DTCC

2017第八届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2017

金融行业基于容器技术的OLTP数据库技术

余军

上海富麦信息科技有限公司

关于我

- 余军
- 开源技术[忠实]粉丝 (20y+)
- 在HP和Red Hat 为企业提供开源解决方案(12y+)
- 现任富麦信息科技有限公司技术负责人(5y+)
- 为传统企业(以金融行业为主)提供企业级开源技术和服务

金融行业开源数据处理(OLTP类)技术的应用现状和挑战

应用现状

- 在基于互联网环境的创新业务场景中逐渐开始应用开源数据处理技术
- MySQL，Redis, MongoDB 三大技术为主, IMDG类为辅
- 单一集群规模不大(~10 节点以内)，但随业务规模的集群数量较多
- 通常运行环境与云环境(IaaS) 关联度较高
- 相比于性能，更关注高可用和安全性

挑战

- 创新业务发展规模和上线敏捷要求 的加速度提升
- 基础资源池规模扩展和效率要求上的提升
- 自动化，资源池，服务化等相关体系建设刚刚起步
- 运维团队的规模非常有限
- 生产安全管理和监管要求非常严格

DBScale

数据处理相关
底层资源池化

数据处理服务化
多租户的灵活业务场景

高可用和安全

高效运维管理

系统研发背景

- 2013 年 为大型金融支付企业研发基于cgroup 的DBaaS 原型系统
- 2014 年 进行产品化重构，迁移到Docker环境
- 2015 年 完善产品配套系统的设计和实现
- 2016 年 在大型金融支付企业进行生产环境的需求落地和生产上线
- 2017 年 微服务化改造，及OLTP数据服务能力扩展支持

DBScale

快速服务

- 面向租户自助申请
- 分钟级完成服务交付
- 灵活支持多种数据处理服务和服务拓扑

弹性资源

- 按需对服务进行纵向和横向的资源伸缩
- 资源有效隔离和计量

安全及运维

- 基础设施到服务层的安全性设计
- 规模化，自动化，智能化，可视化运维

为什么要使用容器?

1. 资源成池后，多租户数据服务开通和运行的隔离控制要求
2. OLTP数据处理服务对稳定性和性能要求高, 容器的性能损耗可以忽略不计
3. 数据服务规模化后，基于容器可以让运维管理有更小的颗粒度控制
4. 部署和运行状态调整速度快，非常敏捷

OLTP数据处理适合用容器吗?

1. 适合，尤其是越来越多的分布式OLTP数据处理服务
2. 和自动化脚本/工具+传统虚拟机仅解决部署问题来比, 基于容器的数据处理技术有更强大的软件服务化能力(云化)，是真正的平台即服务PaaS 运行态管控技术
3. 容器SDN和SDS的发展进度目前是一个比较重要的影响因素，不过也有成熟的变通解决方案
4. 科学选择OLTP类数据处理服务所需要的容器集群底层基础框架(docker/k8s/mesos)是极为重要的工作

