

数据库常见性能故障应急场景



原创 符涣涣 2024-12-04

1437

扁鹊三兄弟的故事是中医文化中的一个经典典故，主要阐述了预防医学和早期干预的重要性。医术层次的划分：大哥：医术最高，擅长“治未病”。二哥：医术次之，专注于“治欲病”。扁鹊：医术在公众眼中被认为最出色，但实际上他专注于“治已病”。

扁鹊三兄弟的故事启发了数据库应用性能“三级预防”理念，即一级预防是防止事件发生，二级预防是早期发现和治理，三级预防是故障时快速应急。

- 第一道防线，应用遵守规范，例如SQL审核、性能测试，代价小收益大；
- 第二道防线，生产隐患一体化治理，例如问题治理，有代价可防隐患进一步恶化；
- 第三道防线，自动化自愈和故障应急，例如小表突变自动收执行计划，代价大收益小，并非“包治百病”。

本文以第三道防线为主，展开探讨下数据库应用性能应急故障场景。

数据库应用性能应急故障场景有哪些？

根据历史应用问题引发的数据库性能事件，总结出以下五大应用SQL性能故障场景，针对这些场景开发监控和自动化工具，以便应用引发数据库性能故障时可快速应急。



1、小表突变统计信息收集不及时

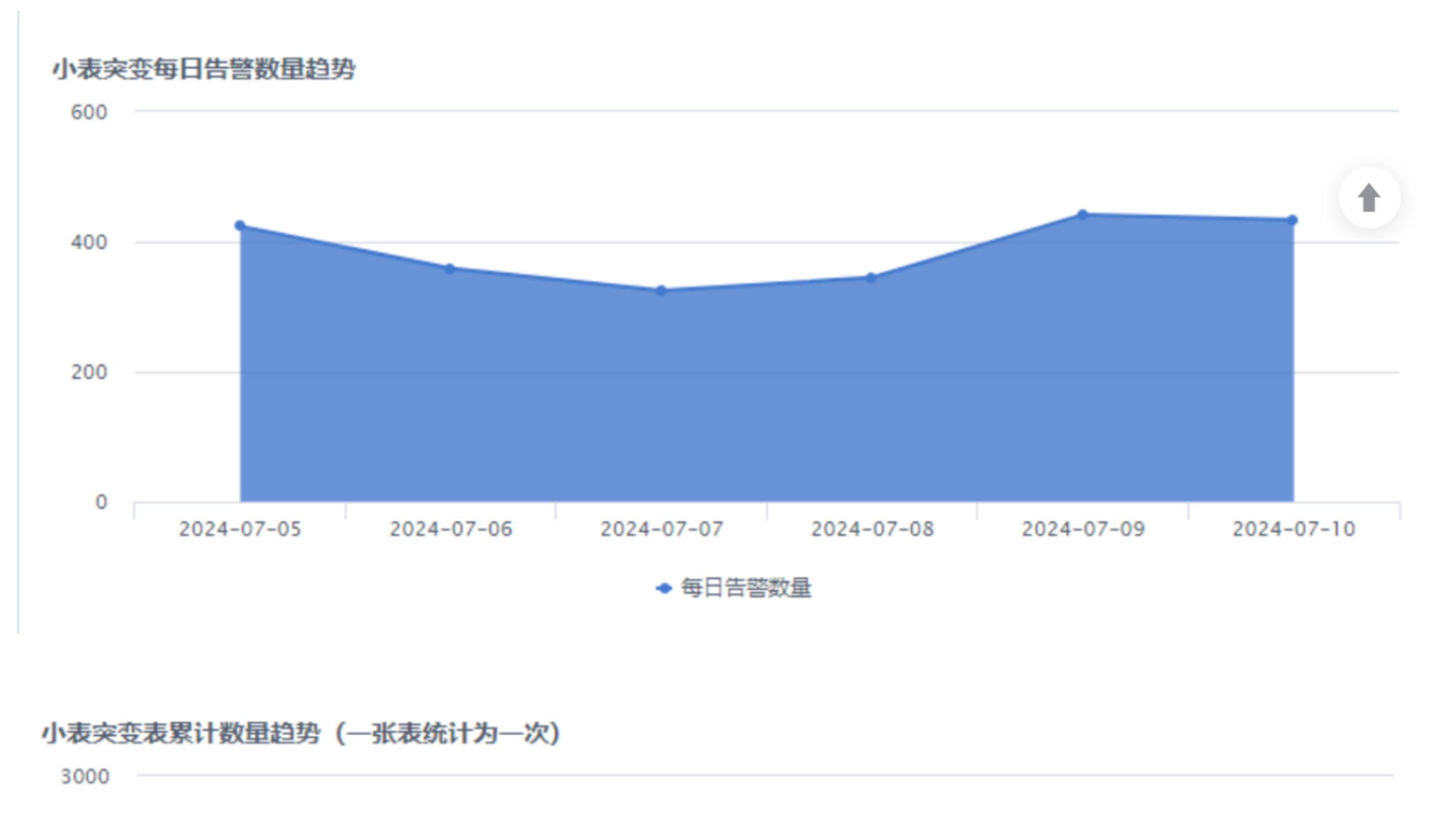
故障场景

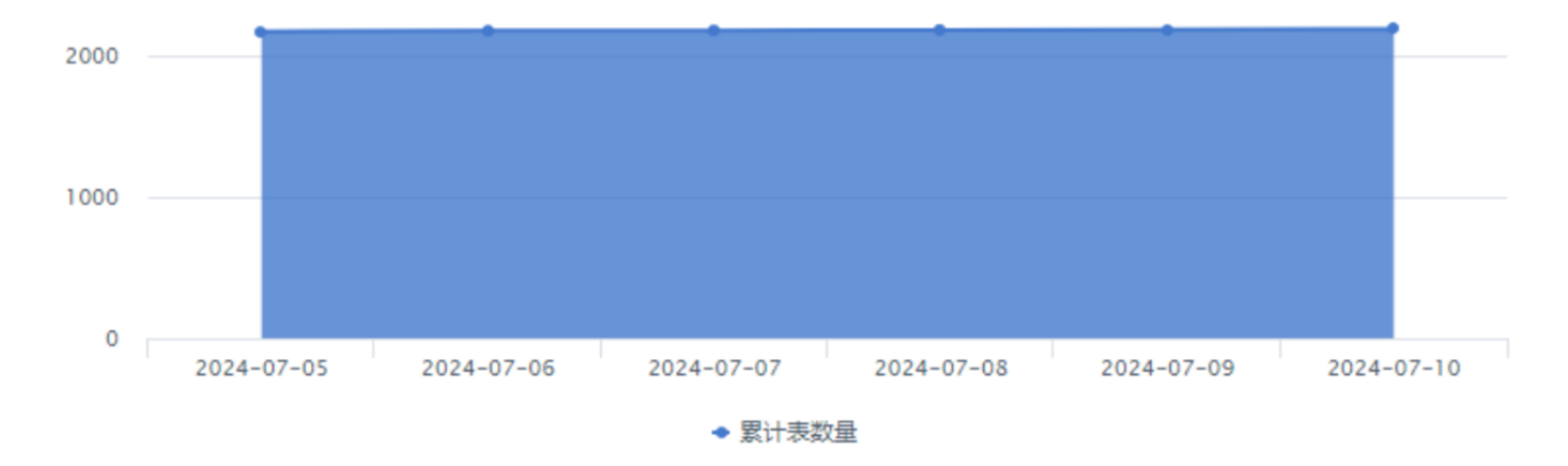
某应用对表加载了大量数据（百万级），但是没有及时对表收集统计，统计信息记录该表的行数为0，导致优化器选择走全表扫描，随着数据量增加sql性能下降，最终影响业务。

规避方案

- 1.应用加载数据后需要及时收集统计信息写入开发规范。
- 2.数据库监控平台实时监测小表DML操作行为，通过滑窗统计单位时间内数据变化量，超过设定阈值触发告警并自动收集统计信息。
- 3.通过运营推动开发固化索引或者及时收集统计信息。

