经验分享

构建OpenStack高可用云服务平台

Geekbang.
极客邦科技

整合全球最优质学习资源，帮助技术人和企业成长

Growing Technicians, Growing Companies

InfoQ queue

专注中高端技术人员的
技术媒体

EGO EXTRA GEEKS' ORGANIZATION NETWORKS

高端技术人员
学习型社交网络

StuQ queue

实践驱动的
IT职业学习和服务平台

GiT GEEKBANG INTERNATIONAL TRAINING 极客邦培训

一线专家驱动的
企业培训服务

地址：北京市朝阳区洛娃大厦C座6层1607室　　网址：www.geekbang.org

目录

- 我们是谁
- 高可用介绍
- OpenStack高可用方案
- 案例分享

海云是谁

- 北京海云捷迅科技有限公司，简称AWcloud海云，国内领先的企业级OpenStack云服务提供商
- AWcloud海云成立于2010年，2012年开始专注OpenStack私有云服务
- 核心成员来自IBM/Red Hat/甲骨文/绿盟/东软
- 公司总部位于北京，在深圳、武汉、上海等地设有分支机构
- 2013年10月获得宝德科技A轮1500万人民币融资
- 2015年06月获得INTEL领投的B轮数千万人民币融资

海云业务

- **私有云解决方案（Private Cloud Solution)：**基于海云OpenStack发行版，支持KVM、Hyper-V、VMware等异构虚拟化平台，为企业客户构建和管理私有云平台。
- **私有云托管（Private Cloud Hosting)：**海云为企业客户提供私有云托管服务，帮助客户一站式解决数据中心、硬件、存储、云平台、运维等所有问题。客户按需租赁整体云平台，按需付费。
- **融合一体方案：**海云为企业客户提供AWcloud超融合一体机，帮助企业实现IT系统快速部署、自动化运维。
- **IDC数据中心云平台联合运营：**联合国内知名IDC企业，整合优势资源，为IDC企业提供基于OpenStack技术的公有云平台，为IDC企业迅速转型为VDC，提供平台支持、技术支持和服务支持。
- **OpenStack咨询与培训：**为企业客户提供OpenStack运维、开发的咨询和培训。海云的技术团队，具备丰富的一线运营经验，可以为企业客户提供成熟的OpenStack运营和运维服务。

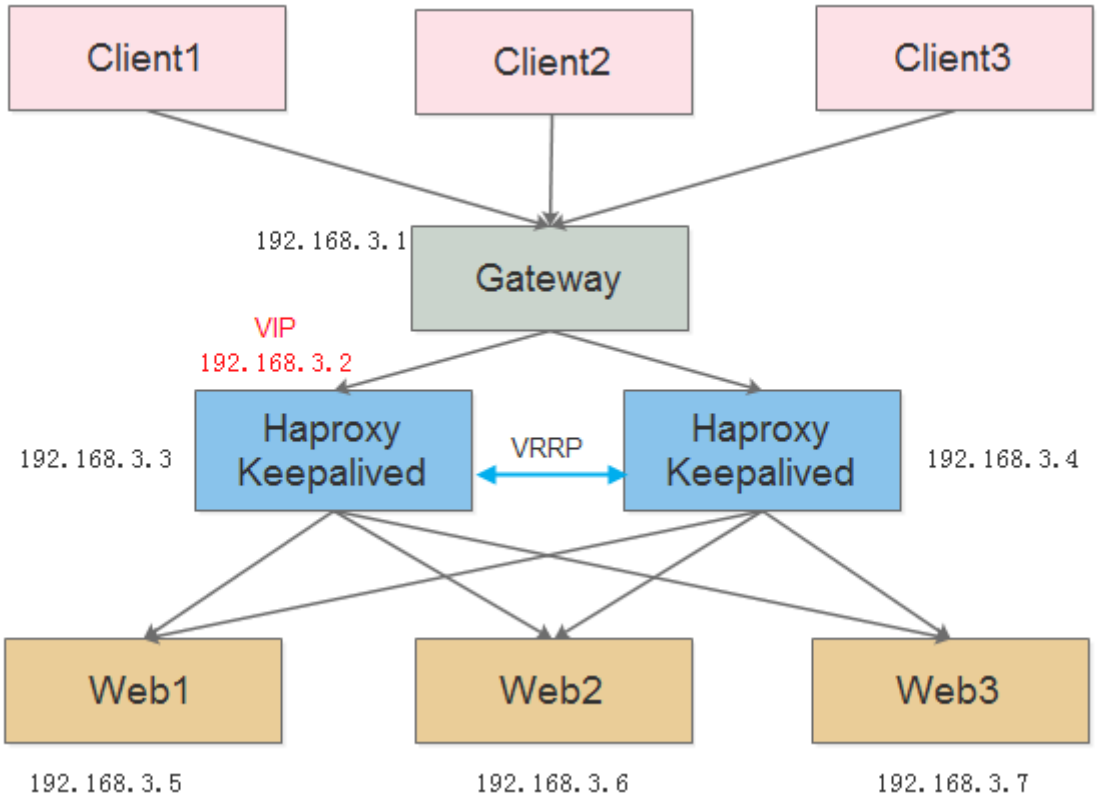
高可用介绍

- Availablity = Fn(MTBF, MTTR)
- 通过避免单点故障来减少停机时间
- 冗余服务
  - Active-Active
    - 无状态应用
    - 应用内置支持
  - Active-Passive
    - 通过外部集群软件

可用性百分比	每年停机时间
99%	87.6小时
99.5%	43.8小时
99.9%	8.8小时
99.95%	4.4小时
99.99%	53分钟
99.999%	5.3分钟

