MYSQL在兴业数金的优化实践

兴业数金
CIB FINTECH

个人介绍

ORACLE WDP OCM讲师

DB2CHINA性能调优版版主

兴业数金首席DBA

QQ:1819442969

服务覆盖全国——国内最大的银行信息系统云服务平台

NO.1

全国最大的核心系统托管服务商

400+

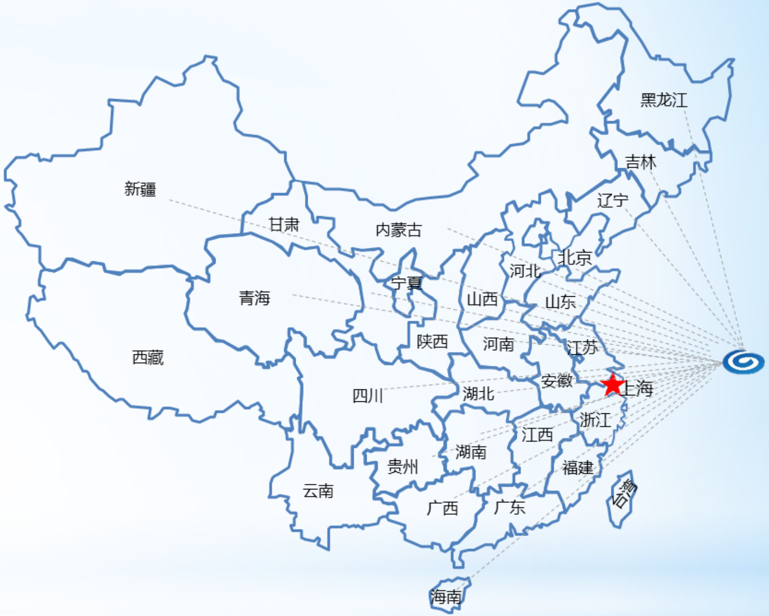
超过400项全方位金融行业云服务

358家

其中城商行17家、村镇银行332家、民营银行7家、外资银行2家