实施运营





生产环境小表突变统计信息累计自愈了2897次，反映出应用在SQL开发规范的遵循上还需要提升，“应在大数据量操作之后立即人工或应用自动进行相关表的统计信息收集”。

2、统计信息不准确导致执行计划突变

故障场景

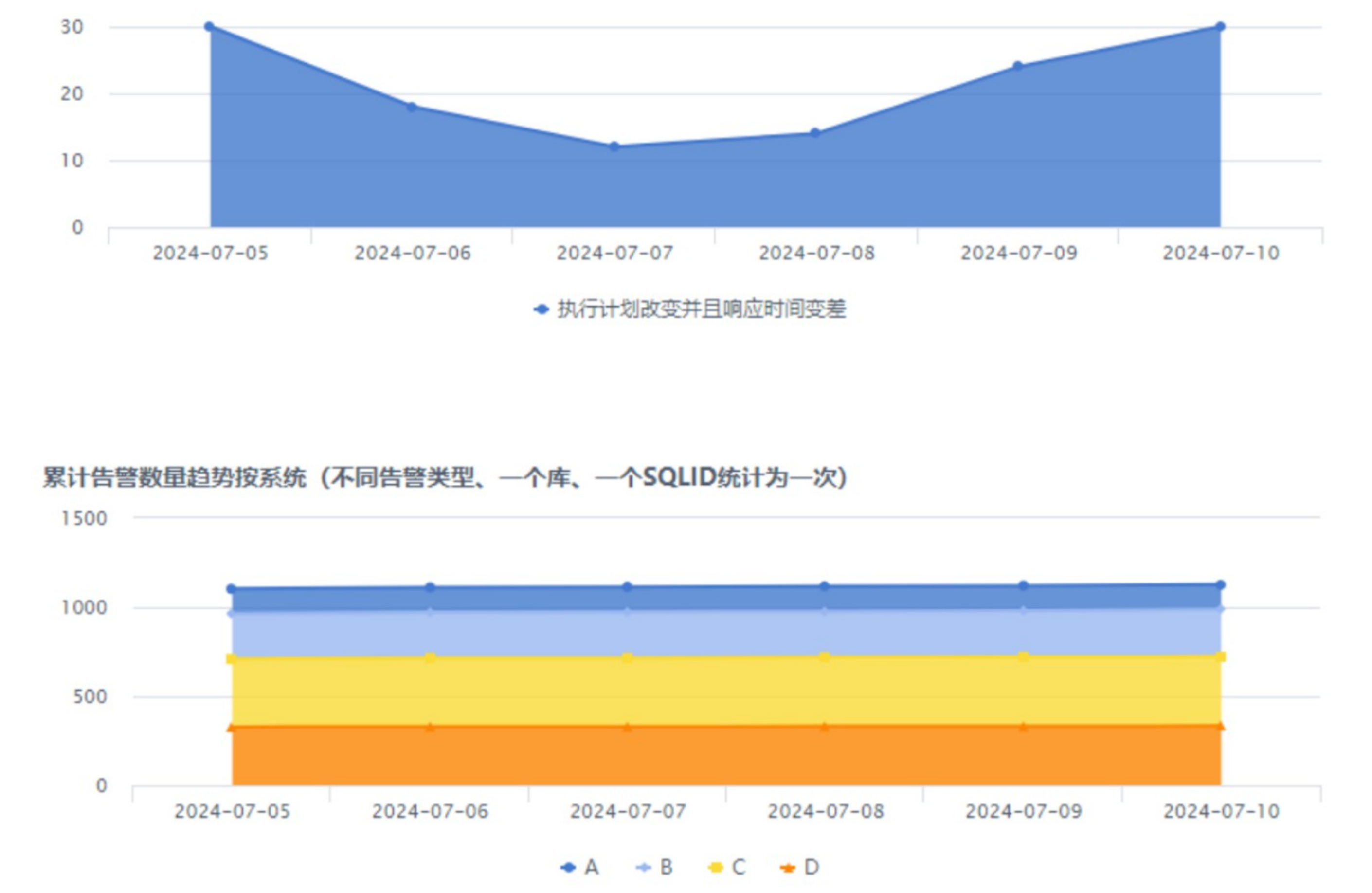
开发人员对数据表收集统计信息，导致**sql执行计划发生改变**，但是由于数据库优化器评估该表存在两个成本较接近的索引，选择了一个实际成本较差的，导致sql性能下降。

