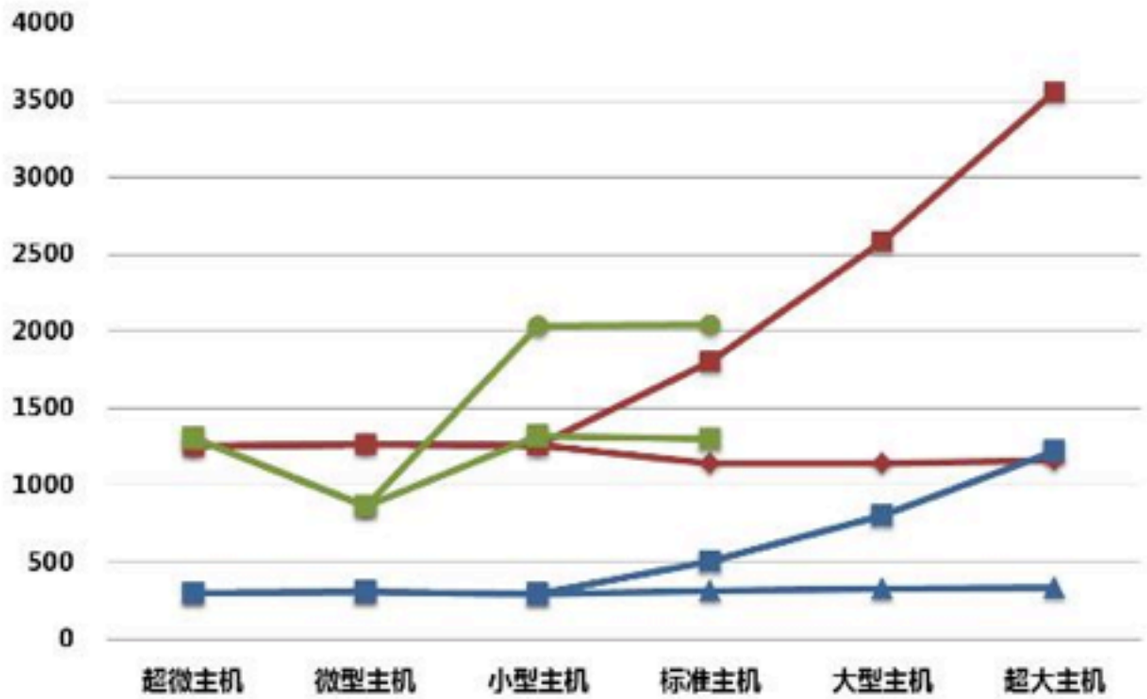
    可用性：云主机、关系型数据库、分布式数据库、云搜索等
- 平台视角
  - 控制用户资源占用，避免相互影响
- 需求分析和产品策略是关键，技术跟着产品策略走

性能QoS

- 目标：提供性能指标明确、稳定、丰富、灵活可配的资源规格
- 挑战：资源共享，由此带来：
  - 性能指标体系需要重定义（要有利于用户使用、提高资源利用率、资源调度和运维）
  - 探索新指标体系的实现方式
- 措施
  - 制定ECU策略作为计算能力衡量标准
  - NOS资源隔离
  - 网络带宽QoS
  - NBS QoS

ECU：VM配置符号化

基准测试 – UnixBench



	CPU	内存	月价格
超微	1	0.5 GB	34.75
微型	1	1 GB	69.5
小型	1	2 GB	139
标准	2	4 GB	278
大型	4	8 GB	556
超大	8	16 GB	1112
阿里EA	1	0.5 GB	99
阿里EB	1	1.5 GB	199
阿里A	2	1.5 GB	399
阿里B	2	2.5 GB	599