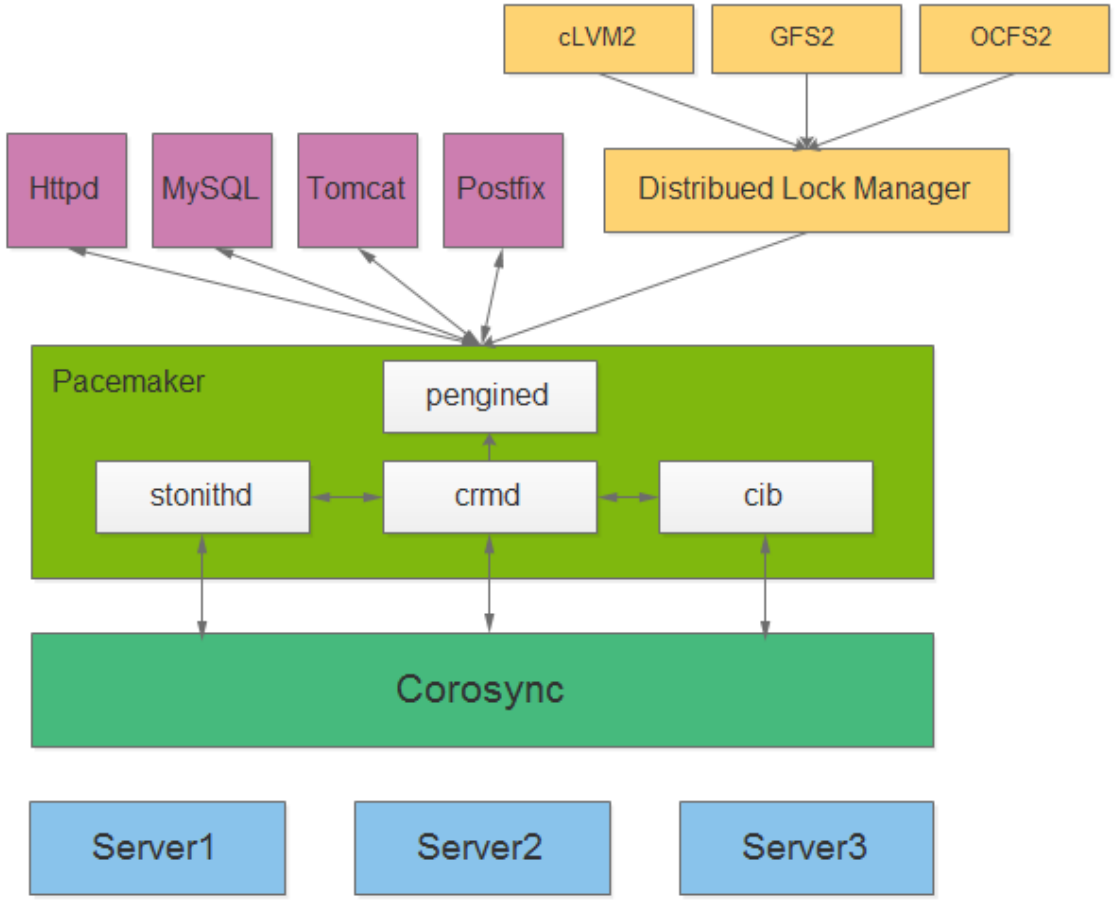
Haproxy + Keepalived

- Haproxy 负载均衡
- Keepalived 切换虚IP
  - 基于VRRP
  - 高版本可以配置单播
- 避免Keepalived “脑裂”
  - ping网关
  - 只有从节点可以failover，发生切换后通知管理员
- 配置简单，适合于切换虚IP
- 没有完整的服务管理和Fence机制

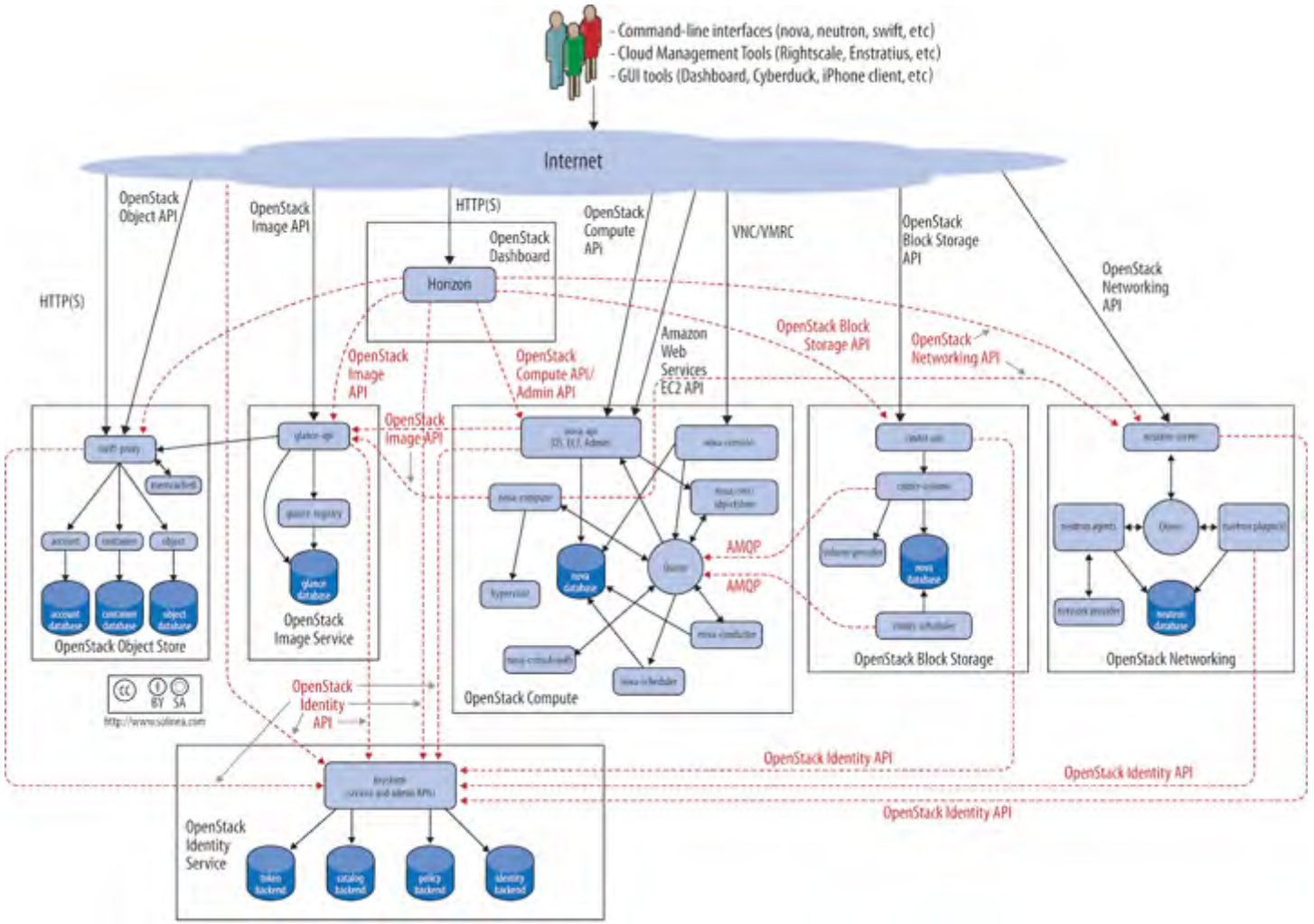


Pacemaker

- 使用Corosync维护成员关系
- Quorum机制
- 通过STONITH支持Fence
- 丰富的服务管理脚本
- 控制服务依赖，主机亲和力
- 丰富的文档