1

架构相关

2

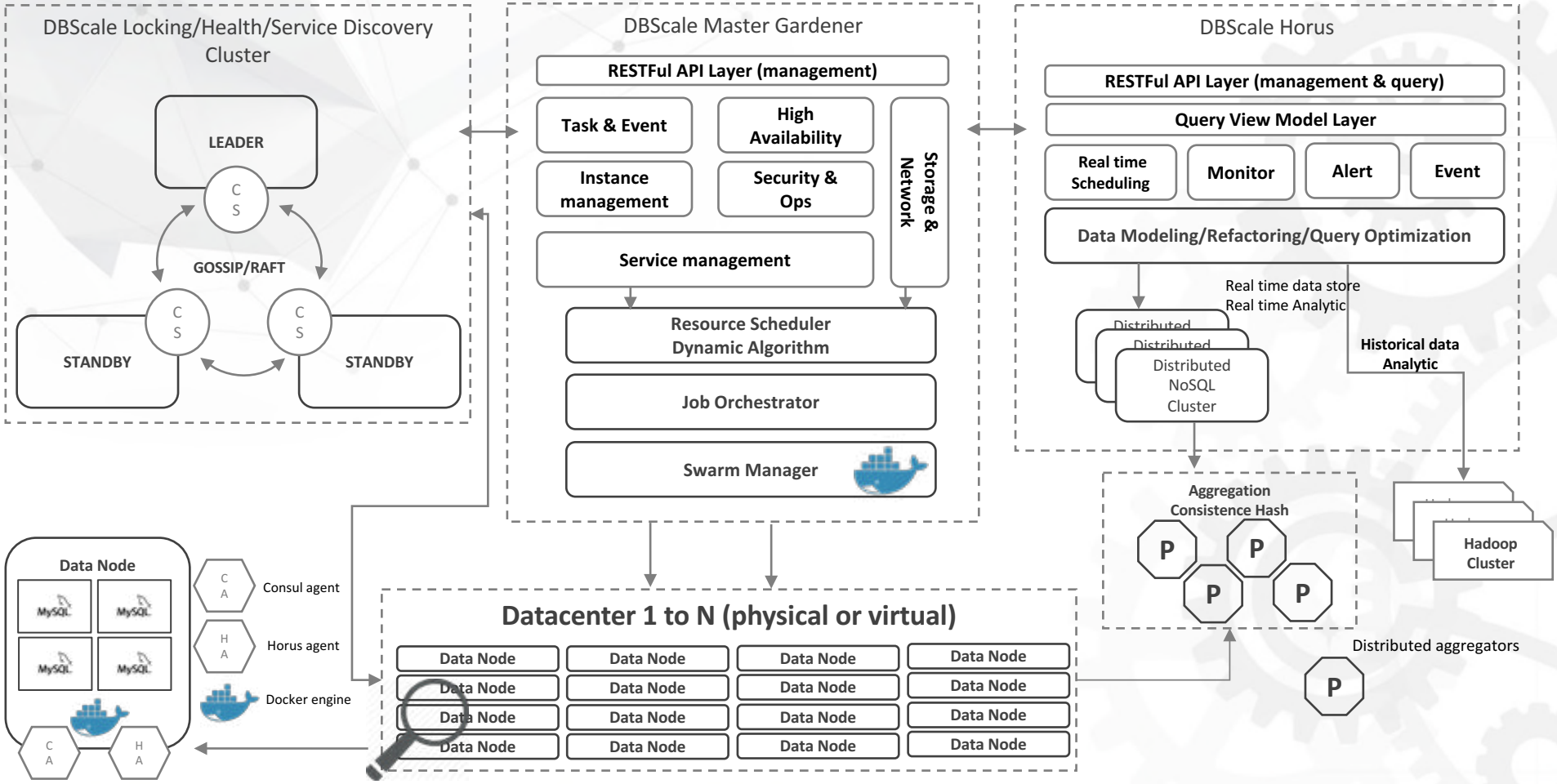
服务化相关

3

高可用与安全

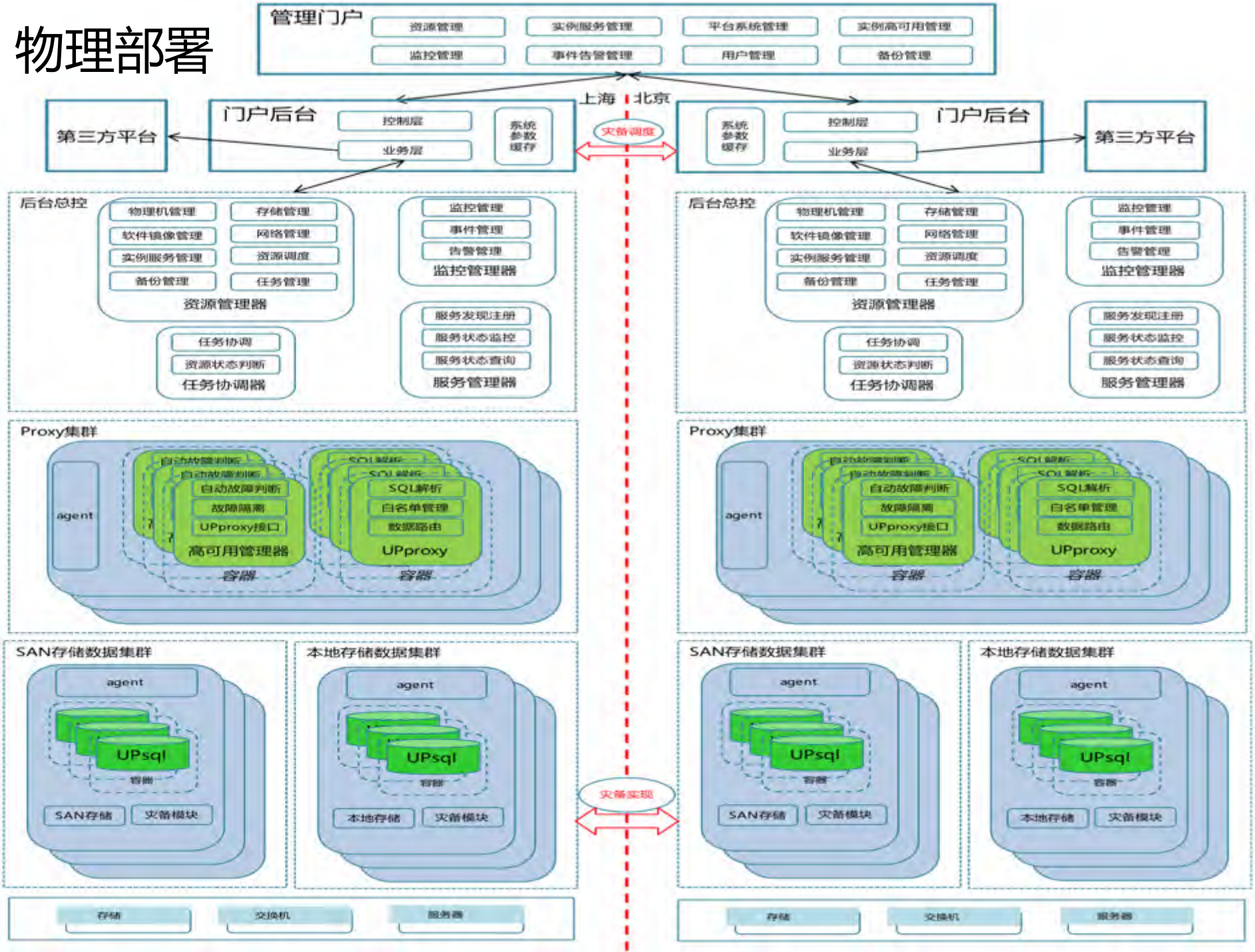
架构相关

DBScale 的逻辑架构

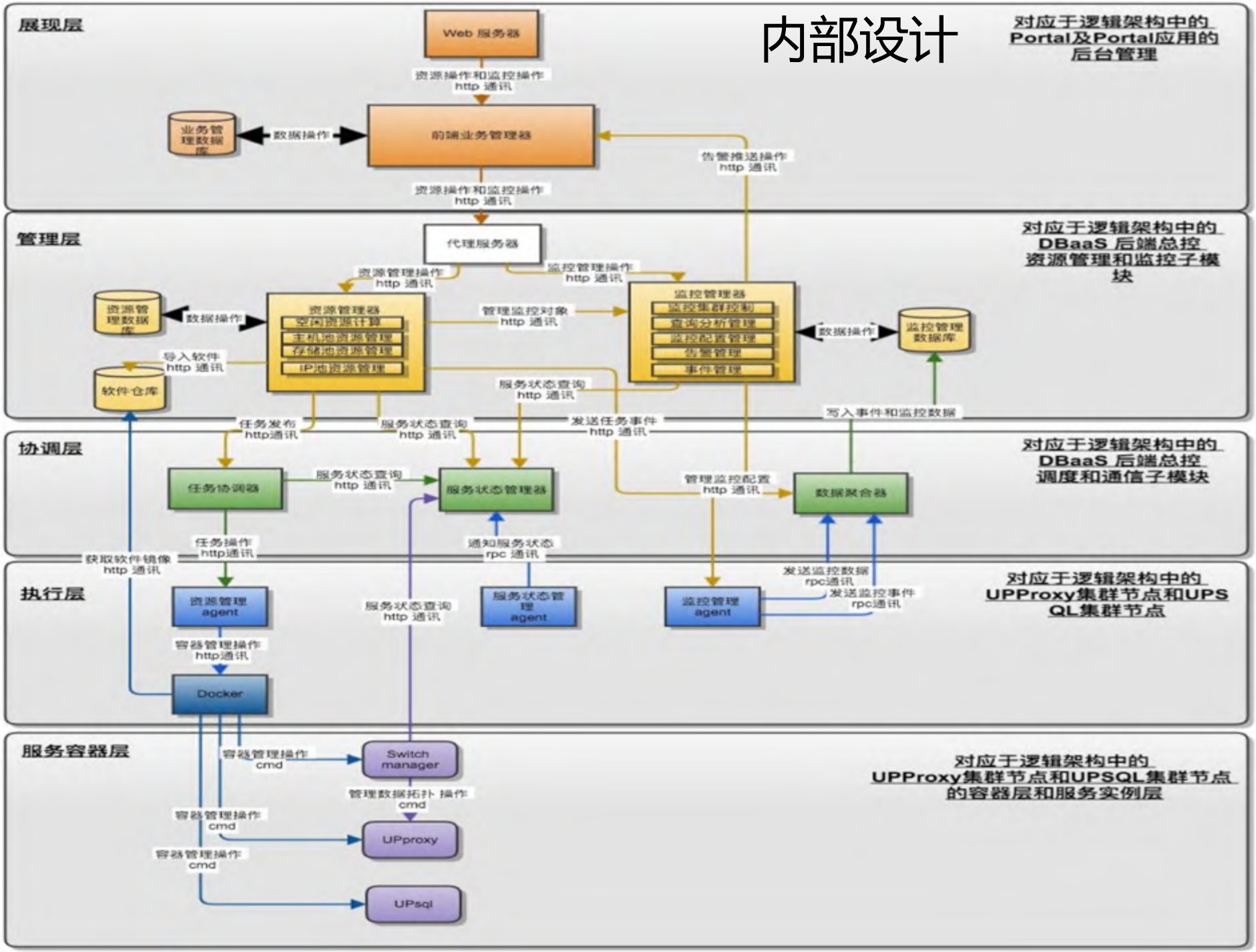


DBScale 的功能结构

物理部署


管理门户
资源管理
实例服务管理
平台系统管理
实例高可用管理
监控管理
事件告警管理
用户管理
备份管理
上海
北京
第三方平台
门户后台
控制层
业务层
系统参数缓存
灾备调度
后台总控
物理机管理
存储管理
软件镜像管理
网络管理
实例服务管理
资源调度
备份管理
任务管理
资源管理器
监控管理
事件管理
告警管理
监控管理器
任务协调
资源状态判断
任务协调器
服务发现注册
服务状态监控
服务状态查询
服务管理器
Proxy集群
agent
自动故障判断
故障隔离
UPproxy接口
高可用管理器
SQL解析
白名单管理
数据路由
UPproxy
容器
SAN存储数据集群
本地存储数据集群
UPsql
SAN存储
本地存储
灾备模块
灾备实现
存储
交换机
服务器