* 阿里云包含带宽价格

实验云（红色系列）
Xeon E5620 @ 2.40 GHz
4 核心 x 2 线程
性能：4693 （ 586 / 线程）

盛大云（蓝色系列）
Opteron 6172 @ 2.10 GHz
12 核心
性能：7784 （ 648 / 核心）

阿里云（绿色系列）
Xeon E5645 @ 2.40 GHz
6 核心 x 2 线程
性能：6906 （ 576 / 线程）

云主机的配置已经退化成一个符号，而不是具有明确含义的参数。

by 蒋清野

ECU：需求与目标

- 用户期望：使用ECU精确衡量云主机的计算能力
  - 已有云主机的计算性能无大幅波动

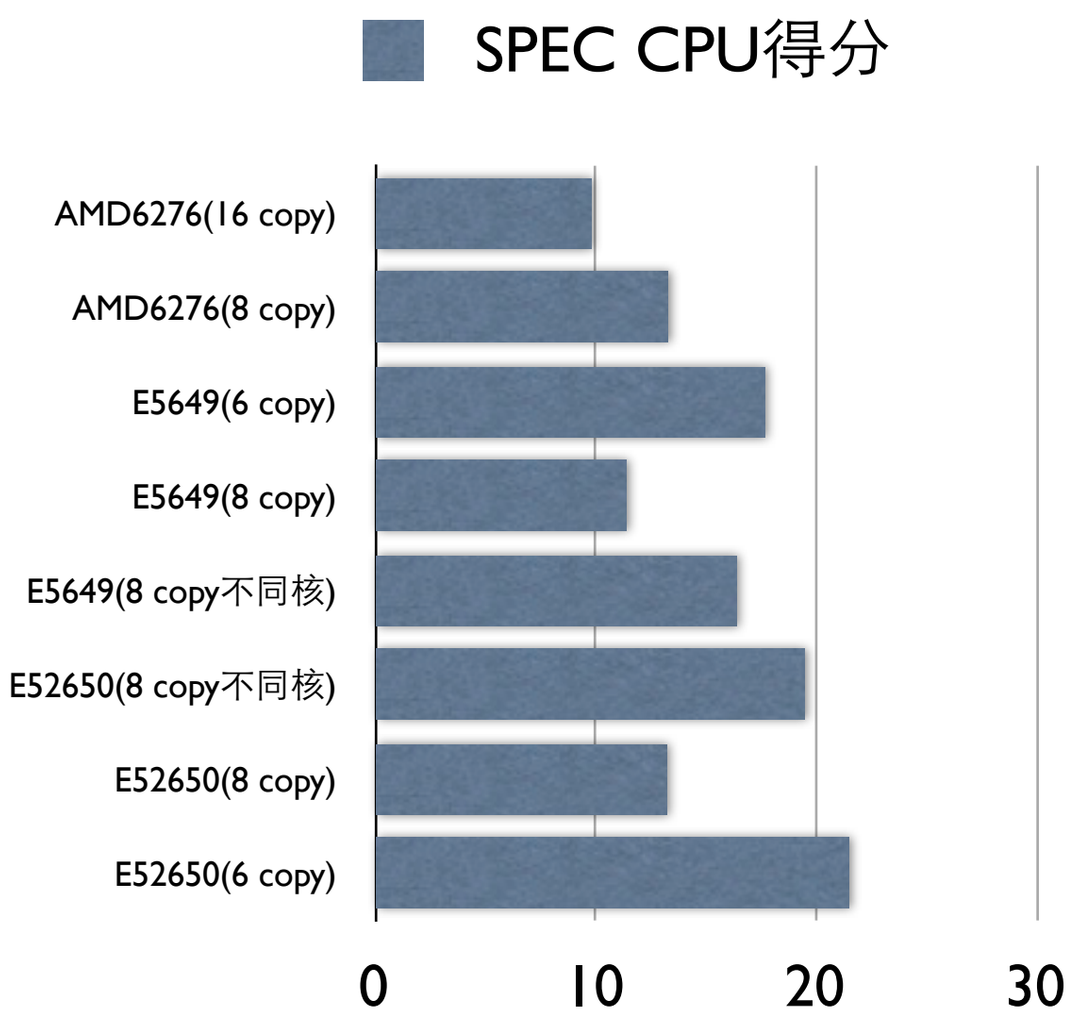
    保障服务质量，降低运维成本
  - 无论用什么机型、CPU、内存，相同ECU的云主机的计算性能相近