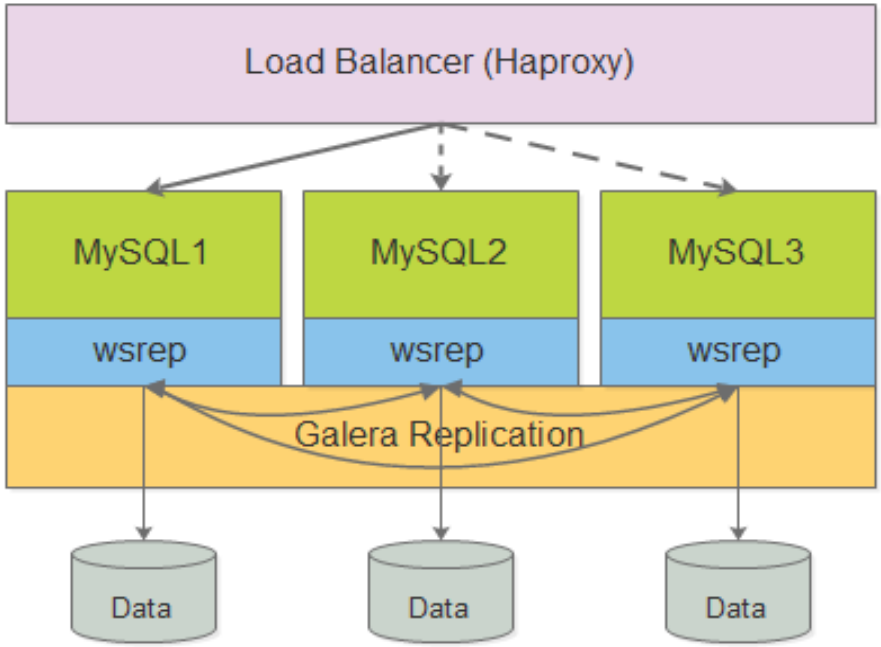
OpenStack架构



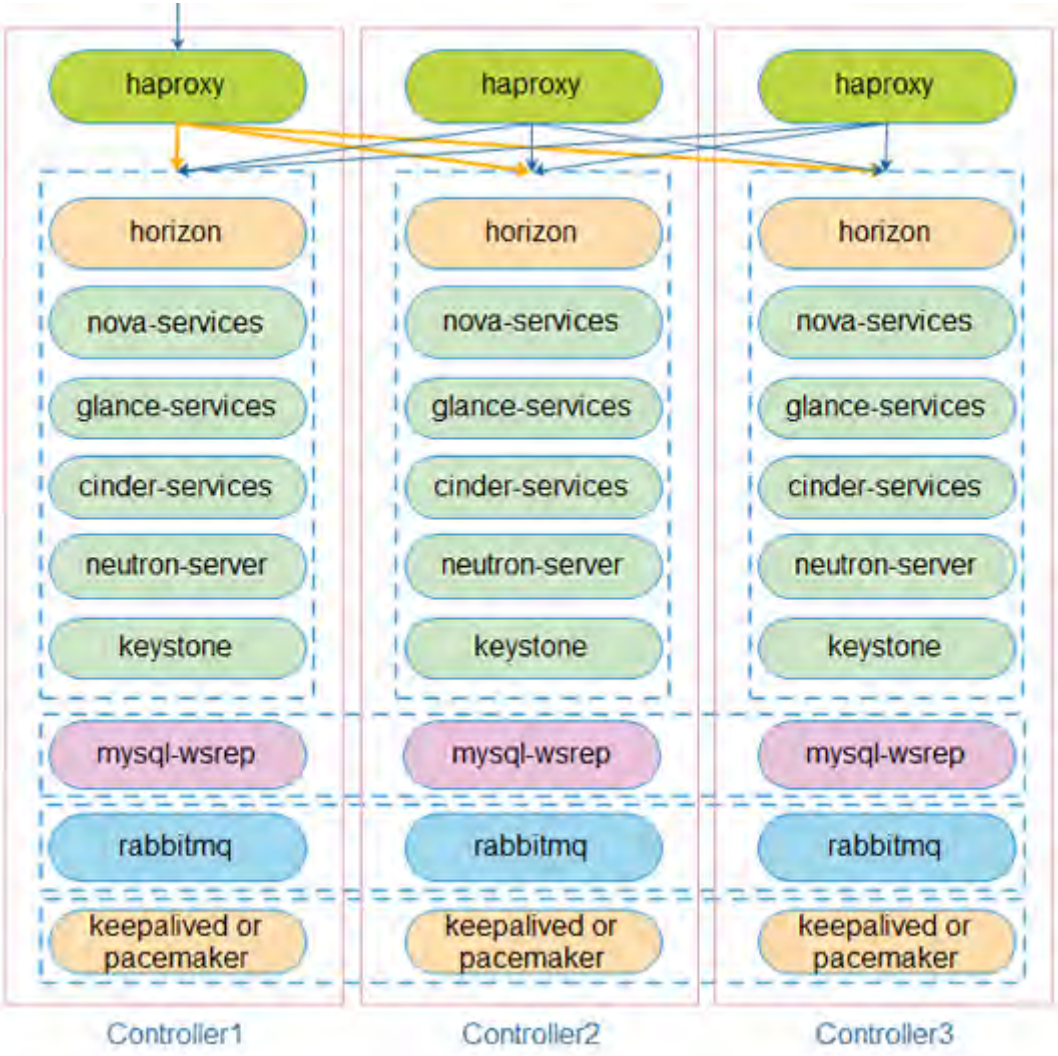
http://docs.openstack.org/openstack-ps/content/figures/2/figures/osog_0001.png

基础服务的高可用

- MySQL
  - Galera
    多主，但只有一个可写，避免死锁
    同步复制，运维方便，吞吐量有影响
  - 原生的主从方案 + MHA
- RabbitMQ
  - RabbitMQ内置集群机制
  - 不要使用Haproxy，直接使用oslo.messaging的驱动