内部设计


展现层
Web 服务器
资源操作和监控操作
http 通讯
业务管理数据库
数据操作
前端业务管理器
告警推送操作
http 通讯
资源操作和监控操作
http 通讯
对应于逻辑架构中的
Portal及Portal应用的
后台管理
管理层
代理服务器
资源管理操作
http 通讯
监控管理操作
http 通讯
资源管理数据库
数据操作
资源管理器
空闲资源计算
主机池资源管理
存储池资源管理
IP池资源管理
管理监控对象
http 通讯
监控管理器
监控集群控制
查询分析管理
监控配置管理
告警管理
事件管理
数据操作
监控管理数据库
导入软件
http 通讯
软件仓库
服务状态查询
http 通讯
写入事件和监控数据
对应于逻辑架构中的
DBaaS 后端总控
资源管理和监控子模块
任务发布
http通讯
服务状态查询
http 通讯
发送任务事件
http 通讯
协调层
任务协调器
服务状态查询
http 通讯
服务状态管理器
管理监控配置
http 通讯
数据聚合器
对应于逻辑架构中的
DBaaS 后端总控
调度和通信子模块
获取软件镜像
http 通讯
任务操作
http通讯
通知服务状态
rpc 通讯
执行层
资源管理
agent
服务状态查询
http 通讯
服务状态管理
agent
监控管理
agent
发送监控数据
rpc通讯
发送监控事件
rpc通讯
对应于逻辑架构中的
UPProxy集群节点和UPSQL集群节点
容器管理操作
http通讯
Docker
服务容器层
容器管理操作
cmd
Switch
manager
管理数据拓扑 操作
cmd
容器管理操作
cmd
UPproxy
容器管理操作
cmd
UPsql
对应于逻辑架构中的
UPProxy集群节点和UPSQL集群节点
的容器层和服务实例层

核心调度设计

调度算法分为过滤器（Filters）和策略器（Strategy）两部分：
Filters:根据约束条件，排除不满足条件的物理机。
Strategy:对物理机的资源状况和订单需求的匹配情况进行打分，根据打分结果排序，选出在最佳的物理机上进行容器创建，使得集群的资源使用状态趋向均衡。

