

# 数据库治理最佳实践

泮圣伟（十眠）

微服务治理研发





**基于SQL的流控降级与容错：**提供基于 SQL 洞察的完整流量防护能力，流量控制、并发隔离、熔断降级、热点流控等



**连接池治理：**提供完整实时的连接池指标，以及完整的连接池治理方案，“坏”连接剔除、提前建连、访问控制。



**数据库灰度：**全链路灰度、开发环境隔离必备场景**无需代码改造**



**动态读写分离：**弱读请求动态隔离至只读实例，提升主库稳定性，提升应用吞吐量

业务场景/痛点

- 场景一：SQL语句处理时间比较长导致线上业务接口出现大量的慢调用，需要快速定位有问题的慢SQL。
- 场景二：在项目初期没有对SQL的性能做好考量，随着业务的发展，用户量级的增加，线上遗留老接口的SQL逐渐成为性能瓶颈。
- 场景三：某系统对外提供某查询接口，SQL语句涉及多表join，某些情况下会触发慢查询，耗时长达30s，最终导致 DB 连接池/Tomcat 线程池满，应用整体不可用。

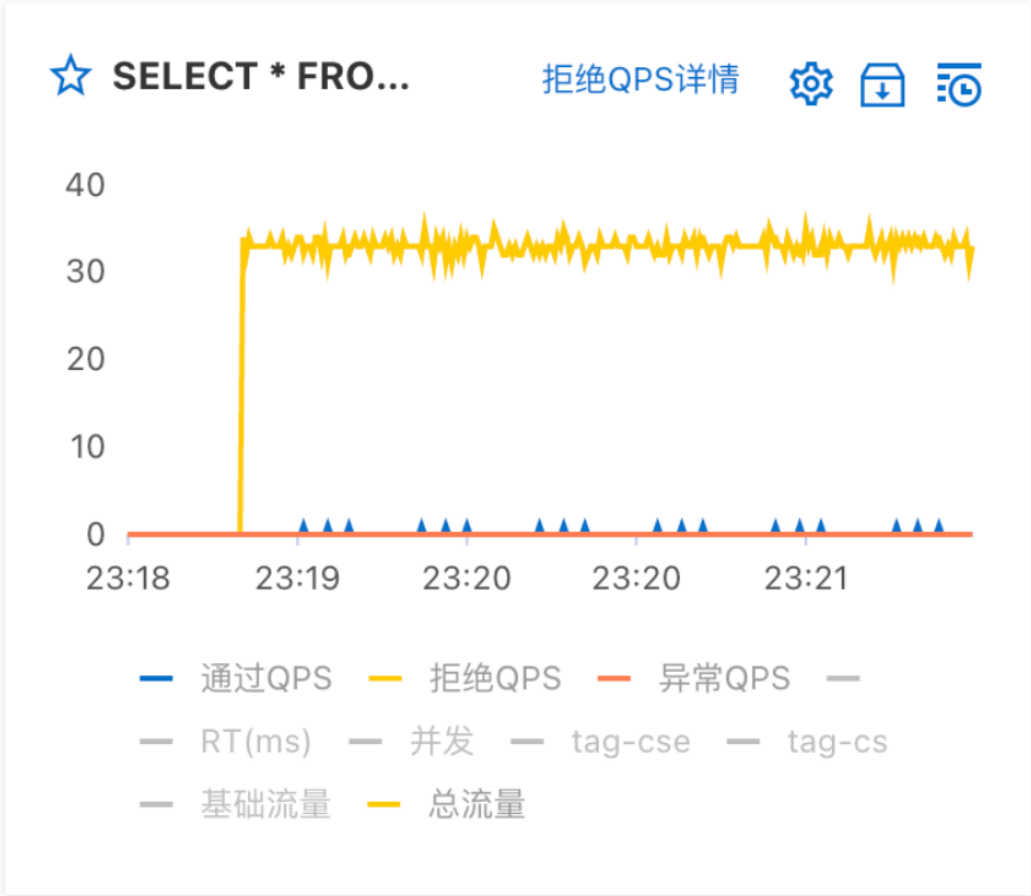
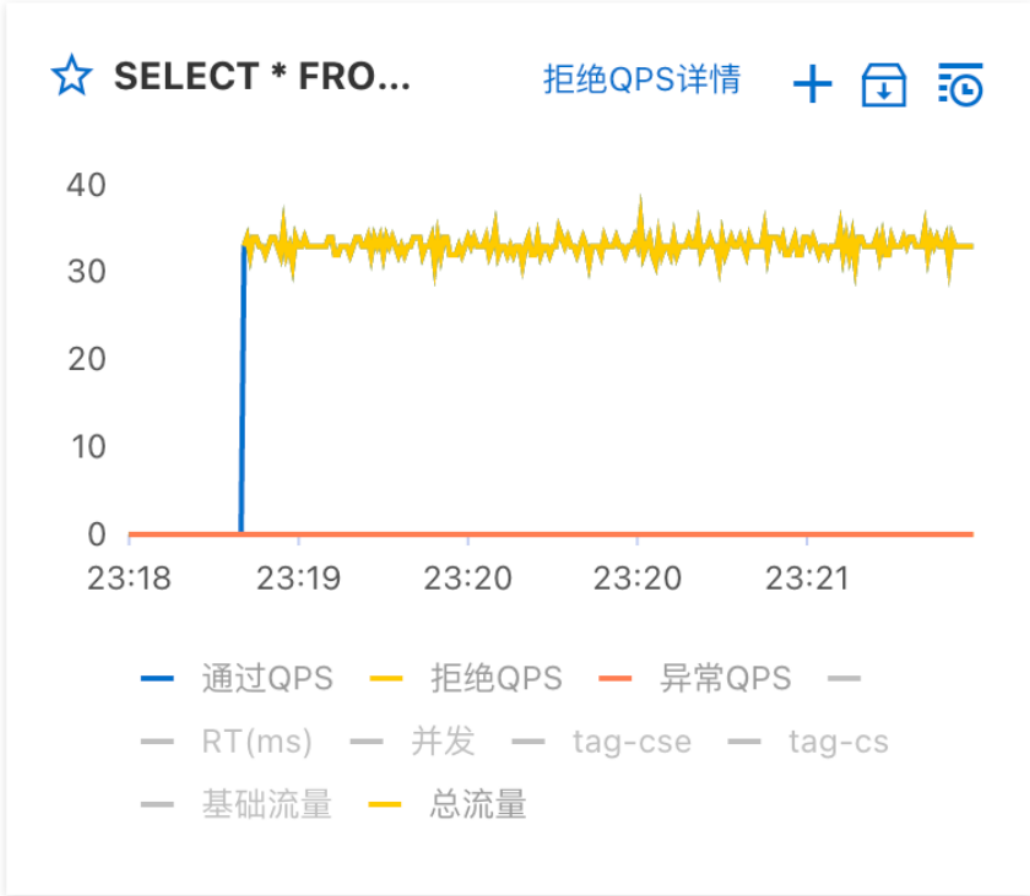
解决方式

