



GOPS2018  
Shenzhen

# GOPS

全球运维大会 2018

2018.4.13-4.14

中国·广东·深圳·南山区 圣淘沙大酒店（翡翠店）





GOPS2018  
Shenzhen

# 负重前行--顺丰数据库运维的求变之路

刘力 顺丰科技



GOPS2018  
Shenzhen

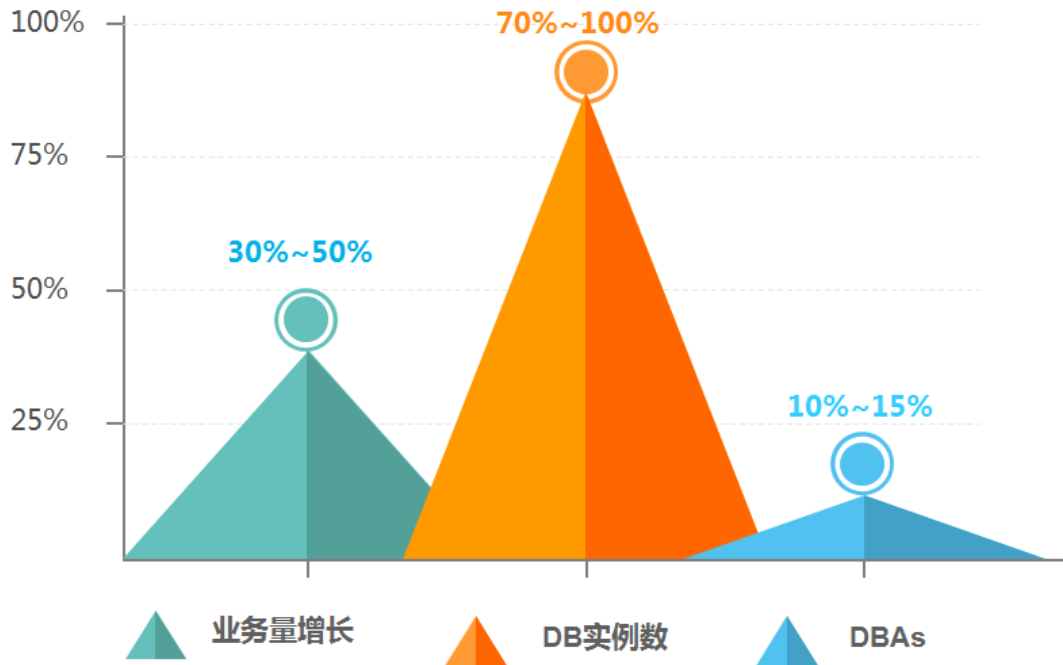
# 负重前行 高速列车换轮子

## 1. 业务增长迅猛

- 业务量增长快速
- 运维对象逐年翻倍

## 2. 技术变革

- 主动：非标到标准
- 被动：传统到开源
- 求变：运维开发&智能运维





GOPS2018  
Shenzhen

# 目录



**1** 非标到标准

**2** 传统到开源

**3** 集中到分布式

**4** 智能运维

# 非标到标准 洪荒年代



GOPS2018  
Shenzhen

## 1. 数据库类型繁多

- 商用开源混杂
- 高可用方案五花八门

## 2. DBAs被技术割裂

- 专业能力：广而不精，技术无法形成聚力
- 个性化运维：压力不均，异常处理效率低

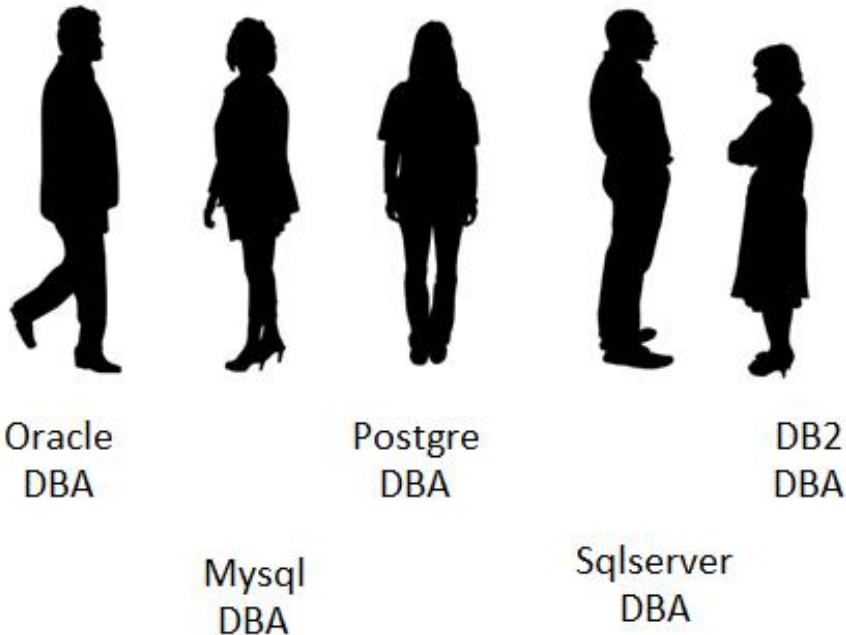
## 3. 忽略用户需求

- 运维：提供稳定数据库服务难吗？

(DBA：主机/存储/网络的问题，为什么都算在我头上)

- 研发：总出问题了，才告诉我代码质量有问题

(DBA：几个月前就发邮件了，没时间天天追你)