控制节点高可用架构

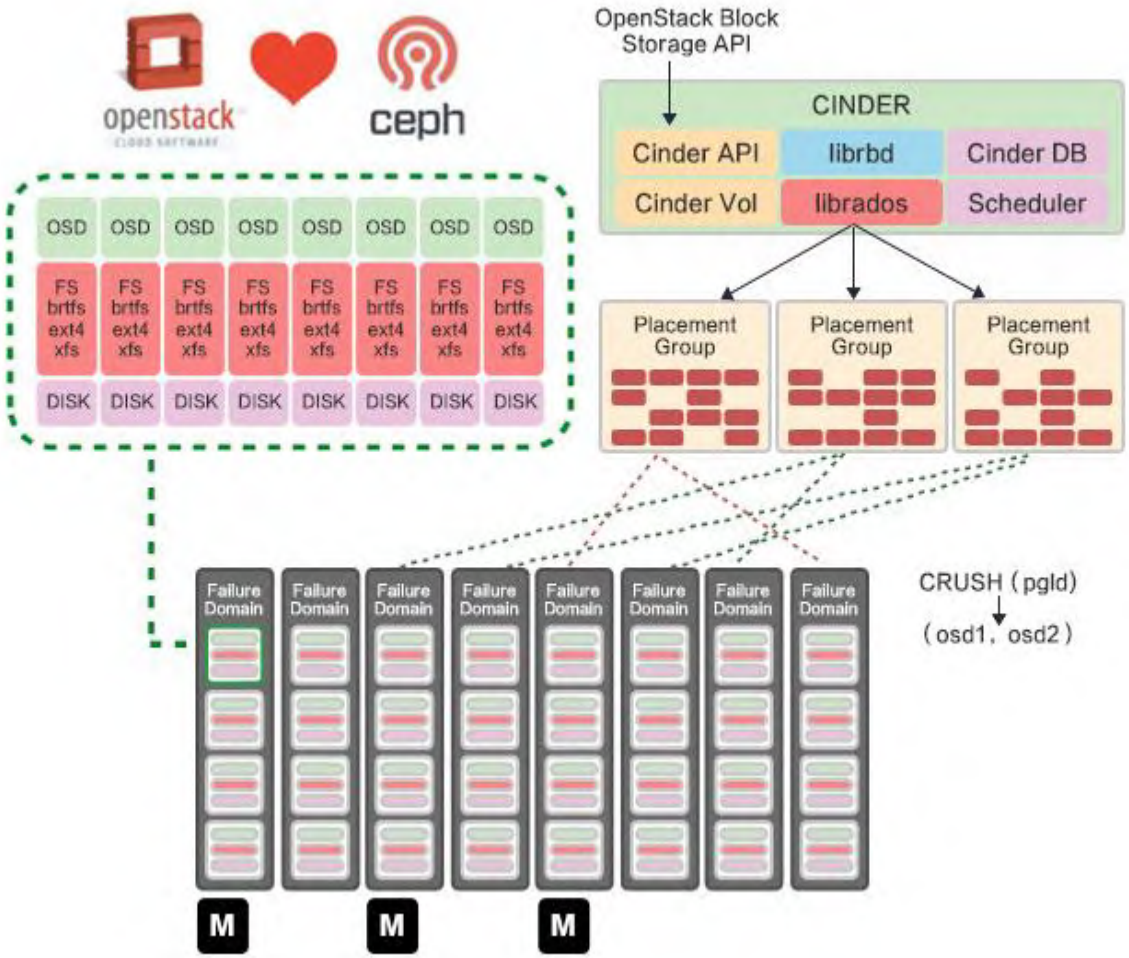

haproxy
haproxy
haproxy
horizon
nova-services
glance-services
cinder-services
neutron-server
keystone
horizon
nova-services
glance-services
cinder-services
neutron-server
keystone
horizon
nova-services
glance-services
cinder-services
neutron-server
keystone
mysql-wsrep
mysql-wsrep
mysql-wsrep
rabbitmq
rabbitmq
rabbitmq
keepalived or pacemaker
keepalived or pacemaker
keepalived or pacemaker
Controller1
Controller2
Controller3

问题

- 单节点故障恢复时间，大概在一分钟左右，这期间的客户端请求可会出错
- RabbitMQ节点的消息同步是异步的，所以有可以碰到丢消息的情况
- RabbitMQ出现过悬空consumer的问题，自制监控脚本，用删队列并让客户端自动重建队列的方法解决。高版本增加客户端应用层面心跳有可能解决
- 使用memcached作为Keystone的token存储时，发现客户端集群没有fence/unfence机制，导致在节点故障时仍然连接故障节点，延长重试时间后，又发现有丢token的现象。改为使用MySQL存放token，后来开发了Redis驱动
- 服务启动顺序有依赖关系，需要保证按顺序启动
- 经过优化后，全部控制节点从断电到完成恢复需要5分钟时间

存储 ： 数据平面(Ceph)