- 通过秒级的 SQL 洞察能力，观察应用和资源API维度的实时数据，从而可以有效地评估系统的整体表现。
- SQL 洞察能力，可以有效分析SQL语句是否写得合理，通过SQL执行的 TopN 列表可以快速定位并发、RT、请求量过大的SQL请求。

全部接口

节点对比 集群统计

展示指标 ☒ 总流量 ☒ 通过QPS ☒ 拒绝QPS ☒ 异常QPS ☐ RT(ms) ☐ 并发



Top SQL 洞察

SQL语句	通过QPS⬆️	拒绝QPS⬆️	异常QPS⬆️	平均耗时(ms)⬆️	最大并发⬆️	操作
INSERT INTO mse_demo_table (location) VALUES (?)	1123	0	0	231	12	SQL防护
INSERT INTO mse_demo_table(location) VALUES(?)	1101	0	3	220	3	SQL防护
INSERT INTO mse_demo_table_gray (location) VALUES ('0\$')	110	0	0	0	1	SQL防护



## 业务场景/痛点

场景一：SQL语句处理时间比较长导致线上业务接口出现大量的慢调用，需要快速定位有问题的慢SQL，并且通过一定的治理手段进行隔离，将业务快速恢复。

场景二：某系统对外提供某查询接口，SQL语句涉及多表join，某些情况下会触发慢查询，耗时长达30s，最终导致 DB 连接池 /Tomcat 线程池满，应用整体不可用。

