



第九届中国系统架构师大会
SYSTEM ARCHITECT CONFERENCE CHINA 2017

58速运数据库降压优化实践

58沈剑

关于-我

- 百度 - 高级工程师
- 58同城 - 高级架构师，技术委员会主席，技术学院优秀讲师
- 58到家 - 高级技术总监，技术委员会主席
- 58速运 - CTO
- “架构师之路”作者，深夜写写技术文章
- 本质：技术人一枚



目录

- DBA的定位
- DB优化方法
- 58速运优化实践
- 总结

SACC2017

一、DBA的定位

DBA的职责

- **设计与定义**数据库系统
- **监控**数据库系统的运行
- **线上数据操作**与维护
- 数据**安全性**保障
- 数据库系统**性能调优**
- ...

SACC2017

实际情况

- **数据库设计**：执行研发哥哥提交的建库/建表/建索引语句，提出质疑时，被通知今天要上线
- **监控**：研发哥哥把数据库玩挂掉的时候，被质问为什么没有报警
- **线上操作**：执行研发哥哥提交的增删查改语句，提出质疑时，被催促业务需求紧急
- **安全性**：研发哥哥发现执行错了，被要求回滚数据，还被质疑恢复数据慢
- **性能调优**：被研发哥哥要求加内存，加连接数，加从库（心里想着这么低的并发，动不动就是join，批量扫库，加多少从库也没用）
- ...

DBA千万不要沦为“工单执行工具”

业务DBA，从专业的角度带给业务价值

业务DBA

- **数据库设计**：业务DBA，从专业的角度，帮助研发做好早期设计
- **线上操作**：业务DBA，需要了解被执行工单的业务背景，来龙去脉，做好把关
- **性能调优**：业务DBA，从专业的角度，结合业务进行优化，给出优化建议
- ...

研发哥哥：好，不是你的锅，你说我怎么改？

二、业务DBA系统优化方法

找主要矛盾，针对性优化

怎么找主要矛盾？

找到主要矛盾，要量化

- **哪些是主要矛盾？**

- (1) 执行次数多的SQL
- (2) 执行时间长的SQL
- (3) 扫描行数多的SQL

- **怎么获取这些数据？**

- (1) 获取所有SQL，执行时间
- (2) 按照某个维度排序
- (3) 得到主要矛盾

落地技巧

- Mysql分析工具：**pt-query-digest**，用于分析慢查询
- **把慢SQL的时间设为0**，从slowlog中获取所有SQL的相关信息，对性能的影响在10%以内
- 可以获取**总体分析结果**，**分组排序的分析结果**，**单Query ID的分析结果**

主要矛盾分析

- 按照总执行时间【time】排序
- 需要重点关注调用次数【Calls】和单次调用时间【R/Call】
- 通过【Query ID】可以得到具体SQL语句
- 通过【explain】，可以得到具体SQL的【索引命中】情况以及【扫描行数】情况

#	Rank	Query ID	Response	time	Calls	R/Call	V/M	Item	
1	#	1	0x88802F8ED1258B55	8365.1346	19.2%	4234	1.9757	0.01	SELECT t_driver_census
2	#	2	0x80EDAA0F880B5181	7112.7886	16.4%	271680	0.0262	0.00	SELECT t_app_user_history_info
3	#	3	0x40E39FC5E2D41BA0	6965.3222	16.0%	286	24.3543	0.21	SELECT t_driver_allowance
4	#	4	0xC75E2842C533E27A	4695.2653	10.8%	2172	2.1617	0.11	SELECT t_driver_allowance_log
5	#	5	0xFEB0D11B78D2A9EB	3840.4539	8.8%	286	13.4282	0.04	SELECT t_driver_allowance
6	#	6	0x442B5532969A6C0E	1989.3281	4.6%	16222	0.1226	0.00	SELECT t_app_user_line
7	#	7	0x4422740EF9104C70	1530.9208	3.5%	2502733	0.0006	0.00	SELECT t_app_announcement
8	#	8	0xD3A45D0937CF7D32	716.1940	1.6%	341859	0.0021	0.00	SELECT t_app_product
9	#	9	0x90556938050BB9CD9	605.9973	1.4%	748	0.8102	0.01	SELECT t_pusher_tradelog
10	#	10	0x3391A9BB24F91ABB	508.9085	1.2%	3591975	0.0001	0.00	SELECT t_app_text_config
11	#	11	0xE3D3A73EF3E3C369	474.2242	1.1%	1972039	0.0002	0.00	SELECT t_app_allowance_distace
12	#	12	0x510154CBD44A5685	470.9233	1.1%	32	14.7164	7.64	SELECT t_app_code t_app_pusher
13	#	13	0x2623A3F21D979DBD	363.4323	0.8%	5	72.6865	50.07	SELECT t_app_log

