同程艺龙

QCon 全球软件开发大会
INTERNATIONAL SOFTWARE DEVELOPMENT CONFERENCE

BEIJING 2018

《MySQL 容器化部署实践》

演讲者 / 王晓波

主办方 Geekbang. 极客邦科技 InfoQ

背景

- 同程旅游早期的数据库都以单库的MySQL。
- MySQL的单库，导致TPS最终还是会成为一个瓶颈。
- MySQL+DB中间件解决水平拆分问题。
- MySQL水平拆分的引入会使数据库实例数量大幅上升,传统运维手段维护成本高,交付能力差。

MySQL数据库为何要Docker化

1.MySQL数据库迅速爆炸式增长后，服务器规模不断增大，快速部署是个问题。
2.随着业务的发展，扩容数据库的不方便不快捷，也是个问题。
3.大量数据量小的数据库系统也单独部署在物理机，浪费问题突出。
4.DBA的数据库自动化标准化运维的需求。
5.Docker在同程的大规模使用，应用部署环境100%容器化，有Docker丰富的经验
。

让数据库的部署点单化开启

配置	DB架构	硬件选型	机房
2核4G 4核4G	一主一从	SATA-SSD	A机房
4核8G 8核8G	一主多从	PCIE-SSD	B机房
8核16G 16核16G	分片集群	大容量磁盘SAS	C机房
16核64G 32核64G			D机房
32核128G			

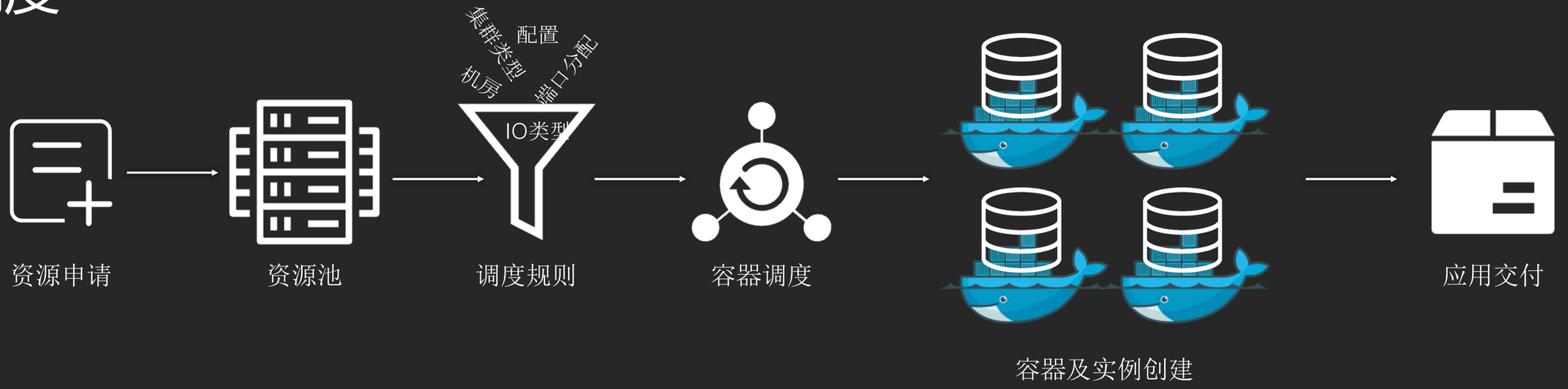
容器化之后的MySQL就是一个私有DB云

- 过载保护机制
- 数据库及实例迁移
- 扩容缩容
- 集群节点管理
- 高可用切换
- 资源池管理
- 自动化备份
- 慢日志分析及查看
- 便捷的告警管理
- 精美的图形展示
- 巨细无遗的监控项
- 集成高可用方案
- 分片集群创建
- 主从集群创建

总体架构


应用
应用
VIP
DB中间件
资源管理|资源调度
复制与迁移|高可用保障|备份恢复
数据存储
分布式监控
自动化运维
告警分析
告警发送
指标画图
趋势预测
自定义脚本
Open API
数据校验
实例迁移
秒级监控诊断
慢日志分析

资源池调度



为了保证MySQL的高可用，需要在Docker容器分配时如何保障主从不在同一宿主机上。我们通过自研Docker容器调度平台管理所有宿主机和容器,自定义Docker容器的分配算法。实现了MySQL的高密度,隔离化,高可用化部署。

调度规则：

1.同一复制集群的实例在不同主机上。

2.优先分配CPU、内存、磁盘空间资源最空闲的主机。

3.根据IO需求调度容器创建在不同IO类型的主机。

4.申请新集群时，若IO要求高则按照宿主机的IO情况，优先选择IO最空闲的主机。

5.VIP集群必须主从端口一致，Proxy接入的集群端口无需一致。

6.VIP集群端口基于网段递增，Proxy集群端口基于IP递增

Docker里放了什么

内核版本

Kernel版本 4.7

操作系统

CentOS 7.2

宿主机

部署服务器监控、容器监控agent容器

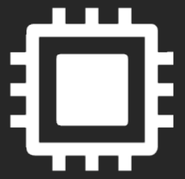
容器

Docker版本 1.12,部署监控及系统服务agent

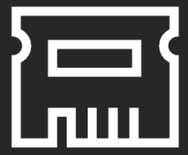
镜像

MariaDB镜像(按产品)、MySQL5.7镜像(按产品)、监控容器镜像、HA管理系统镜像、实例迁移服务镜像、监控服务端镜像

资源隔离

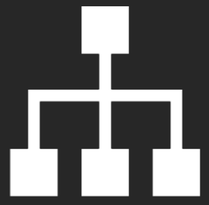


CPU最大超卖3倍，通过cpu-period配合cpu-quota一起使用，来限制容器的CPU的使用量。比cpuset-cpus绑定CPU的方式灵活。



限制容器内存，且内存不超卖。通过—memory限制内存，同时结合MySQL自身参数控制几个内存大户(比如buffer_pool等)，最后配合lxcfs增强隔离性。

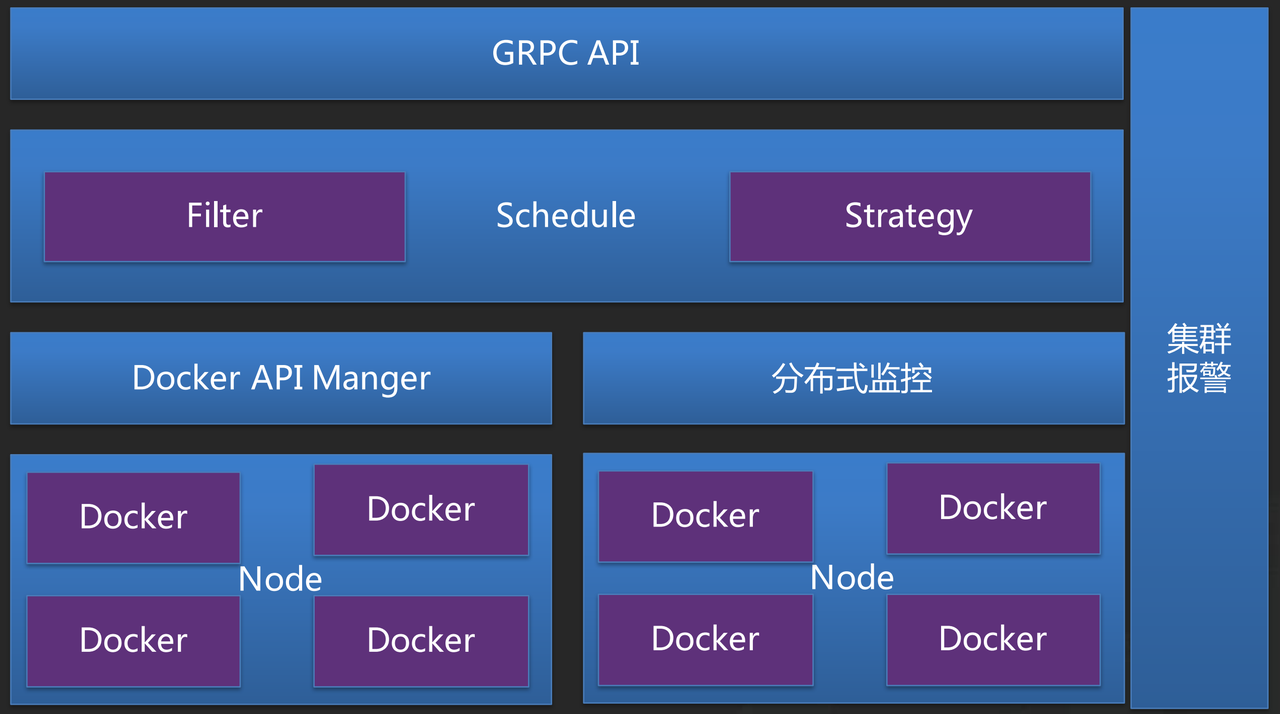
IO方面由于我们采用挂载宿主机本地的磁盘设备，还不能做到彻底隔离。所以对于高IO的实例使用的是PCIE-SSD。磁盘空间方面，我们在申请时会预估出一个量，使用超过80%的时候会结合本地磁盘空间评估是否有足够空间扩容，若宿主机剩余空间不足会启动迁移扩容流程。