    可以自由的调整云主机物理位置而不担心性能不足
  - 云主机的计算性能近似与ECU成正比

    可以根据现有负载决策scale-up/down策略
- 目标
  - SPEC CPU整型计算性能得分与ECU成正比，波动幅度不超过±10%

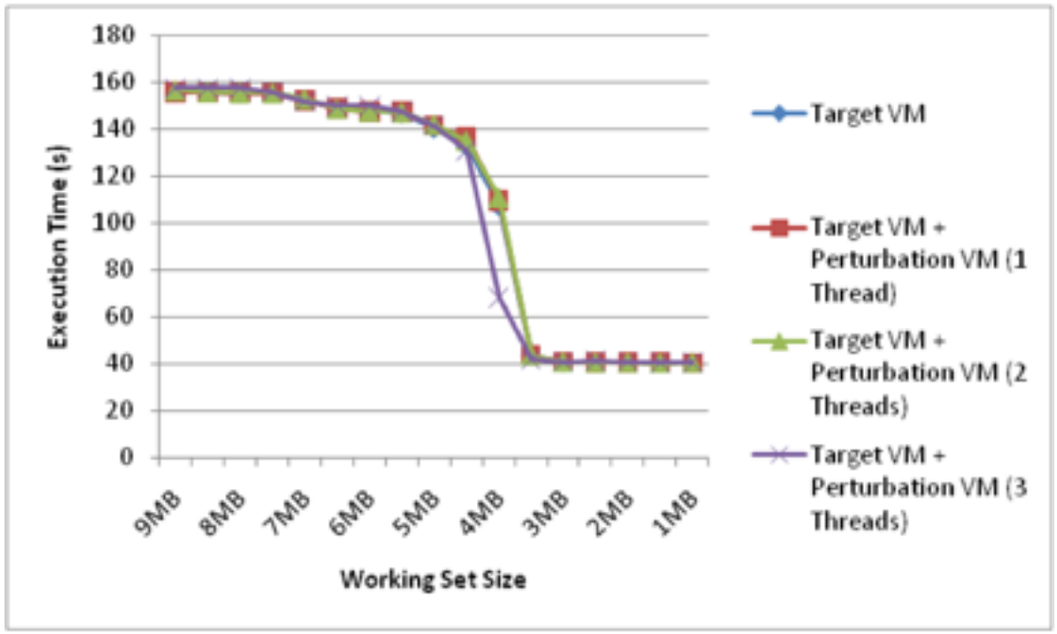
ECU：技术挑战

- 不同CPU的每核(每线程)计算能力不同
- 相同CPU在不同的系统总体负载下每核(每线程)计算能力不同
- 相同总体负载相同CPU，独占核还是两线程共享核计算能力不同
- 波动幅度远远超过10%



ECU：实现机制

- cgroup控制策略
  - cpu.cfs_quota_us/ cpu.cfs_period_us
  - cgroup机制能够全范围控制VM性能，关键是具体策略
- CPU L3 cache isolationg/page coloring
  - 能有效解决VM性能受其他VM和Host总体负载影响问题
  - 难以实施
  - 较大幅度的性能下降