GOPS2018  
Shenzhen

# 非标到标准 穷则思变

## 1. 认识到问题

- 基础架构的稳定性和性能瓶颈对DB影响，是DBA需要承担的
- 应用架构以及SQL质量对DB影响，也是DBA需要承担
- DBAs需要更多时间服务用户

## 2. 标准化改造

- 领导支持
- 尽量降低上层应用感知





GOPS2018  
Shenzhen

# 目录

1 非标到标准

➔ 2 传统到开源

3 集中到分布式

4 智能运维



GOPS2018  
Shenzhen

# 传统到开源 去商化

## 1. 自上而下，技术转型

- 开源风暴来临，鸵鸟还是水手？
- 突破舒适区&空杯心态

## 2. 步步为营

- 简单用：关系型数据的容器
- 控制规模：垂直拆分
- 数据一致性：先用SAN解决
- 建立运维研发能力，自研HA方案



OR







GOPS2018  
Shenzhen

# 传统到开源 运维研发

## 1. 打开窗，就是新世界

- 正视开源风暴带来的影响
- 从被动执行，到主动求变
- 思维一旦开打，技术视野更广阔

## 2. 研发能力建设

- 靠自己：降低业厂商的依赖
- 团队建立：外招+内部引导+大学生
- 正规化：需求牵引，架构设计，研发过程管控，灰度上线，生产发布

### 团队建立

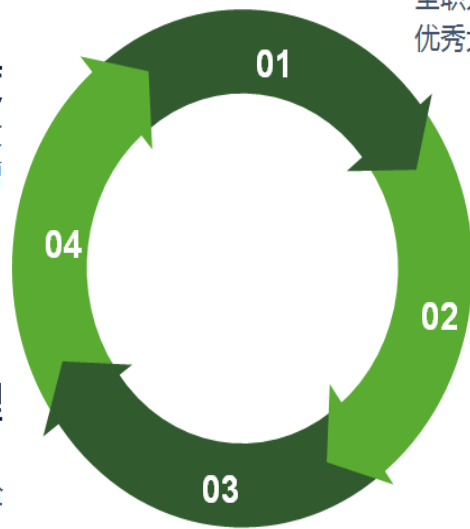
全职开发，外部引入，  
优秀大学生

**增加运维参与度**  
需求分析，原子操作开发  
用户验收和吐槽

**研发过程管理**  
拒绝小作坊，  