## 解决方式

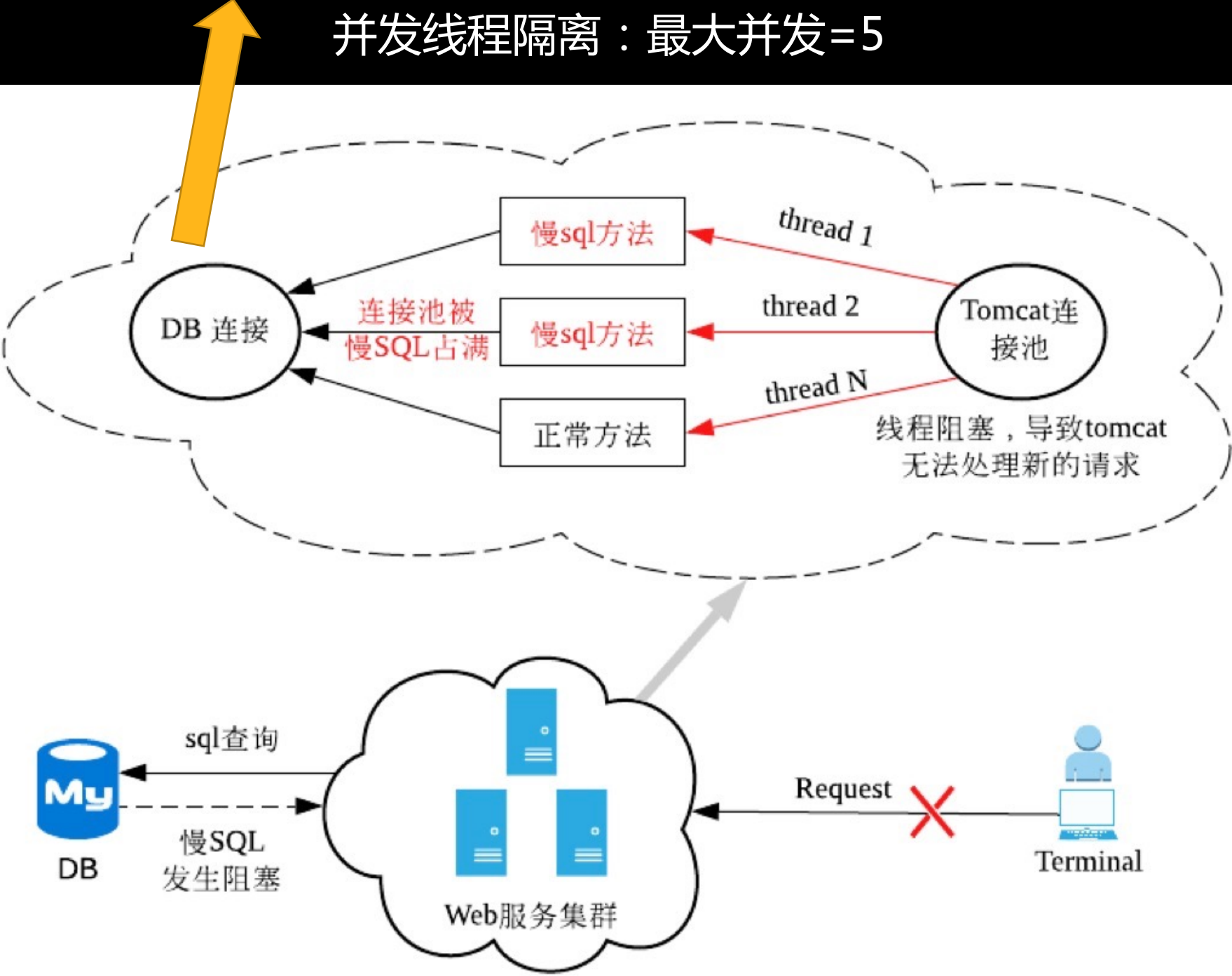
当流量近似稳态时： $\text{并发线程数} = \text{QPS} * \text{RT(s)}$ 。RT 升高，并发线程数升高，代表服务调用出现堆积。

通过SQL洞察快速定位并发大的SQL，并配置服务并发隔离能力，相当于一道“软保险”，防止满调用的SQL挤占正常服务资源，保证服务整体的稳定性。

## Thread Pool



并发线程隔离：最大并发=5





# *MSE* 产品升级发布会