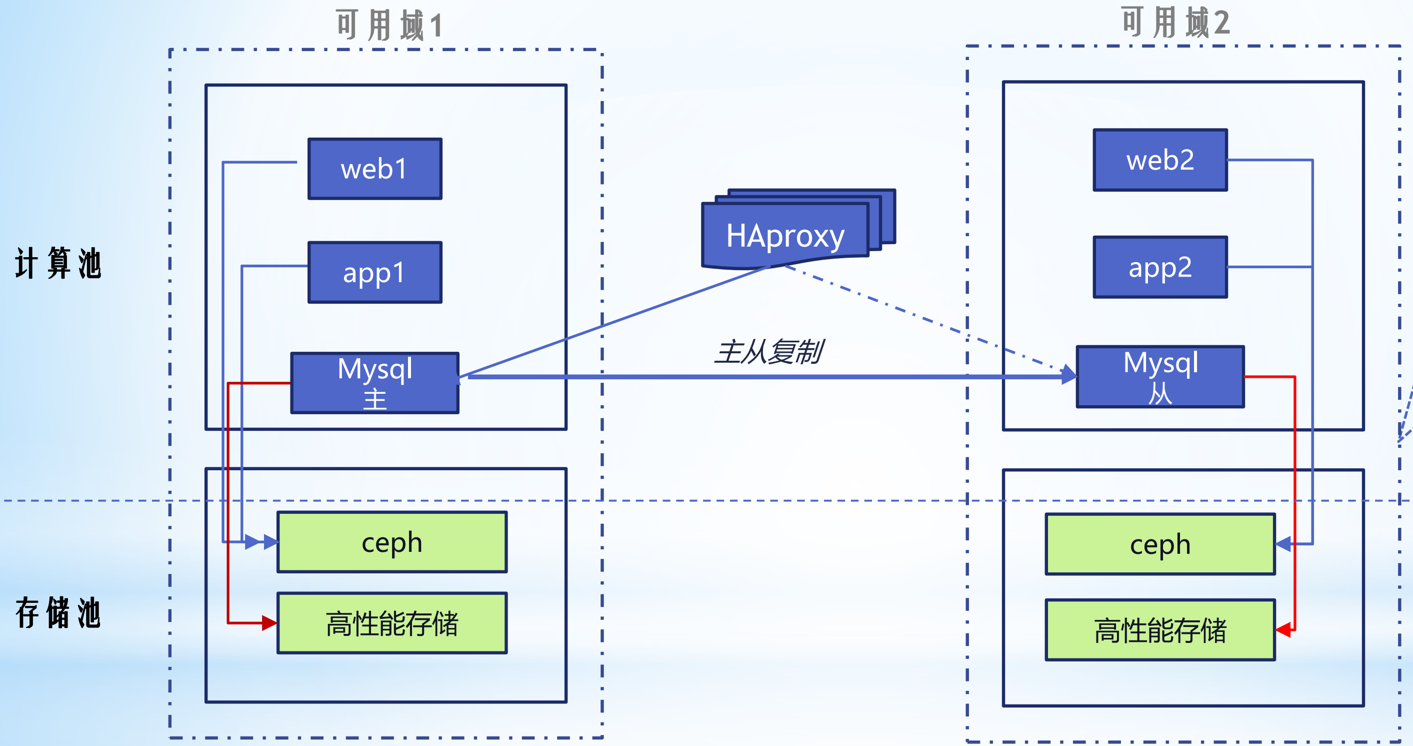
213家

213家已上线或启动实施



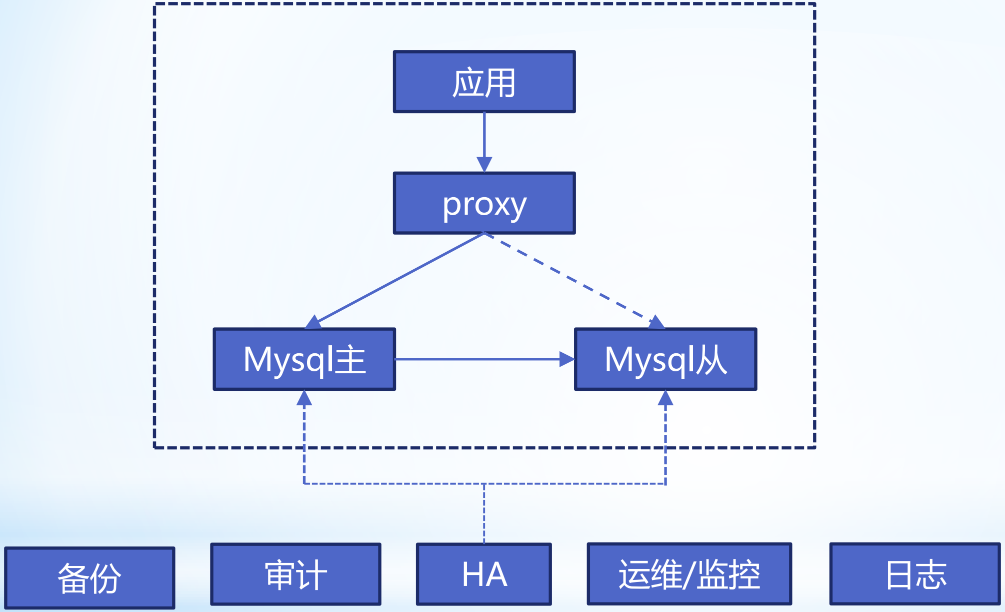
业务已覆盖**26个**省、直辖市、自治区

基于云平台架构演进的Mysql设计



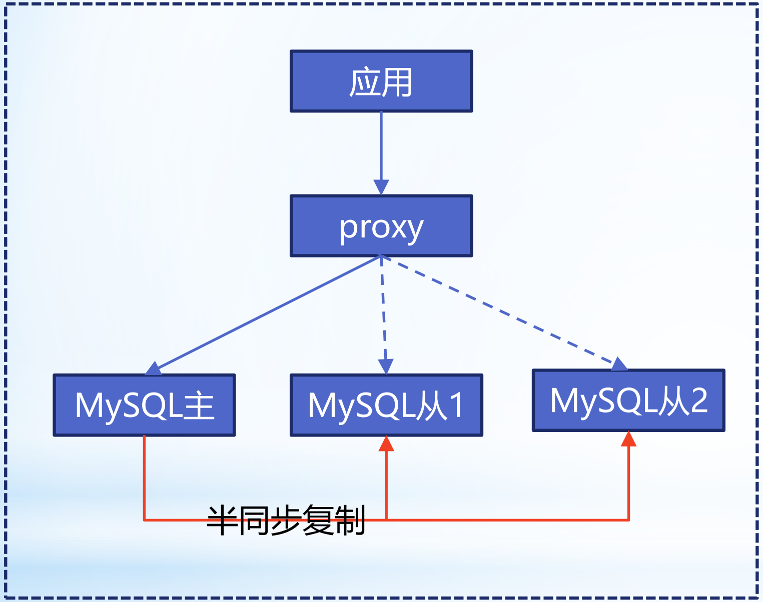
- 可用域之间是独立的资源池，**最大不相关**
- 应用跨可用域部署
- 数据库通过**主从复制**、**HA**组件实现
- 规避池级别的故障，进一步提升**高可靠性**
- **银行云100%的应用**