麻雀虽小五脏俱全

### 需求驱动

以解决一线运维，  
实际问题为抓手





GOPS2018  
Shenzhen

# 目录

1 非标到标准

2 传统到开源

➔ 3 集中到分布式

4 智能运维



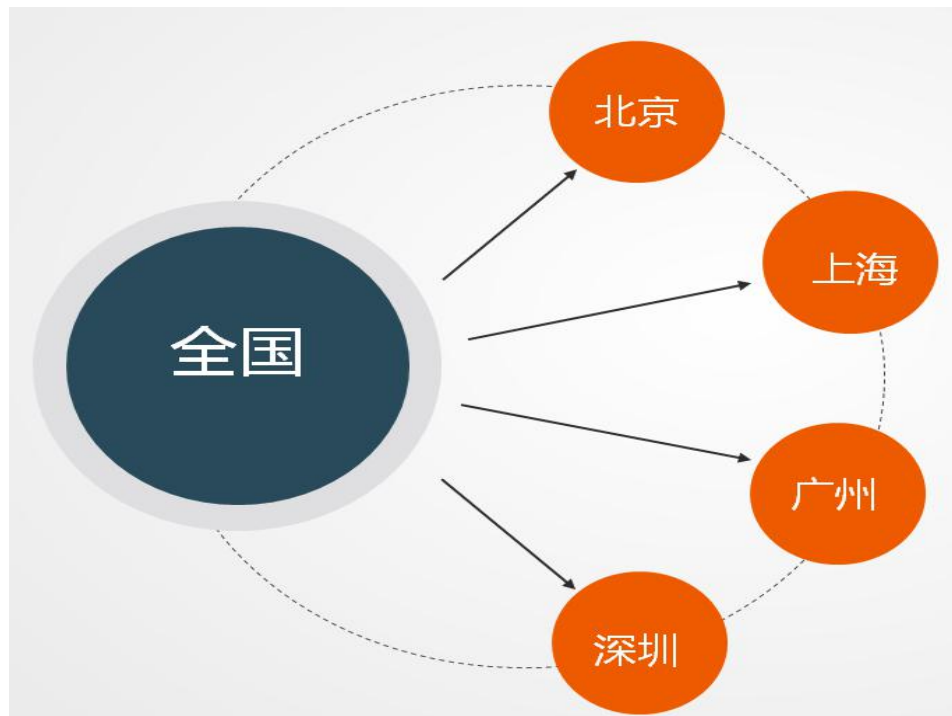
# 集中到分布式 单实例性能瓶颈

## 1. 单实例性能瓶颈

- 单Mysql TPS上限1w+
- 紧耦合逻辑，垂直拆分难度大
- 从应用架构上，按区域做水平拆分

## 2. 用户吐槽

- 研发：应用架构要改造，关联数据要处理：  
不能给我一个强大的数据库吗？
- 运维：到底需要部署多少套应用？





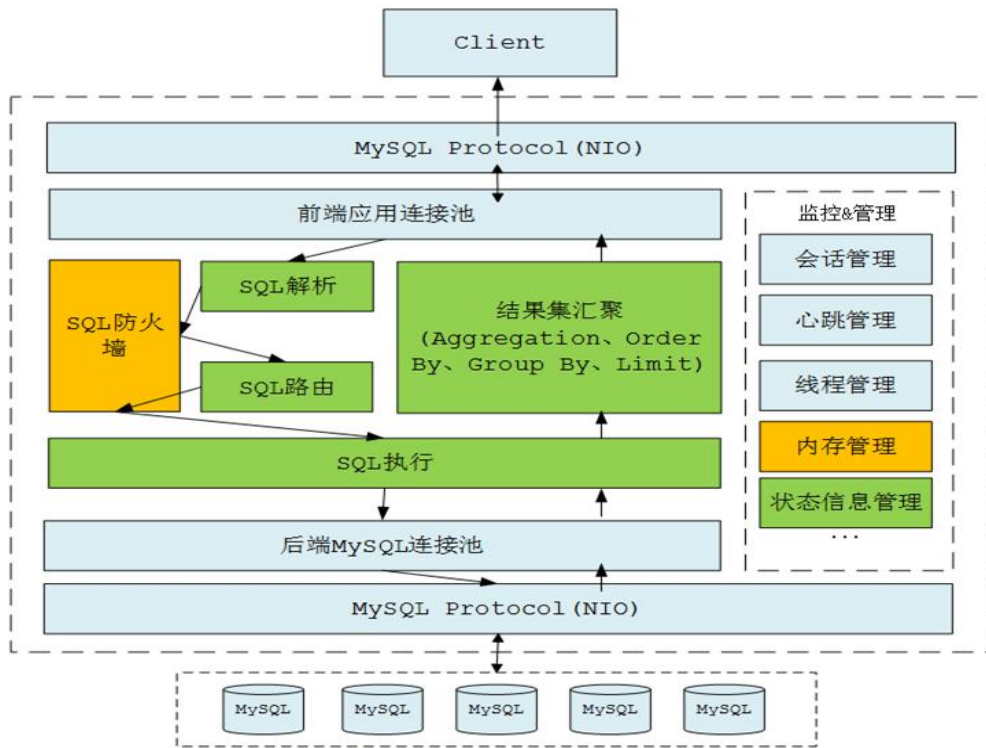
# 集中到分布式 Mysql\_proxy

## 1. 基于开源软件二次开发

- 核心系统TPS吞吐20w+
- SQL防火墙
- 大结果汇聚

## 2. 取之开源，用之开源

- 贡献代码量30%
- 修复关键bug 10+





GOPS2018  
Shenzhen

# 集中到分布式 大数据汇聚

## 1. 大数据汇聚策略

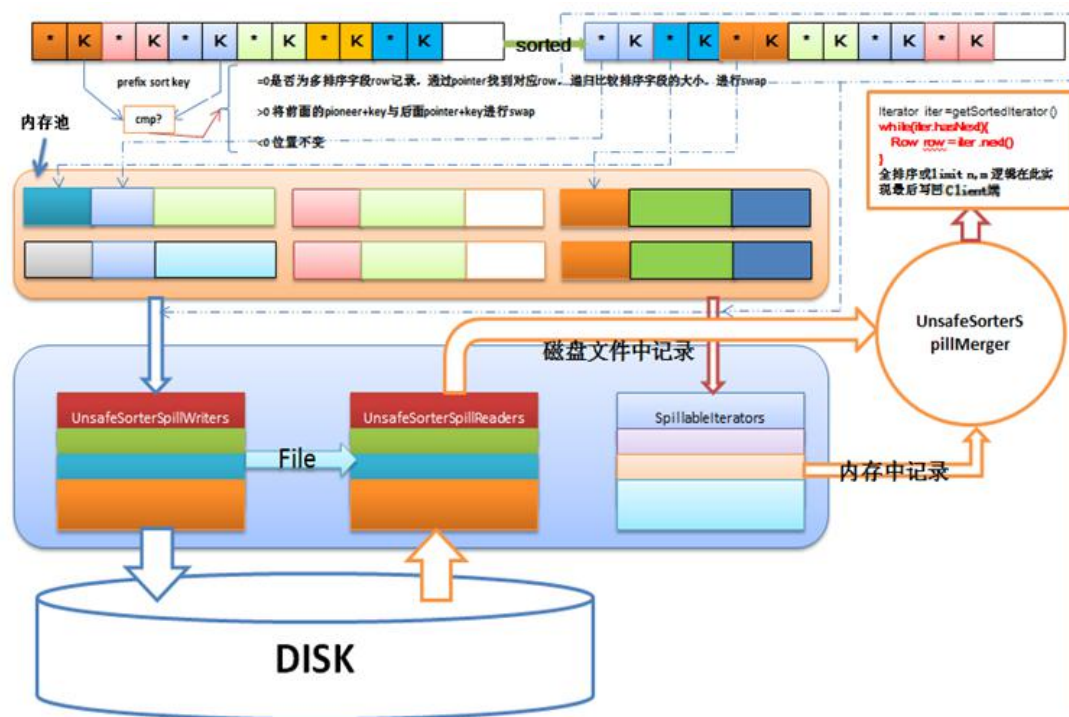
- Off-heap&DISK排序

## 2. 效果

- 实现亿级规模结果集操作  
Order/Merge/Group, 避免OOM
- 减少Full GC的时间, 提高应用程序性能

## 3. 应用

- OLTP应用中, 主要是避免OOM