怎么针对性优化？

三、58速运优化实践

扫描行数多

- WHERE uid **not in** (select userid from t_app_pusher)
- WHERE note LIKE '**%XX**'
- WHERE **YEAR(date)** <= '2017'
- WHERE state IN (0, 1) AND createtime >= '**2016011-01**'
- 建立20+以上的索引，删除无效索引30+

#	Rank	Query ID	Response	time	Calls	R/Call	V/M	Item
1	#	1	0x88802F8ED1258B55	8365.1346	19.2%	4234	1.9757	0.01
2	#	2	0x80EDAA0F880B5181	7112.7886	16.4%	271680	0.0262	0.00
3	#	3	0x40E39FC5E2D41BA0	6965.3222	16.0%	286	24.3543	0.21
4	#	4	0xC75E2842C533E27A	4695.2653	10.8%	2172	2.1617	0.11
5	#	5	0xFEB0D11B78D2A9EB	3840.4539	8.8%	286	13.4282	0.04
6	#	6	0x442B5532969A6C0E	1989.3281	4.6%	16222	0.1226	0.00
7	#	7	0x4422740EF9104C70	1530.9208	3.5%	2502733	0.0006	0.00
8	#	8	0xD3A45D0937CF7D32	716.1940	1.6%	341859	0.0021	0.00
9	#	9	0x90556938050BBCD9	605.9973	1.4%	748	0.8102	0.01
10	#	10	0x3391A9BB24F91ABB	508.9085	1.2%	3591975	0.0001	0.00
11	#	11	0xE3D3A73EF3E3C369	474.2242	1.1%	1972039	0.0002	0.00
12	#	12	0x510154CBD44A5685	470.9233	1.1%	32	14.7164	7.64
13	#	13	0x2623A3F21D979DBD	363.4323	0.8%	5	72.6865	50.07
14	#	14						SELECT t_driver_census
15	#	15						SELECT t_app_user_history_info
								SELECT t_driver_allowance
								SELECT t_driver_allowance_log
								SELECT t_driver_allowance
								SELECT t_app_user_line
								SELECT t_app_announcement
								SELECT t_app_product
								SELECT t_pusher_tradelog
								SELECT t_app_text_config
								SELECT t_app_allowance_distace
								SELECT t_app_code t_app_pusher
								SELECT t_app_log

处理时间长

- WHERE phone=13800001234
- SELECT * FROM order WHERE date < = CURDATE()

优化为：

```
$curDate = date('Y-m-d');
```

```
$res = mysql_query(
```

```
    ' SELECT * FROM order WHERE date < = $curDate');
```

#	#	Rank	Query ID	Response	time	Calls	R/Call	V/M	Item
1	#	1	0x88802F8ED1258B55	8365.1346	19.2%	4234	1.9757	0.01	SELECT t_driver_census
2	#	2	0x80EDAA0F880B5181	7112.7886	16.4%	271680	0.0262	0.00	SELECT t_app_user_history_info
3	#	3	0x40E39FC5E2D41BA0	6965.3222	16.0%	286	24.3543	0.21	SELECT t_driver_allowance
4	#	4	0xC75E2842C533E27A	4695.2653	10.8%	2172	2.1617	0.11	SELECT t_driver_allowance_log
5	#	5	0xFEB0D11B78D2A9EB	3840.4539	8.8%	286	13.4282	0.04	SELECT t_driver_allowance
6	#	6	0x442B5532969A6C0E	1989.3281	4.6%	16222	0.1226	0.00	SELECT t_app_user_line
7	#	7	0x4422740EF9104C70	1530.9208	3.5%	2502733	0.0006	0.00	SELECT t_app_announcement
8	#	8	0xD3A45D0937CF7D32	716.1940	1.6%	341859	0.0021	0.00	SELECT t_app_product
9	#	9	0x90556938050BBCD9	605.9973	1.4%	748	0.8102	0.01	SELECT t_pusher_tradelog
10	#	10	0x3391A9BB24F91ABB	508.9085	1.2%	3591975	0.0001	0.00	SELECT t_app_text_config
11	#	11	0xE3D3A73EF3E3C369	474.2242	1.1%	1972039	0.0002	0.00	SELECT t_app_allowance_distace
12	#	12	0x510154CBD44A5685	470.9233	1.1%	32	14.7164	7.64	SELECT t_app_code t_app_pusher
13	#	13	0x2623A3F21D979DBD	363.4323	0.8%	5	72.6865	50.07	SELECT t_app_log