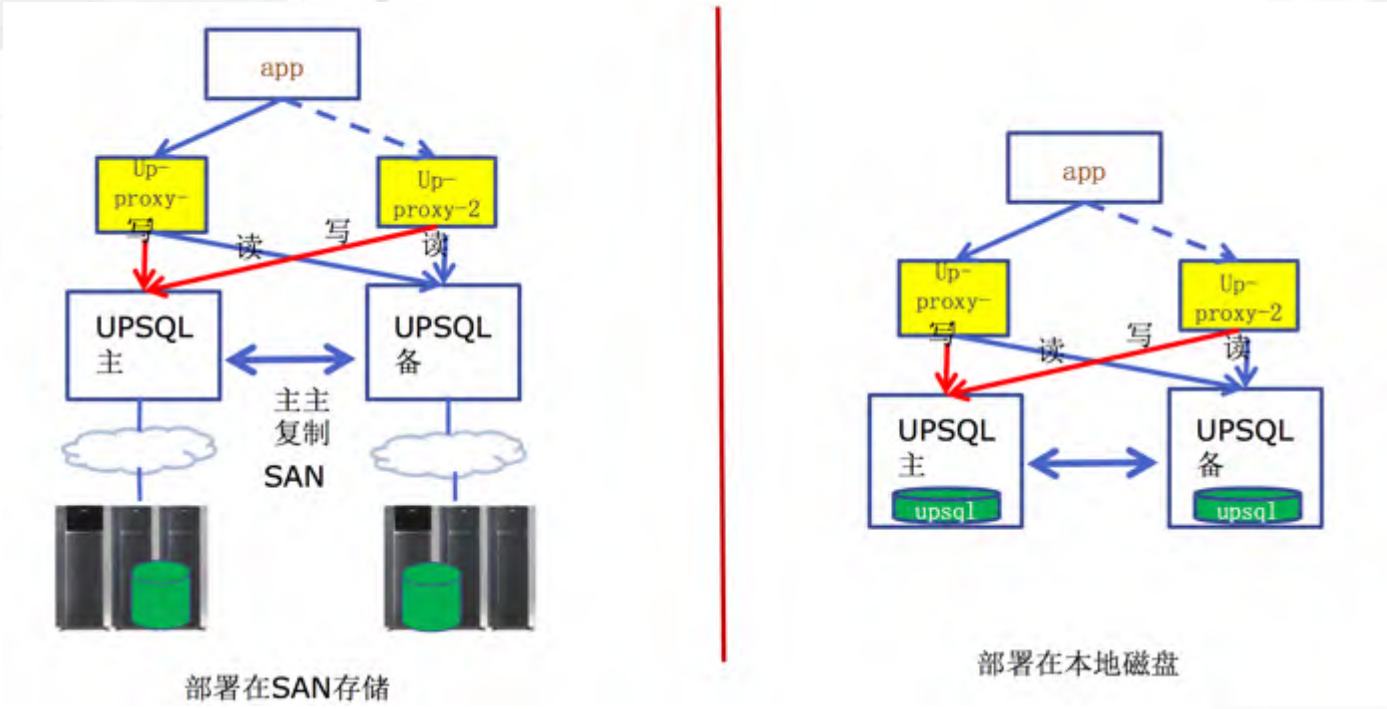
约束过滤条件

- 物理机的空闲资源满足需求规格
- 一个集群使用一个存储
- 主从实例部署在不同集群中
- Upsql集群节点分布在不同物理机上

- 对物理机的资源剩余及负载情况打分排序
- 选择最佳服务器部署

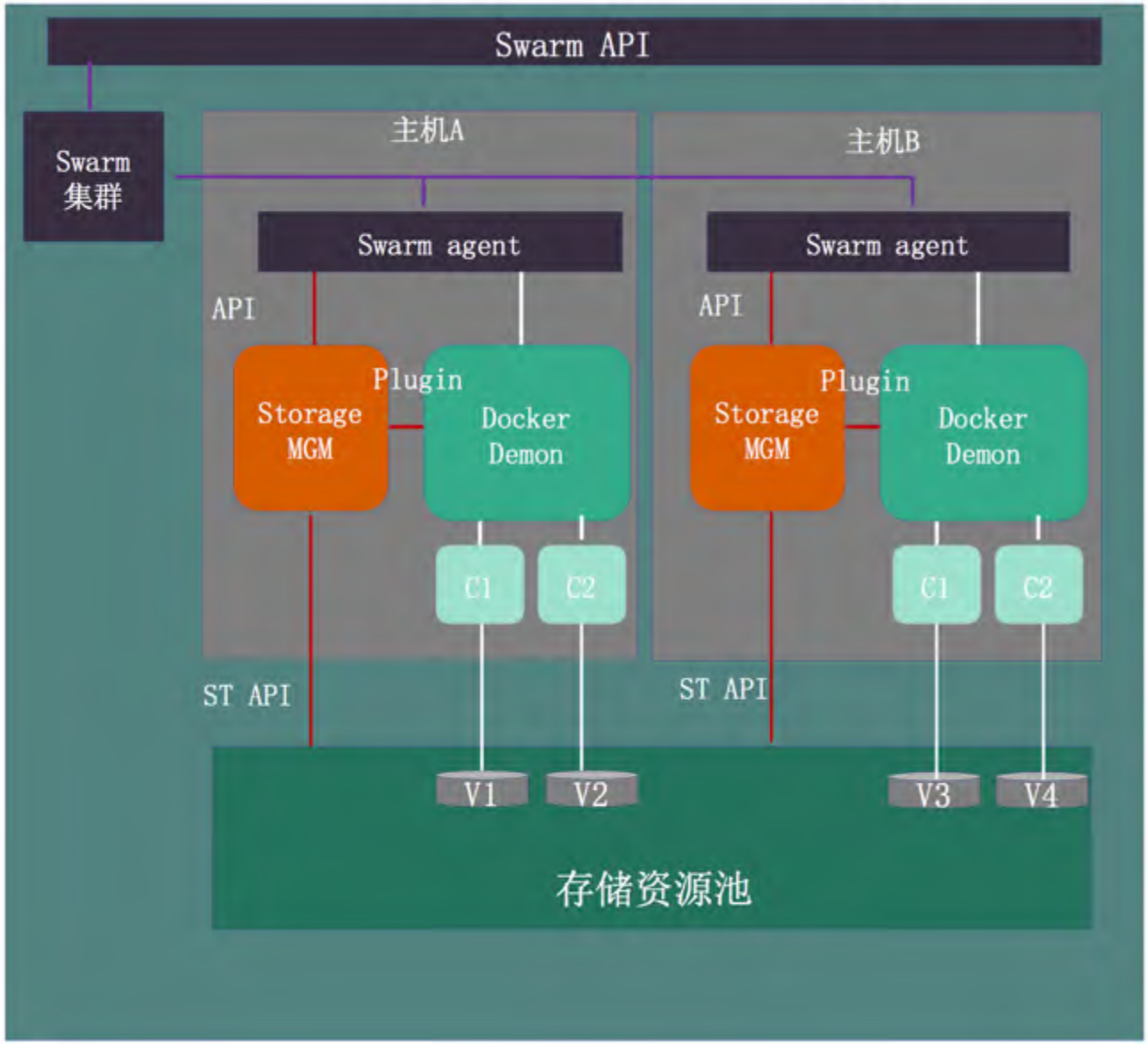
支持多种存储架构

1、SAN存储：数据服务部署在SAN上,主从节点部署在不同SAN存储上
2、本地存储：数据服务部署在本地磁盘上,同时支持使用SSD盘、FLASH 卡作为高 速BINLOG日志目录,提升性能。



容器存储方案

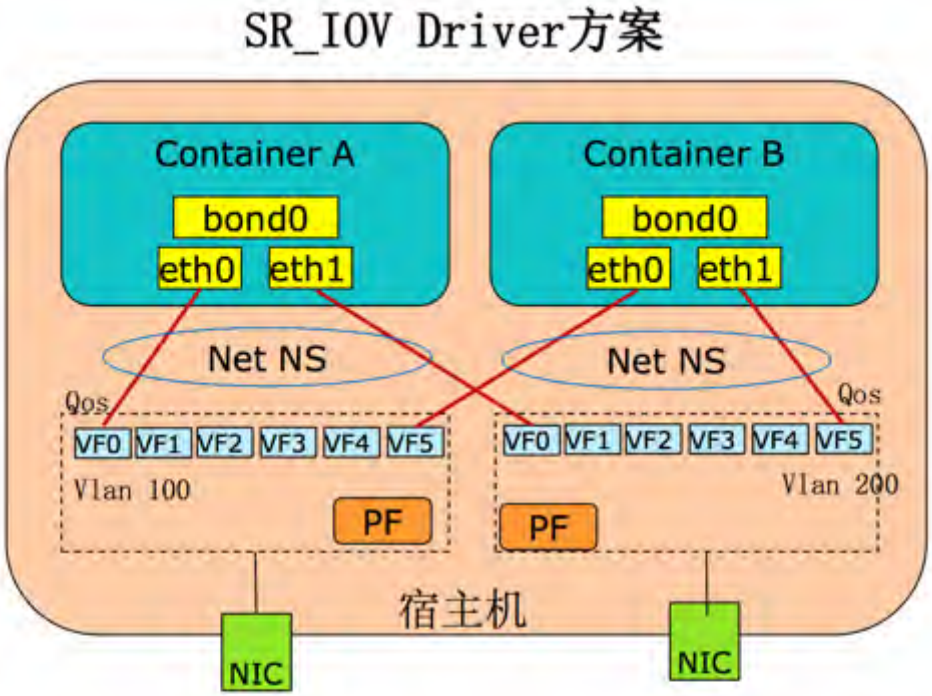
Volume Plugin：StorageMGM



平台存储管理解决方案

- 自主研发容器数据卷管理器StorageMGM，与docker集群管理框架Swarm集成，可以便捷完成Docker集群的存储管理（业界有Flocker）
- StorageMGM以模块化方式提供各种存储的管理API接口，可以便捷扩展
- StorageMGM作为Docker的标准Volume Plugin实现容器的volume管理
- 通过StorageMGM实现了大规模容器集群、多存储平台集成管理；实现带卷容器迁移；实现容器和卷的松耦合；支持迁移服务器之间的本地数据卷

高性能容器网络方案



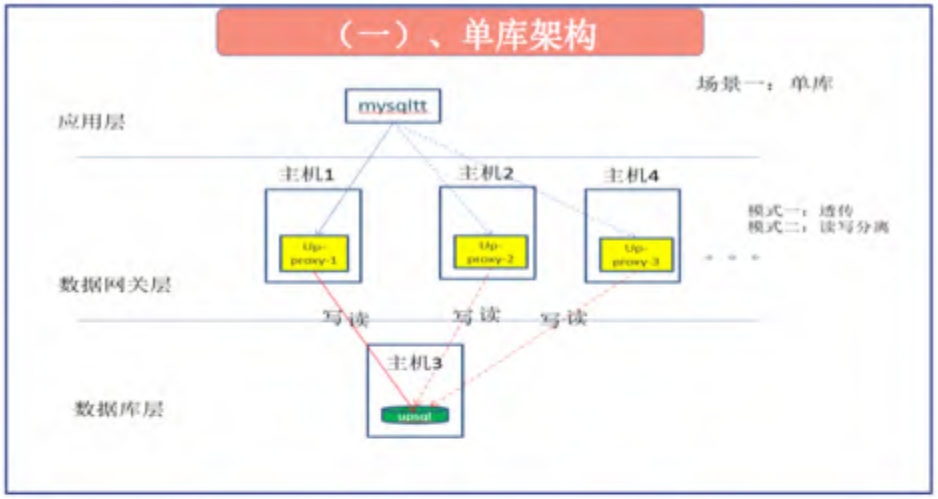
平台内部远期网络架构

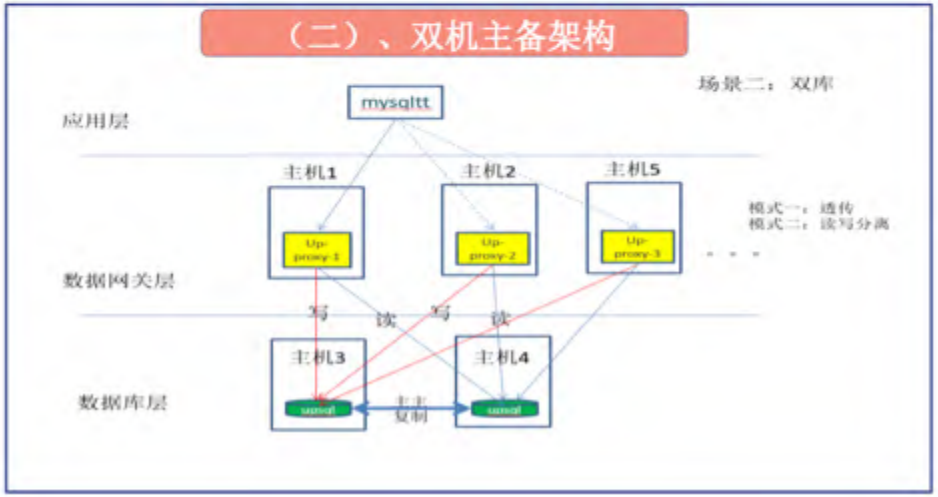
- 主机层面采用SR_IOV Driver网卡硬件虚拟化技术满足“网络融合+SDN”网络虚拟化技术架构
- 宿主机上包含管理网和业务网，平台管理控制通过管理网，业务用于挂载共享目录；容器内部只设业务网
- 通过支持SR_IOV的网卡PF虚拟出多块VF，VF/PF通过不同vlan进行网络分层、隔离容器和宿主机网络
- VF通过Linux的Net NS直接映射给Docker容器内部，并进行双网卡bonding
- VF上配置Qos策略，进行流控
- 该网络架构性能好，相较于物理环境下性能损耗非常少，适合对网络带宽、网络时延要求较高的数据库环境，同时实现了对网络的隔离及Qos等安全性要求。