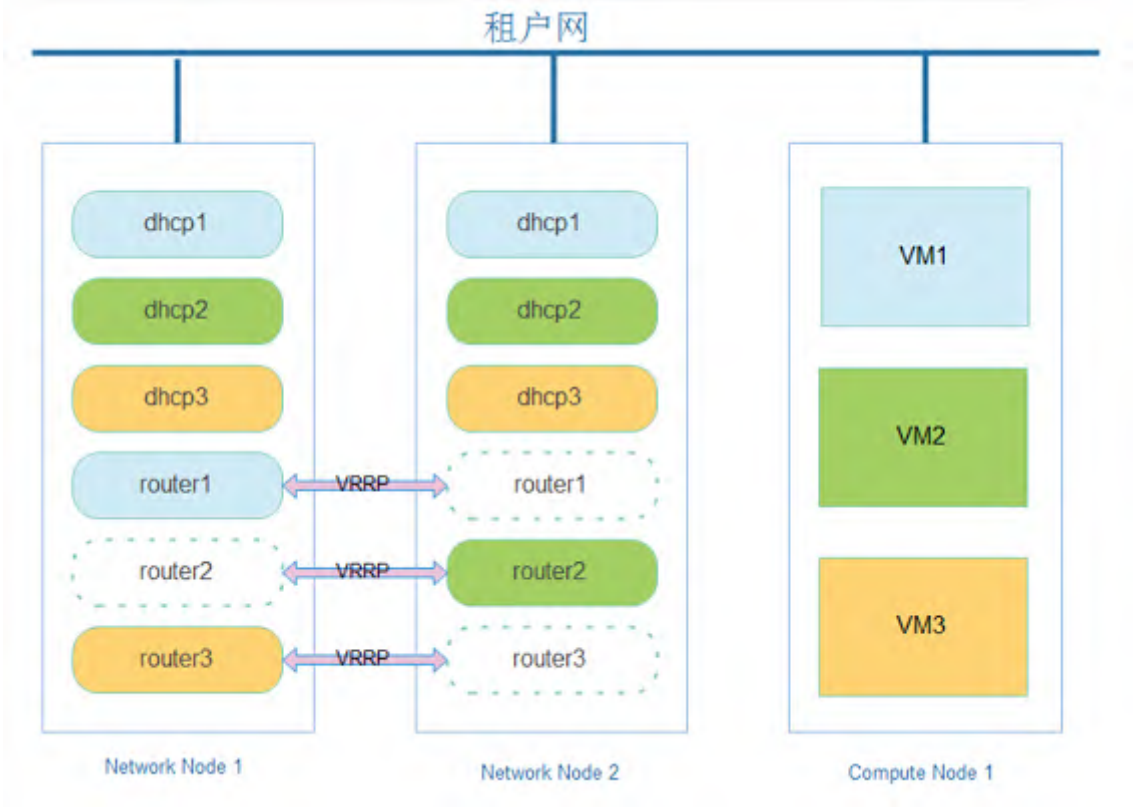
- 支持数据多副本
- 数据按照CRUSH算法存放，可以考虑物理拓年
- 客户端通过CRUSH算法来计算数据存储位置
- Ceph Monitor使用Paxos算法保证一致性



http://pinrojas.com/2014/05/20/ceph-you-will-love-the-way-it-flies/

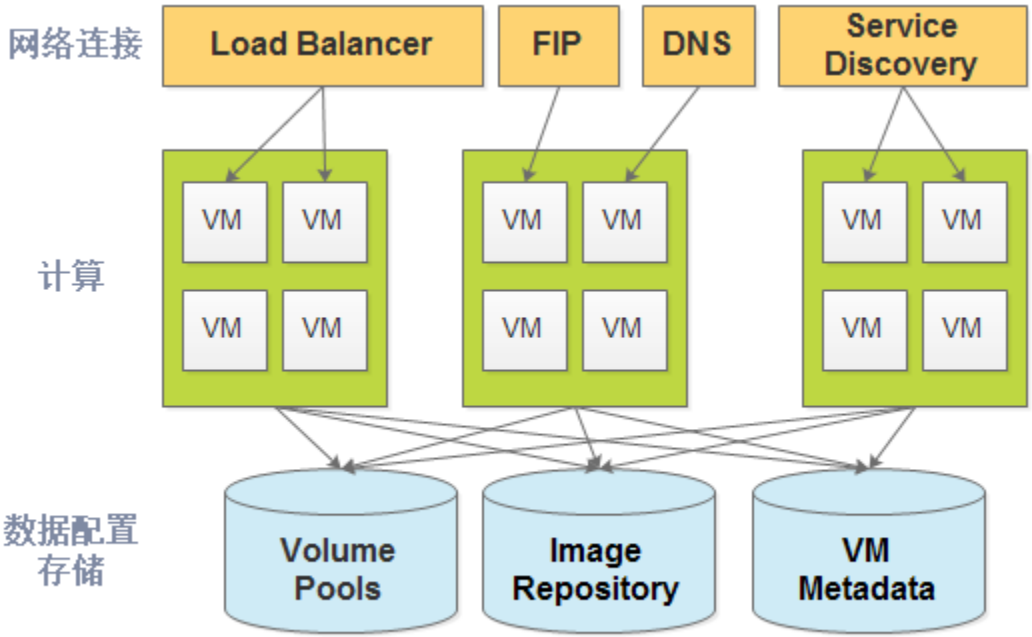
网络节点

- DHCP 运行多个Agent，地址分配由Neutron控制
- 虚拟路由器通过Keepalived互备
- 在L版本里修复和L2pop的冲突
- 不支持连接跟踪信息failover
- 单个网络节点在可用性和性能都可能成为瓶颈
- 社区DVR功能并不成熟
- 可以根据需求考虑使用物理网络设备管理三层



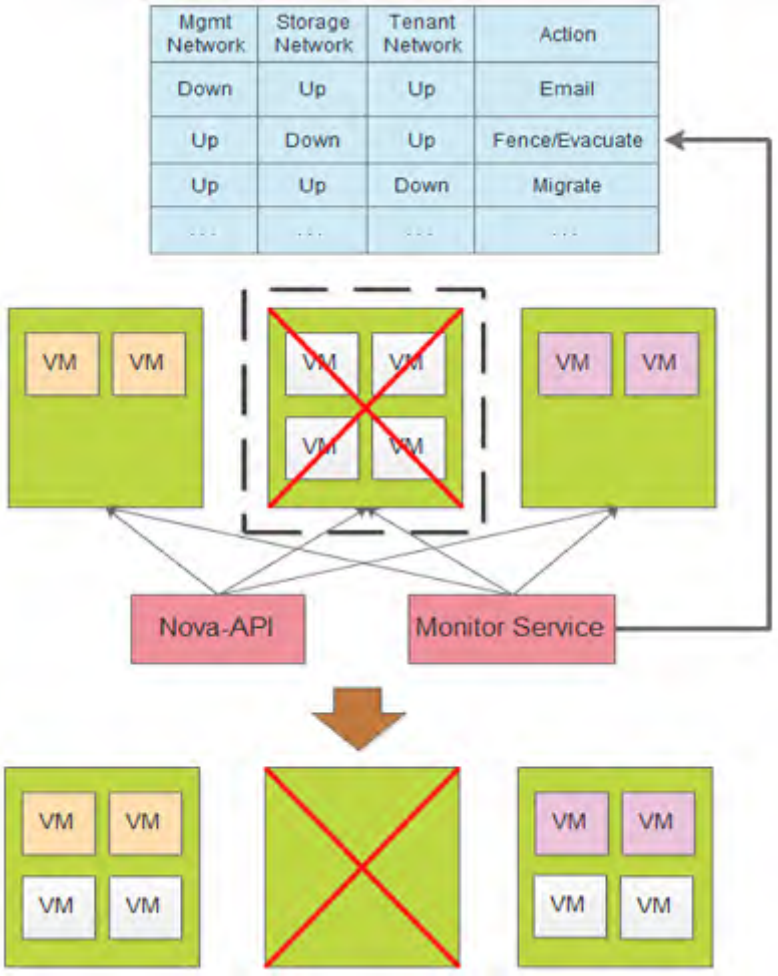
虚拟机高可用方案

- 在虚拟机内运行集群
  - SDN的限制
    - 多播
    - Neutron的allowed-address-pairs
    - 虚IP的切换和L2pop冲突
  - 冷备没有太大意义
- 虚机保证
  - 无状态应用 – 直接重建
  - 有状态应用
    - 解耦IP地址和数据盘
    - 在其他其点启动或者新建并关联浮动IP地址和数据盘
- 由基础设施保证