目前使用的host模式，无法隔离网络。但是考虑到10G接入，且单机密度可控的情况下，网络消耗不会过载。另外目前我们已经在线下尝试结合Ovs+Dpdk的方案实现网络隔离。

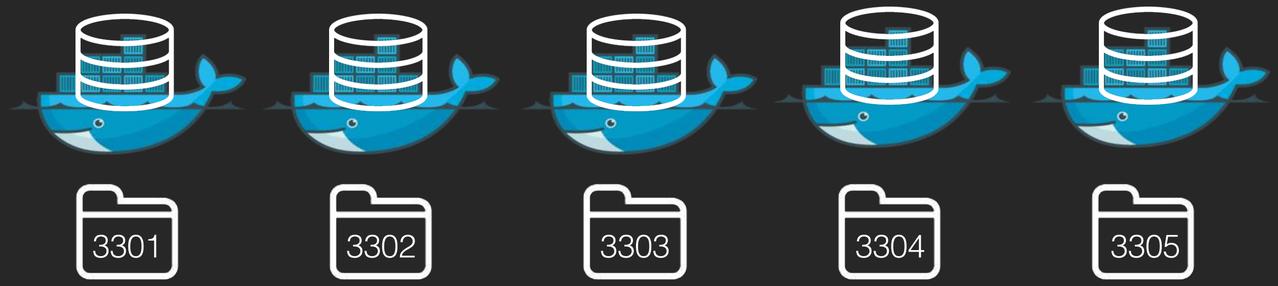
PS：容器虚拟化带来轻量高效，快速部署的同时，docker容器在隔离性方面也存在一些缺陷。例如，在容器内部proc文件系统中可以看到Host宿主机上的proc信息。这样就导致了一些问题，比如监控信息不准确、限制内存会导致应用程序OOM等。我们基于lxcfs组件来增强容器的隔离性。

容器的调度



- 提供两种API：
- Docker API的封装，用于创建,删除,修改。
- 集群管理API，用于集群管理。
- Scheduler调度：选择最优节点创建容器。
- Agent：用于连接监控模块，上报系统运行状况。

磁盘挂载



宿主机磁盘

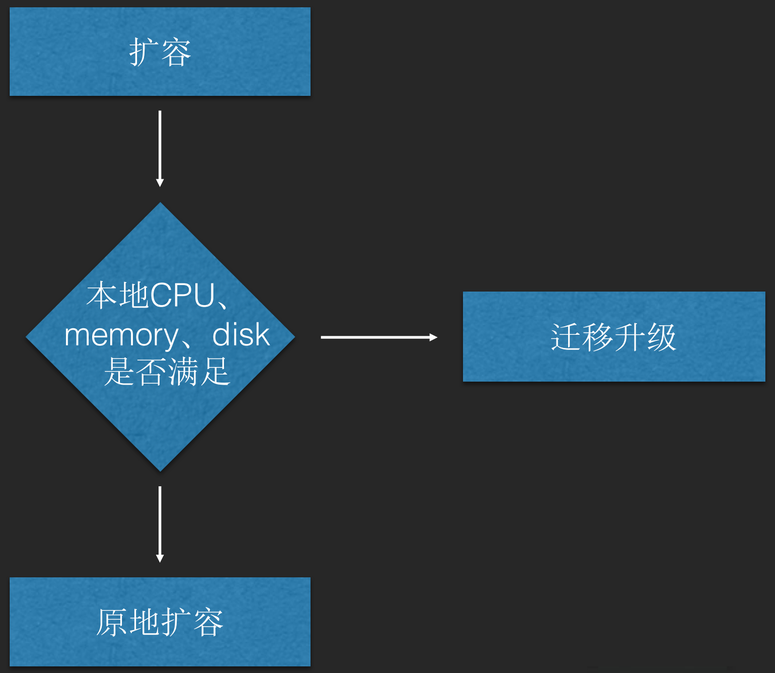
- 为了保证容器内的MySQL实例有更好的磁盘IO性能。采用了本地宿主机磁盘挂载到容器内的方式(每个实例对应一个文件夹)。

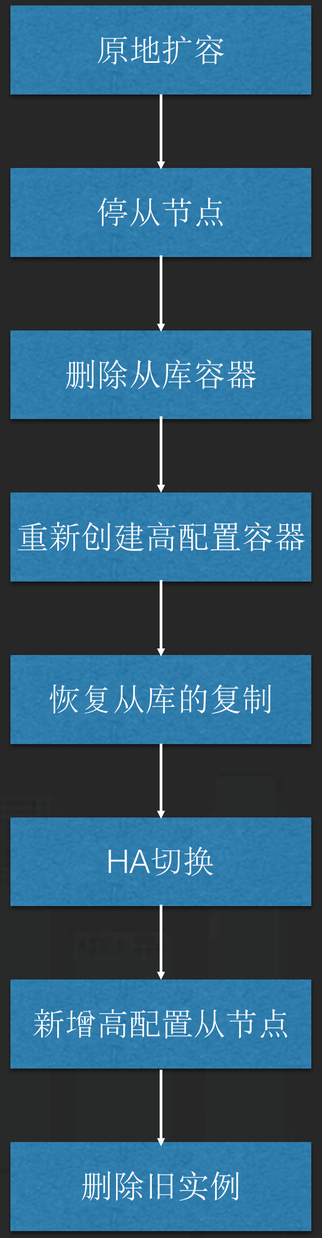
这种方式的优势是IO性能最佳，随之而来的缺点是磁盘容量不好估算，有可能在使用了一段时间后出现磁盘空间不足的问题，这个时候则会启动迁移扩容的流程。

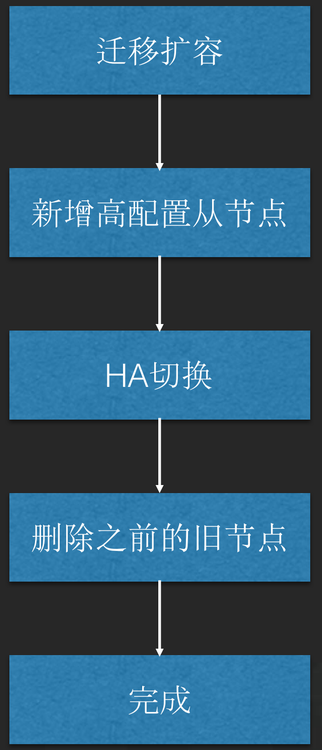
已经在和提供高密度IO分布式存储解决方案厂商接触，计划测试平台接入分布式存储的方案。

