



—— 开源与架构技术沙龙-北京站 ——

关联与下钻分析

--如何快速定位MySQL性能瓶颈

李季鹏

1

MySQL性能管理需求与现状

2

MySQL性能分析建设

现实需求

如何用技术解决当前数据库管理的难题？

简单化
自动化
智能化

利用技术经验沉淀解决人少事多专业性强的数据库管理工作，构建自动化帮助企业构建良好的数据库原型环境；

如何让少量的DBA管理大量的数据库？

如何通过简单的方式改善和优化数据库性能问题？

如何持续保障数据库的稳定性/可用性

如何快速发现数据库性能、故障问题并且定位

如何持续跟踪数据库性能优化的效果

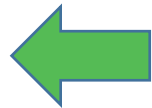
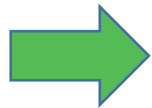
如何自动检查数据库健康、自我诊断、自我优化

.....

传统MySQL性能管理手段的限制

分析手段多样

- 运行状态
 - ◆ Global Status
 - ◆ Session Status
- 语句分析
 - ◆ Slow日志 + pt-query-digest
 - ◆ profiling
 - ◆ performance_schema
- 会话分析
 - ◆ processlist
 - ◆ pstack
- 第三方工具集
 - ◆ percona-toolkit
 - ◆ ...



分析管理复杂

- 指标增量获取繁琐
 - ◆ Global Status增量需要mysqladmin
- 严重依赖语句执行分析
 - ◆ Session Status分析依赖语句重新执行
 - ◆ profiling分析依赖语句重新执行
- 欠缺增量语句记录的手段
 - ◆ performance_schema语句只存储累积值
 - ◆ 基于slow日志信息较少，入库环节不易维护

依赖手段多样，管理工具化整合难以做到轻量级、简单化

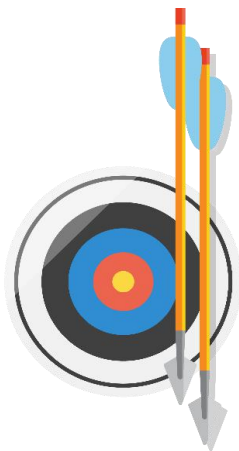
缺乏增量记录手段，难以做好历史性能的回溯分析

更多的语句执行细节严重依赖于语句的手工执行



轻量化采集

丰富采集信息，但确保统一且非侵入式性能指标获取手段，避免自动化采集过程对目标端的性能影响。



增量化记录

状态指标、语句信息实现增量化记录，使性能管理具备丰富的时段展示手段和历史回溯手段。



经验沉淀

沉淀MySQL性能分析经验，重点在于沉淀指标关联、下钻到问题语句分析过程。

1

MySQL性能管理需求与现状

2

MySQL性能分析建设

MySQL性能下钻分析路线图

性能入口

核心指标上浮

阶段核心：快速定位性能问题节点

关键问题：选取少量具有代表性的核心指标

关联分析

指标分解

指标对比

阶段核心：提供关联、对比等分析手段

关键问题：指标增量采集，语句增量采集

指标语句关联

时段执行回溯

问题点定位

语句负载

响应时间监控

语句阻塞

大范围查询

多范围查询

大批量变更

临时表情况

语句关联情况

语句排序情况

全表扫描语句

语句执行报错

... ..

阶段核心：提供丰富多样的语句执行信息

关键问题：避免需要额外执行语句的手段

性能入口——选取适当的上浮核心指标

选取原则

- 直观反映数据库负载
- 横向反映数据库间性能问题程度

性能公式参考

时段负载：

$$\text{Load} = (\text{QPS} * \text{avg_Latency}) * \text{time}$$

引申问题：由于采集有周期性，慢查询越多的数据库，通过该公式计算的负载值与OS统计的负载值相差越大

公式修正：

$$\text{ASPS} = (\text{QPS} * \text{avg_Latency}) + \text{SUM}(\text{cycle_inc_time})/\text{time}$$

$$\text{ASPS} \approx \text{System Load}$$

性能问题影响程度：

$$\text{QRTi} = \text{Optimum} * 1 + \text{Acceptable} * 0.5 + \text{Unacceptable} * 0$$

参考指标

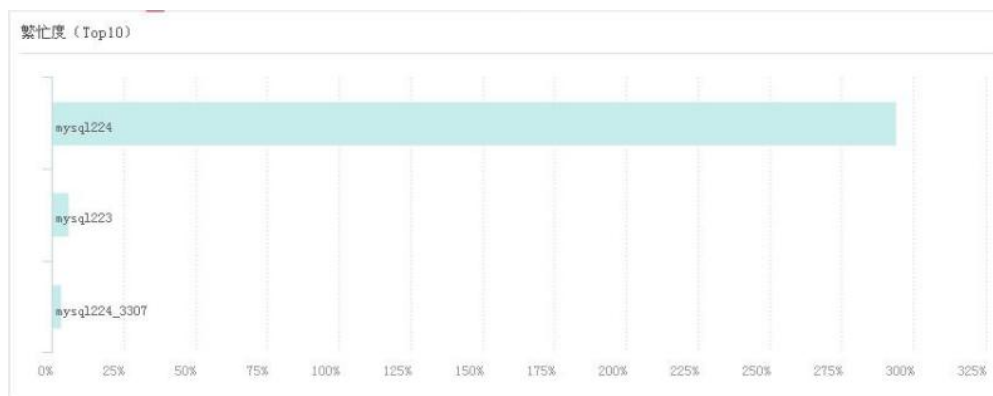
- QPS (Query Per Second) 每秒查询(完成)数
- ASPS (Active Statement per Second) 繁忙度，每秒同时活跃操作数
- QRTi (Query Response Time index) 查询响应有效率