Mysql HA服务云化



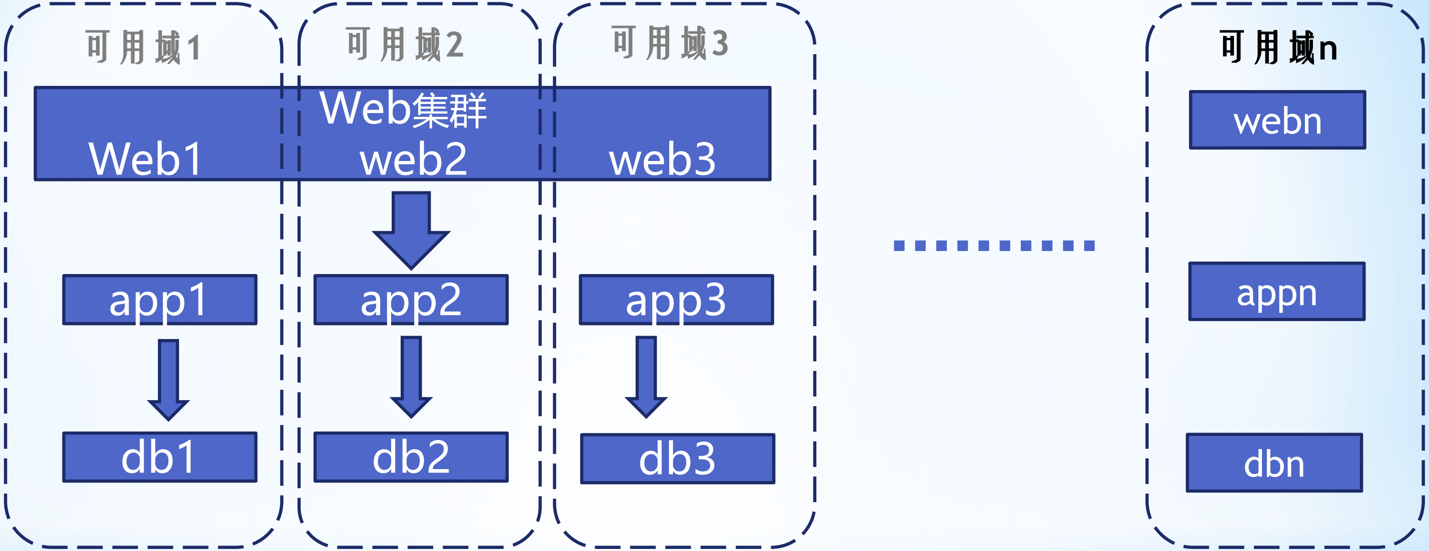
- **Mysql HA**服务标准化、云化，作为云服务的一个产品
- **数据库服务**方便可得，但云内实现了HA切换、备份、监控等多种功能，解决了传统数据库运维中60-80%的工作量
- 对不熟悉MySQL的客户提供快速上线的可能