规避方案

- 1.应用对于关键交易的SQL语句、涉及表数据量变化比较大的SQL语句建议采用添加**提示（hint）** 固定。
- 2.数据库监控平台实时采集Top SQL的执行态信息，执行计划发生改变时，进行前后对比，如果变差触发告警并**回退执行计划**。
- 3.通过运营，推动开发将Top SQL的执行计划通过提示固定住，确保执行计划稳定。

实施运营





生产环境执行计划回退累计自愈了1114次，反映出应用在SQL开发规范的遵循上还需要提升，“对于关键交易的SQL语句、涉及表数据量变化比较大的SQL语句建议采用添加提示（hint）的写法来保证执行计划的准确性”。

3、大事务导致锁资源占用过高

故障场景

开发人员对某系统数据库九千多万行数据大表进行表字段更改出现异常，导致锁表，影响部分业务，回滚花费了十几个小时。

规避方案

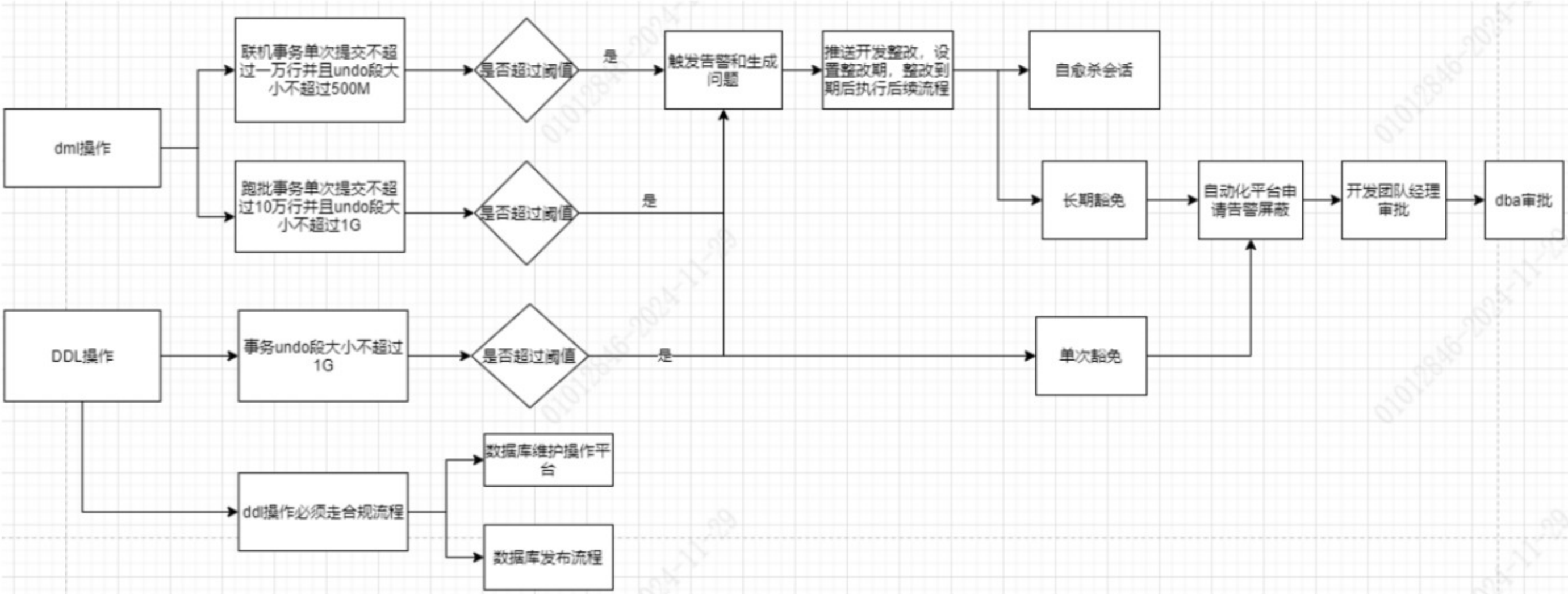
制定《大事务强制熔断机制方案》，生产识别到大事务立即触发熔断，防止事务继续扩大造成影响。