集群扩容

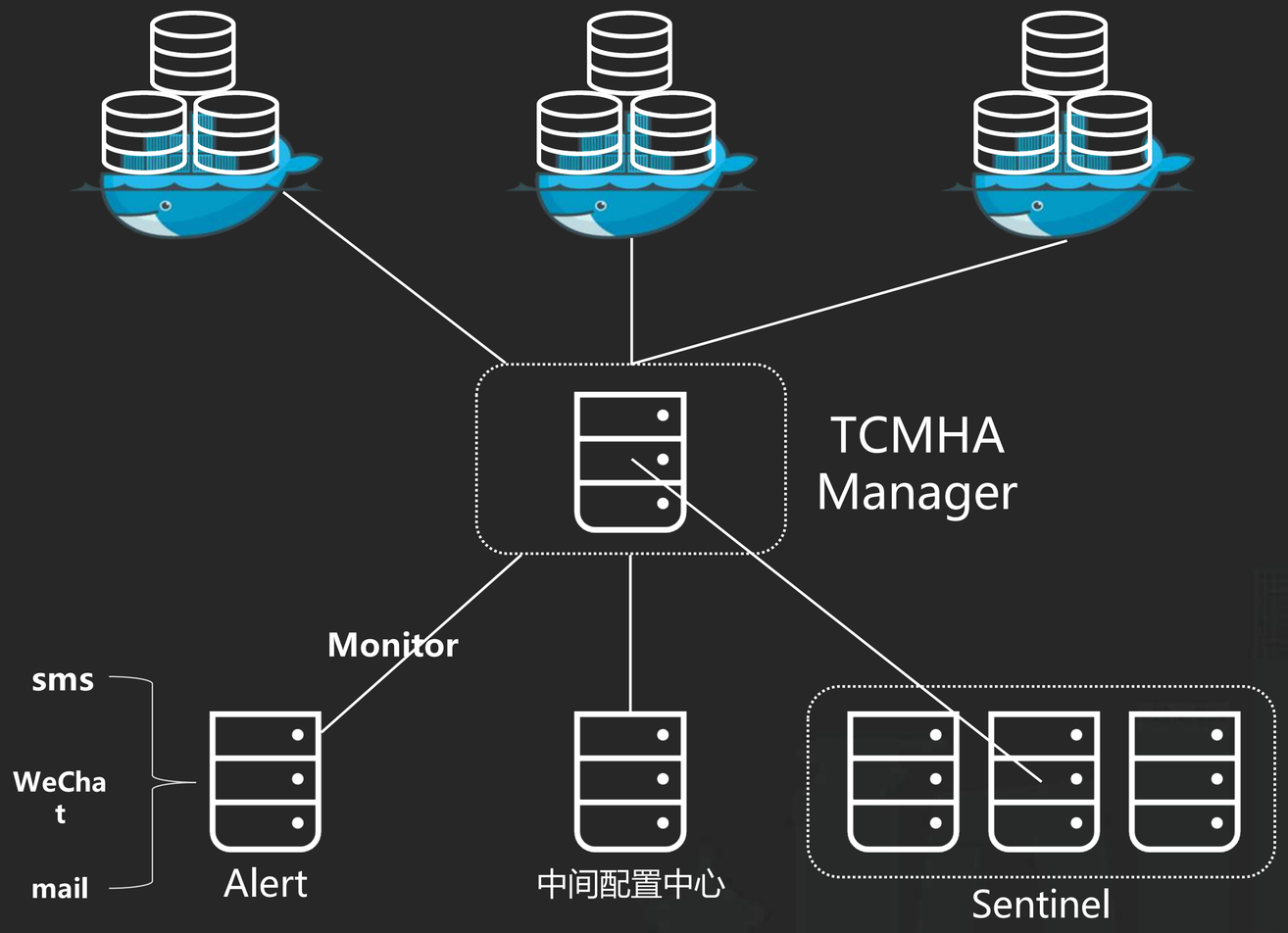
扩容逻辑


扩容
本地CPU、memory、disk是否满足
迁移升级
原地扩容


原地扩容
停从节点
删除从库容器
重新创建高配置容器
恢复从库的复制
HA切换
新增高配置从节点
删除旧实例


迁移扩容
新增高配置从节点
HA切换
删除之前的旧节点
完成

高可用管理



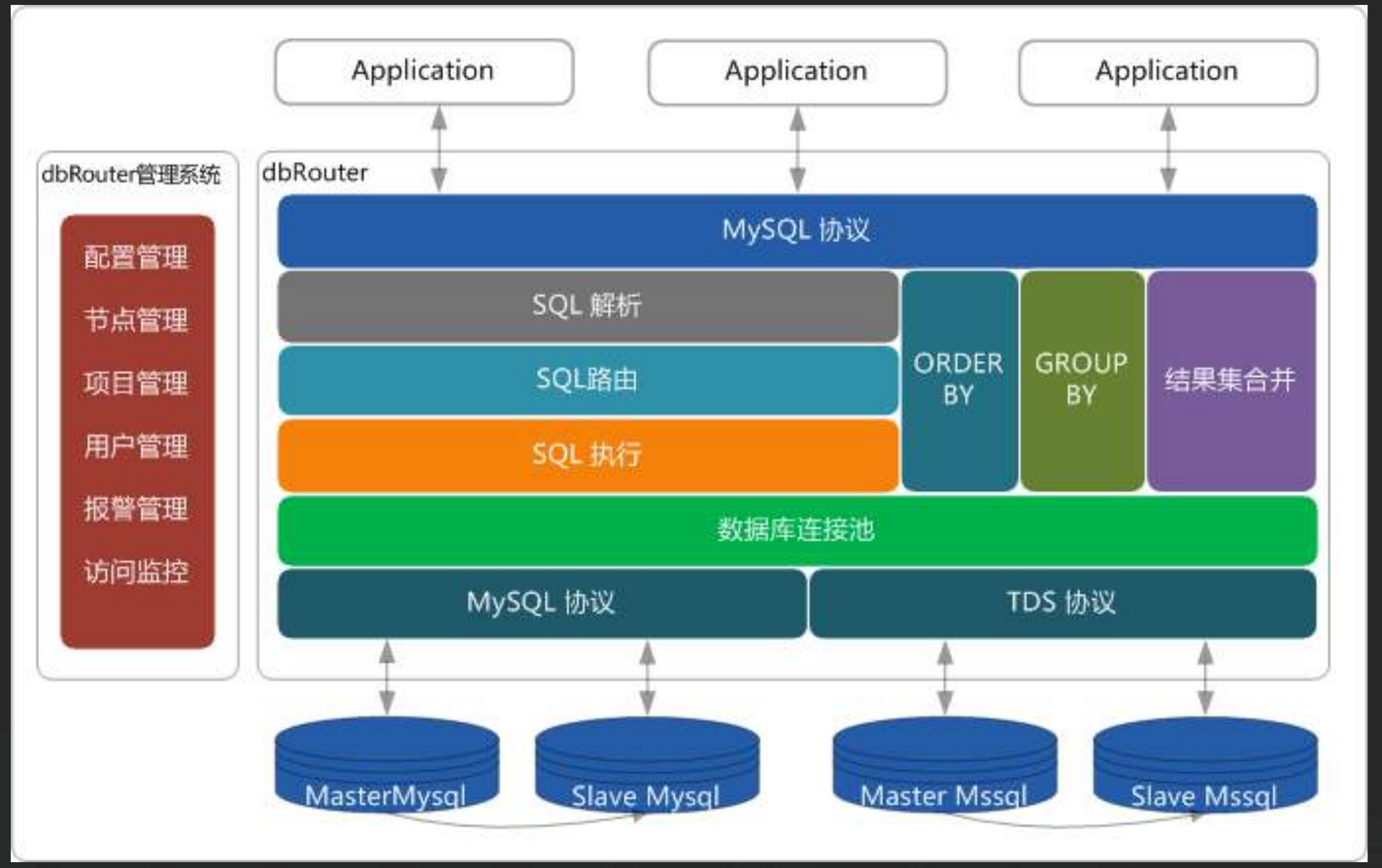
TCMHA 管理分为两个部分：

- MariaDB、MySQL5.6的高可用管理是基于开源工具MHA定制开发后的工具完成的，该工具支持了MariaDB的gtid、更完善的哨兵检测机制、对接了DB中间件等定制化功能。
- MySQL5.7的MGR复制集群是我们自己写的一套高可用组件配合DB中间，实现无感知的高可用切换。