QPS

- 特点:
 - ◆ 基于执行次数的性能指标
 - ◆ 反映业务每秒发起的查询次数
- 优势:
 - ◆ 可关联下钻时段内语句执行数
- 不足:
 - ◆ 无法直观横向比较数据库性能问题

ASPS

- 特点:
 - ◆ 复合来源的MySQL性能指标
 - ◆ 反映语句语句对系统造成负载的程度
- 优势:
 - ◆ 同时反映时段内执行统计以及当前查询的问题
 - ◆ 可作为横向性能对比指标
- 不足:
 - ◆ 并不直接与数据库状态指标等同



QRTI

- 特点:
 - ◆ 基于语句执行时长的性能指标
 - ◆ 直观反映不可接受的慢语句数
- 优势:
 - ◆ 可直接下钻执行过程的问题语句
 - ◆ 可作为横向对比指标
- 不足:
 - ◆ 仅简单慢查询程度

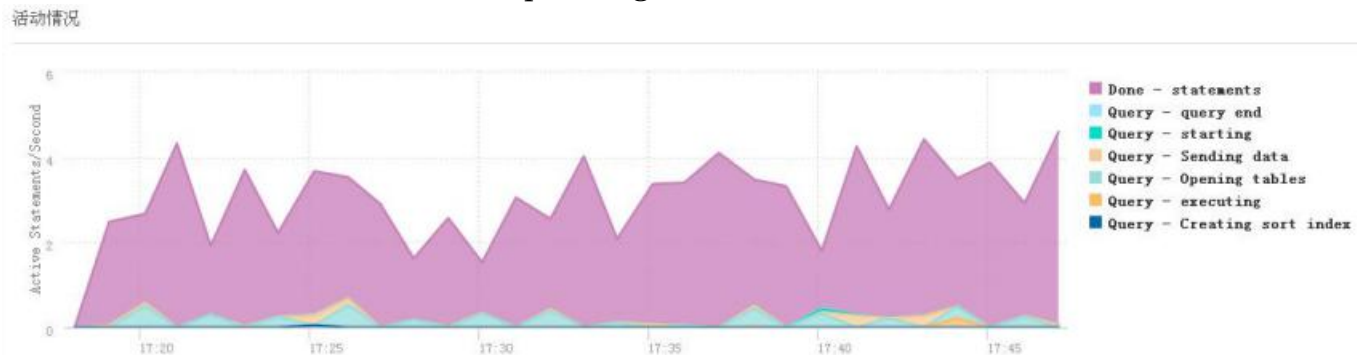
查询响应指数 (平均)



关联分析——提供多种场景的分析通路

指标分解

- 分解获取到性能指标的构成详情
- 场景举例
 - ◆ 分解ASPS，发现opening table占比较多

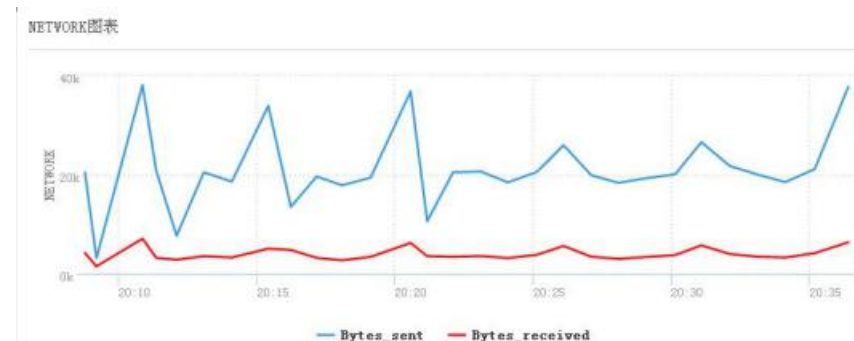


指标对比

- 选取不同性能指标按维度进行对比：
- 对比方式
 - ◆ 同库不同指标间对比：对比Com_delete与Innodb_rows_deleted
 - ◆ 异同库相关指标对比：对比不同库的innodb buffer pool命中率
 - ◆ 同库不同时段指标对比：对于语句优化后临时表创建改善情况

指标语句关联

- 提供性能指标关联到语句的分析通路
- 场景举例
 - ◆ 通过临时表创建率高定位到创建临时表的语句
 - ◆ 发现时段内bytes较高，关联到rows_sent高的语句



时段执行回溯

- 提供历史时段内的指标和语句回溯
- 场景举例
 - ◆ 回顾过去故障时段的语句执行情况

语句来源对比

●performance_schema > slow日志

●采集信息对比：

●P_S优势：
丰富信息，更多下钻维度
采集更轻量更可靠
可配置instrument筛选

● P_S劣势：
增量存储需要处理实现
根据设置，要做溢出处理
digest的样例需要关联

维度	对比项	P_S	Slow日志
指标丰富程度	count	有	结合统计工具
	query_time	有	有
	lock_time	有	有
	rows_sent	有	有
	rows_examined	有	有
	rows_affected	有	无
	errors/warnings	有	无
	tmp_tables	有	无
	关联方式	有	无
	排序方式	有	无
	索引使用	有	无
易用性	sample样例获取	麻烦	简单
	sample执行时间	麻烦	简单
	获取方式	jdbc直采	工具处理+文件
	配置方式	丰富	简单



The logo for DBAplus, featuring the letters 'DBA' in red, blue, and orange respectively, followed by 'plus' in green. The background is a blue geometric pattern of triangles and a faint city skyline.

DBAplus

www.dbaplus.cn

THANK YOU!