大事务判断标准：

- 联机交易事务涉及的记录数量不得超过**10000**条并且占用的undo大小不超过**500M**；
- 批量事务涉及的记录数不得超过**100000**条并且占用undo大小不得超过**1G**；

实施方案

实时监控大事务情况，触发告警后自愈；推动应用限期整改，并分级分类实施。流程如下图：



4、缺索引导致全表扫描

故障场景

某系统应用表数据量逐渐增大，但是缺少索引，导致sql性能下降全表扫，影响业务。

规避方案

提供智能索引推荐工具sql access advisor，在sql出现全表扫描性能问题时可给出建索引的语句。

实施方案

- 1、实时监控全表扫语句，高并发触发告警；
- 2、封装“oracle sql adviser”实现自动化，一键处理。

5、高并发慢SQL导致资源不足

故障场景

- 应用高并发锁等待导致业务异常
- 应用突发大流量导致数据库资源被打满
- 高并发慢sql短时间无法快速优化，例如需新建索引
- 慢sql源头无法快速识别，立即止血。

规避方案1

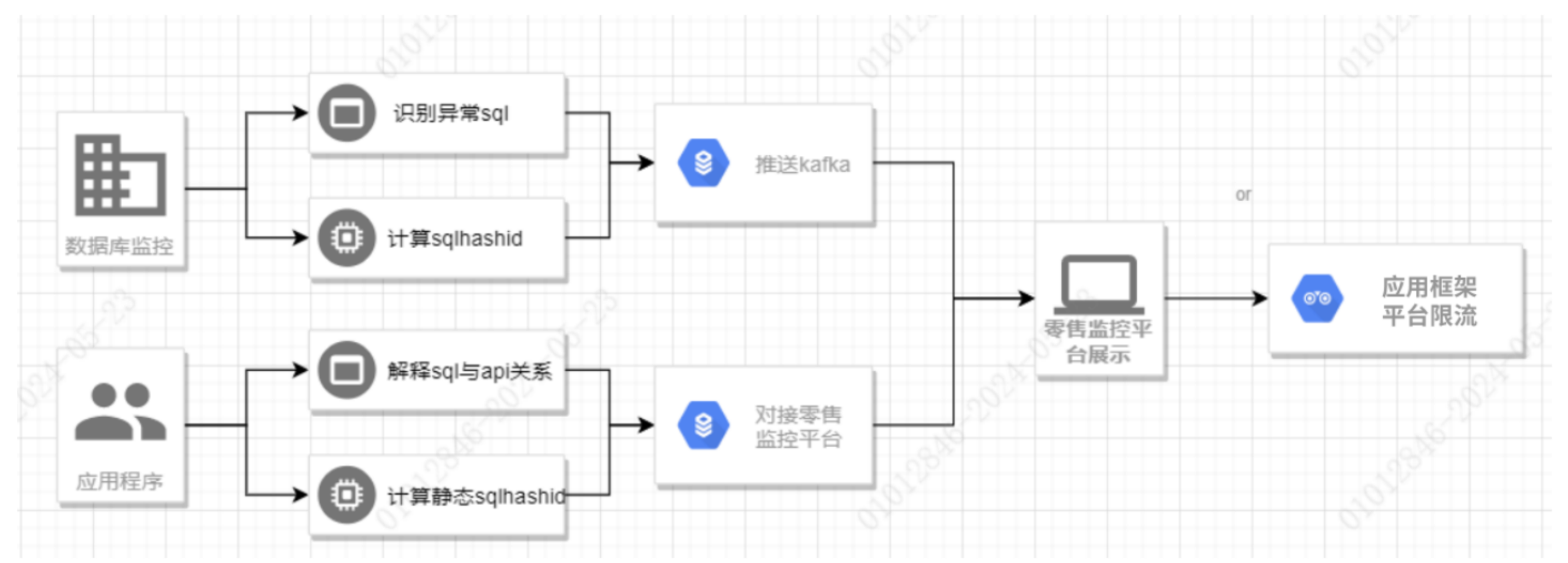
数据库性能故障告警关联链路追踪应用告警，数据库和应用同时异常触发性能应急三板斧自愈。自愈处理点：