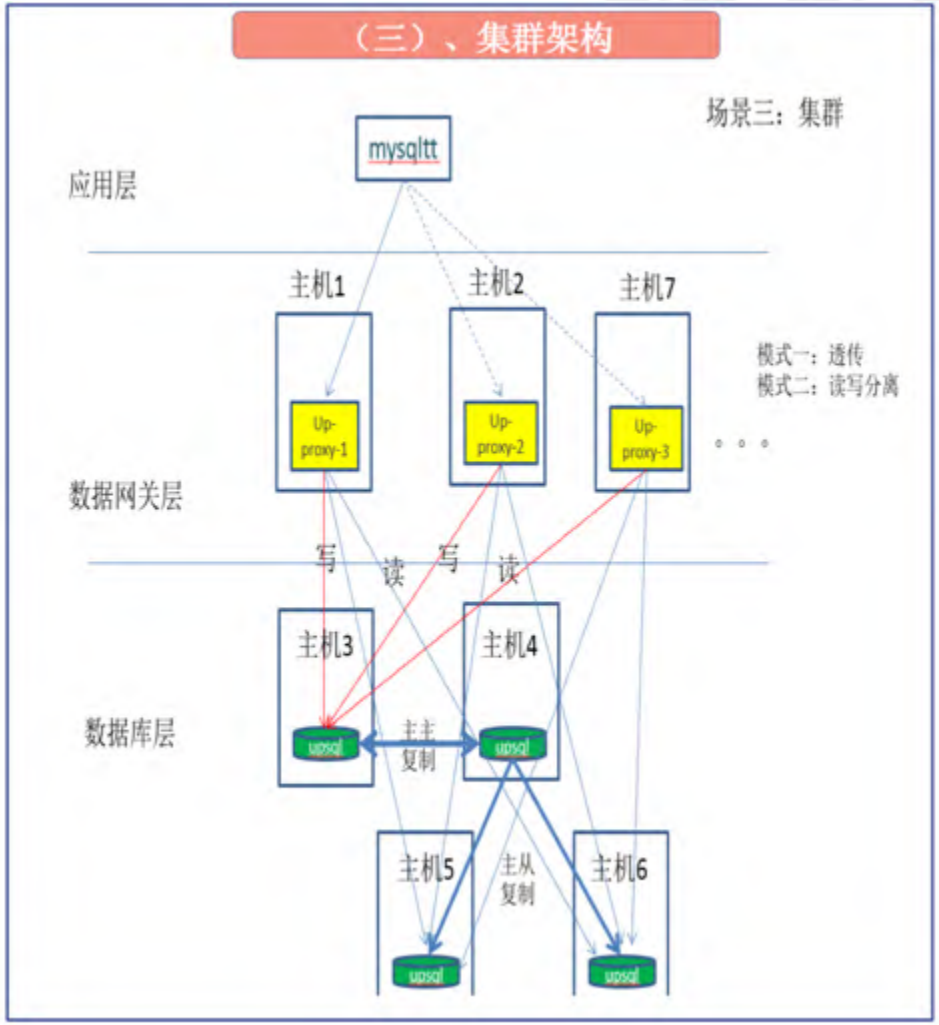
服务化相关

支持灵活的服务拓扑

1、单库架构,只有一个UPSQL节点,一次可以申请N个单库(1XN)
2、双机架构,提供双机主主复制2个UPSQL节点实现高可用保护,一次可以申请N对双机((1X1)XN)
3、集群架构,提供master-standby-slave多节点集群实现高可用保护和性能扩展,slave可以N个(1X1`XN)
4 、Sharding 分片架构



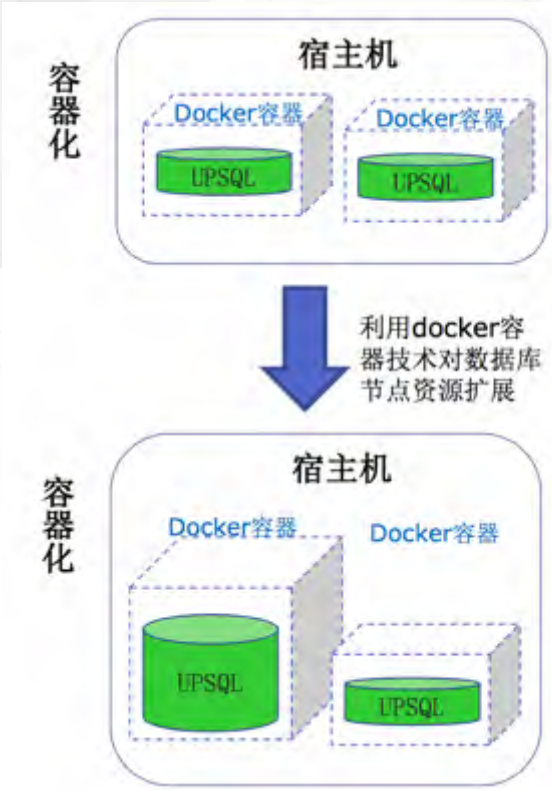


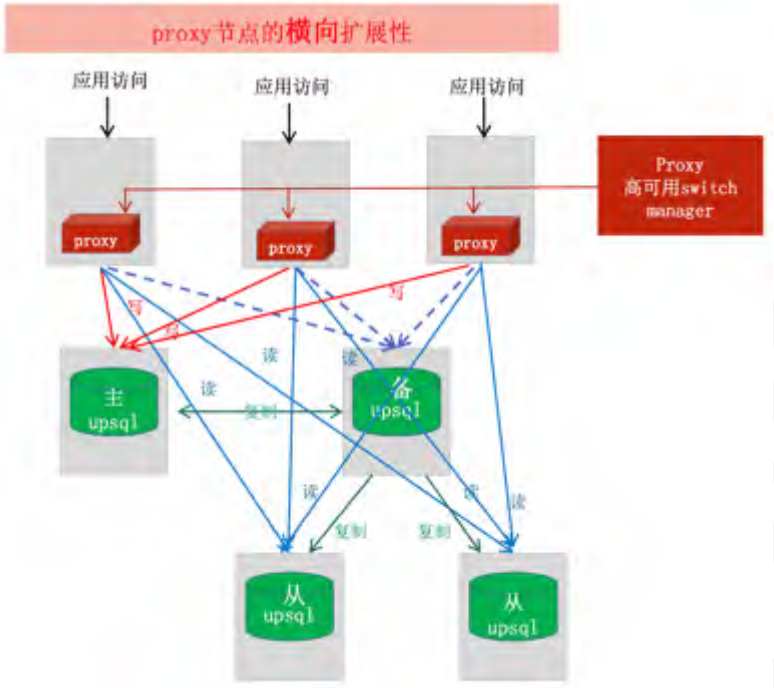


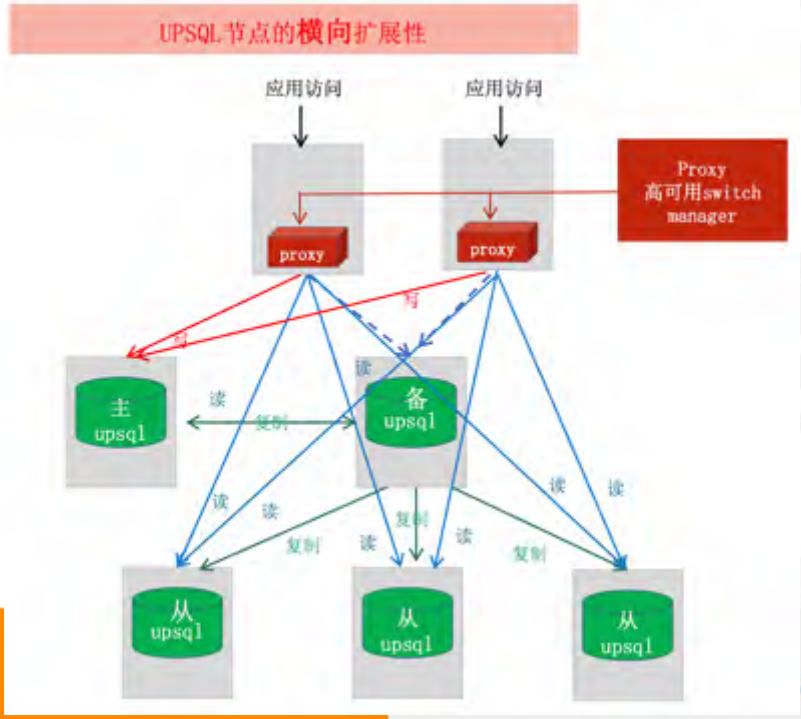
支持从SQL,NoSQL, NewSQL 完整而灵活的服务拓扑

- 支持MySQL 所有常规集群拓扑(5.6/5.7)
- 支持用户自定义的MySQL集群拓扑
- 支持MySQL 5.7.17开始的的MGR结构
- 支持Redis Sentinel 结构，Redis Proxy结构和全对等的Sharding Cluster 结构
- 即将完成对主流IMDB内存NoSQL数据库的支持
- 计划年内完成主流的NewSQL集群的服务和服务拓扑支持

