

TAKINTALKS

部分精彩活动节选



扫码关注公众号

- 讲师课件
- 优质技术文章
- 技术资料包/干货模版

业内TOP
专家与会分享探讨

专注
大型营销活动保障领域

8000+
相关从业者从中获益

提升MySQL性能可观测性 之TopSQL

全面洞察和优化SQL性能

冯光普 – DMALL数据库负责人

2 0 2 4 T a k i n T a l k s

About Me



冯光普

多点DMALL，数据库团队负责人

MySQL、TiDB、OB、Redis、MongoDB稳定性

DB PaaS平台，研发侧标准流程及快速交付

DB中间件，多活架构

更早，阿里巴巴数据库AliSQL团队

CONTENTS

- 01 慢SQL的局限性**
难以充分发现问题、阈值配置难
- 02 TopSQL原理及实现**
全量SQL统计、分析排序、图表展示
- 03 TopSQL性能诊断实践**
直观发现多种SQL性能问题
- 04 性能观测组合拳：慢SQL+TopSQL**
相辅相成、全面洞察、精准监控

业务背景

SQL性能问题，是生产数据库崩溃的头号因素



慢SQL

换道变多

应用建新连接

SQL running增长

MySQL CPU 100%

所有车道蠕行

应用重试SQL

导航上是深红色

MySQL崩溃

熄火打牌唠嗑

全面洞察SQL性能

到底是什么SQL导致了问题

大规模数据库 (1000+)

自动化工具、平台

高效支撑研发体系

研发可理解，能自助

方法及挑战

流程管控

开发 -> 测试 -> 生产
提前优化性能

监控告警

资源：CPU、Mem、IO、Net
QPS、InnoDB rows
连接数、活跃线程数
复制延迟
...

Slow Log分析

可定位到问题SQL
pt-query-digest
...

上线前完全规避问题，太过理想

SQL生成的，数据也是动态的...

无效

可观测

可判断

可优化

有效

Part one

慢SQL的局限性

难以充分发现问题、阈值配置难

回复“进群”
交流群与听众/分享者互动



MySQL Slow Log分析工具

pt-query-digest


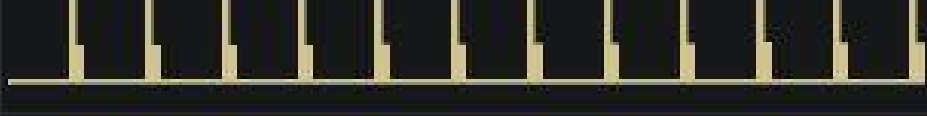
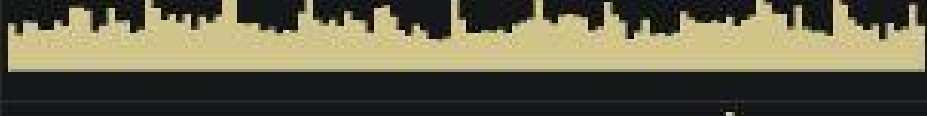