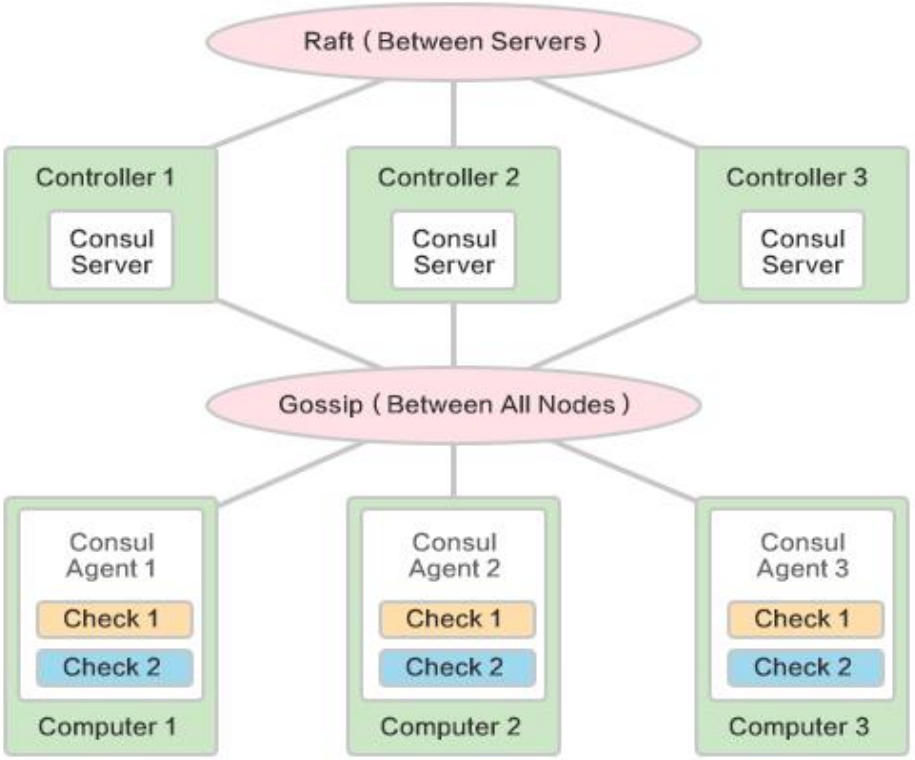
计算节点 ： 高可用方案v1

- 需求
  - 心跳监测，维护成员关系
  - 故障处理动作
  - 故障域隔离机制
- 实现v1
  - 轮循计算节点不同网络接口
  - 根据策略矩阵执行动作
  - 发邮件，IPMI Fence，Evacuate
- Nova evacuate接口依赖
  - force-down
  - 在Liberty API Microversion 2.11中支持
- 实现v1的问题
  - 监控服务本身的高可用
  - IPMI网络失联后无法处理，如断电
  - 扩展性问题，计算节点数量受限



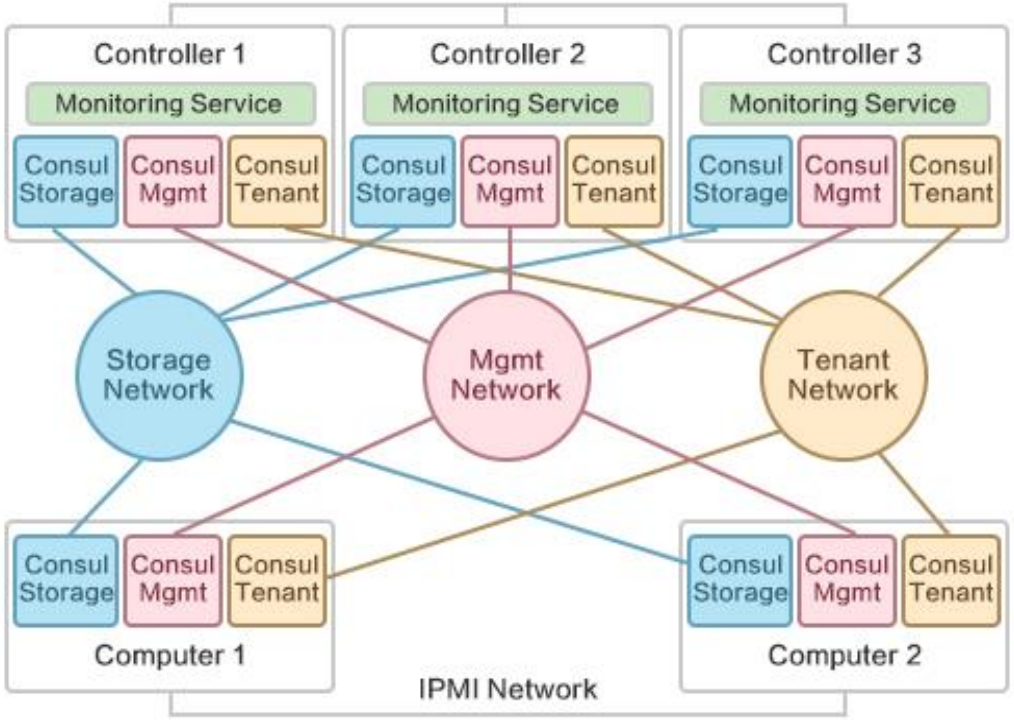
计算节点 ： Consul

- 由Hashicorp开发
- 服务发现协议
  - 服务注册
  - DNS接口
  - 配置模板
- 分布式 K/V 存储
- 节点健康检查
- 大规模集群
  - 使用Gossip协议
  - 分布式成员检查
- 会话和锁
  - 提供Leader选举机制
- REST API