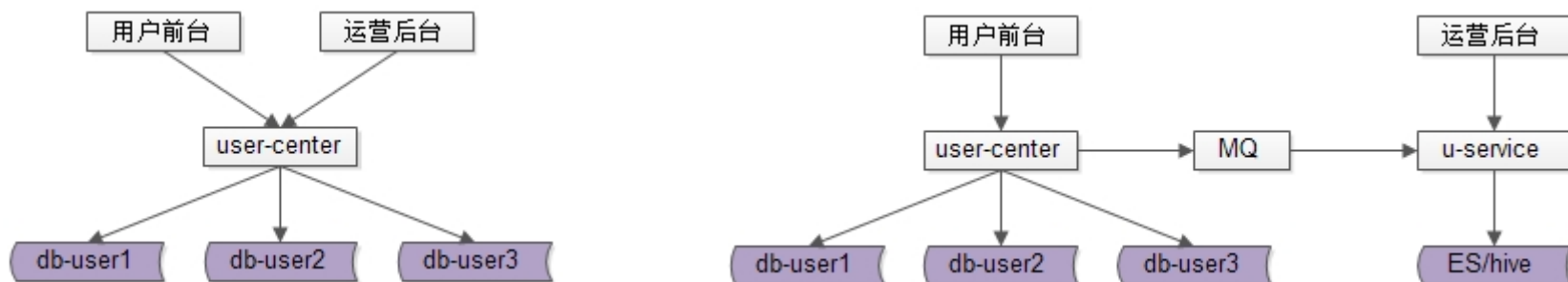
访问次数多

- 可以发现TOP-SQL中都是读请求
- 增加缓存能大大降低数据库压力
 - (1) 用户缓存
 - (2) 司机缓存
 - (3) 推送订单缓存
- 为什么没有加缓存？

#	#	Rank	Query ID	Response	time	Calls	R/Call	V/M	Item
1	#	1	0x88802F8ED1258B55	8365.1346	19.2%	4234	1.9757	0.01	SELECT t_driver_census
2	#	2	0x80EDAA0F880B5181	7112.7886	16.4%	271680	0.0262	0.00	SELECT t_app_user_history_info
3	#	3	0x40E39FC5E2D41BA0	6965.3222	16.0%	286	24.3543	0.21	SELECT t_driver_allowance
4	#	4	0xC75E2842C533E27A	4695.2653	10.8%	2172	2.1617	0.11	SELECT t_driver_allowance_log
5	#	5	0xFEB0D11B78D2A9EB	3840.4539	8.8%	286	13.4282	0.04	SELECT t_driver_allowance
6	#	6	0x442B5532969A6C0E	1989.3281	4.6%	16222	0.1226	0.00	SELECT t_app_user_line
7	#	7	0x4422740EF9104C70	1530.9208	3.5%	2502733	0.0006	0.00	SELECT t_app_announcement
8	#	8	0xD3A45D0937CF7D32	716.1940	1.6%	341859	0.0021	0.00	SELECT t_app_product
9	#	9	0x90556938050BBCD9	605.9973	1.4%	748	0.8102	0.01	SELECT t_pusher_tradelog
10	#	10	0x3391A9BB24F91ABB	508.9085	1.2%	3591975	0.0001	0.00	SELECT t_app_text_config
11	#	11	0xE3D3A73EF3E3C369	474.2242	1.1%	1972039	0.0002	0.00	SELECT t_app_allowance_distace
12	#	12	0x510154CBD44A5685	470.9233	1.1%	32	14.7164	7.64	SELECT t_app_code t_app_pusher
13	#	13	0x2623A3F21D979DBD	363.4323	0.8%	5	72.6865	50.07	SELECT t_app_log

为什么调用次数Calls差异这么大

- 前台访问量大，查询条件简单，返回数据量小
- 后台访问量少，查询条件复杂，返回批量数据
- 架构优化，前台与后台架构分离



举个写请求的例子

- 写操作耗时不长，但大量单条写操作容易引发大量磁盘IO
- 司机地理位置，经纬度存储，以方便计算路径
- 数据库存储
- 批量数据库存储，降低写压力
- 用hive/ES存储，降低数据库压力

数据库瓶颈？

- 这个锅，DBA不背
- 数据库QPS低于千级别，就有瓶颈，八成是研发哥哥的问题（索引，代码，架构）
- 改进推不动？就直接告诉他问题在哪里

四、总结

总结

- **pt-query-digest** , 慢SQL设为0
- **获取分组排序分析结果** , 得到**次数多** , **耗时长** , **行数多**的主要矛盾SQL
- **针对主要矛盾进行优化**
 - (1) 索引添加
 - (2) SQL优化
 - (3) 缓存增加
 - (4) 架构优化
- **观念的改变** : 由 “工单型DBA” 转变为 “业务伙伴DBA”

“架构师之路” 公众号



Q&A

谢谢！

THANKS

The background features a dark, almost black space filled with intricate patterns of glowing blue particles. These particles form dense, swirling trails that resemble nebulae or data streams. A bright, white-to-blue gradient light source is positioned behind the word 'THANKS', creating a lens flare effect that illuminates the surrounding particle trails. The overall aesthetic is futuristic and digital.