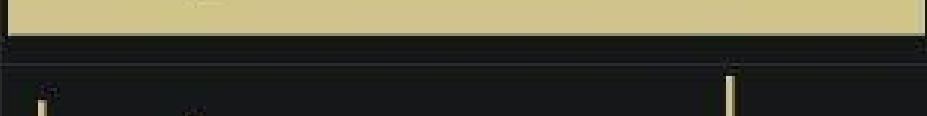

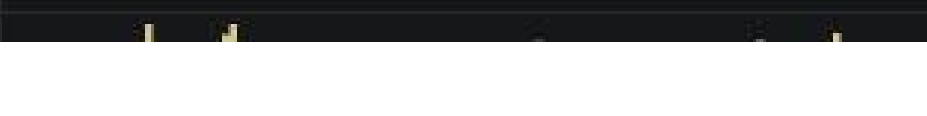
PMM Query Analytics

| Show / hide columns | | | | | | First | Previous | 1 | 2 | Next | Last | | |
|---------------------|-------|-------------|----------|------------|------------|---|-------------|-------------|----------|------|------|--|--|
| Checksum | Count | Total ms | Avg ms | First Seen | Last Seen | Query Fingerprint | Reviewed On | Reviewed By | Comments | | | | |
| 361364721930840825 | 15 | 231810.00 | 15453.99 | 2012-08-24 | 2012-08-31 | SELECT p.post_id FROM phpbb_posts p WHERE p.topic_id = ? AND p.post_approved = ? ORDER BY p.post_time LIMIT ? | | | | | | | |
| 4696394594996027782 | 40 | 486910.00 | 12172.82 | 2012-08-11 | 2012-08-31 | UPDATE phpbb_users SET user_lastvisit = ? WHERE user_id = ? | | | | | | | |
| 4724454430261861928 | 13 | 251590.00 | 19353.00 | 2012-08-14 | 2012-08-29 | SELECT COUNT(p.post_id) AS prev_posts FROM phpbb_posts p WHERE p.topic_id = ? AND p.post_approved = ? AND (p.post_time < ? OR (p.post_time = ? AND p.post_id <= ?) | | | | | | | |
| 5905094567261964269 | 26 | 674220.00 | 25931.45 | 2012-08-13 | 2012-08-30 | SELECT username, user_id, user_viewemail, user_posts, user_regdate, user_from, user_website, user_email, user_icq, user_aim, user_yim, user_msnm, user_avatar, user_avatar_type, user_allowavatar FROM mad_users WHERE user_id <> ? ORDER BY username LIMIT ? | | | | | | | |
| 6188895525816298663 | 38 | 514420.00 | 13537.40 | 2012-08-12 | 2012-08-31 | SELECT username FROM mad_users WHERE reduced_username = ? | | | | | | | |
| 6857733383566028078 | 158 | 2150090.00 | 13608.17 | 2012-08-10 | 2012-08-31 | UPDATE phpbb_sessions SET session_time = ?, session_last_visit = ?, session_admin = ?, session_page = ?, session_forum_id = ? WHERE session_id = ? | | | | | | | |
| 6911144973463760610 | 4079 | 84130780.00 | 20625.34 | 2012-08-09 | 2012-08-31 | (SELECT h.author_user_id AS author, h.message_id AS id, h.message_body AS message, h.message_date AS DATE, GROUP_CONCAT(u.username SEPARATOR ?) AS recipients FROM mad_chat_history AS h LEFT OUTER JOIN mad_chat_history_recipients AS hr ON hr.message_id = h.message_id INNER JOIN mad_users AS u ON u.user_id = hr.user_id WHERE h.private = ? AND h.author_user_id = ? GROUP BY id) UNION ALL (SELECT h.author_user_id AS author, h.message_id AS id, h.message_body AS message, h.message_date AS DATE, GROUP_CONCAT(u.username SEPARATOR ?) AS recipients FROM mad_chat_history AS h LEFT OUTER JOIN mad_chat_history_recipients AS hr ON hr.message_id = h.message_id INNER JOIN mad_chat_history_recipients AS hr ON hr ? message_id = h.message_id INNER JOIN mad_users AS u ON u.user_id = hr ? user_id WHERE h.private = ? AND hr.user_id = ? GROUP BY id) ORDER BY DATE DESC LIMIT ? | | | | | | | |
| 7428101489402929338 | 19 | 447890.00 | 23572.95 | 2012-08-10 | 2012-08-27 | SELECT COUNT(post_id) AS total FROM phpbb_posts WHERE post_time > ? | | | | | | | |
| 7467891370387641567 | 144 | 4381760.00 | 30428.92 | 2012-08-10 | 2012-08-31 | mysqldump | | | | | | | |
| 8318308310479237169 | 15 | 224620.00 | 14974.42 | 2012-08-13 | 2012-08-23 | SELECT u . * , s . * FROM phpbb_sessions s, phpbb_users u WHERE s.session_id = ? AND u.user_id = s.session_user_id | | | | | | | |