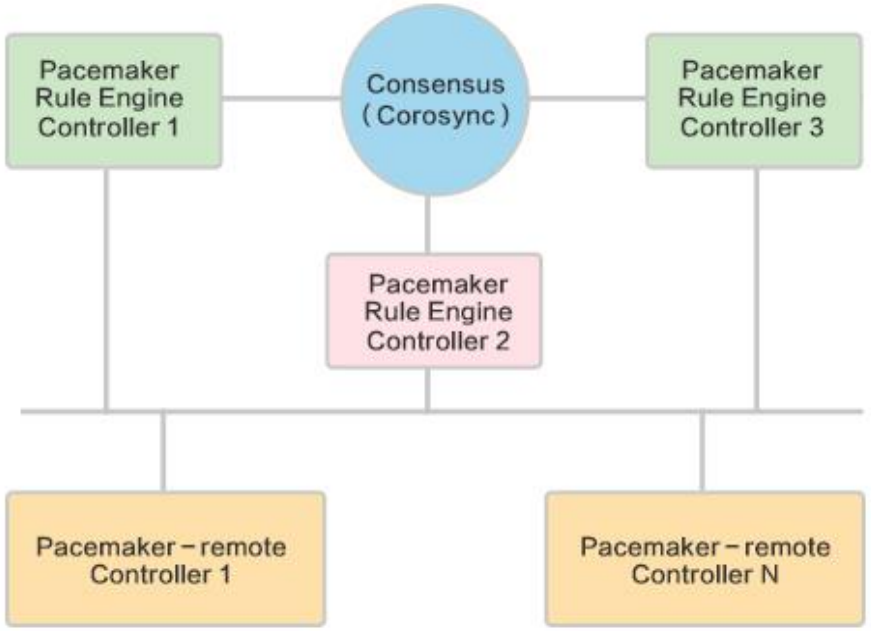
计算节点 ： 基于Consul的分布式健康检查方案

- 健康检查
  - 通过Consul成员关系来确定网络连接
  - 计算节点通过Consul来注册健康检查
- 监控服务的高可用
  - Leader选举和释放
  - 使用Consul的会话和锁
- Fence
  - IPMI
  - Consul事件广播
- Fenced节点列表
  - 避免重复Fence
  - 手工Unfence
- 网络分区的处理
  - 限制同时fence数量
  - 只解决计算节点故障