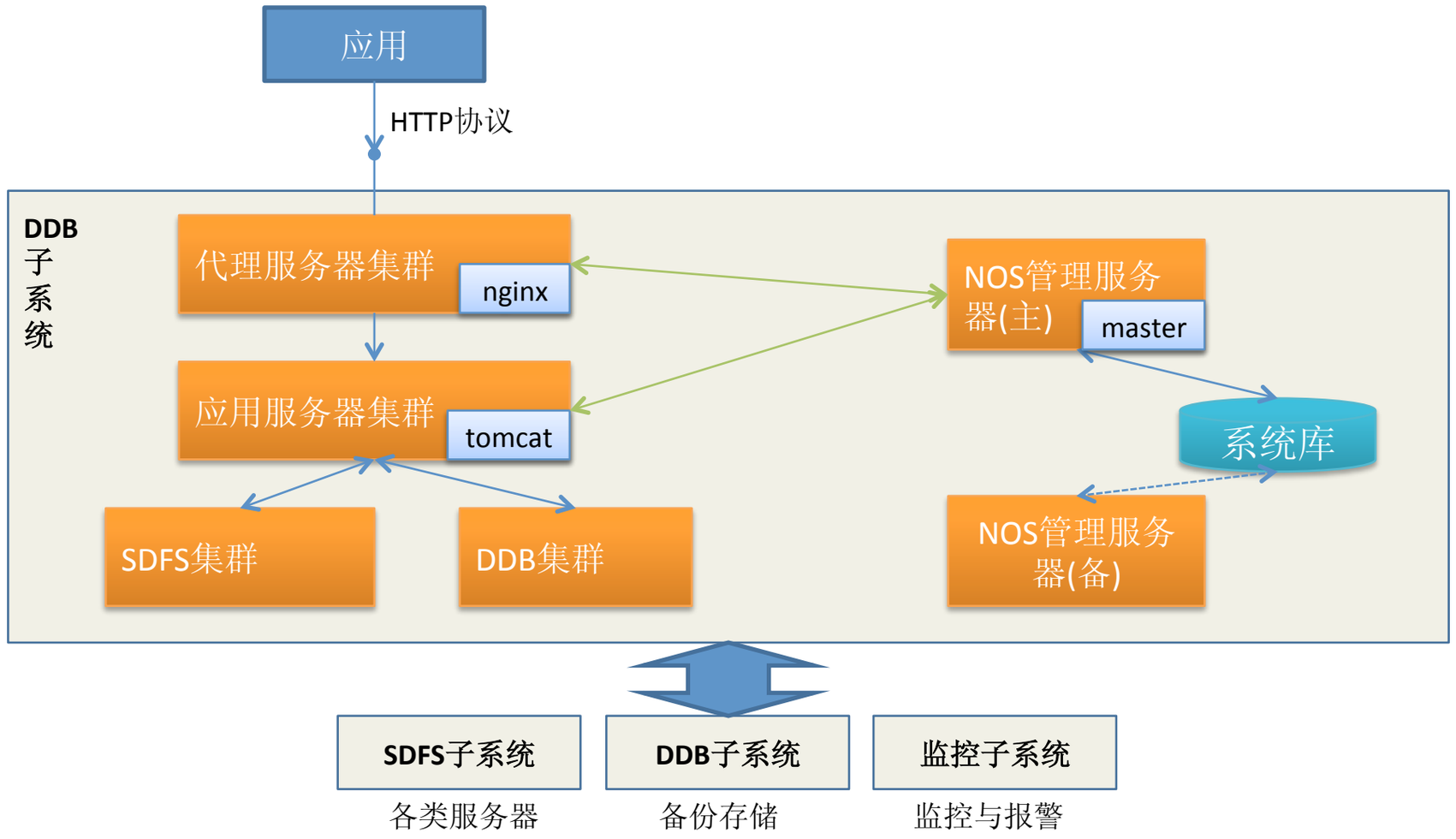


Himanshu Raj et al, Resource Management for Isolation Enhanced Cloud Services

ECU：解决方案

- VCPU范围绑定到PCPU
  - VCPU绑定到PCPU会大幅降低计算性能，幅度达20%以上
  - 给Host预留一些核以保障系统控制、网络、云硬盘IO等性能
- 以典型Host CPU负载（30%）为测试基准
  - 使用SPEC CPU 2006测试VM整型计算能力得分，测试时Host的CPU负载30%左右，被测VM CPU负载100%
- 进行大量测试，确定各种规格VM在各种机型上的cpu.cfs_quota_us参数值
  - cpu.cfs_period_us统一设置为100ms，测试发现设置为10ms时系统不能准确控制VM的CPU占用和性能
  - 一般小规格VM要在理论值之上调高cpu.cfs_quota_us才能达到预期性能
  - 打开THP可提高10%左右性能
  - cgroup控制在某些情况下可能影响到VM网络性能稳定性，还在继续研究

NOS资源隔离：架构


应用
HTTP协议
DDB
子
系
统
代理服务器集群
nginx
应用服务器集群
tomcat
SDFS集群
DDB集群
NOS管理服务
器(主)
master
系统库
NOS管理服务
器(备)

SDFS子系统
各类服务器
DDB子系统
备份存储
监控子系统
监控与报警

NOS资源隔离：方案

- 需求
  - 限制指定“桶”总下行带宽、并发连接数、QPS和单连接下行带宽
  - 实时统计指定“桶”总下行带宽、并发连接数、QPS
- 技术路线：在nginx层实现
- 普通方案及其问题
  - limit_conn_zone/limit_conn, limit_req_zone/limit_req, limit_speed_zone/limit_speed
  - 问题：

    只能限制单台nginx服务器

    修改限制后，需要reload nginx配置才能生效，无法做到及时生效

    运维人员难以获得限制信息

NOS资源隔离：方案

- 改进方案
  - 基于ngx_lua模块，便于实现复杂逻辑与架构
  - 使用ngx.share.DICT在worker间共享内存，以便统计“桶”级并发连接数等汇总信息
  - 通过在access_by_lua/log_by_lua中注入hook代码统计并发连接数
  - 通过在access_by_lua注入代码统计QPS
- 全局控制
  - 设计全局LimitServer统计汇总、下发限制指令
  - 各nginx与LimitServer定期心跳汇报统计信息
  - LimitServer不可用时，各nginx遵循既有指令正常工作
  - 先平均分配（可增加一定余量），根据汇总信息发现负载不均智能调整

网络带宽QoS

- 需求分析
  - 外网带宽：成本高，需要精细控制，用户有认知
  - 内网带宽：成本低，用户缺乏认知，无需精细控制
- 设计
  - 外网带宽：配额控制，创建云主机时由用户指定
  - 内网带宽：无配额，根据VCPU与内网带宽统计数据制定较宽松默认控制策略，创建云主机时无需指定
- 实现
  - nova.conf配置每台服务器可用内外网总带宽
  - nova调度带宽filter
  - 通过tc控制VM内外网带宽： rate/ceil/burst