MySQL Slow Log分析工具

pt-query-digest

PMM Query Analytics

| # | Query | Load | Query Count | Query Time | |
|---|---|---|-------------|------------|----------|
| | TOTAL |  | 18.80 load | 7.52k QPS | 2.50 ms |
| 1 | UPDATE `sbtest1` SET `k` = `k` + ? WHERE `id` = ? |  | 5.25 load | 93.45 QPS | 56.18 ms |
| 2 | SELECT `c` FROM `sbtest1` WHERE `id` = ? |  | 2.49 load | 200.61 QPS | 12.40 ms |
| 3 | UPDATE `sbtest1` SET `k` = `k` + ? WHERE `id` = ? |  | 1.82 load | 144.07 QPS | 12.63 ms |
| 4 | update sbtest1 set k=k? where id=? |  | 1.70 load | 50.05 QPS | 34.00 ms |
| 5 | insert into michaelc.t1 values() |  | 0.66 load | 21.04 QPS | 31.24 ms |
| 6 | SELECT `c` FROM `sbtest1` WHERE `id` = ? |  | 0.41 load | 195.04 QPS | 2.08 ms |
| 7 | SELECT `c` FROM `sbtest1` WHERE `id` BETWEEN ? AND ? |  | 0.34 load | 20.06 QPS | 16.83 ms |
| 8 | SELECT SUM (`K`) FROM `sbtest1` WHERE `id` BETWEEN ? AND ? |  | 0.33 load | 20.06 QPS | 16.60 ms |
| 9 | SELECT `c` FROM `sbtest1` WHERE `id` BETWEEN ? AND ? ORDER BY `c` |  | 0.33 load | 20.06 QPS | 16.43 ms |

MySQL Slow Log的局限性

```
# Time: 2024-01-21T16:02:31.686154Z
# User@Host: dmall_sale[dmall_sale] @ [10.28.91.244] Id: 4904887
# Query_time: 0.221763 Lock_time: 0.000055 Rows_sent: 0 Rows_examined: 1276
SET timestamp=1705852951;
delete from sale_detail_his WHERE sv_no = '1000000113503914';
```

有限的指标

Query_time
Lock_time
Rows_sent
Rows_examined

难以发现所有的SQL性能问题

非全量SQL流量

仅记录超过long_query_time

大规模业务，阈值配置难度大

阈值配置与业务相关

难以标准化
业务持续迭代，动态变化

Part two

TopSQL原理及实现

全量SQL统计、分析排序、图表展示

帮助研发更高效定位问题，比解决问题更关键

回复“进群”
交流群与听众/分享者互动



TopSQL原理

0

数据输入

SQL流量、binlog

1

分析计算

协议解析、SQL指纹、指标统计

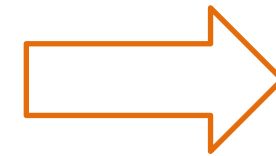
2

结果输出

指标存储、查询排序、图表展示

TopSQL实现

语法解析 (<https://github.com/pingcap/parser>)
MySQL协议 (<https://github.com/zr-hebo/sniffer-agent>)



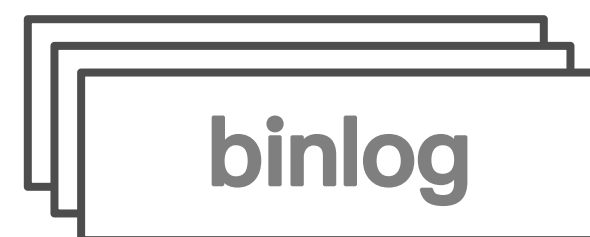
1. 旁路网络监听

2. MySQL协议解析

3. 语法解析

4. SQL指纹提取

5. 数据统计



1. Event解析

`binlog_rows_query_log_events`

2. SQL指纹提取

3. 更新模式识别

`diff(before, after)`

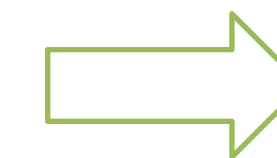
4. 数据统计



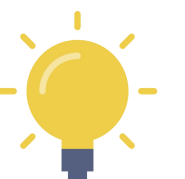
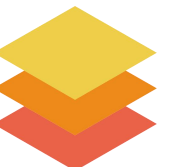
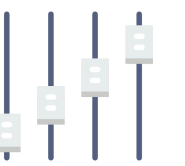
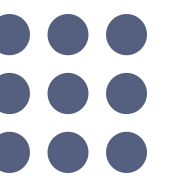
分析
计算