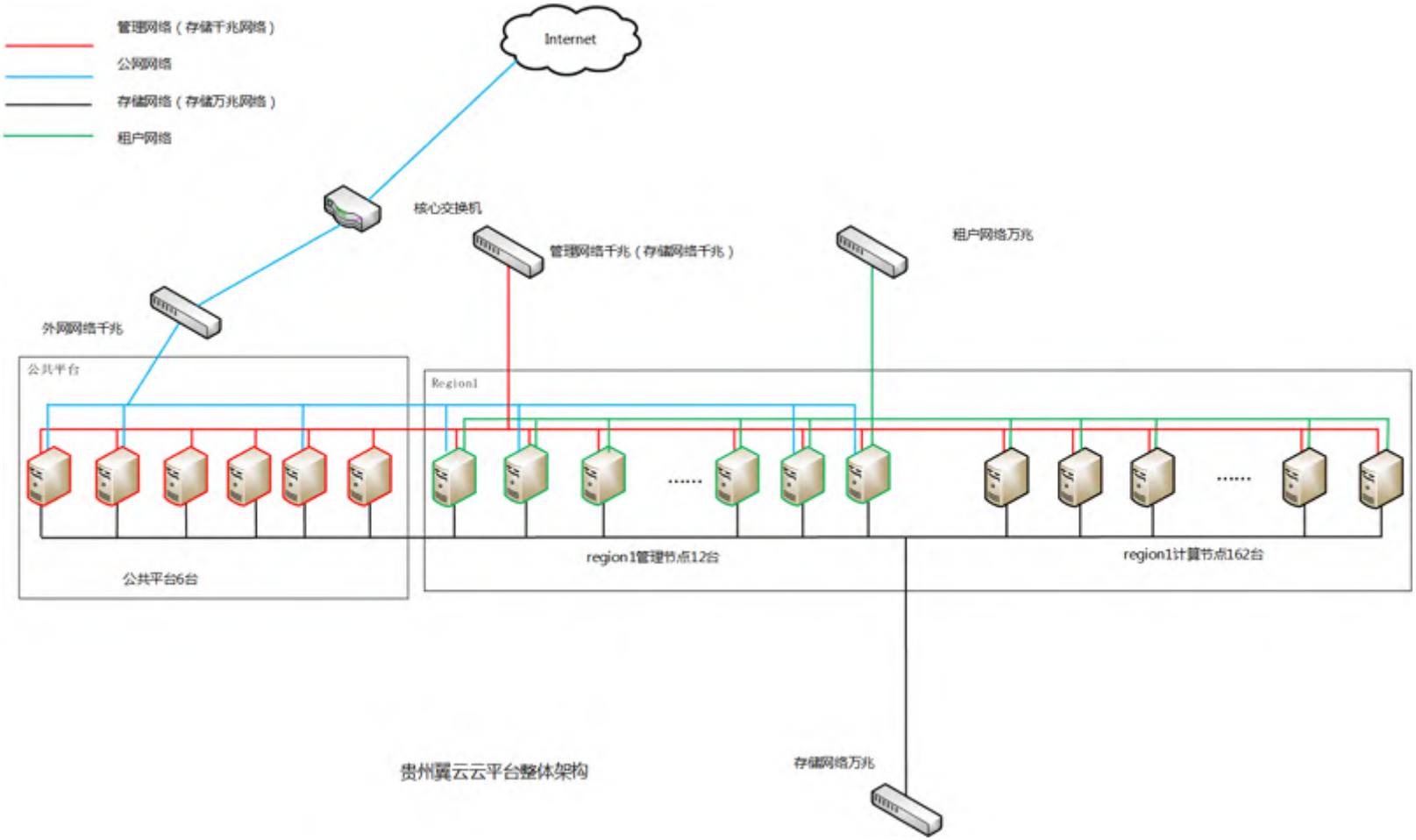
计算节点 ： 为什么不使用Pacemaker-remote

- 只能有一个心跳网络
  - 使用bonding?
- 不能控制不同条件的执行动作，只能Fence



案例介绍 ： 贵州翼云

案例介绍 ： 物理拓扑

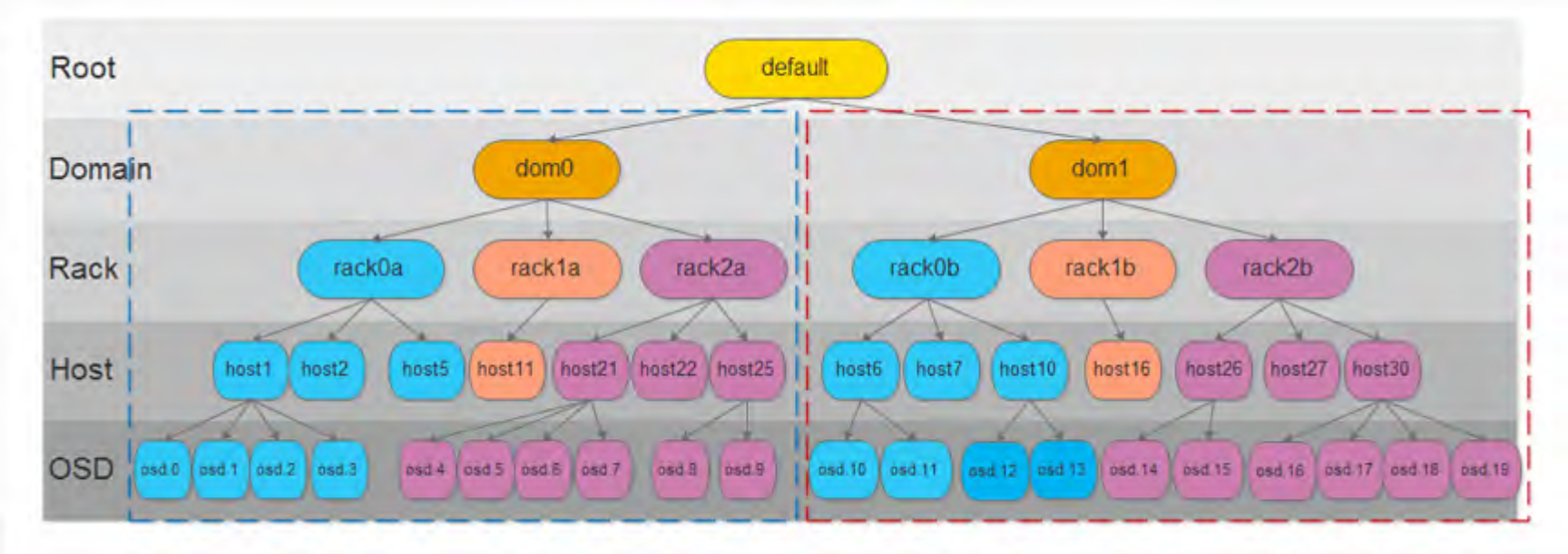


贵州翼云云平台整体架构

案例介绍 ： 管理结点部分部署架构

- 使用 Haproxy + Keepalived
- 使用ZeroMQ代替RabbitMQ
- MySQL原生主从
- 使用Redis作为Keystone Token驱动

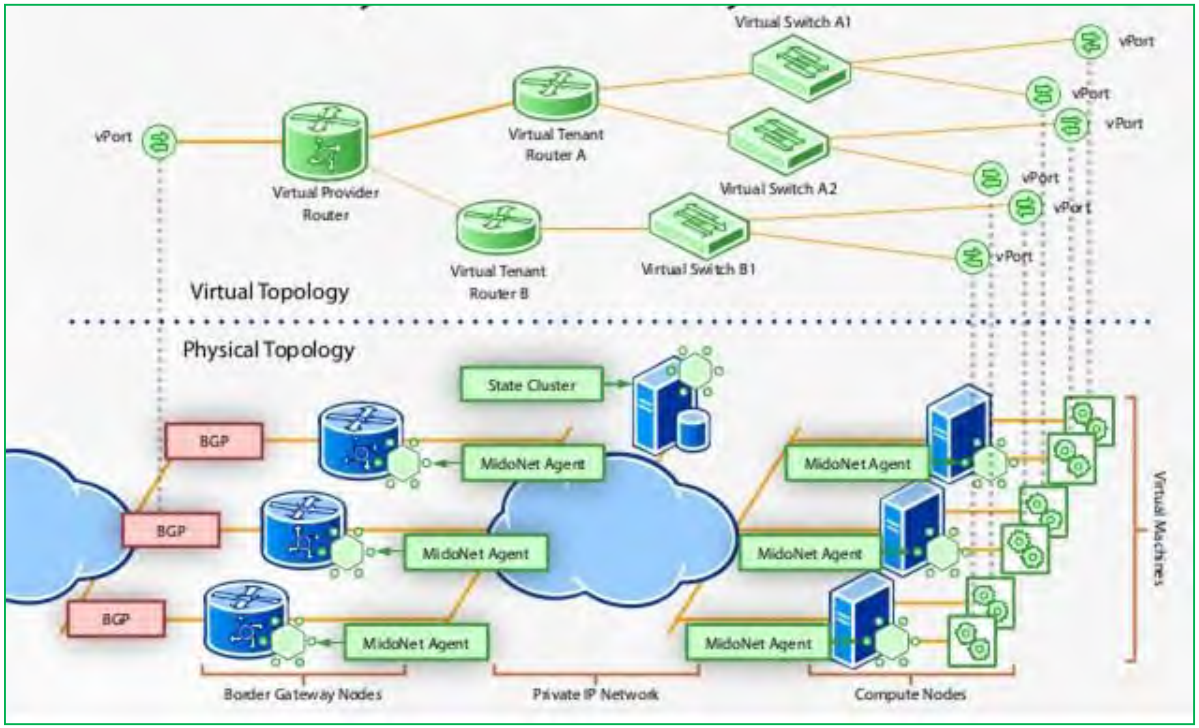
Ceph OSD Map



- 3复本存放
- 单个数据复制域 3 * 0.5rack * 10host * 4osd = 60 osd
- 减少OSD之间关联度

网络 Midonet

- 网络配置数据库
  - ZooKeeper
  - Cassandra
- 东西向流量分布式
- 外网网关使用BGP



http://image.slidesharecdn.com/midonet101-150221232947-conversion-gate01/95/midonet-101-face-to-face-with-the-distributed-sdn-12-638.jpg?cb=1424561601

总结

- 使用Pacemaker 或者 Keepalived 做心跳管理
- 使用Haproxy做负载均衡
- 使用MySQL和RabbitMQ的集群
- 存储和网络数据平面高可用由后端方案保证
- 虚拟机高可用监控物理机或者虚拟机心跳，做虚机疏散或重建


Q&A

Thanks!