

DTCC

数 / 造 / 未 / 来

第十二届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2021



2021 年 10 月 18 日 - 20 日 | 北京国际会议中心

分享者: 陈阳



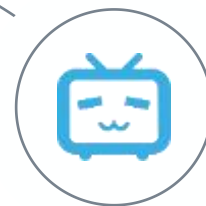


B 站业务场景

点播类业务
对可用性要求比较高



直播类业务
高并发场景、热点数据



电商类业务
数据一致性、热点数据



游戏类业务
对性能要求比较高



业务场景





面临的挑战

- 急速业务增长的背后带来大量的变更需求
- 数据量急速膨胀, 频繁资源扩容、拆分、迁移
- 告警治理, 关键时刻最重要的那条被淹没
- 异常场景下管控能力不足
- 数据安全隐患



服务治理的思考

治理什么



如何治理

3

安全

- 存储：数据分级、存储加密、备份
- 网络：网络隔离、安全策略
- 认证：账号密码存储、权限隔离
- 审计：日志、行为审计能力
- 异常：异常流量拦截能力

2

质量

- 流程标准：专业化、标准化、精细化
- 质量管理：可量化、运营化
- 容量管理：服务分级、可预测性
- 监报告警：分级、收敛、自愈
- 容灾预案：预案、灾演
- 问题总结：事后复盘、Todo追踪

1

效率

- 人效：自动化、自助化、智能化
- 交付：工单化、平台化
- 响应：Oncall、问题上升



数，造，未，来



按照执行频次、耗时、综合收益排序去逐步实现自动化治理



提高效率

方向: 平台化、自助化、智能化

挑战 1: 把人的经验转化为平台规则

挑战 2: DBA 角色转化

DBA --> SRE? DBRE? DBIE? PE?

As a DBRE you will:

- Work on database reliability and performance aspects for GitLab.com from within the SRE team as well as work on shipping solutions with the product.
- Analyze solutions and implement best practices for our main PostgreSQL database cluster and its components.
- Work on observability of relevant database metrics and make sure we reach our database objectives.
- Work with peer SREs to roll out changes to our production environment and help mitigate database-related production incidents.
- OnCall support on rotation with the team.
- Provide database expertise to engineering teams (for example through reviews of database migrations, queries and performance optimizations).
- Work on automation of database infrastructure and help engineering succeed by providing self-service tools.
- Use the GitLab product to run GitLab.com as a first resort and improve the product as much as possible.
- Plan the growth and manage the capacity of GitLab's database infrastructure.
- Design, build and maintain core database infrastructure pieces that allow GitLab to scale to assist hundreds of thousands of concurrent users.
- Support and debug database production issues across services and levels of the stack.
- Make monitoring and alerting alert on symptoms and SLOs, and not on outages.
- Document every action so your learnings turn into repeatable actions and then into automation.