- 用于大数据数据抽取





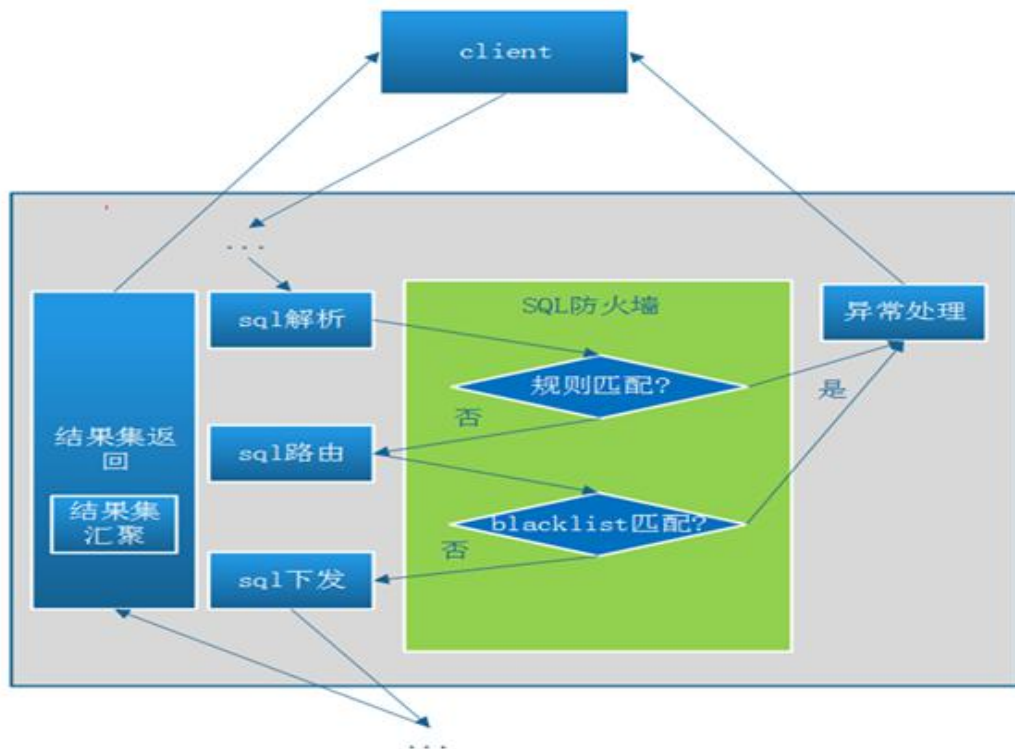
# 集中到分布式 SQL防火墙

## 1. 思路

- 规范和要求要落地在平台上，文档和培训永远跟不上研发人员的代谢速度
- 降低DBA和研发之间的交互

## 2. 运用

- 开发环境启动功能，直接拒绝异常SQL；在开发阶段，过滤明显不合规的SQL
- 生产环境关闭功能，做信息收集，以便性能分析；应急情况下，启动开关





GOPS2018  
Shenzhen

# 目录

1 非标到标准

2 传统到开源

3 集中到分布式

➔ 4 智能运维

# 智能运维



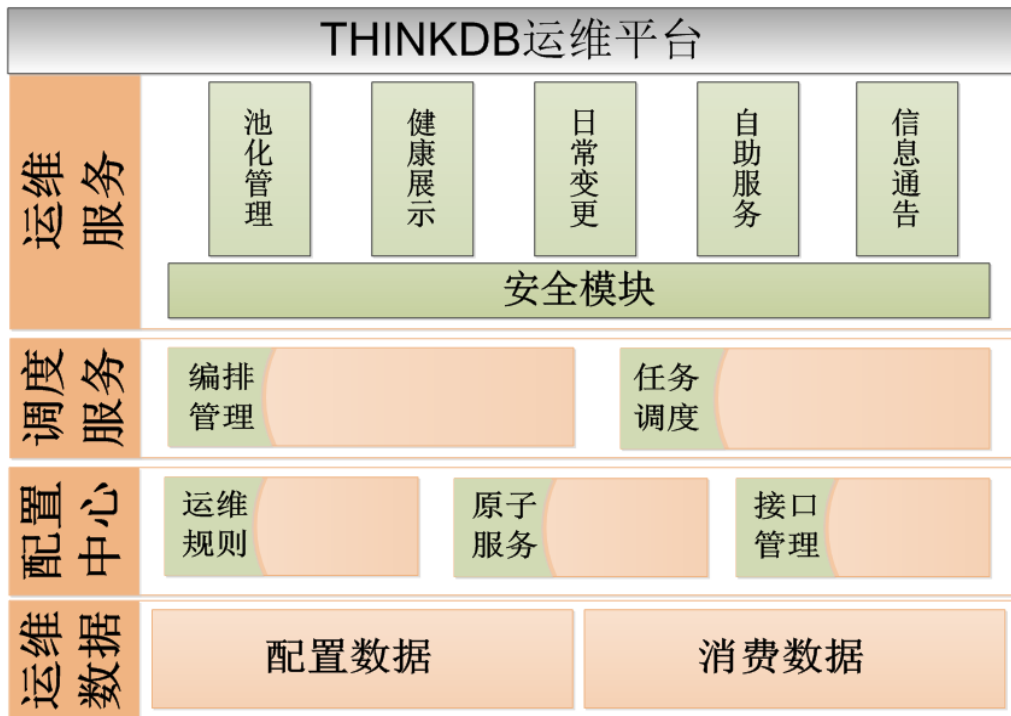
GOPS2018  
Shenzhen

## 1. 矛盾转移

- 研发&运维的需求基本满足
- 面对膨胀的DB数量，运维复杂度大幅提升

## 2. 解放DBA双手

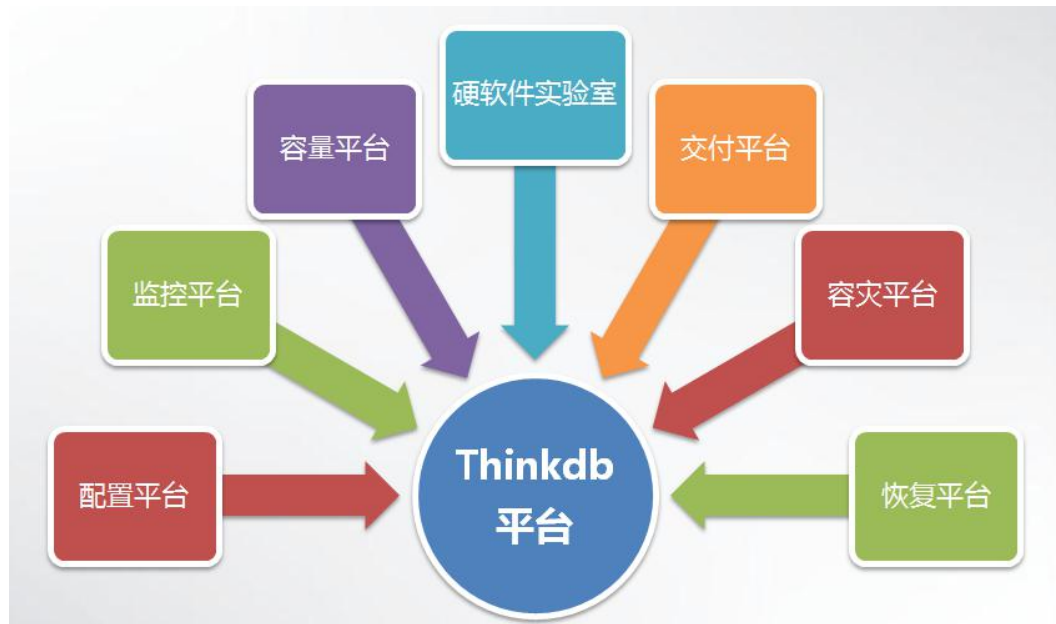
- ITIL：服务自助；标准报警/事件自愈；变更自动化；信息透明化
- 资源池动态管理：HA逻辑化，资源动态平衡，硬件资源热增减
- 怎么做？





## 运维平台系统群

- 配置平台：以应用和组件2个维度自动获取
- 监控平台：组件状态实时数据采集
- 容量平台：日常容量管理，高峰容量预测
- 硬软件实验室：输出硬软件最佳实践
- 交付平台：面向用户，自助交付
- 容灾平台：应用视角以及组件批量处理
- 恢复平台：指定时间点自动恢复