NBS QoS

- 需求
  - 大容量存储，可靠性、性能要求不高
  - 开发、测试所用，容量、可靠性、性能要求不高
  - 线上数据库、搜索所用，可靠性、性能要求高
  - 可用性要求不是太高
- 困难
  - 磁盘共享时性能相互影响严重
  - 网络带宽比较紧张
- 方案
  - 提供“独占式”和“共享式”两种云硬盘
  - 使用单机RAID 1提供可靠性，降低网络带宽占用
  - 创建云硬盘时要求指定要挂载的云主机，以便调度云硬盘到与云主机网络条件较好的主机
  - 区分需要挂载和不需要挂载云硬盘的云主机，不需要挂载云硬盘的云主机优先调度到刀片等网络带宽低的主机

可靠性可用性QoS

- 目标：满足不同应用的不同可靠性和可用性需求
- 措施
  - 云主机可用域
  - RDS高可靠高可用
  - 存储可靠性

云主机可用域：需求

- 上层应用或系统需要规避单点失效
  - 两台Tomcat云主机互备
  - 数据库Master与Slave不希望一起挂掉
  - 平台运维时避免影响到产品服务可用性
- 资源隔离
  - 易受攻击的UGC类应用（如博客）需要与其他类型的应用部署于不同物理机
  - 某些非常重要的大型应用希望独占物理机
  - RDS等上层服务使用的云主机，可能希望与普通产品使用的云主机隔离
  - 用云主机进行性能测试怕影响到产品服务