自动化运维演进方向

人肉运维

DBA 人工完成数据库部署、监控配置、SQL 审核、优化

半自动化运维

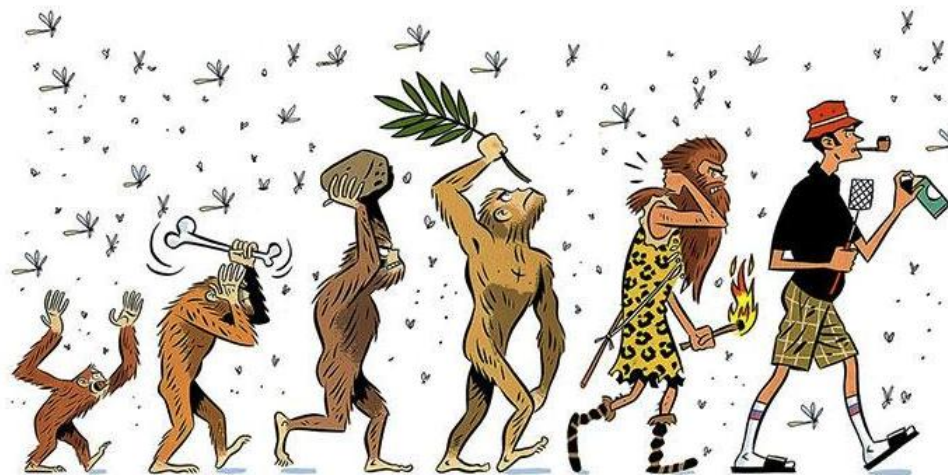
常见运维操作脚本化、平台化，比如自动化部署、审核上线、故障切换

全自动化运维

运维平台具备高级管控能力的自动化，比如自动资源分配、优化分析、异常诊断辅助、流量管控等能力

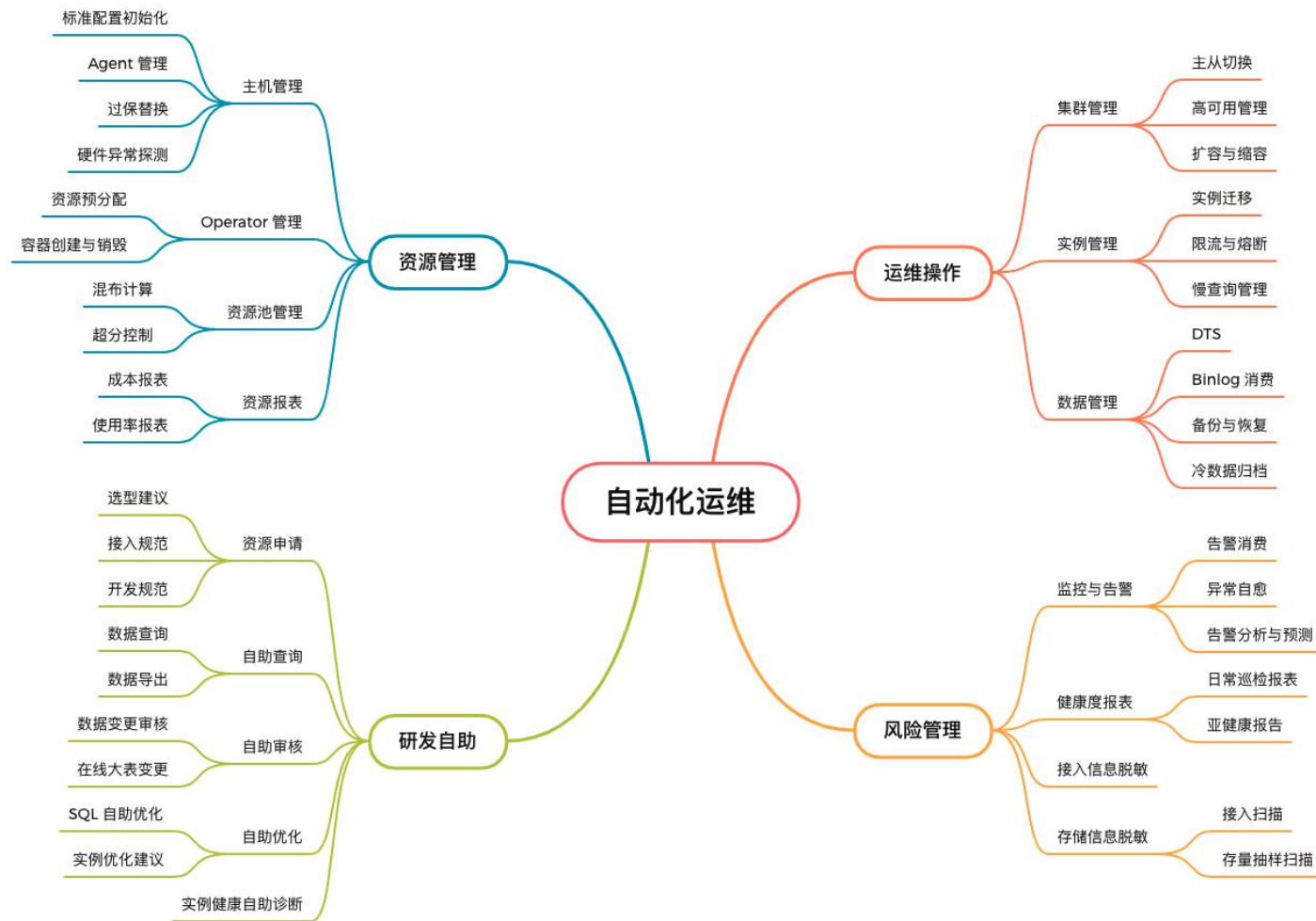
智能化运维

服务完全由数据库运维系统接管，几乎不需要人工干预，比如自动决策、高度自愈、自动扩缩容，并结合 AIOps 有一定的前瞻预测性





自动化运维平台



高速发展下, 稳定性治理是重中之重





稳定性治理

- **高可用:** 同机房、跨机房、投票决策
- **资源隔离:** 按照服务等级水位线计算、容器化
- **慢查询:** 可视化、指纹分析、趋势分析、快速止损
- **中间件:** 异常服务感知, 透明伸缩、拦截、限流与熔断
- **自愈:** 简单异常自愈
- **拆分与聚合:** 千亿级别数据量如何选型
- **应急响应:** SOP、OnCall、限流、熔断、降级、演练
- **大型活动:** 备战、全链路压测、应急预案



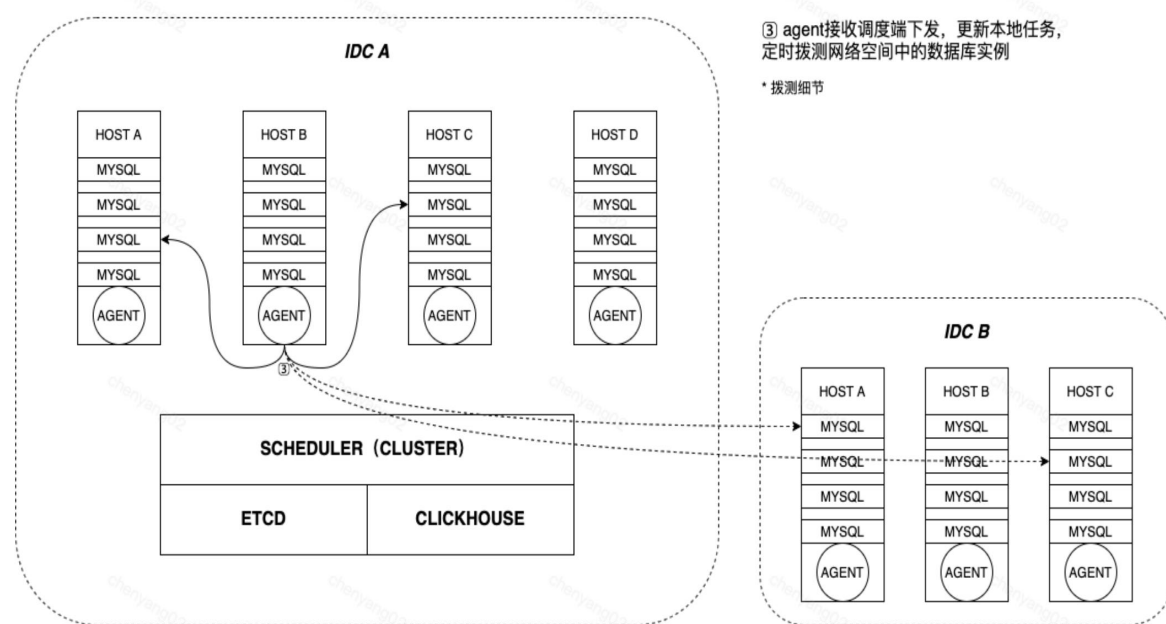
The diagram illustrates the architecture of IDC A, which is enclosed in a dashed rounded rectangle. Inside, there are four host groups labeled HOST A, HOST B, HOST C, and HOST D. Each host group contains a vertical stack of components: four MySQL databases and one AGENT at the bottom. The AGENT of each host group is connected to a central SCHEDULER (CLUSTER) block via a line labeled ②. The SCHEDULER (CLUSTER) block is divided into two sections: ETCD and CLICKHOUSE. A self-loop arrow labeled ① is shown on the SCHEDULER (CLUSTER) block. An arrow labeled ① points from the SCHEDULER (CLUSTER) block to a separate block labeled CENTRAL IDC, which contains AKSO MNG and META DATA.