服务伸缩


容器化
宿主机
Docker容器
Docker容器
UPSQL
UPSQL
利用docker容器技术对数据库节点资源扩展
容器化
宿主机
Docker容器
Docker容器
UPSQL
UPSQL

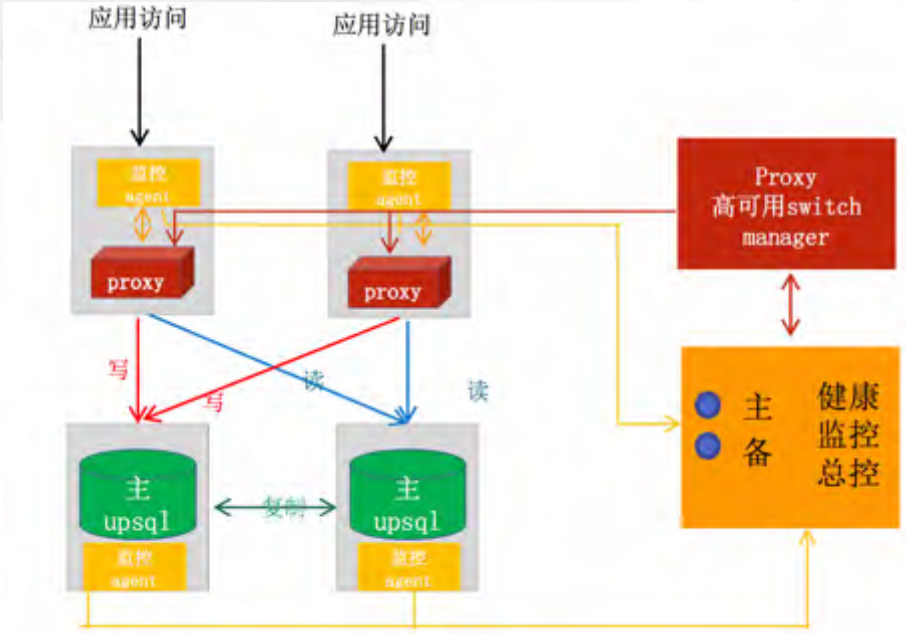

proxy节点的横向扩展性
应用访问
应用访问
应用访问
Proxy
高可用switch
manager
proxy
proxy
proxy
写
写
读
读
主
upsql
读
复制
备
upsql
读
读
复制
复制
从
upsql
从
upsql

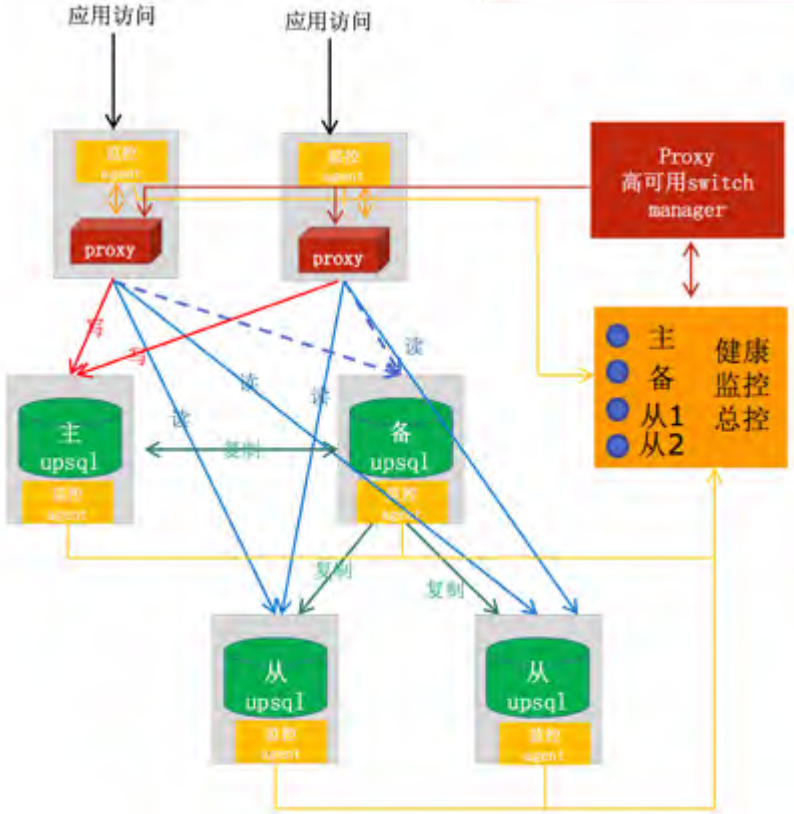

UPSQL节点的横向扩展性
应用访问
应用访问
Proxy
高可用switch
manager
proxy
proxy
写
写
读
读
主
upsql
读
复制
备
upsql
读
读
复制
读
复制
读
读
复制
从
upsql
从
upsql
从
upsql

3

高可用与安全

服务的高可用设计(以MySQL服务为例)