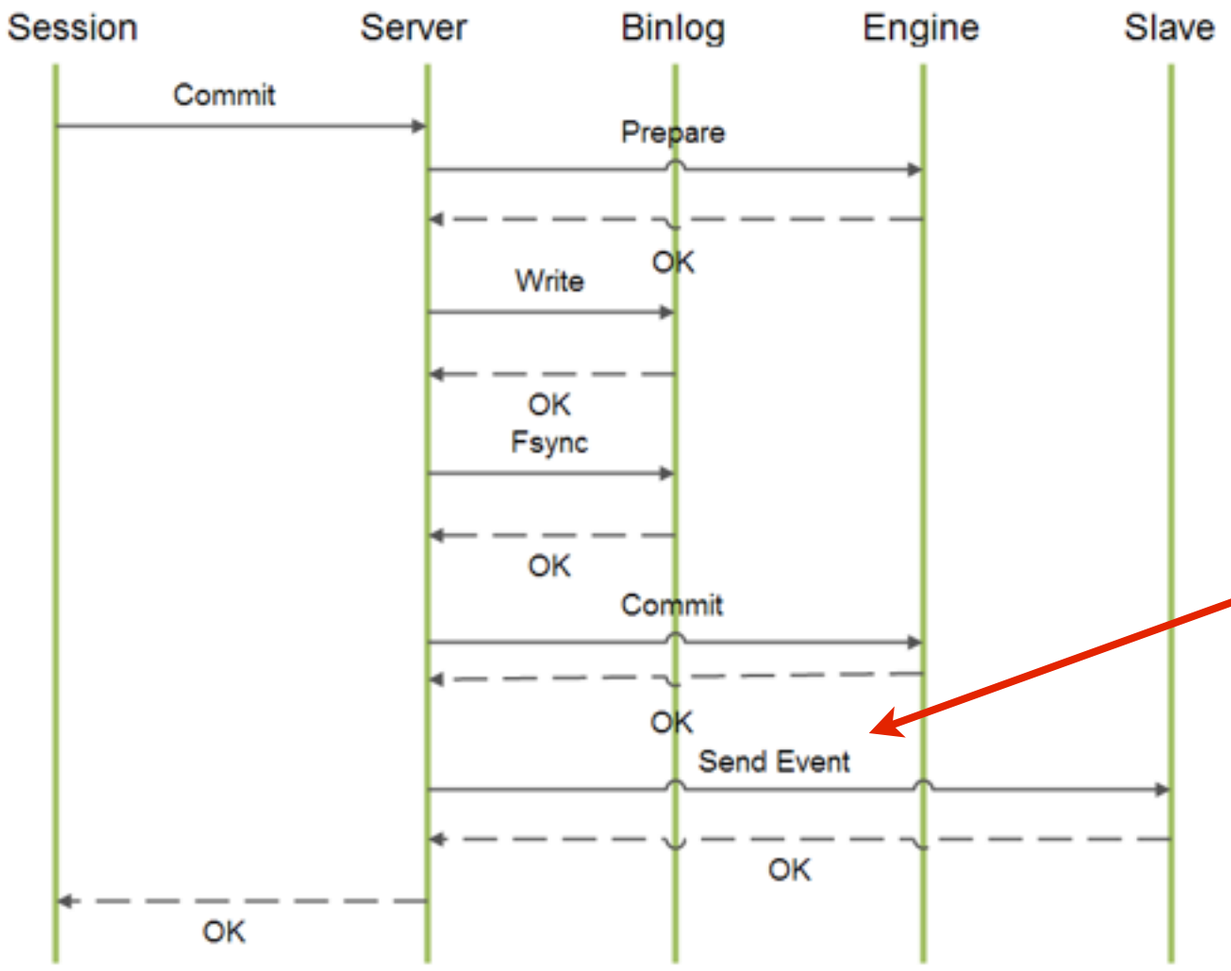
云主机可用域：实施策略

- OpenStack已有功能
- 承诺
  - 不同可用域的云主机不会从同一物理机分配
  - 云主机从属的可用域保持不变
  - 平台运维时不会关闭超过一个可用域中的云主机
- vs AWS Availability Zone
  - AWS Availability Zone：可用性隔离性好（一个AZ包含1到多个数据中心，供电、网络、消防等都隔离），不同AZ互联带宽受限制，EBS不可跨AZ挂载
  - 网易云主机可用域：只做物理机级别的隔离，不同可用域互联带宽不受影响，云硬盘可跨可用域挂载（便于实现基于共享存储的高可用服务）