结果
存储

查询
聚合



页面
展示



Part three

TopSQL性能诊断实践

直观发现多种SQL性能问题

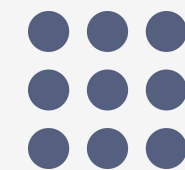
回复“进群”
交流群与听众/分享者互动



TopSQL能力

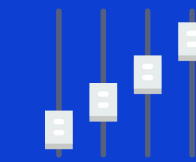
全量SQL模式

业务SQL模式全景图



Top排序展示

次数、总耗时、平均耗时、流量



热力图

直观发现热点表



物理更新统计

比对字段变化、定位IO瓶颈、
识别无效更新



TopSQL-总耗时

时间都去哪儿了?

概况

商品 _ware (在线)

登录数据库

集群运维维护

刷新

返回应用列表

基本信息

JDBC关闭连接SSL配置(useSSL=false)可以看到全量TOP SQL, SQL统计精度为30分钟

实例信息

实例 3306

时间区间 2024-02-28

18:30

19:00

搜索

单击以总耗时排序

监控

性能监控

NEW Top SQL

Slow SQL

中风险 巡检

管理

账号信息

库表信息

连接管理

错误日志

快照查询

工具

数据闪回

数据安全

| SQL指纹 | 执行总耗时 ↓ | 执行平均耗时 | SQL总流量(字节) | SQL执行总次数 | SQL执行时间最大值 | SQL执行时间最大值发生时间 |
|---|----------|----------|------------|----------|------------|---------------------|
| <code>select `id`, `item_num` as `itemnum`, `matnr` as `matnr` from `ware_matnr_item_num` where `vender_id` = ? and `yn` = ? and `ic`</code> | 24.47min | 1.68s | 706004219 | 876 | 4.55s | 2024-02-28 18:40:10 |
| <code>select `id`, `scale_code` as `scalecode` from `ware_scale_code` where `vender_id` = ? and `rf_id` = ? and `yn` = ? order by `id`</code> | 28.31s | 6.26ms | 1950592 | 4524 | 26.73s | 2024-02-28 18:54:08 |
| <code>select `r`.`id` as `id`, `r`.`vender_id` as `venderid`, `r`.`sku_id` as `skuid` from `dmall_ware`.`ware_sku_rule` as `r` where `r`.`s`</code> | 1.54s | 85.46ms | 8160870 | 18 | 106.26ms | 2024-02-28 18:30:13 |
| <code>select `id`, `cn_name` as `cnname`, `en_name` as `enname`, `first_char` as `firstchar`, `logo`, `company_info` as `companyinfo`</code> | 1.16s | 193.28ms | 10757486 | 6 | 201.68ms | 2024-02-28 18:47:00 |
| <code>select `id`, `item_num` as `itemnum` from `ware_matnr_item_num` where `vender_id` = ? and `matnr` = ? and `yn` = ? order by</code> | 1.08s | 240.55us | 1673713 | 4484 | 31.00ms | 2024-02-28 18:53:07 |
| <code>select `cm_cat_id` as `cmcatid`, `parent_id` as `parentid`, `cat_name` as `catname`, `cat_name_en` as `catnameen`, `cat_image`</code> | 458.33ms | 4.88ms | 182973 | 94 | 8.80ms | 2024-02-28 18:33:17 |
| <code>select `b`.`id` as `id`, `b`.`vender_id` as `venderid`, `b`.`vender_name` as `vendername`, `b`.`start_time` as `starttime`, `b`.`c`</code> | 385.52ms | 64.25ms | 14460 | 6 | 97.25ms | 2024-02-28 18:40:36 |
| <code>select `b`.`id` as `id`, `b`.`vender_id` as `venderid`, `b`.`vender_name` as `vendername`, `b`.`start_time` as `starttime`, `b`.`c`</code> | 313.51ms | 52.25ms | 14484 | 6 | 60.48ms | 2024-02-28 18:40:52 |
| <code>select `id`, `locnr_id` as `locnrid`, `shop_id` as `shopid`, `locnr`, `matnr`, `mstae`, `mmsta`, `mmsta_time` as `mmstatime`, `mst`</code> | 101.79ms | 90.16us | 4220396 | 1129 | 2.12ms | 2024-02-28 18:48:19 |

案例: 业务数据库cpu利用率高, 同时网络IO流量高, 慢SQL中仅有少数几条记录, 难以定位原因

TopSQL-平均耗时

慢SQL阈值到底应该设置为多少?