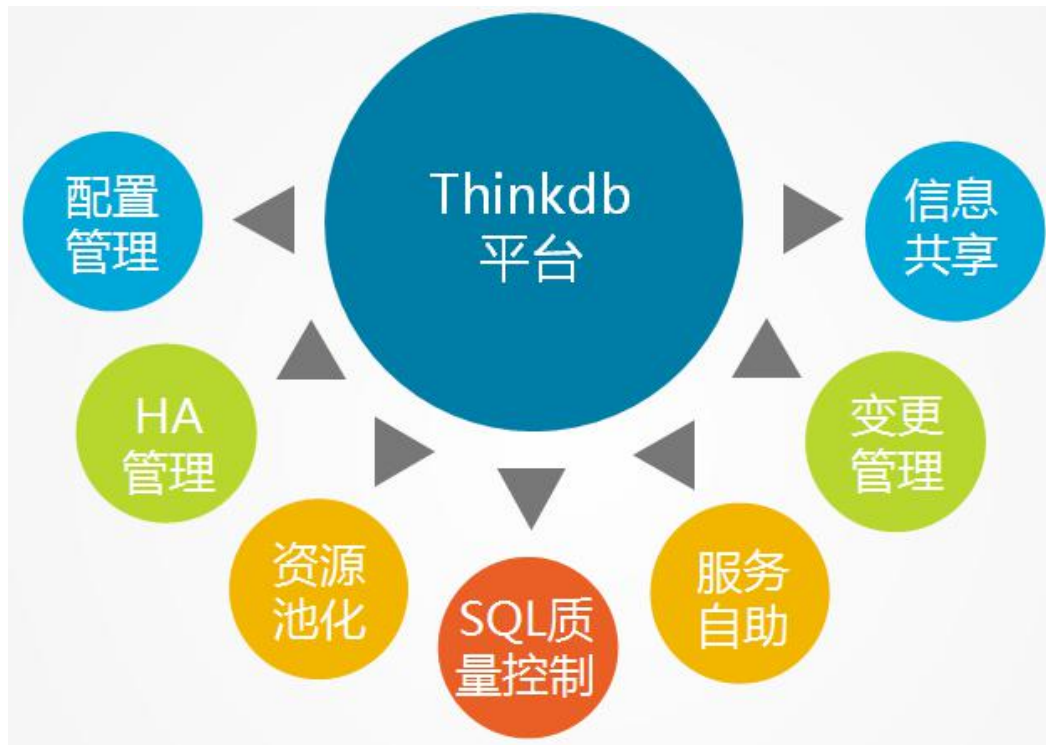


## 1. 配置准确性100%

- 将物理操作转化平台上逻辑操作
- 1%的配置数据和实际状况的差异，都可能导致故障

## 2. 如何保障准确性

- 从外部接入配置平台的数据
- 从平台内置独立的DB信息收集模块
- 2方面数据互相校验，要求耦合度100%
- 所有命令下发前，实时校验实例状态

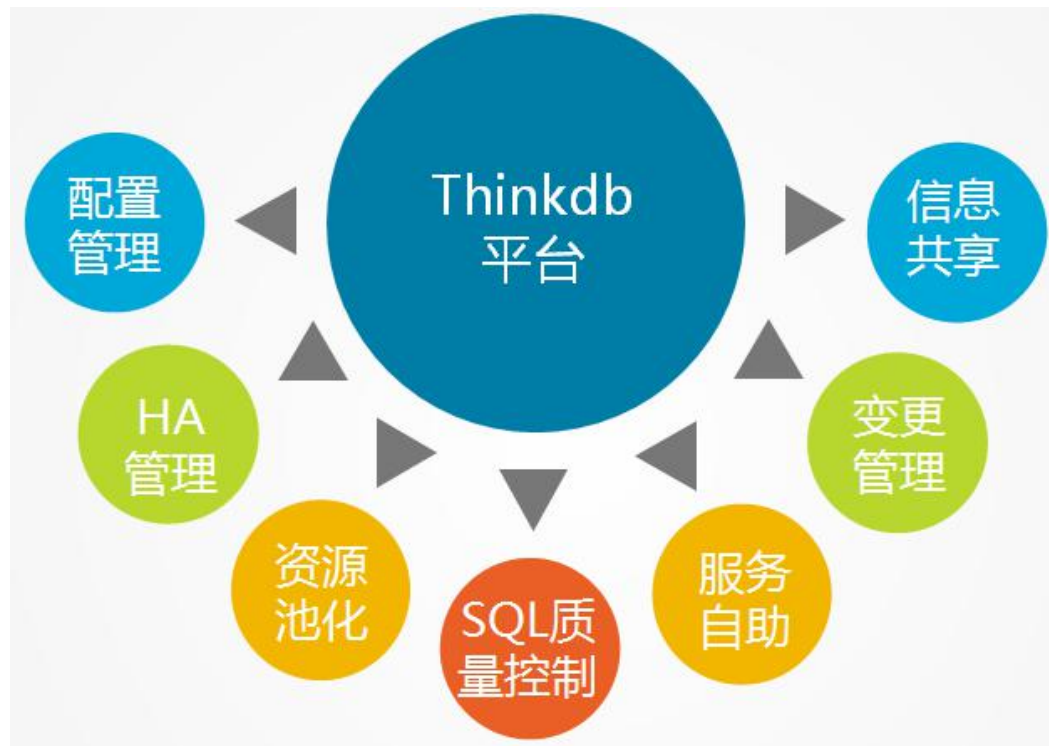


## 1. HA自动管理

- 数据一致性要求高:MGR+ OS/DB双心跳确认+服务恢复
- 数据一致性要求中:半同步+OS/DB双心跳确认+服务恢复