① 基于etod 调度端选主，对外提供服务

① 调度端定时拉取中心机房数据库实例及主机信息

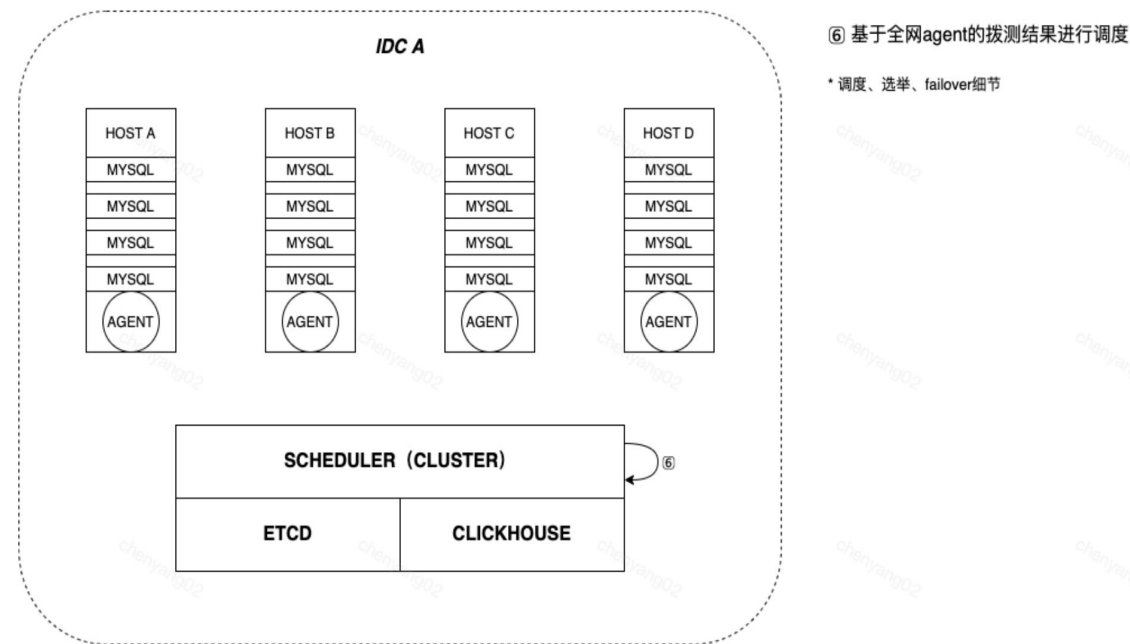
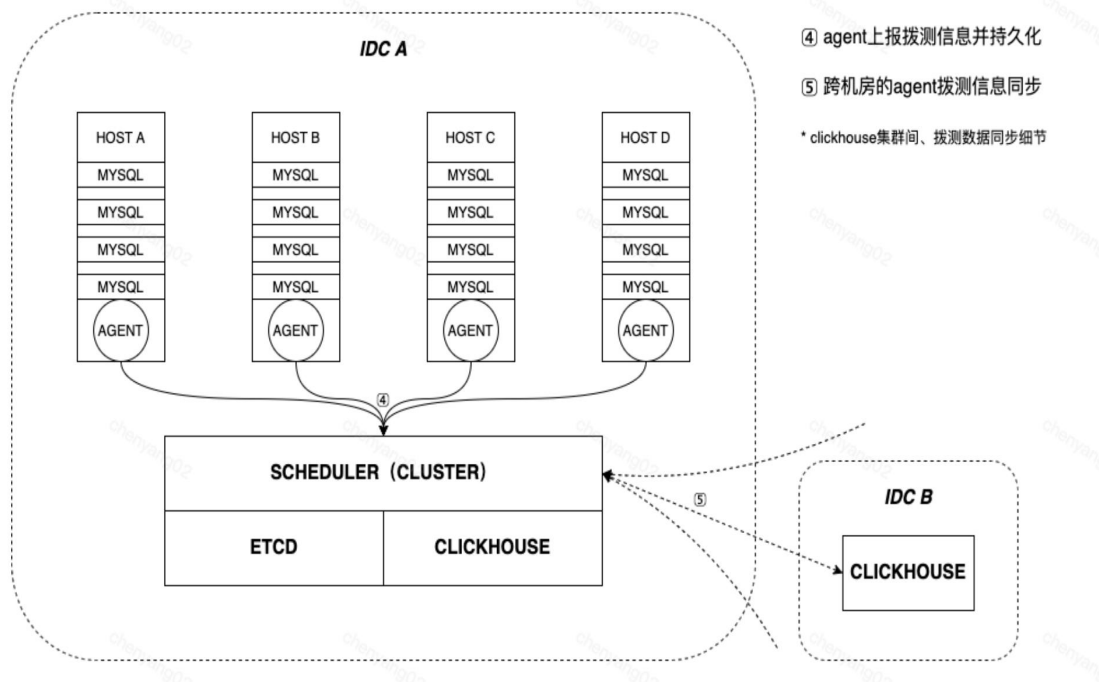
② 调度端拨测任务下发本机房内所有agent