DB中间件

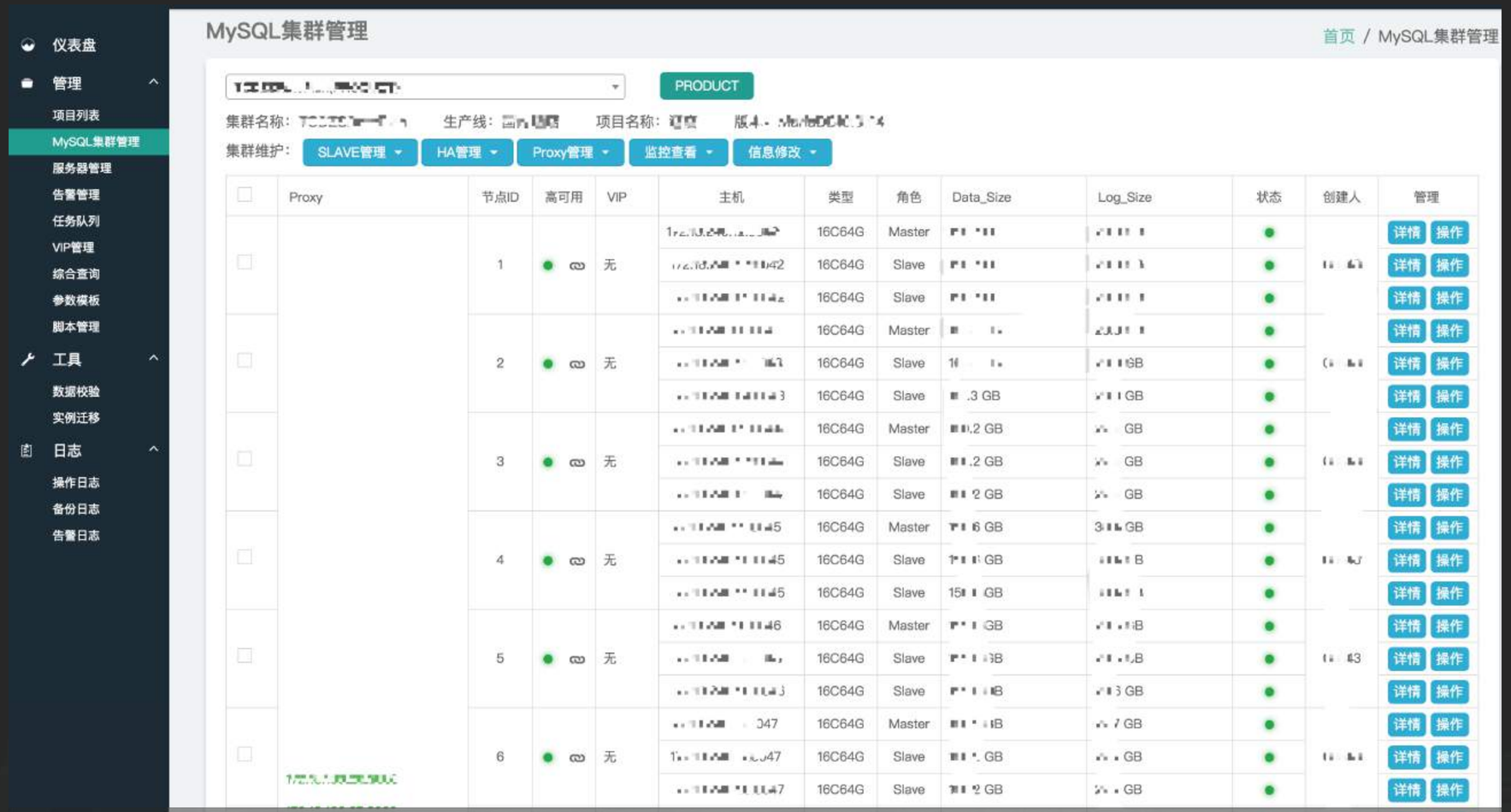
兼容mysql协议
支持SELECT/INSERT/UPDATE/DELETE语句
支持单DB实例上的inner join
支持单DB实例上的事务
支持聚合函数：max、min、sum、avg、count
支持：distinct、order by、group by、limit、
top:definition text