## 2. HA自动修复

- 尝试建立异常节点和新主动的同步关系
- 自动重建异常节点，并恢复同步
- 容灾节点自动和新节点建立同步关系

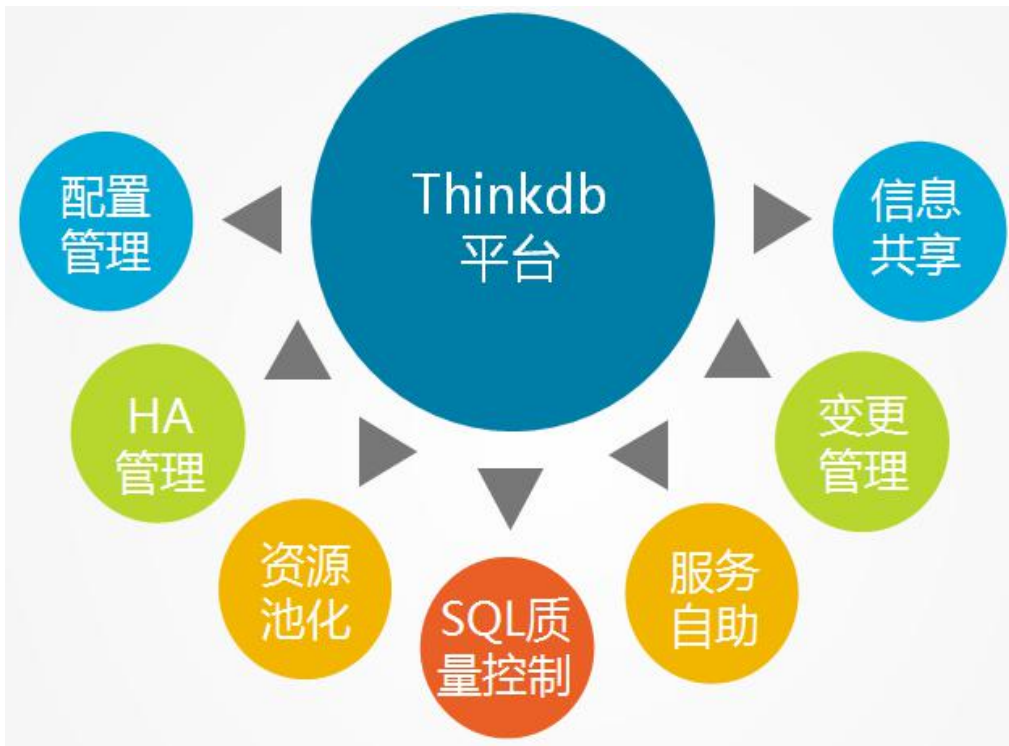


## 1. 资源池化管理

- 对接监控平台，获取资源状态数据
- 资源管理逻辑：Pctfree/Pctused资源使用率上下限
- 动态资源调整：定期自动触发物理节点间实例再分配

## 2. 物理资源自动扩容

- 对接容量平台，获取容量预估数据
- 物理主机管理：库存管理，热增减



## 1. SQL代码质量控制

- 开发环境：SQL防火墙

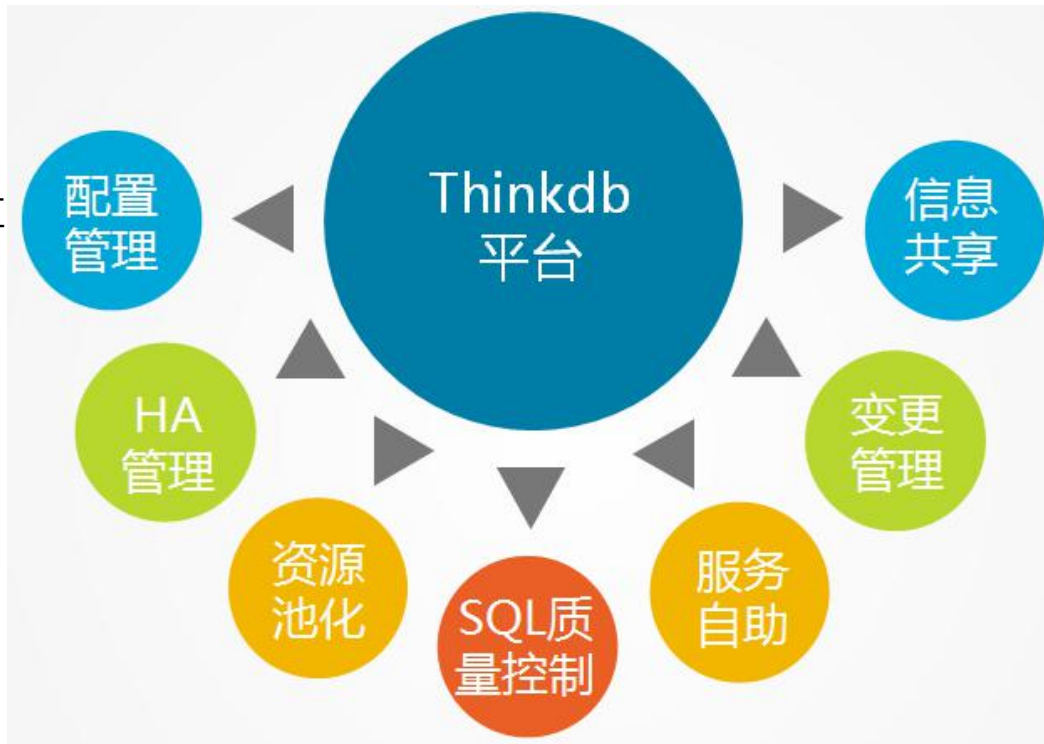