* 确认同步频次、缓存时效及告警；
除了主动拉取外需支持被动接受推送
元数据涉及版本；
任务下发及分配；



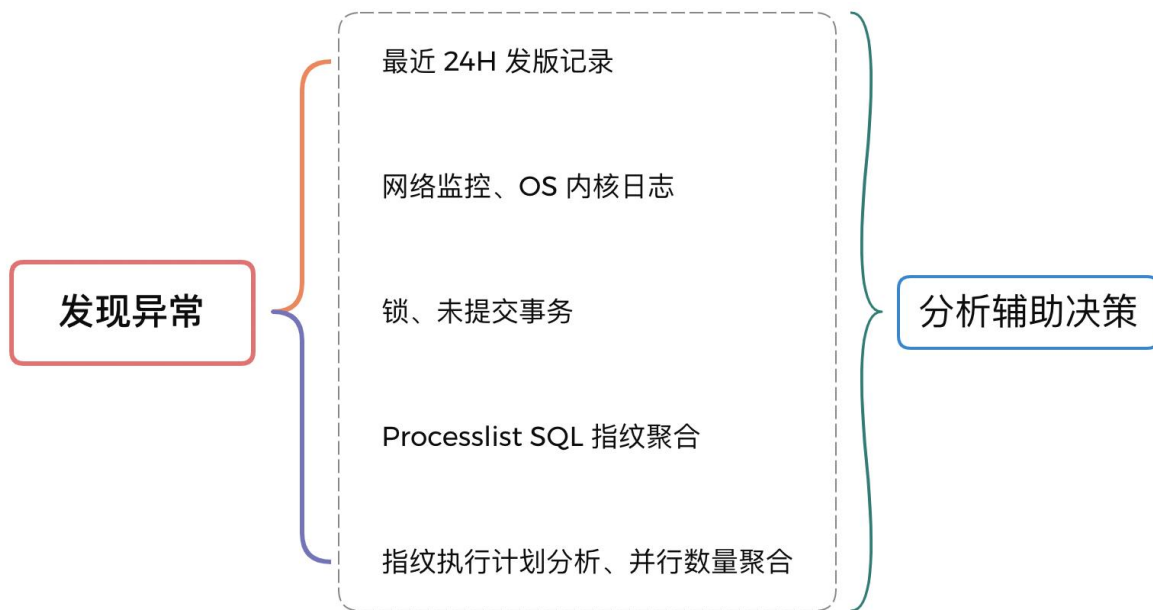
高可用 BRM – Bilibili Replication Manager

治理关键词: 同机房 -> 跨机房、多节点投票决策、专线抖动误切预防、熔断机制



慢查询

治理关键词: 可视化、指纹分析、趋势分析、快速止损



Analyze Result

Dangerous SQL

```
select max(dyn_ctime) as dyn_ctime, campus_id from t_campus_pool where mid = ?  
### Execution Plan ###  
[  
  {  
    "rows":664487,  
    "select_type":"SIMPLE",  
    "Extra":"Using where",  
    "ref":null,  
    "key_len":null,  
    "possible_keys":null,  
    "key":null,  
    "table":"t_campus_pool",  
    "type":"ALL",  
    "id":1  
  }  
]
```

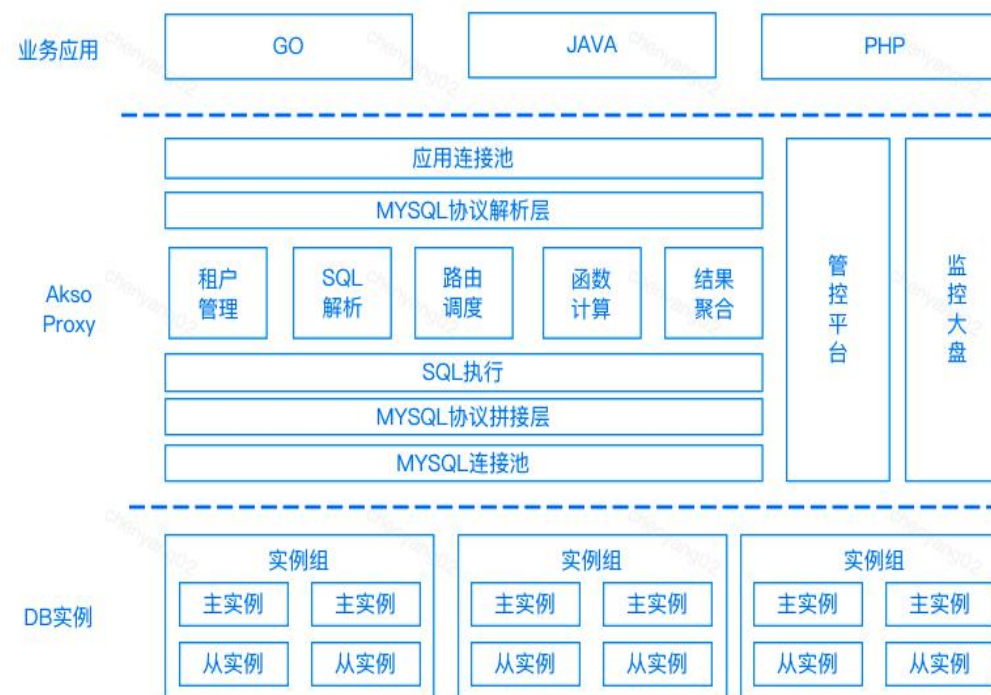
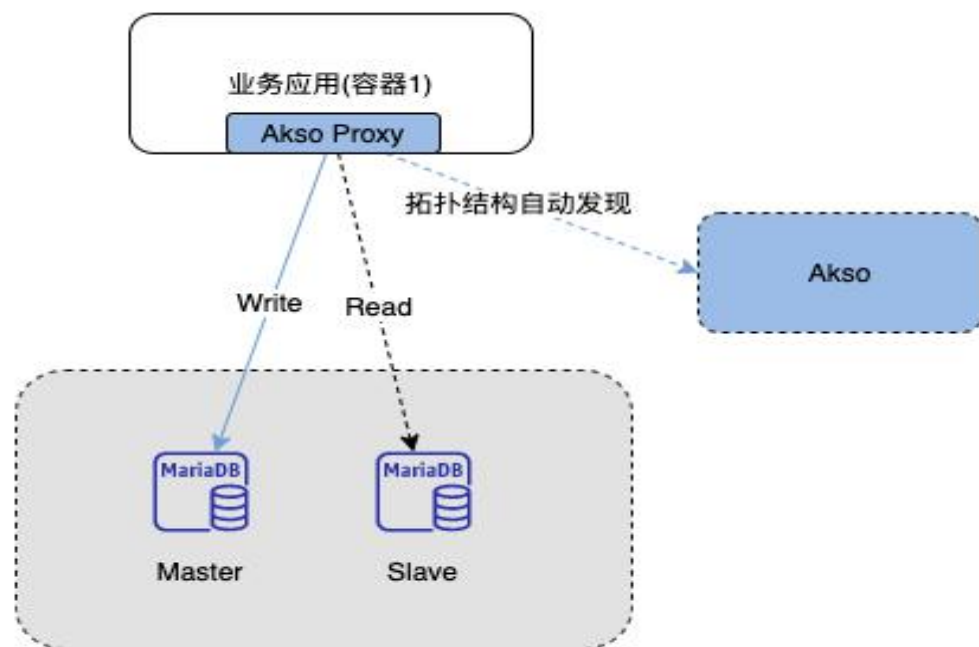
Recommended actions