- 概况
- 基本信息
- 实例信息
- 监控
- 性能监控
- NEW Top SQL
- Slow SQL
- 巡检
- 管理
- 账号信息
- 库表信息
- 连接管理
- 错误日志
- 快照查询
- 工具
- 数据闪回
- 数据安全

采销供 _data_mart (在线)

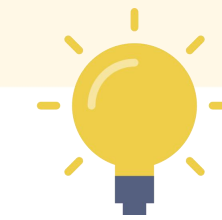
登录数据库

集群运维维护

刷新

返回应用列表

JDBC关闭连接SSL配置(useSSL=false)可以看到全量TOP SQL, SQL统计精度为30分钟



可能的问题时间点

实例: :4134 时间区间: 2024-02-29 17:00 17:30 搜索

单击以平均耗时排序

| SQL指纹 | 执行总耗时 | 执行平均耗时 ↓ | SQL总流量(字节) | SQL执行总次数 | SQL执行时间最大值 | SQL执行时间最大值发生时间 |
|---|--------|----------|------------|----------|------------|---------------------|
| <code>select sum(`sale_qty`) as `sale_qty_sum`, sum(`cost_unamt`) as `cost_unamt_sum`, sum(`cost_amt`) as `cost_amt_sum`, sum(`s</code> | 30.00s | 30.00s | 1369 | 1 | 30.00s | 2024-02-29 17:10:30 |
| <code>select count(?) from `sale_shop_goods_202202` where `sale_date` between ? and ? and `supplier_code` in (...) and `yn` = ? and `ve</code> | 30.00s | 30.00s | 396 | 1 | 30.00s | 2024-02-29 17:10:30 |
| <code>select count(?) from `shop_goods_sale_stock_019` where `vendor_id` = ? and `supplier_code` in (?) and `day` between ? and ?</code> | 21.56s | 10.78s | 420 | 2 | 16.55s | 2024-02-29 17:02:20 |
| <code>select sum(`sale_qty`) as `sale_qty_sum`, sum(`sale_amt`) as `sale_amt_sum`, sum(`stock_qty`) as `stock_qty_sum`, sum(`stock</code> | 21.14s | 10.57s | 1250 | 2 | 16.14s | 2024-02-29 17:02:27 |
| <code>select count(?) from `shop_goods_sale_stock_032` where `vendor_id` = ? and `supplier_code` in (?) and `day` between ? and ?</code> | 28.83s | 9.61s | 630 | 3 | 17.05s | 2024-02-29 17:13:25 |
| <code>select sum(`sale_qty`) as `sale_qty_sum`, sum(`sale_amt`) as `sale_amt_sum`, sum(`stock_qty`) as `stock_qty_sum`, sum(`stock</code> | 27.80s | 9.27s | 1881 | 3 | 16.03s | 2024-02-29 17:13:27 |
| <code>select `id`, `vendor_id`, `supplier_code`, `supplier_name`, `day`, `shop_code`, `shop_name`, `goods_code`, `goods_name`, `sale_qty`,</code> | 17.10s | 8.55s | 5061988 | 2 | 8.56s | 2024-02-29 17:16:32 |
| <code>select `id`, `vendor_id`, `supplier_code`, `supplier_name`, `day`, `shop_code`, `shop_name`, `goods_code`, `goods_name`, `sale_qty`,</code> | 25.16s | 8.39s | 15255 | 3 | 8.47s | 2024-02-29 17:13:33 |
| <code>select `id`, `vendor_id`, `supplier_code`, `supplier_name`, `day`, `shop_code`, `shop_name`, `goods_code`, `goods_name`, `sale_qty`,</code> | 7.47s | 7.47s | 1435110 | 1 | 7.47s | 2024-02-29 17:03:08 |

案例: 统计分析类AP业务, SQL虽然慢, 但频次低, 慢SQL阈值需调整到30s+

TopSQL-网络流量 (出)

哪个SQL传输了最多的数据?