支持多种拆分方式
不分区
根据关键字段，进行hash分区
根据时间字段，进行时间分区
根据关键字段，进行区间分区
读写分离模式



集群管理

- HA切换
- Slave增加删除
- 创建新集群
- 查看监控
- 慢日志分析
- 慢日志查看
- 手动备份



备份系统

多样化备份：

1.物理备份：

通过xtrabackup备份20G以上的实例。

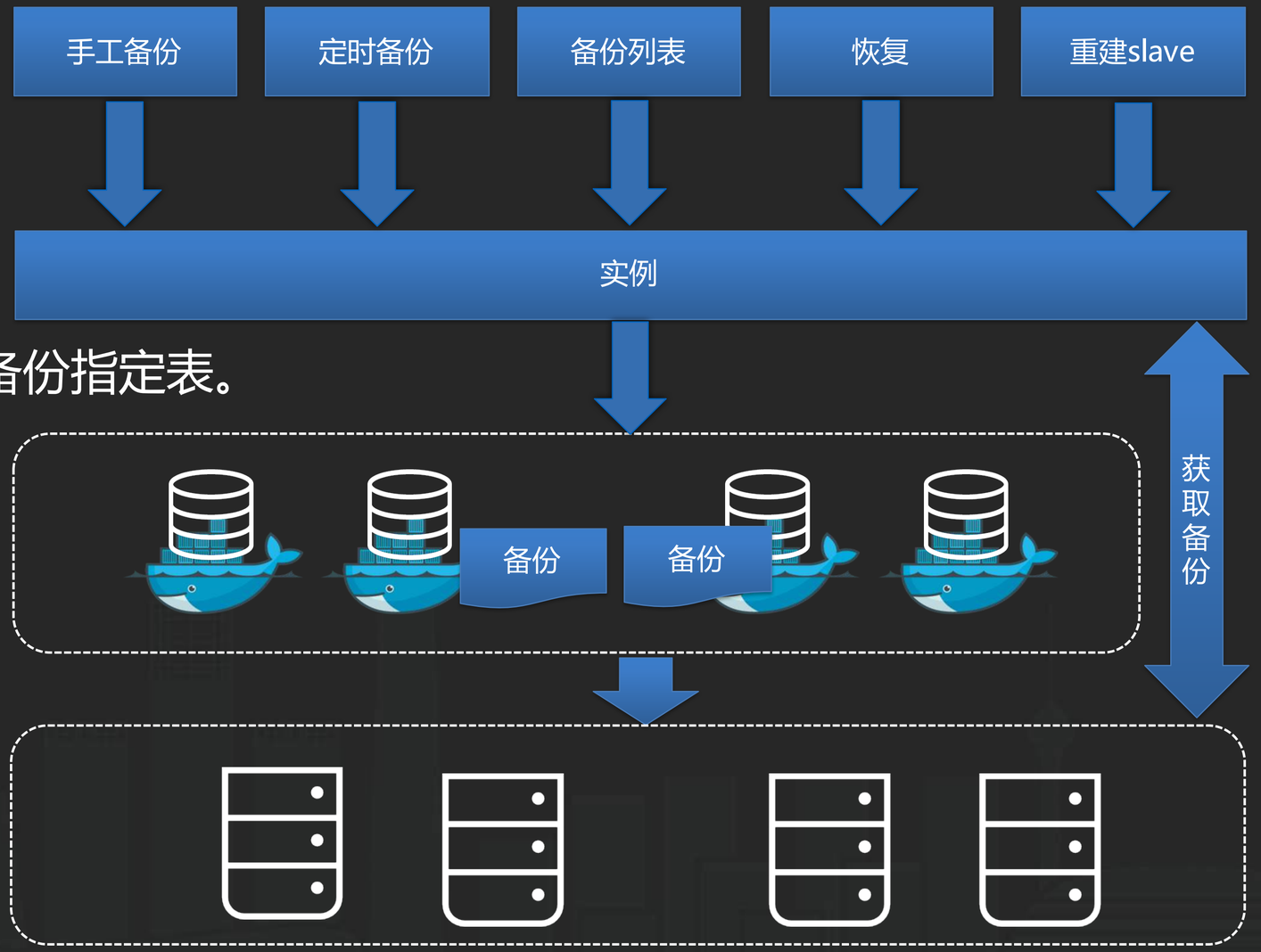
2.逻辑备份：

通过mydumper备份20G以下的实例或备份指定表。

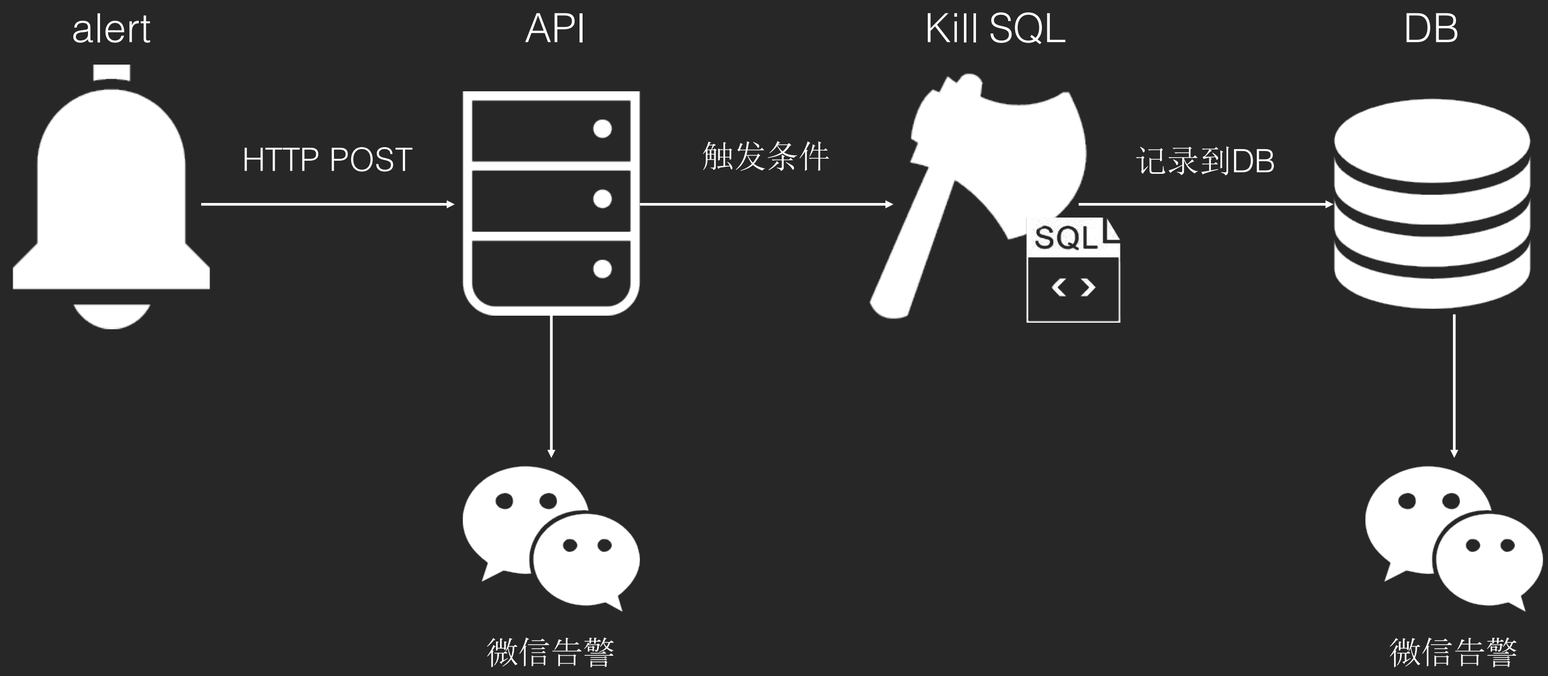
备份策略：

1.每日自动化备份。

2.DBA临时手工备份。



过载保护



目的:

- 当系统数据库较高时确保大多数请求能够够正常访问。

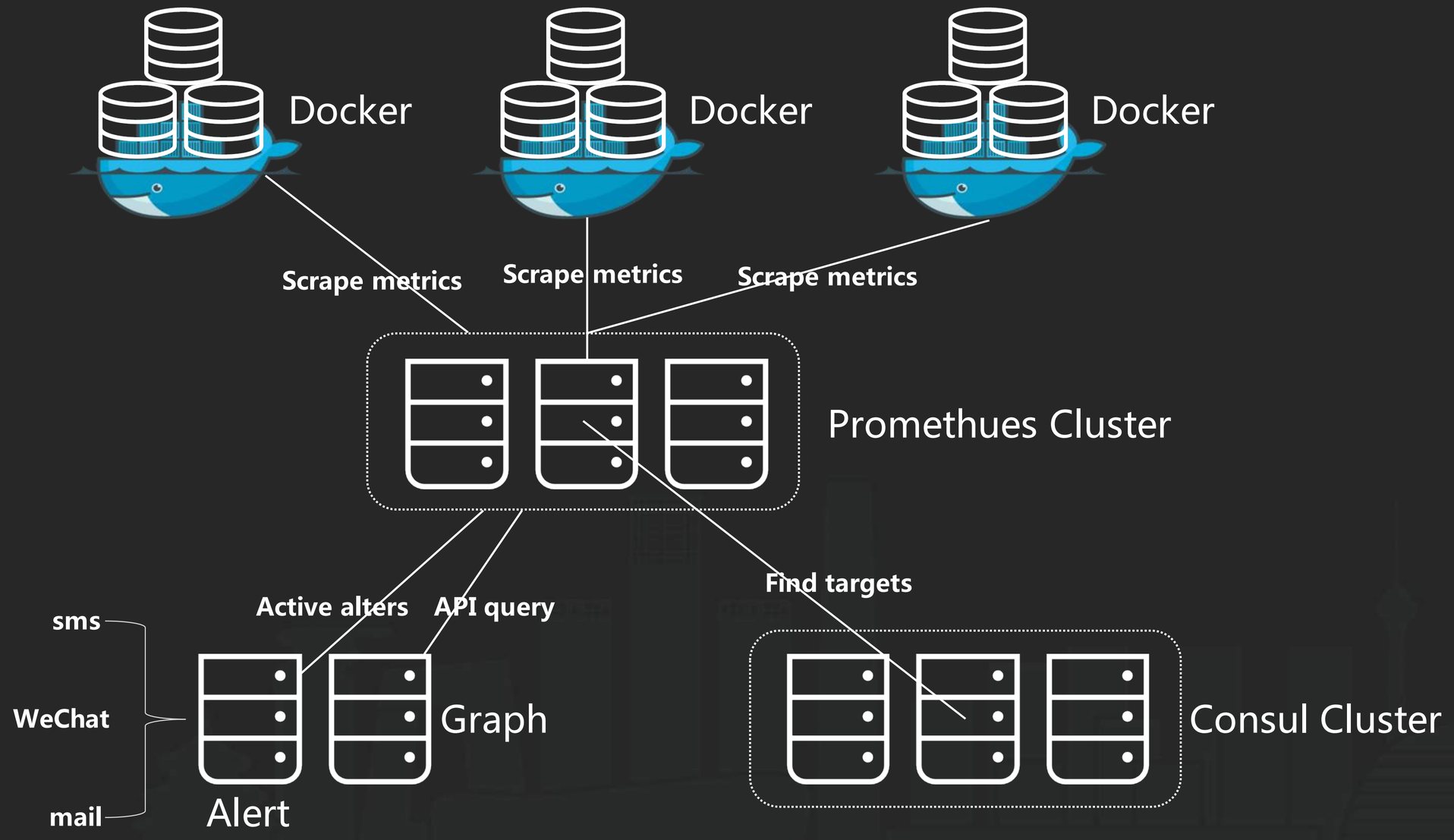
触发条件:

- 依赖监控告警系统,thread_running > 30(可自定义)

kill哪些语句:

- select语句
- 非特定系统账号(可自定义)
- 执行时间超过10S
- 只Kill sql请求，不kill连接

监控告警系统


Docker
Docker
Docker
Scrape metrics
Scrape metrics
Scrape metrics
Promethues Cluster
Active alters
API query
Find targets
sms
WeChat
mail
Graph
Alert
Consul Cluster

MySQL监控

多达上百项的详细的监控信息：

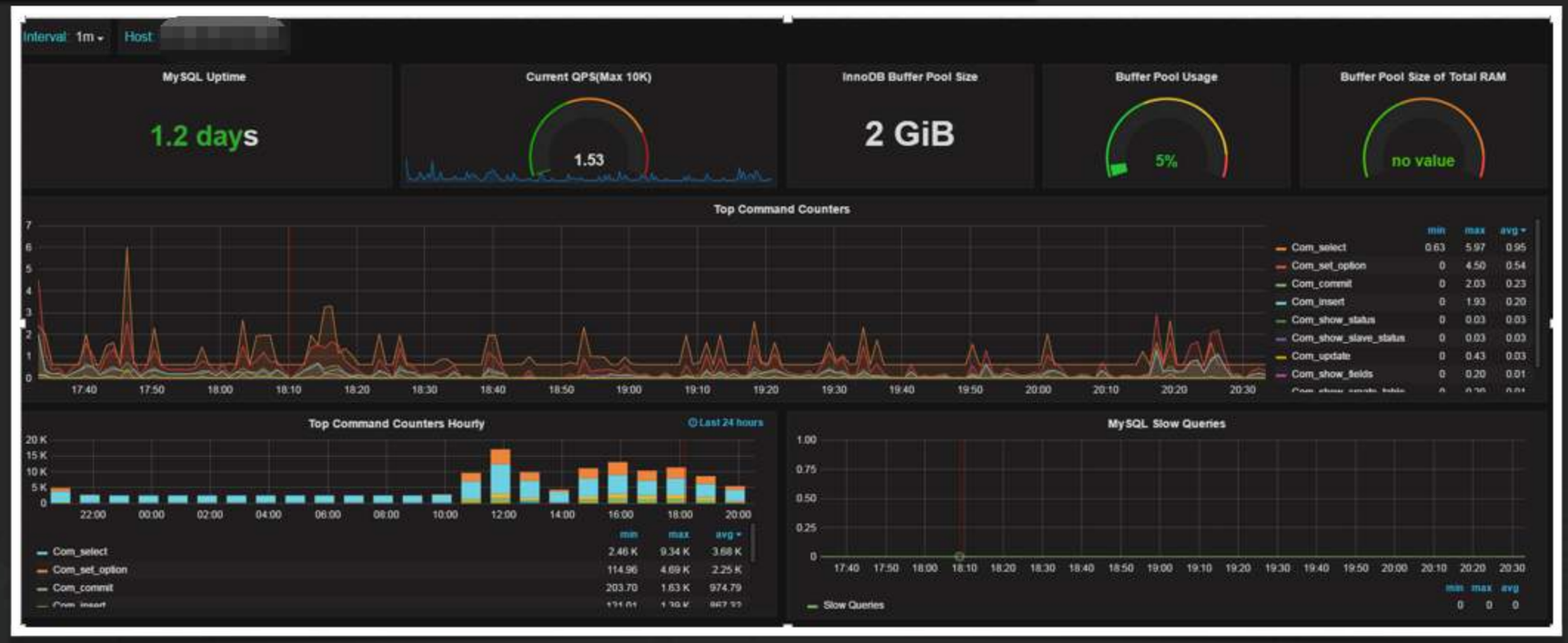
PRODUCT QA UAT TEST

集群名称： 生产线： 项目名称：

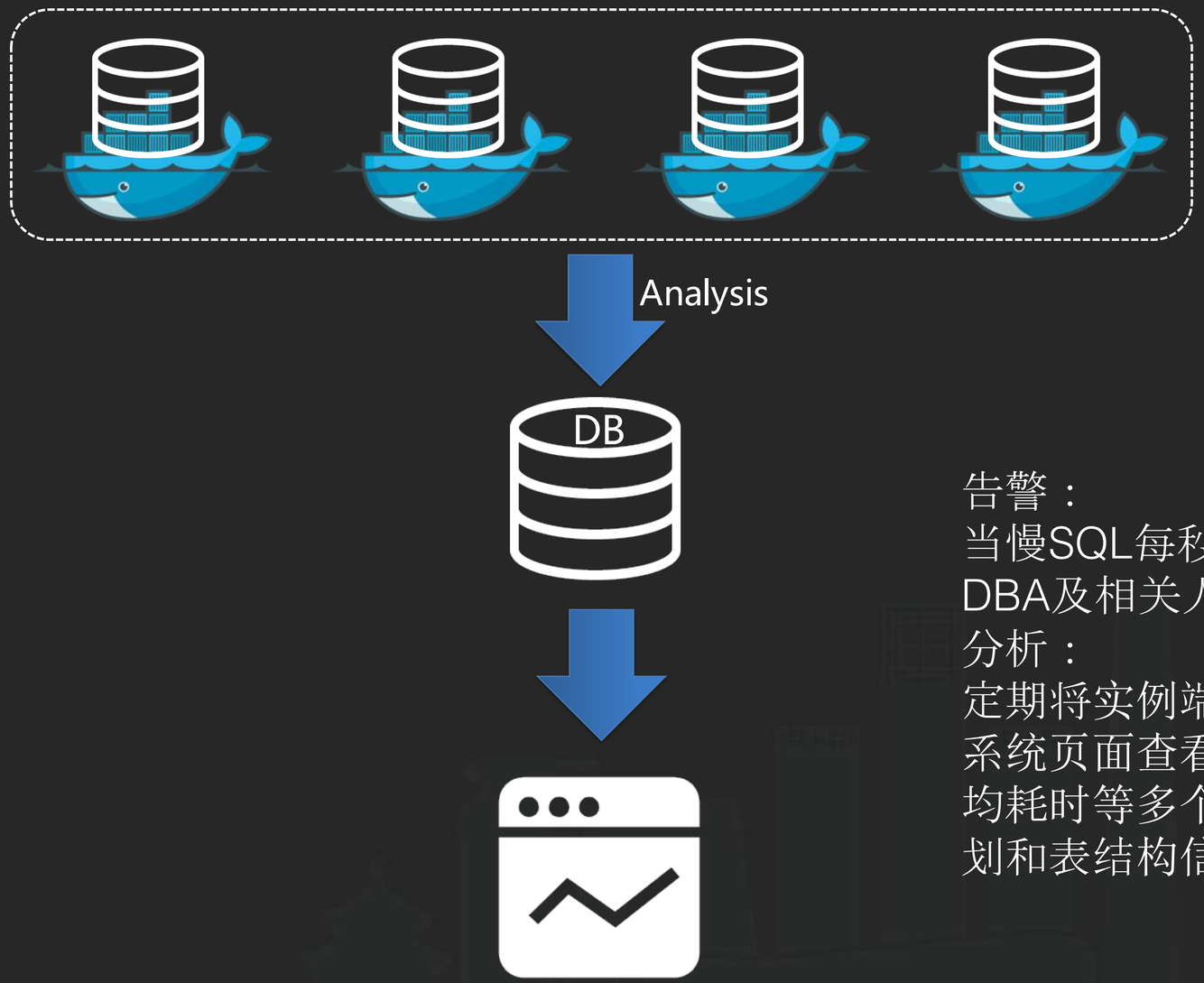
集群维护： SLAVE管理 HA管理 监控查看 信息修改

监控查看 → MySQL / 容器 / 接收人管理

Proxy	节点ID	高可用	VIP	主机	类型	角色	Data_Size	Log_Size	状态	创建人	管理
	1				1C1G	Master					详情 操作
					1C1G	Slave					详情 操作