- 部署
  - 普通产品、开发及功能测试3个可用域
  - UGC类产品2个可用域
  - 性能测试1个可用域

高可用RDS：方案讨论

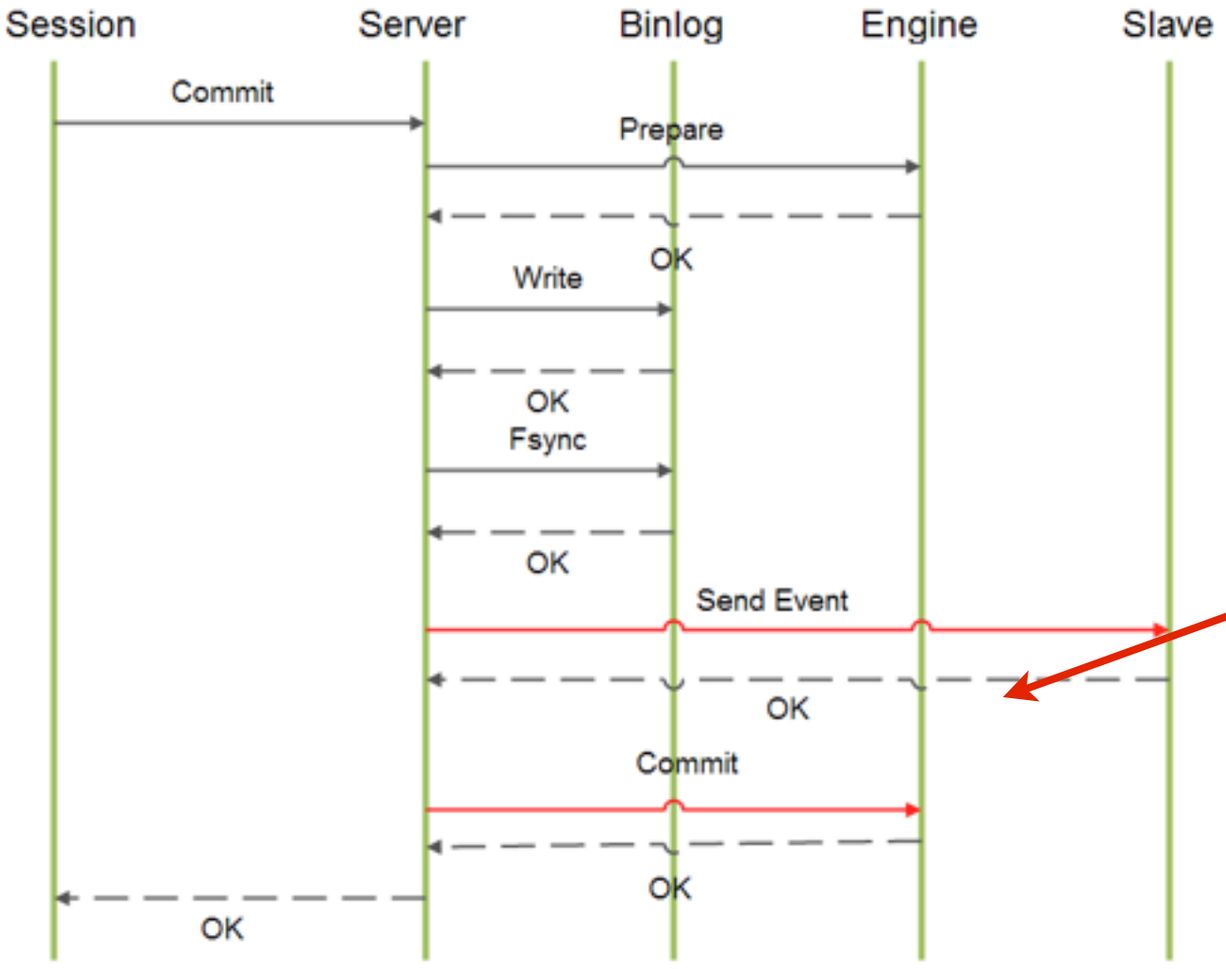
- 基于MySQL复制
  - MHA、MMM、阿里云RDS、腾讯CDB、淘宝RDS?
  - failover时间短，可用性好
  - 性能优秀
  - 可能导致事务丢失，可靠性不佳
- 基于共享存储
  - 共享binlog、共享存储、DRBD、AWS RDS（据说基于DRBD？）
  - 保证数据可靠性
  - failover时间较长（数据库需要recover）
  - 性能损失