- 循环杀并发过高sql；
- 扩数据库规格；
- 杀锁源头；
- 回退topsql的执行计划；

规避方案2

制定数据库遭遇性能故障时从应用层面进行**API限流**的技术方案，从而实现快速识别应用的问题接口，并达成快速隔离、防止出现性能故障蔓延到全库导致系统层面不可用。

方案流程： 数据库sql监控->应用api与sql关系->框架自动限流，全流程自动化。



总结

综合以上各个场景，可以看出引发应用性能类故障或隐患，起因都是应用程序开发没有遵守开发规范、架构要求。如果严格遵守了这些故障或隐患是可以规避的。

“治已病”不如“治未病”，即三道防线，最管用的是事前第一道，应用遵守规范；事中第二道，发现隐患一体化治理；第三道是事后应急，做有限度的补救。

在工具建设方面做了如上的补救措施，这些措施都属于事后，因为资源问题无法覆盖所有场景和范围，即无法保证应用不遵守规范的情况，自动化工具能100%兜底。

◀

墨力计划

sql

性能优化

最后修改时间：2024-12-04 11:22:40

「喜欢这篇文章，您的关注和赞赏是给作者最好的鼓励」


关注作者

赞赏

【版权声明】 本文为墨天轮用户原创内容，转载时必须标注文章的来源（墨天轮），文章链接，文章作者等基本信息，否则作者和墨天轮有权追究责任。如果您发现墨天轮中有涉嫌抄袭或者侵权的内容，欢迎发送邮件至：contact@modb.pro进行举报，并提供相关证据，一经查实，墨天轮将立刻删除相关内容。

评论

分享你的看法，一起交流吧～

wzf00726月前

评论 0


扁鹊三兄弟的故事是中医文化中的一个经典典故，主要阐述了预防医学和早期干预的重要性。医术层次的划分：大哥：医术最高，擅长“治未病”。二哥：医术次之，专注于“治欲病”。扁鹊：医术在公众眼中被认为最出色，但实际上他专注于“治已病”。

扁鹊三兄弟的故事启发了数据库应用性能“三级预防”理

筱悦星辰7月前

评论 0

只要敢于尝试，任何时候开始做一件事都是最佳时间。

星星之火8月前

评论 0

第一道防线，应用遵守规范，例如SQL审核、性能测试，代价小收益大；
第二道防线，生产隐患一体化治理，例如问题治理，有代价可防隐患进一步恶化；
第三道防线，自动化自愈和故障应急，例如小表突变自动收执行计划，代价大收益小，并非“包治百病”。

心静自然凉8月前

评论 0

👍《黄帝内经》“上工治未病，不治已病，此之谓也”，高明的医生都是治未病，提前预防问题的发生，前置运维

章芋文8月前

评论 1

学习了，基本上涵盖了常见的突发性能故障类型，还从事后救火逐渐过渡到了前置运维，体现了dba真正的价值啊

相关阅读

【GaussDB】深入剖析Insert Select慢的定位全过程

深度解析MySQL的半连接转换

巧用json工具解析MySQL优化器追踪文件

SQL Server、MySQL (InnoDB)、PostgreSQL、达梦 (DM)、金仓 (KingbaseES) 在事务日志 (Transaction Log / WAL) 管理架构上的核心差异及简单对比

SQL Server、 MySQL、 PostgreSQL 三大数据库事务日志简单对比及对高可用方案配置粒度的影响

数据压缩60%让“PostgreSQL” SQL运行更快，这不科学呀？

Oracle关于慢sql的优化

在Oracle 11gR2中，一个Session Cursor能够被缓存在PGA中的必要条件时该Session Cursor所对应的SQL解析和执行的次数要超过几次？

SQL成“致命杀手”？DBdoctor【SQL限流新功能】一招精准限流，守护核心业务不宕机

高效压测 Doris 全指南：从指标监控到性能优化实践