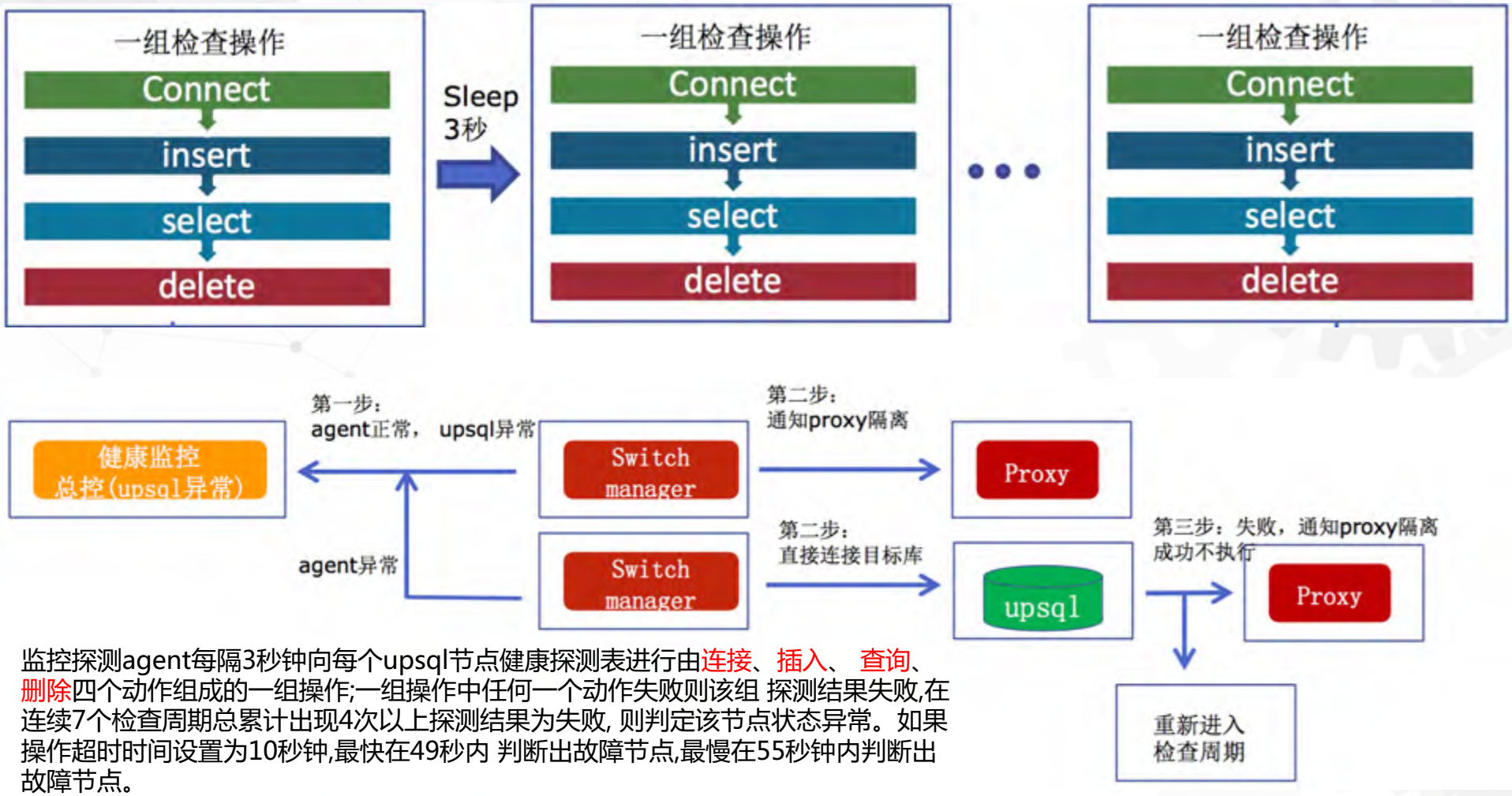


高可用由监控监控、proxy switch manager、proxy三部分构成

- 监控检查模块实时收集集群节点的健康状况

•proxy switch manager实时从健康 监控总控获取每个节点的监控情况, 同时根据主备关系生成新的集群拓 扑关系报文下发给proxy,proxy根 据新集群拓扑关系重新进行访问连 接路由

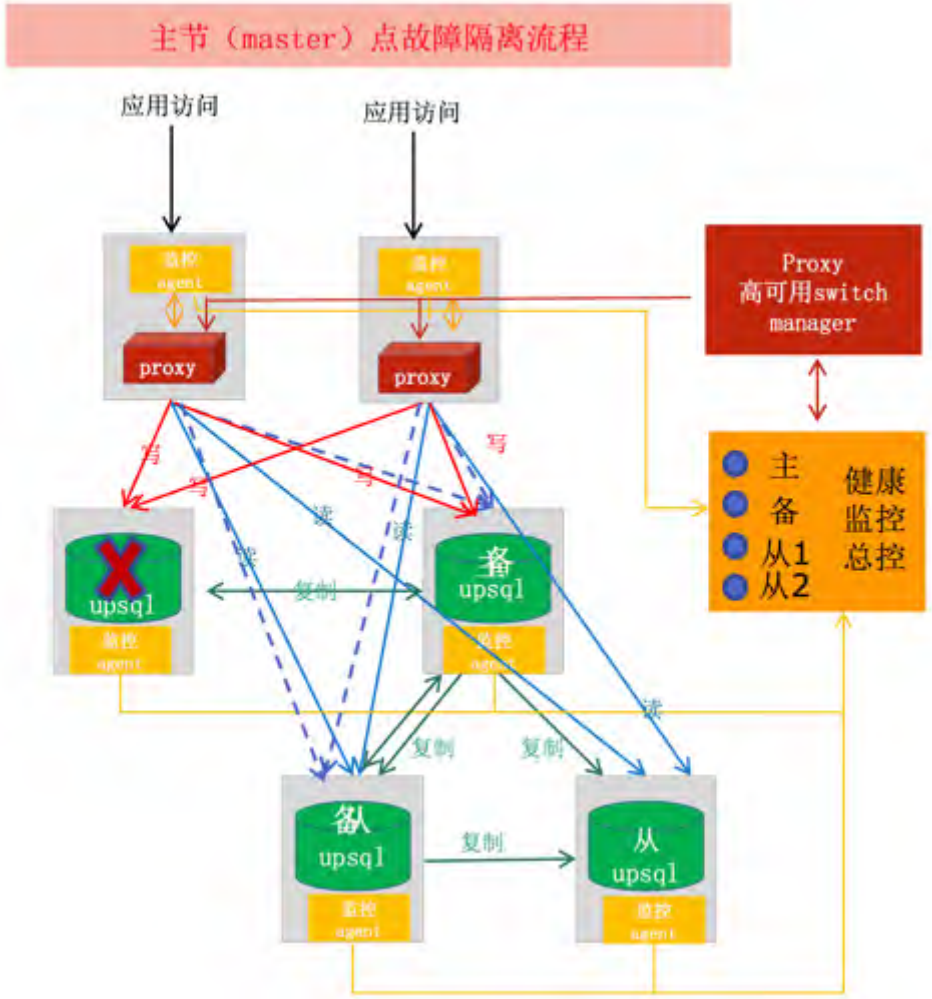
服务故障的可用性检测和故障转移(以MySQL服务为例)



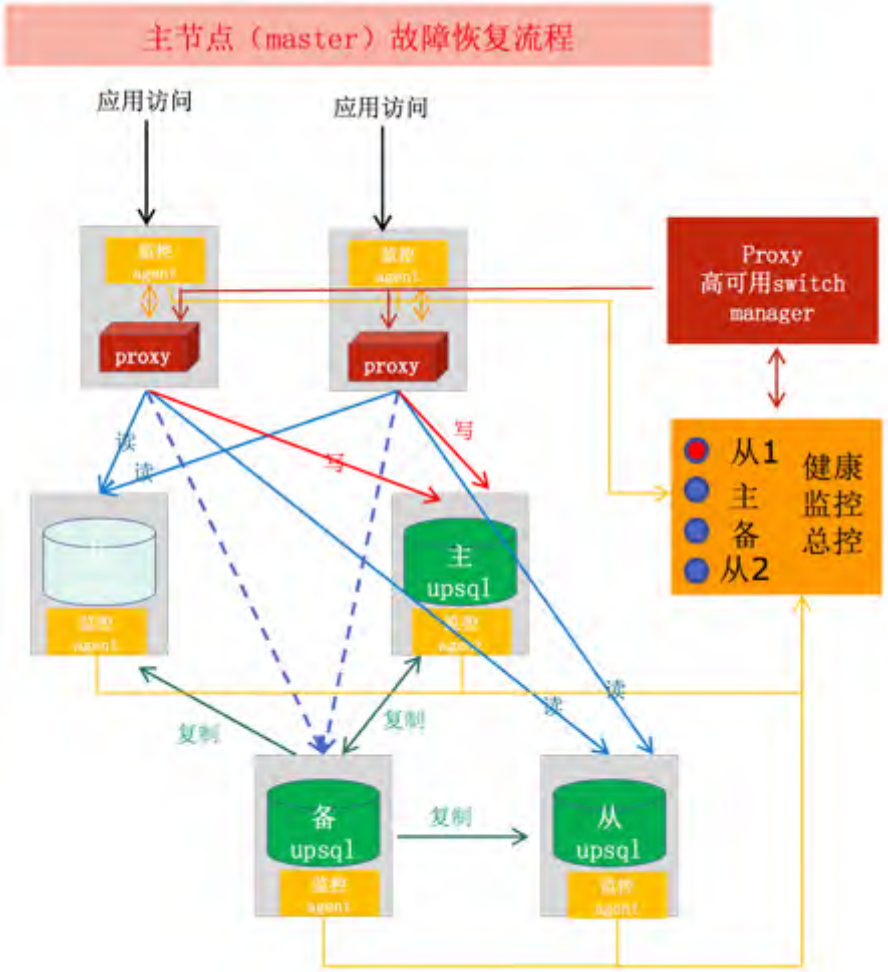
监控探测agent每隔3秒钟向每个upsql节点健康探测表进行由连接、插入、 查询、删除四个动作组成的一组操作;一组操作中任何一个动作失败则该组 探测结果失败,在连续7个检查周期总累计出现4次以上探测结果为失败, 则判定该节点状态异常。如果操作超时时间设置为10秒钟,最快在49秒内 判断出故障节点,最慢在55秒钟内判断出故障节点。