概况

交易 mall_prevention (在线)

登录数据库 集群运维维护 刷新 返回应用列表

基本信息

JDBC关闭连接SSL配置(useSSL=false)可以看到全量TOP SQL, SQL统计精度为30分钟

实例信息

实例 3465 时间区间 2024-02-28 17:00 17:30 搜索

单击以流量排序

监控

性能监控

Top SQL

Slow SQL

中风险 巡检

管理

账号信息

库表信息

连接管理

错误日志

快照查询

工具

数据闪回

数据安全

| SQL指纹 | 执行总耗时 | 执行平均耗时 | SQL总流量(字节) ↓ | SQL执行总次数 | SQL执行时间最大值 | SQL执行时间最大值发生时间 |
|---|--------|----------|--------------|----------|------------|---------------------|
| <code>select `id`, `order_id`, `prev_order_type`, `order_user_id`, `order_user_phone`, `order_user_phone_suffix`, `prev_user_id`, `order_s</code> | 6.28s | 103.16us | 225468526 | 60926 | 4.22ms | 2024-02-28 17:15:00 |
| <code>select `id`, `task_key` as `taskkey`, `business_no` as `businessno`, `task_type` as `tasktype`, `param`, `task_name` as `taskname`,</code> | 65.18s | 1.02ms | 154573898 | 63798 | 10.91ms | 2024-02-28 17:15:00 |
| <code>select @@session.tx_read_only</code> | 13.99s | 22.82us | 65604910 | 613130 | 12.51ms | 2024-02-28 17:00:00 |
| <code>insert into `order_index` (`order_id`, `prev_order_type`, `order_user_id`, `order_user_phone`, `order_user_phone_suffix`, `prev_use</code> | 22.60s | 371.02us | 60898694 | 60924 | 57.89ms | 2024-02-28 17:13:48 |
| <code>update `order_index` set `order_id` = ?, `order_user_id` = ?, `order_status` = (case when `order_status` <= ? then ? else `order_st</code> | 15.49s | 770.27us | 42385051 | 20112 | 46.64ms | 2024-02-28 17:16:52 |
| <code>select * from `prevention_blacklist` where `prevention_blacklist`.`yn` = ? and `prevention_blacklist`.`order_user_id` = ? and `preve</code> | 1.50s | 74.35us | 42110845 | 20221 | 4.34ms | 2024-02-28 17:01:08 |
| <code>update `order_index` set `order_id` = ?, `order_user_id` = ?, `prev_user_id` = ?, `order_status` = (case when `order_status` <= ? ti</code> | 17.08s | 846.78us | 38337484 | 20173 | 46.54ms | 2024-02-28 17:22:59 |
| <code>select `id`, `env_type` as `envtype`, `task_key` as `taskkey` from `prevention_task` where (`task_owner` is ? or `task_owner` = ?) a</code> | 31.33s | 491.13us | 33869456 | 63798 | 7.66ms | 2024-02-28 17:10:00 |
| <code>insert into `prevention_task` (`task_key`, `business_no`, `task_type`, `param`, `task_name`, `remark`, `execute_times`, `intervals`,</code> | 22.69s | 537.16us | 28943200 | 42249 | 47.59ms | 2024-02-28 17:08:42 |