在Mysql proxy中内嵌SQL防火墙，在开发环境直接拒绝不合理的SQL

- 测试环境：SQL代码审核

版本发布流程，加入SQL代码审核  
不断优化SQL规则引擎

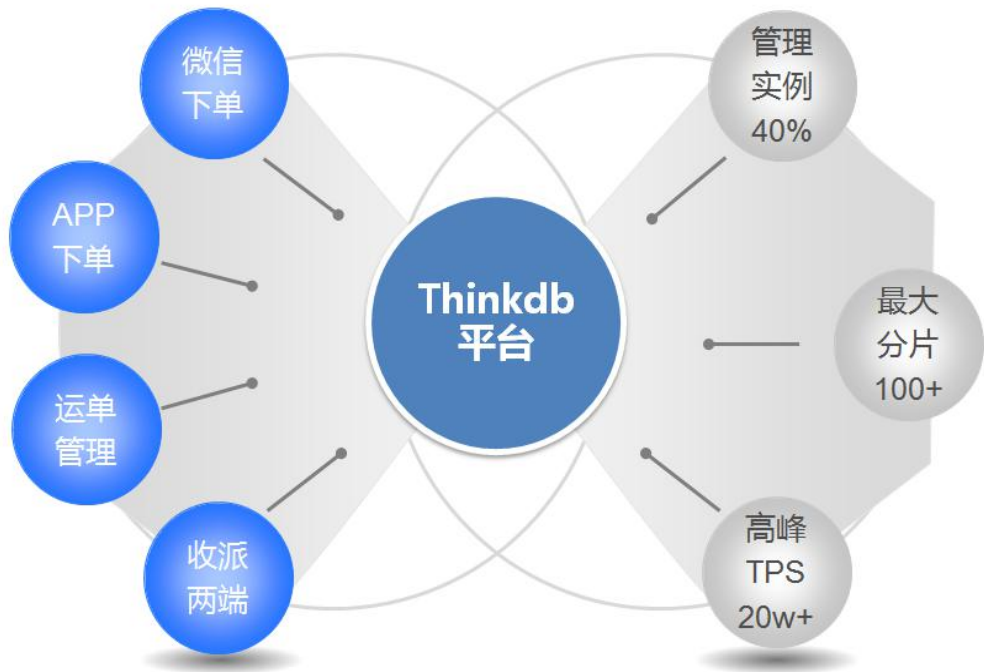
- 生产环境：SQL质量评分排名

定期分析慢SQL，基于应用系统做汇总排名



## 1. 生产实践

- 核心应用：APP/微信下单，运单系统，收派系统
- 新实例上线，自动接入Thinkdb平台
- 老实例逐步加入Thinkdb管理，覆盖率40%
- 运单系统：分片100+，高峰TPS 20w+
- 双11高峰：风平浪静