- kill user xxx;
- set global max_statement_time = 3;
- add sql to black list by proxy config

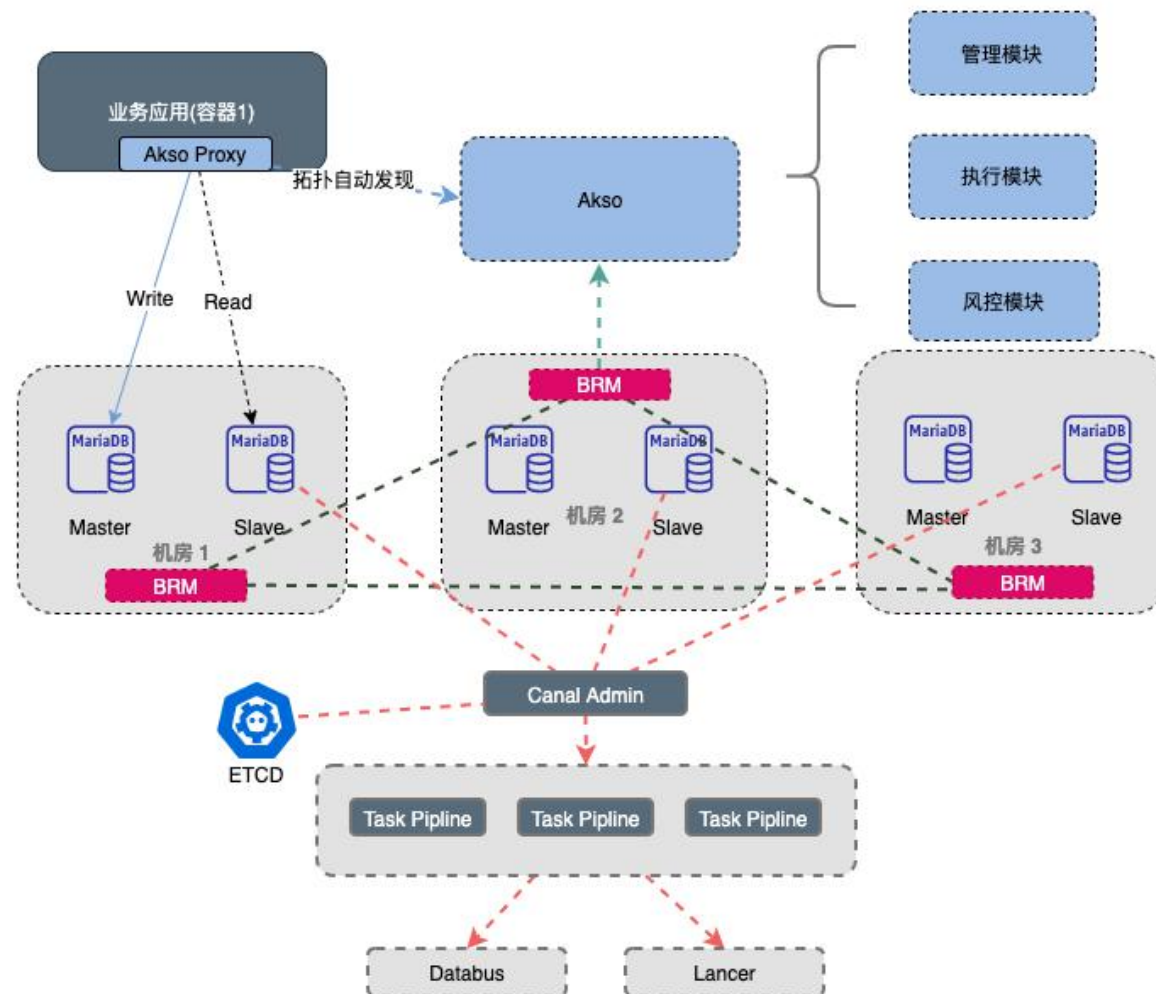


DB Proxy

治理关键词: 微服务、Sidecar、自动发现、限流与拦截、多机房路由

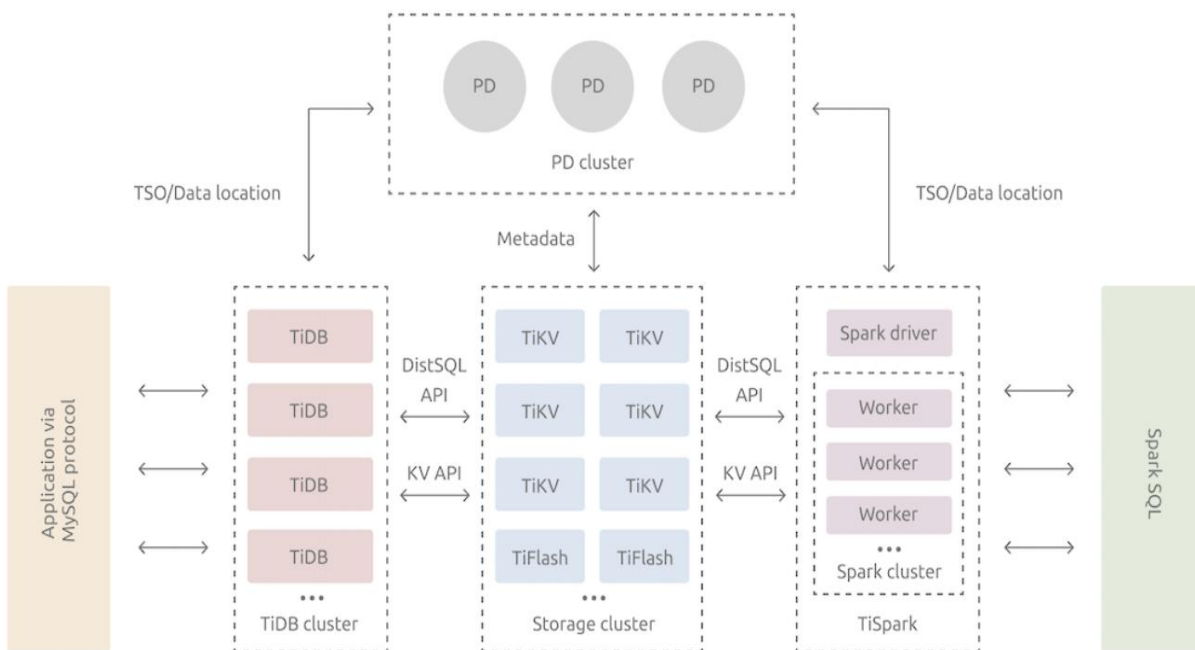
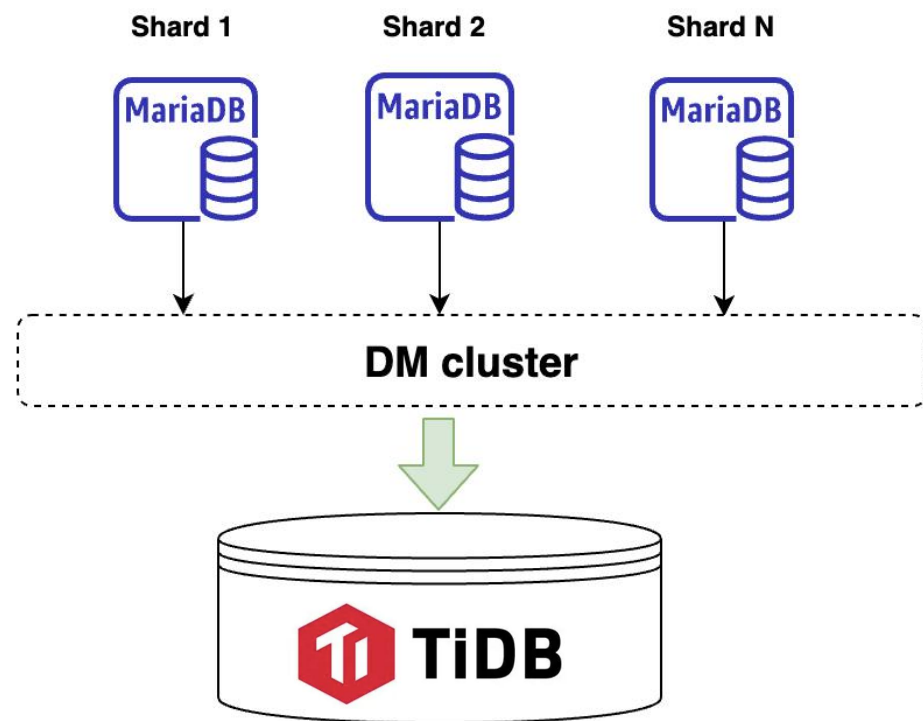


全局视角



拆分与聚合

治理关键词: 千亿数据量、二次拆分 or 合并、水平扩展、自动扩缩容





应急响应

治理关键词: SOP、OnCall、紧急事件响应、故障处理、定期演练

- **SOP:** Standard Operating Procedure
- **OnCall:** SRE 方法论、主备角色
- **紧急事件响应:** 告警治理、分工合作、问题上升
- **过载保护:** 触发阈值、快速扩容、拦截新请求、kill 旧请求、黑名单指纹、服务降级
- **定期演练:** 不可忽视却常常被忽视的一个环节





故障处理

新手

- 慌乱中不知道该怎么下手
- 一头扎入某个问题底层深究, 再也出不来, 不管外界系统死活...