案例: 业务SQL动态生成, 针对性裁剪掉不必要的返回字段 (TEXT、BLOB), 降低IO消耗, 提升性能



诊断实践

热点表

IO瓶颈SQL



降低感知问题的门槛，就是解决问题

- 哪些表IO压力大，DML分布情况？
- 高IO消耗SQL可能是哪些？
- 分表后，压力是否均匀？
- 哪些字段真实被更新？
- 按行数排序SQL模式？



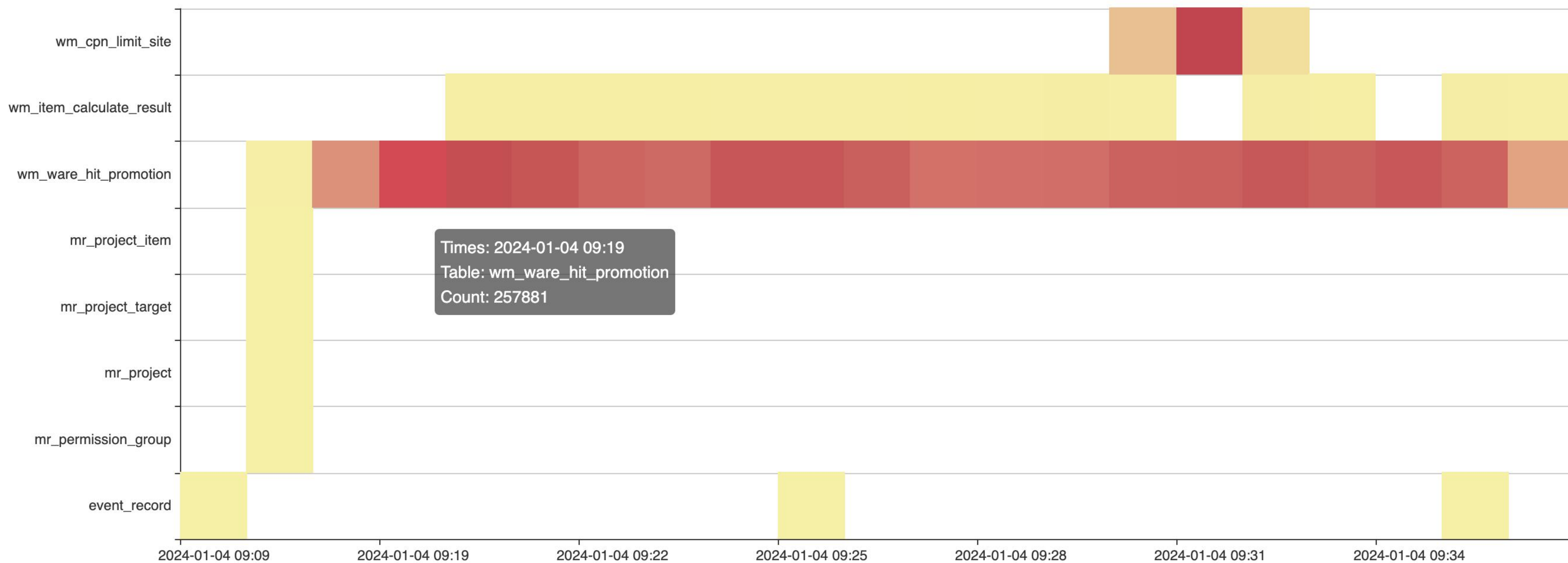
TopSQL-热力图1

哪些表上新增了更多的数据?

分钟总览 表级热力图 物理更新统计 SQL指纹

insert

表级热力图



Times: 2024-01-04 09:19
Table: wm_ware_hit_promotion
Count: 257881

1 257881 257881

TopSQL-物理更新识别

哪些字段被更新了

分钟总览 表级热力图 物理更新统计 SQL指纹

| 表名 | 更新字段 | 更新次数 | binlog长度 |
|---------------------------|---|---------|----------|
| store_score | modified,created | 3757125 | 4.31GB |
| realtime_operate_overview | modified | 574598 | 257.74MB |
| realtime_trade_overview | modified | 494755 | 151.00MB |
| minios_ware_weeksales_top | sku_id,matnr,ware_name,modified | 190146 | 59.06MB |
| realtime_ware_cat_sales | modified | 155840 | 59.33MB |
| realtime_operate_overview | ware_total_amount,ware_num,order_cnt,modified | 145274 | 65.74MB |
| realtime_ware_cat_sales | ware_total_amount,ware_num,order_cnt,gross_profit,modified | 67957 | 25.88MB |
| minios_ware_weeksales_top | modified | 39137 | 12.15MB |
| store_score | onsale_ware_cnt,modified,created | 38957 | 44.80MB |
| realtime_operate_overview | ware_total_amount,coupon_amount,ware_num,order_cnt,modified | 35823 | 16.23MB |