主服务节点故障隔离和恢复 (以MySQL服务为例)

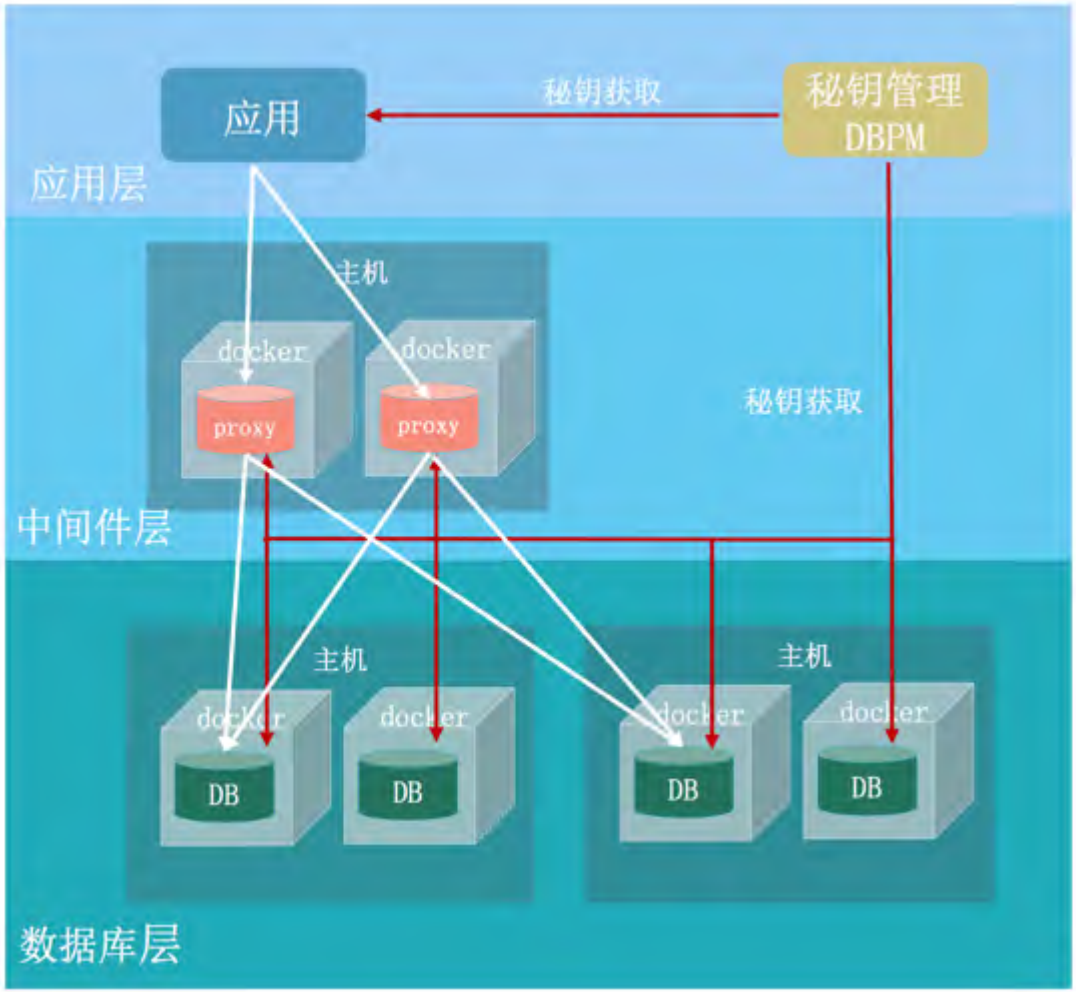
- 健康agent发现节点异常上报至健康 总控
- Proxy连接主库失败
- Proxy switch manager从健康监控 总控发现主节点故障,给proxy发隔 离指令
- Proxy接收到指令后,将主库路由指 向原来的standby数据库,standby 升级为主库
- 将其中一个从库升级为新的standby
- 剩余的从库将从新的standby同步数据



- 故障原来主节点恢复之后,将会以 从库的角色加入集群
- 先从standby节点同步数据至一致
- Proxy switch manager给proxy发从节点加入指令
- Proxy接收到指令后,将读请求分流 送往新加入的从节点



结构化的容器服务安全方案



架构层面

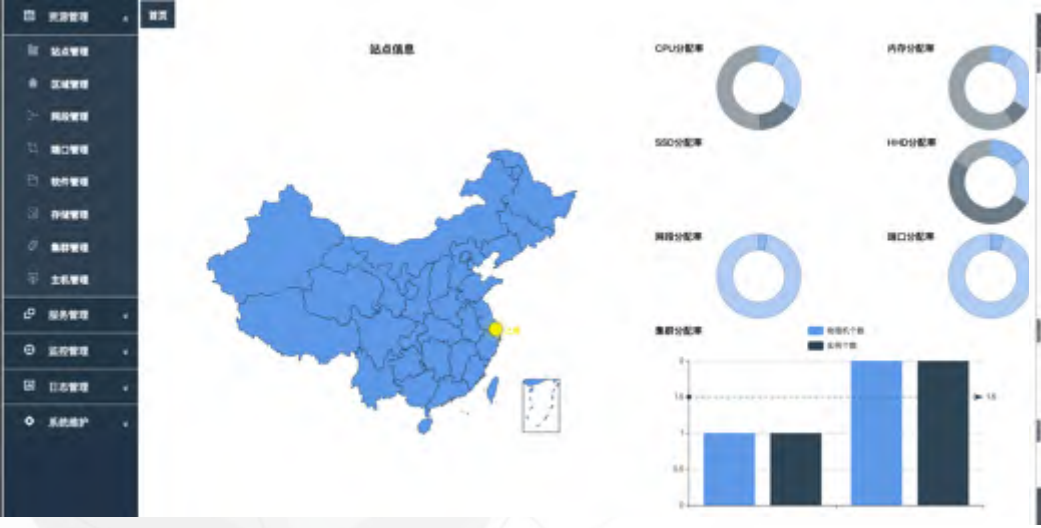
- 多层级防护，防火墙、IDS、IPS、应用、平台
- 采用容器技术，对网络、计算、存储资源完全隔离
- 增加中间件层，隔离了应用与数据库直访

应用层面

- 白名单、用户权限、异常连接、异常SQL
- DB层进行白名单访问管理、用户权限管理、支持数据加密
- 访问秘钥动态管理

基础环境

- 主机安全加固
- Docker、数据库软件安全漏洞加固
- Docker权限控制



点击展开) 单元格背景警示色：danger>80%; warning>60%; success>30%; info 0~30%

所属站点 铭光 | 所属集群 | 使用状态 | 查询 | 重置

当前伐值：100 | 使用伐值修改

主机名	所属站点	所属集群	网络地址	CPU使用率		内存使用率		HDD分配率		SDD使用率		状态	监控
				状态	分配值-总值	状态	分配值-总值	状态	分配值-总值	状态	分配值-总值		
node001	铭光	数据集群001	192.168.2.141	✓	1/4	✓	2/7	✓	9/19	✓	0/0	启用	
node002	铭光	代理集群001	192.168.2.142	✓	1/4	✓	0/7	✓	19/19	✓	0/0	启用	
node003	铭光	代理集群001	192.168.2.143	✓	1/4	✓	0/7	✓	10/19	✓	0/0	启用	

THANKS

SequeMedia 盛拓传媒 ITPUB ChinaUnix