老司机

- 第一时间迅速评估影响范围
- 快照, 止损, 优先保证业务可用性
- 通过有效手段重现问题, 逐步缩小问题范围, 定位根因



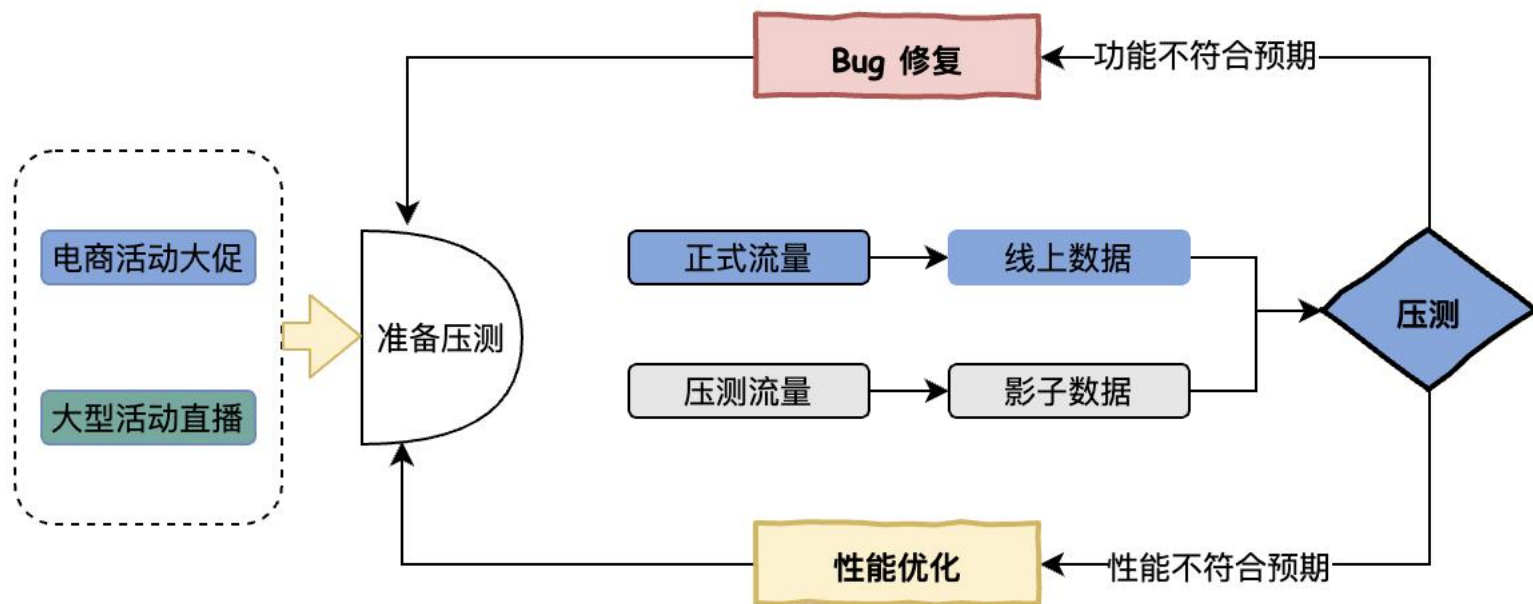


故障处理



大型活动保障

治理关键词: 全链路压测、全链路跟踪、混沌工程、扩容、限流、熔断、降级、容灾



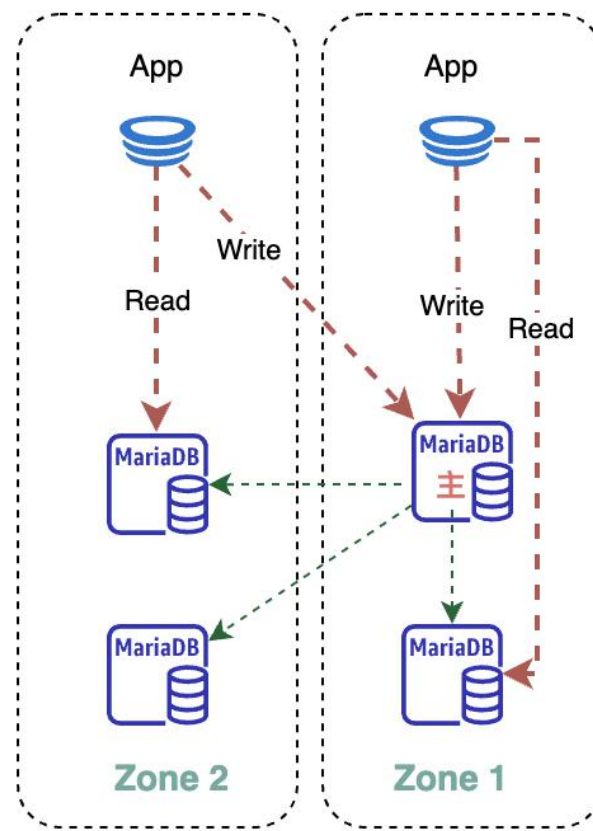
一旦发生数据安全问题, 带来的风险可能是毁灭性的

多活与容灾

治理关键词: 读多活、写多活、单元化改造、流量转发

早期的读多活架构, 类似支付宝的Gzone

- 服务器级别故障: 无数据丢失, 业务无影响
- 机房级别故障: 无数据丢失, 业务降级只读

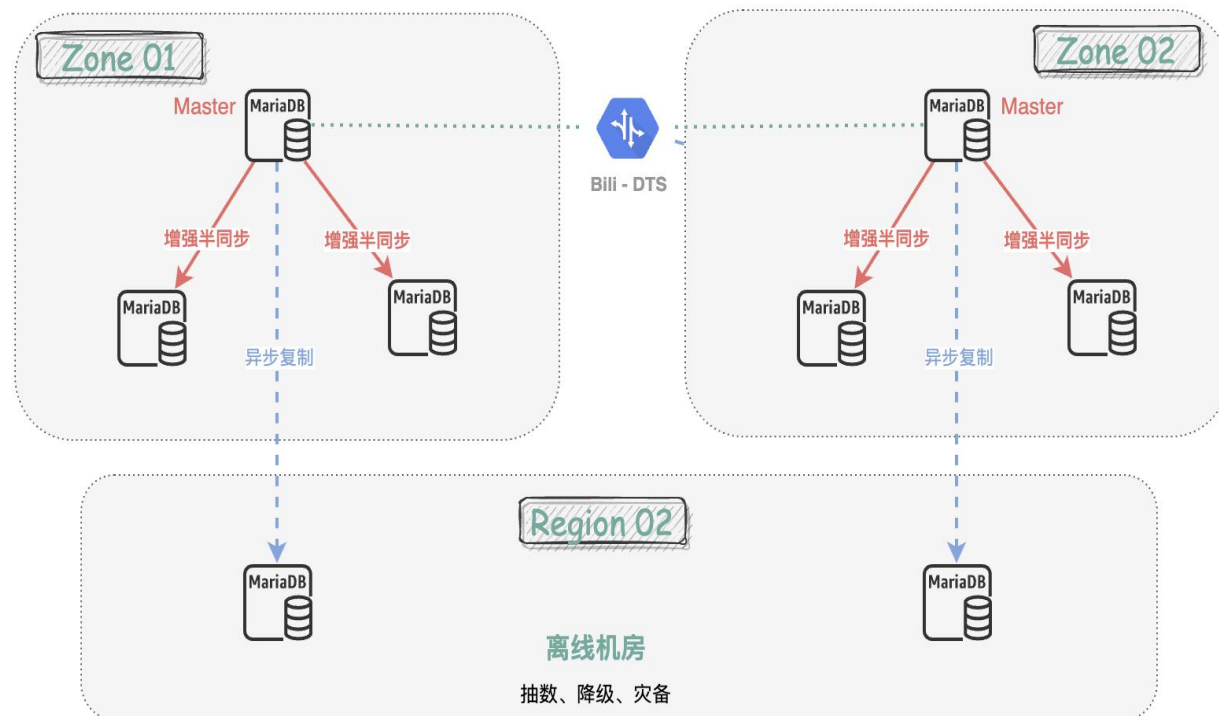


多活与容灾

治理关键词: 读多活、写多活、单元化改造、流量转发