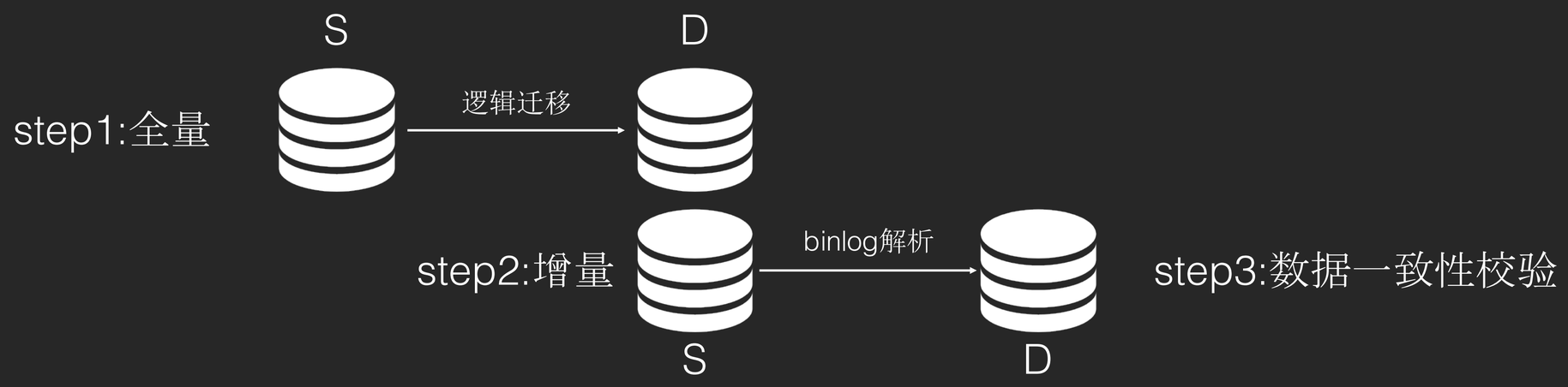


慢日志分析



告警：
当慢SQL每秒产生的数量超过阈值的时触发告警通知DBA及相关人员及时关注问题。
分析：
定期将实例端的慢SQL分析后录入数据库，然后通过系统页面查看慢SQL的执行时长、每日次数、每日平均耗时等多个维度的指标。同时也支持ui上查看执行计划和表结构信息。

数据库及实例迁移



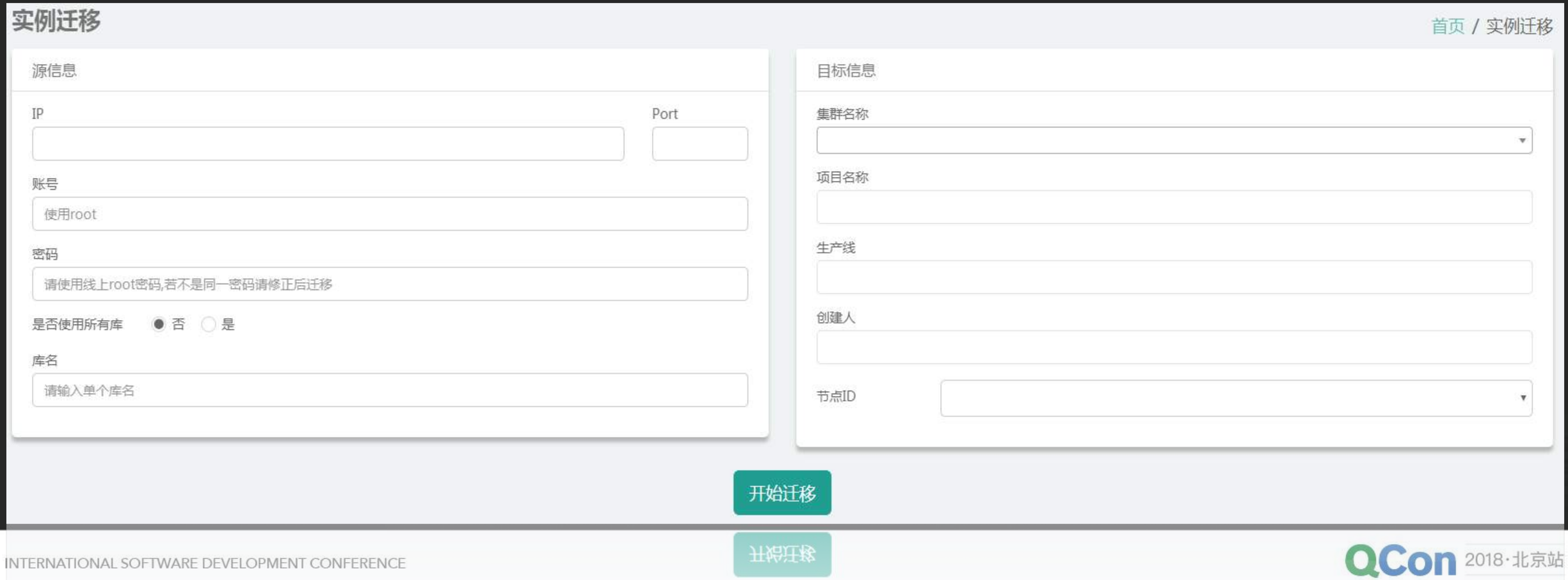
- 实例迁移我们支持了全实例、库、表级别的迁移,并且同时兼容各种MySQL版本。

全量迁移基于开源工具mydumper和myloader做了定制化开发，解决了一些问题，比如utf8mb4导致乱码的问题等。

增量复制使用的是公司自研的基于binlog复制的产品。该工具支持指定库、表级别复制，另外也支持源与目标命名不一致(结构一致)的复制。

实例迁移

实例迁移：	实例、库、表迁移
	平台外的实例迁移至平台内
	平台内数据迁移



成果

- 屏蔽底层物理资源，降低决策时间
- 资源利用率提升30倍
- 90%的基础运维工作完全自动化
- 交付能力提升70倍
- 生产环境近3500个实例在平台上工作