识别无效更新

案例：业务字段value实际没有改变，产生了大量垃圾数据，造成数据同步（基于binlog）到下游延迟

TopSQL-IO大户诊断

按更新行数排序

分钟总览 表级热力图 物理更新统计 SQL指纹

| 表名 | SQL指纹 | 执行次数 | 影响行数 |
|-----------------------------|--|------|---------|
| growth_bill_054 | UPDATE `growth_bill_054` SET `vendor_id`=? | 1 | 8784429 |
| statistics_realtime_consume | UPDATE `statistics_realtime_consume` SET `anonymity_count`=?, `modified`=NOW(), `yn`=? WHERE `id`=? | 245 | 245 |
| statistics_realtime_consume | UPDATE `statistics_realtime_consume` SET `member_count`=?, `modified`=NOW(), `yn`=? WHERE `id`=? | 63 | 63 |
| user_login_history | UPDATE `user_login_history` SET `last_login_time`=?, `last_login_ip`=?, `last_login_ua`=?, `last_login_source`=?, `modified`=NOW(), `yn`=? WHERE `id`=? | 63 | 63 |
| statistics_realtime_member | UPDATE `statistics_realtime_member` SET `order_created_count`=?, `modified`=NOW(), `yn`=? WHERE `id`=? | 60 | 60 |
| statistics_realtime_member | UPDATE `statistics_realtime_member` SET `new_user_count`=?, `modified`=NOW(), `yn`=? WHERE `id`=? | 12 | 12 |
| composite_user_04 | UPDATE `composite_user_04` SET `user_id`=?, `system_code`=?, `instance_code`=?, `instance_name`=?, `state`=?, `vendor_id`=?, `properties`=?, `modified`=NOW(), `yn`=? WHERE `id`=? | 5 | 5 |
| paid_up_member_task | UPDATE `paid_up_member_task` SET `type`=?, `state`=?, `times`=?, `task_id`=?, `modified`=NOW(), `yn`=? WHERE `id`=? | 4 | 4 |
| wechat_marketing_card | UPDATE `wechat_marketing_card` SET `extra_data`=?, `syn_time`=?, `modified`=NOW(), `yn`=? WHERE `id`=? | 3 | 3 |
| user_login_history | UPDATE `user_login_history` SET `last_login_time`=?, `last_login_ip`=?, `last_login_ua`=?, `modified`=NOW(), `yn`=? WHERE `id`=? | 2 | 2 |

IO消耗大户

案例：update SQL动态拼接，漏了其他条件，导致大量数据被更新，数据库IO压力高

Part four

组合拳：慢SQL+TopSQL

相辅相成、全面洞察、精准监控

回复“进群”
交流群与听众/分享者互动



MySQL性能观测组合拳

慢SQL

研发:

订阅慢SQL告警
SQL性能诊断依赖DBA

DBA:

管理慢SQL阈值困难
大量SQL性能诊断优化工作

TopSQL

研发:

SQL性能诊断不依赖DBA
直观 & 高效

DBA:

卸载**90%+** SQL性能诊断
每DBA服务研发人数++

慢SQL+TopSQL

研发:

订阅慢SQL告警
自助诊断优化SQL性能

DBA:

轻松管理慢SQL阈值
运维更大规模MySQL, **10K级**

业务发展前期

服务大规模研发体系

支撑大规模MySQL

CONCLUSION SHARE 总结分享



慢SQL的局限

覆盖不全、指标有限、配置难

TopSQL的原理、实现、实践

多维度Top排序、热力图、IO瓶颈分析

慢SQL + TopSQL

互为补充、全面洞察SQL性能、研发自助

研发自助定位SQL性能问题，DBA运维更大规模MySQL集群

THANK YOU

谢谢你的观看~

冯光普 – DMALL数据库负责人

回复“进群”
交流群与听众/分享者互动