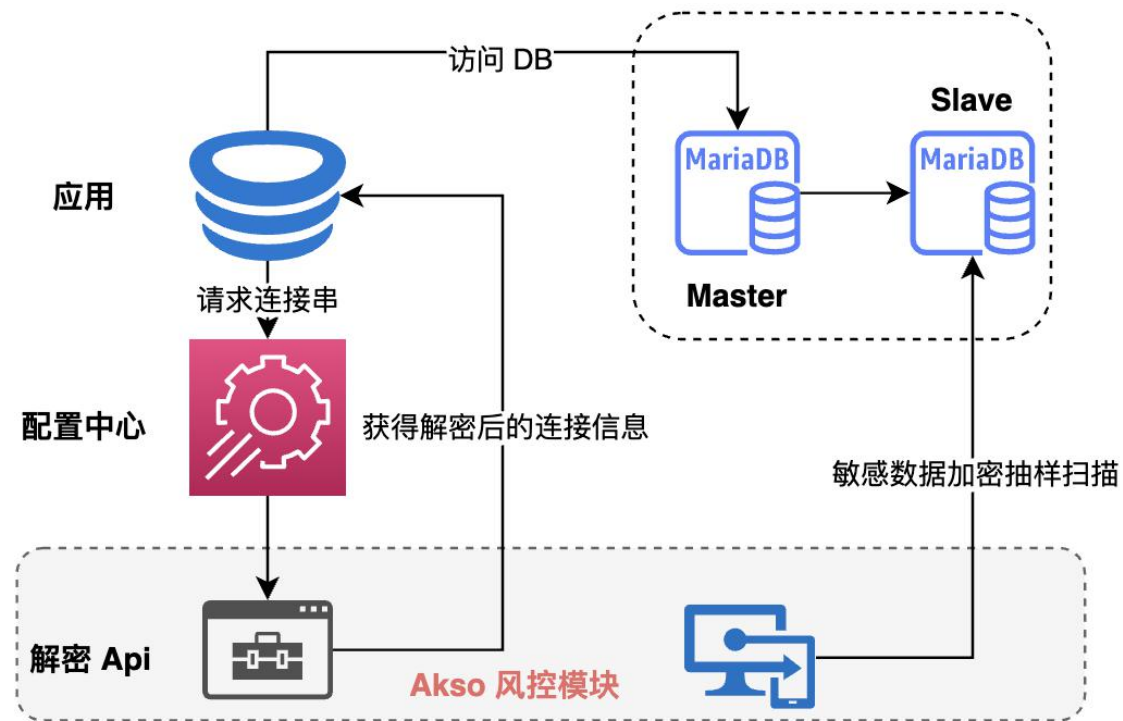
部分集群改造为写多活架构

- 服务器级别故障: 无数据丢失, 业务无影响
- 机房级别故障: 分钟级数据丢失, 业务无影响



数据安全

- 用户行为安全审核
- 异常流量发现
- 账号密码泄漏
- 数据泄漏





数据安全

治理关键词: 怎么防止删库跑路

意外删除

- 备份抽样恢复性验证
- 数据分级, 重要数据物理备份 + 逻辑备份
- 延迟从库
- 堡垒机危险命令审核机制

蓄意破坏

- 堡垒机危险命令审核机制
- 权限隔离, DBA 对备份文件只读权限
- 善待你的 DBA 兄弟们...



感谢观看



数据库岗位热招中, 欢迎自荐或推荐

在这里, 你可以收获一群热爱技术的逗比队友

在这里, 你可以拥有二次元的工作氛围

在这里, 你可以找到自己的星辰大海...



数, 造, 未, 来





THANKS