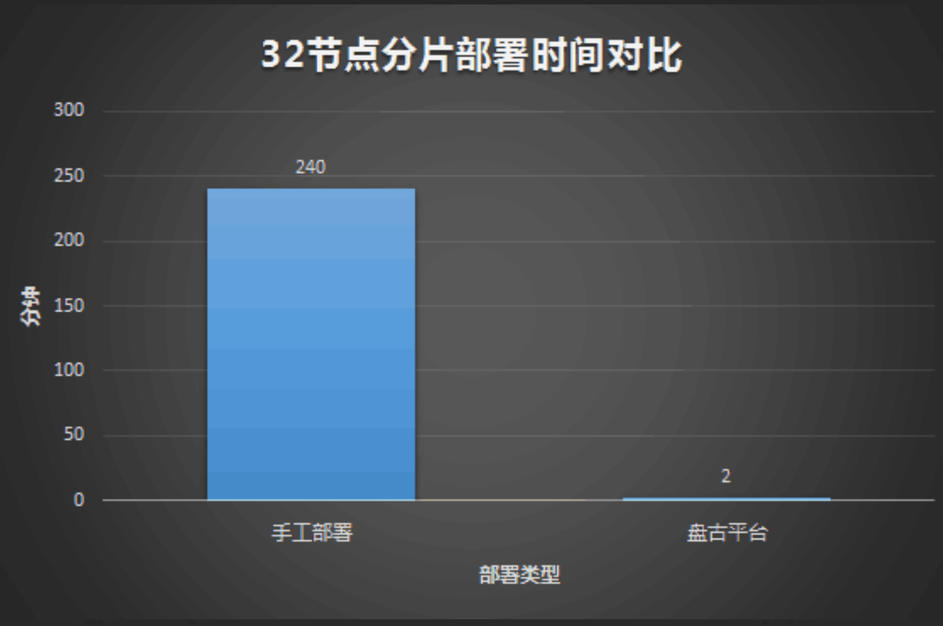
效率提升

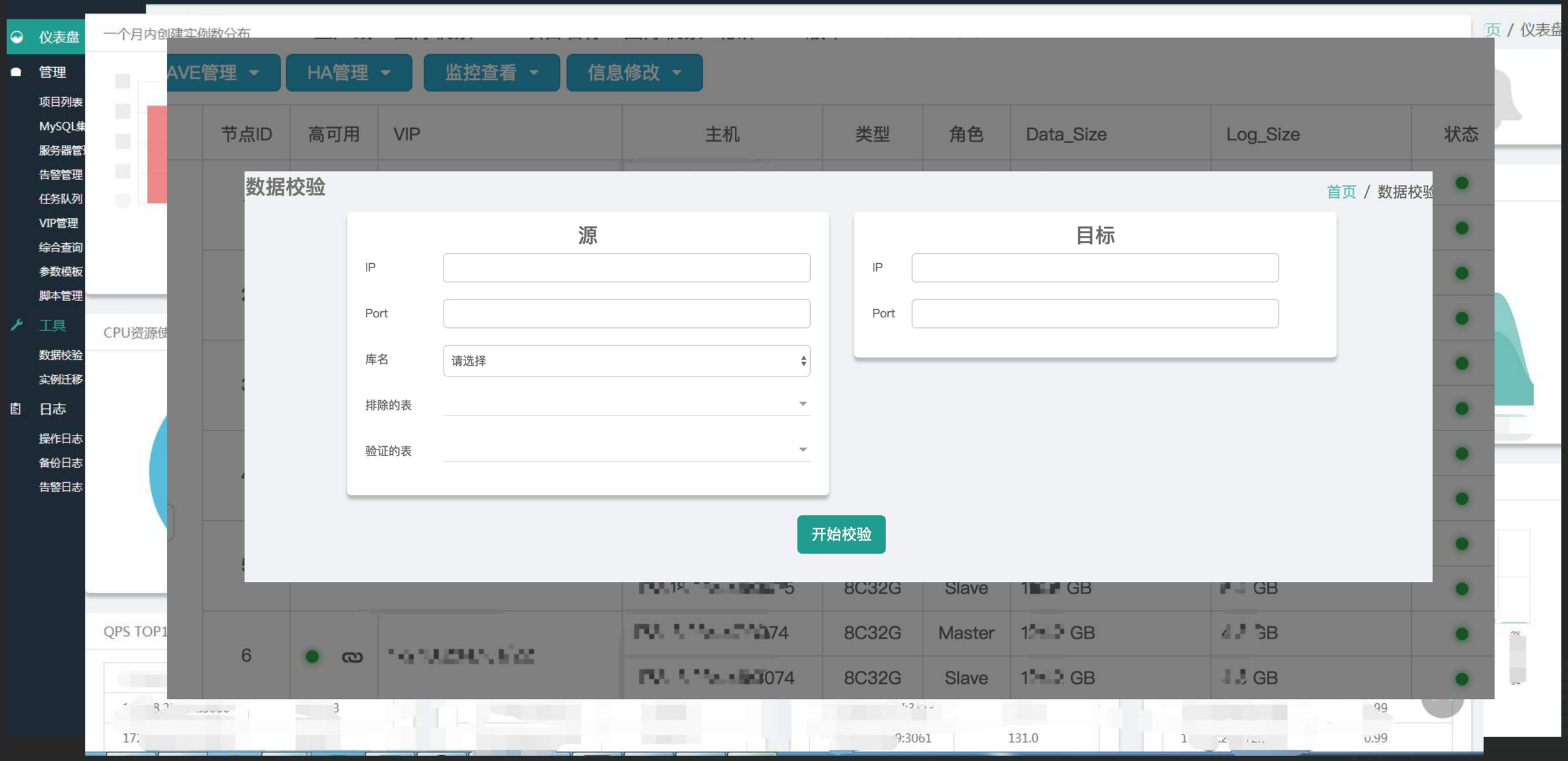
手工部署：

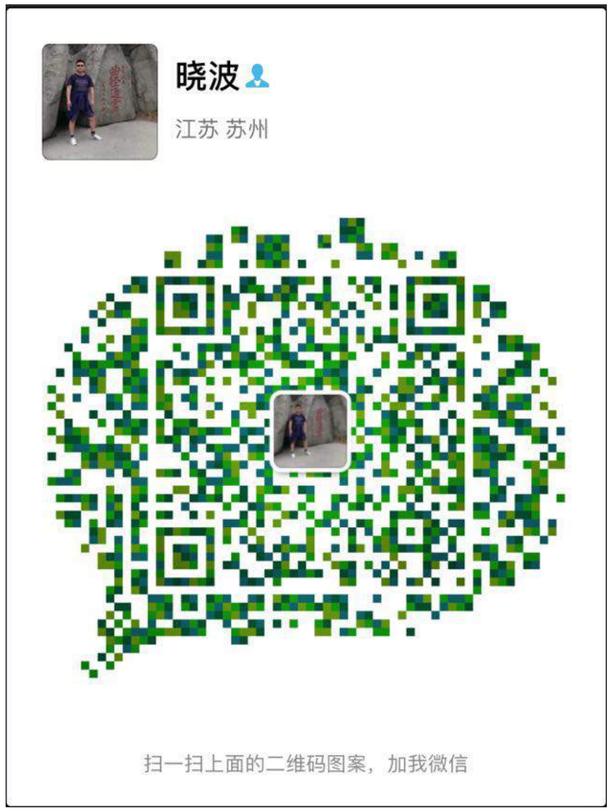
部署一套高可用集群+备份，配置监控。至少30分钟。部署32个节点的分片的集群，至少一个上午。无系统化管理，资源分配情况，无法统一调配，服务器资源利用率低。

MySQL容器平台：

部署一套高可用集群+自动化备份+慢日志分析+监控。用时1-2分钟。部署32个节点的分片集群，只需5分钟。标准化的系统管理，部署环境统一、配置文件统一。系统化的操作降低人为失误和重复劳动。资源使用集中管理，有效利用服务器资源。




仪表盘
管理
项目列表
MySQL集
服务器管
告警管理
任务队列
VIP管理
综合查询
参数模板
脚本管理
工具
数据校验
实例迁移
日志
操作日志
备份日志
告警日志
一个月内创建实例数分布
CPU资源使
QPS TOP1
AVE管理
HA管理
监控查看
信息修改
节点ID
高可用
VIP
主机
类型
角色
Data_Size
Log_Size
状态
8C32G
Slave
8C32G
Master
8C32G
Slave
6
数据校验
首页 / 数据校验
源
IP
Port
库名
请选择
排除的表
验证的表
目标
IP
Port
开始校验
首页 / 仪表盘



Thanks!

INTERNATIONAL SOFTWARE DEVELOPMENT CONFERENCE

主办方 Geekbang. 极客邦科技 InfoQ

同程艺龙