# 智能运维 放飞大脑



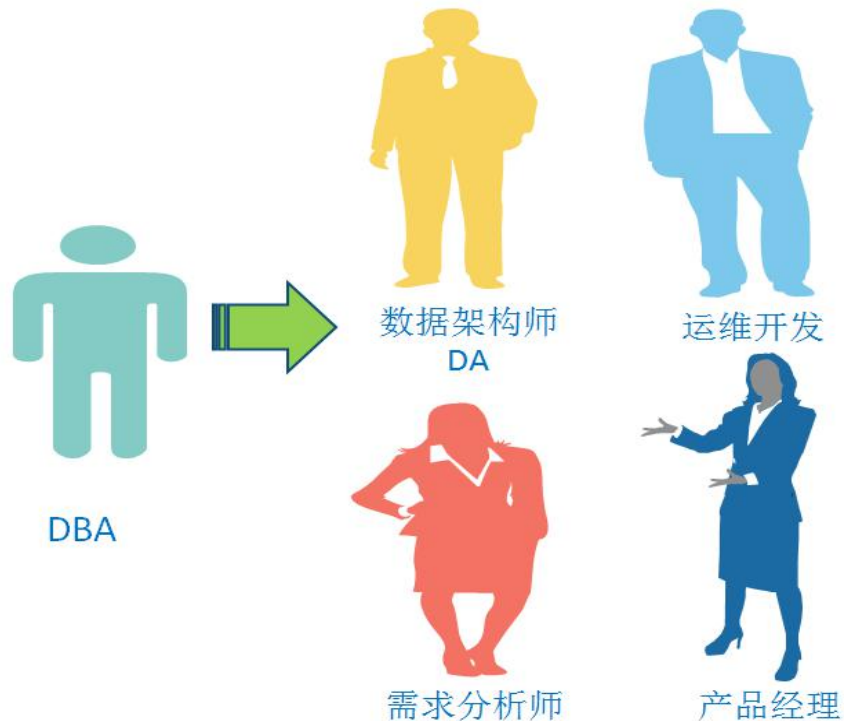
GOPS2018  
Shenzhen

## 1. 不仅仅是DBA

- 沟通协调&逻辑思维是DBA利器
- 开放心态，敢于尝试

## 2. 岗位延伸

- 向上：数据架构师；需求分析师；产品经理
- 向下：运维开发；开源代码开发



# 负重前行 高速列车换轮子



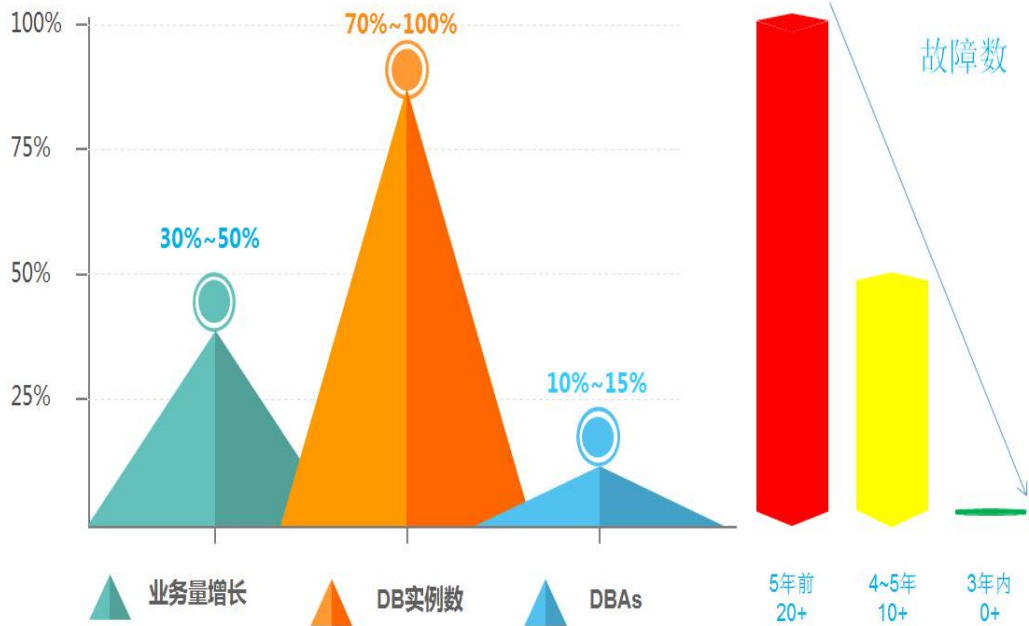
GOPS2018  
Shenzhen

## 1. 勇于承担&问题即机遇

- 把问题看成自己的，你的边界就在不断扩展

## 2. 如果你有一把锤子

- 架构设计对DBA是手到擒来
- 加上运维研发能力
- 所有已知的问题都是钉子



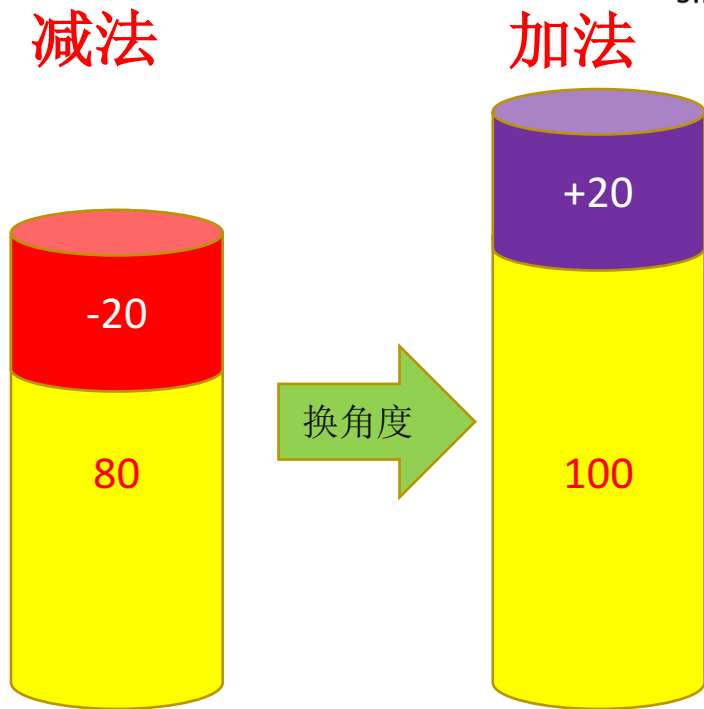


## 1. 运维原罪

- 运维是做减法的团队
- 做到0故障也只能是100
- 好的研发团队，能做到120+

## 2. 换一个角度

- 做好了基础运维的基础上
  - 任何的创新都是加分项
- ( 做不好基础运维，任何创新失去支撑 )





GOPS2018  
Shenzhen



# Thanks

高效运维社区  
开放运维联盟

荣誉出品



GOPS2018  
Shenzhen

想第一时间看到高效运维社区  
的新动态吗？