高可用RDS： semi-sync



事务在binlog得到slave ack之前已经提交
故障时事务durability不能保证

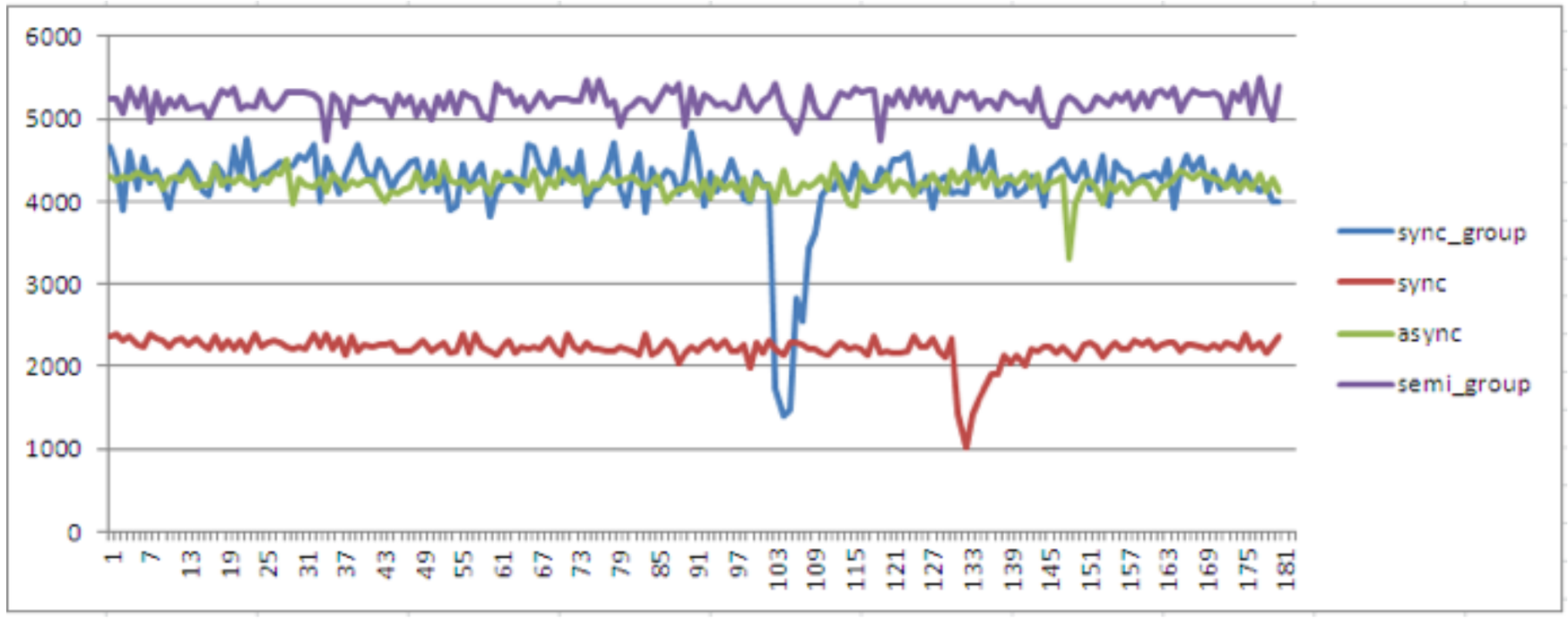
高可用RDS：VSR复制



事务在binlog得到slave ack之后再提交
故障时事务durability能够保证

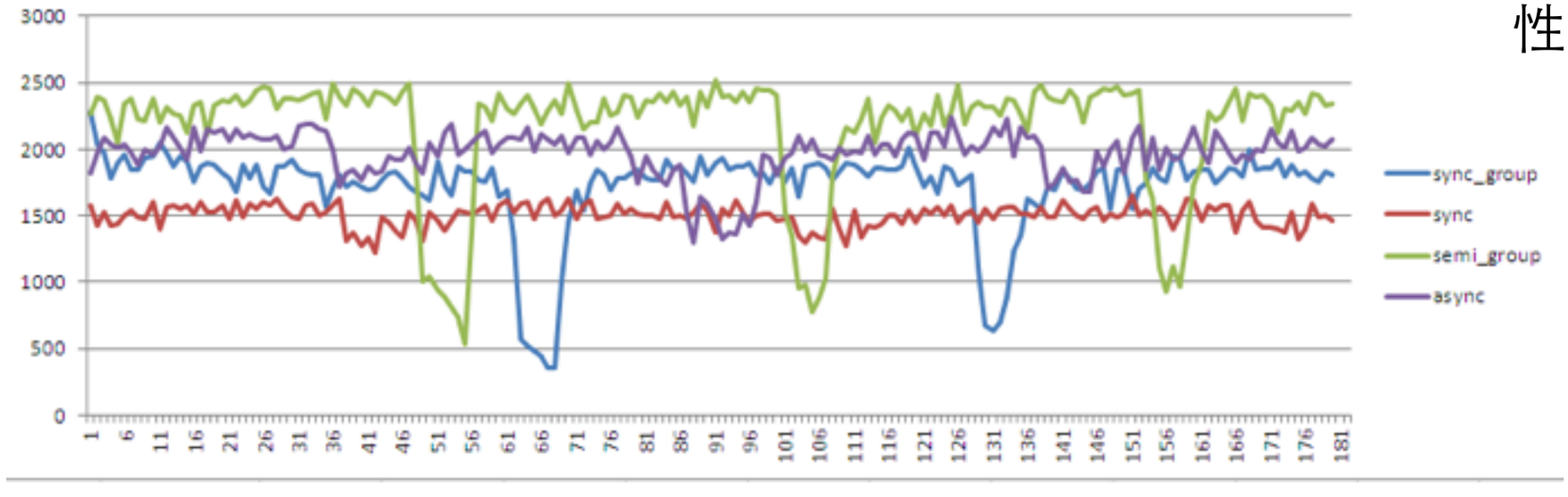
https://github.com/NetEase/InnoSQL

高可用RDS：VSR性能



QPS

使用group commit后VSR性能接近async复制



TPS

高可用RDS：方案

- 基于VSR复制
- NAT实现failover时IP不变
- 很多细节问题
  - 所有状态转换（同步/异步复制、主从切换等）统一由Manager控制
  - failover时要等待slave replay完relaylog再提供服务，此时不能等待Read_Master_Log_Pos和Exec_Master_Log_Pos相等，因binlog中可能存在不完整事务
  - slave重启时如何确定SQL thread执行起点？master.info和relay.info可能不一致。InnoSQL patch
- 更多高可用流程
  - 修改实例规格
  - 修改需要重启的配置参数

存储可靠性

- 云主机本地存储
  - SAS，无RAID
  - 云主机迁移之后丢失
  - 适合性能、可靠性要求不高的应用
- 云硬盘
  - 目前：SAS，RAID 1
  - NBS 2.0：分布式多复本(1-3)

    3复本云硬盘，永远不会损坏的云硬盘
- 对象存储
  - 1-3复本

遗留问题及未来方向

- 减小cpu.cfs_period_us
  - 可望减少对网络IO延迟的影响
- 网络带宽QoS考虑上层网络架构
  - 一筐14片刀片只有2千兆上行
- 不会坏的系统盘
- 不会坏的云硬盘