Mysql HA服务云化



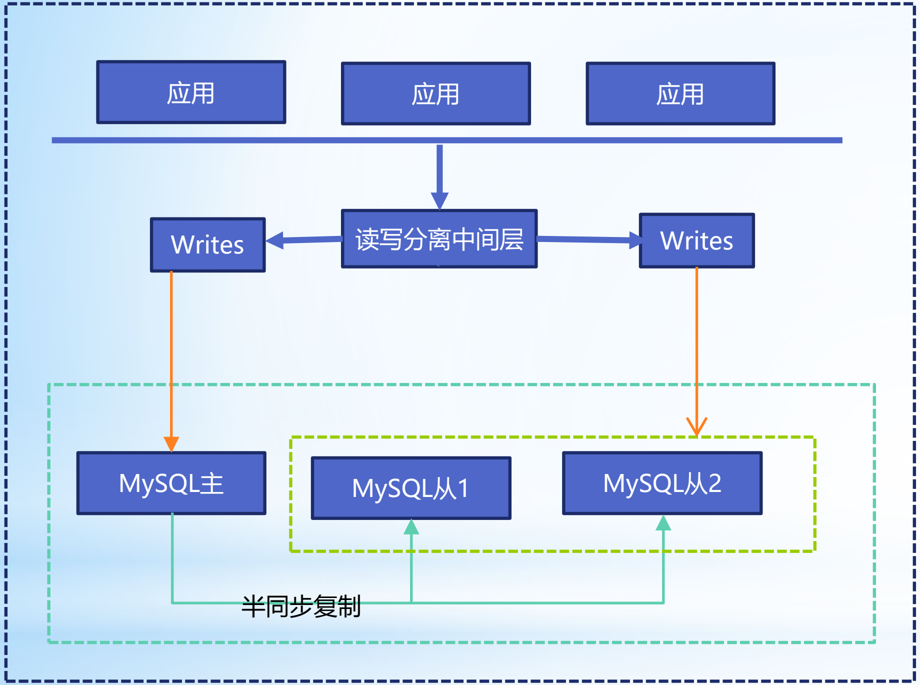
- **一致性保障：**一主两从半同步复制架构，确保数据零丢失
- **备份：**数据自动备份，支持全量增量等备份方式
- **运维监控：**自动化运维管理监控，大幅度提高运维效率
- **日志分析：**可视化日志分析，帮助DBA快速定位SQL问题

云上Mysql读写分离、分库分表



特点：mysql天生适应云的架构，适应横向扩展
读写分离、分库分表无法云服务标准化

MySQL 读写分离



（1）读写分离：

支持自动化读写分离，在主机写，多个从机读，分散主机压力，对应用透明业务无需调整，无需额外修改应用程序，没有迁移带来的开发成本。

（2）热扩容：

生产环境中在线扩容MySQL从机，不影响业务

（3）事务可强制读主库

（4）根据中间层对业务SQL的统计，可按实际情况调整主从数据库的比例

（5）高性能：

支持分布式部署，构建高可用架构，中间层的性能损耗低于5%，且支持基于MySQL5.7版本的高性能读写分离

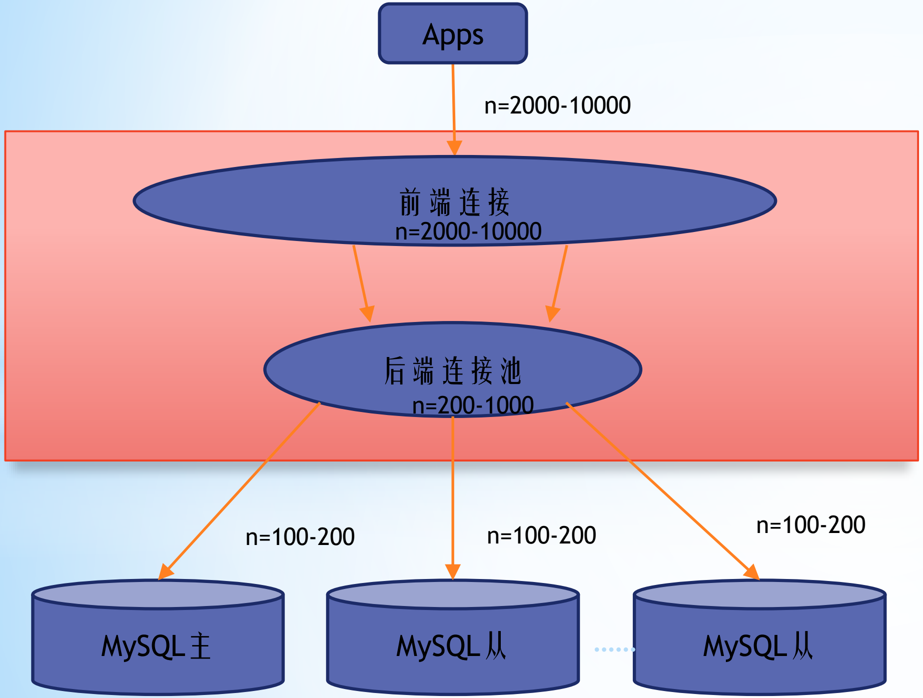
（6）读延迟检测

业务维度的延迟检测，读库延迟阀值自定义，超出阀值的从机集群软件将自动剔除集群，待数据延迟小于阀值时自动加入集群，并自动负载对外提供读服务

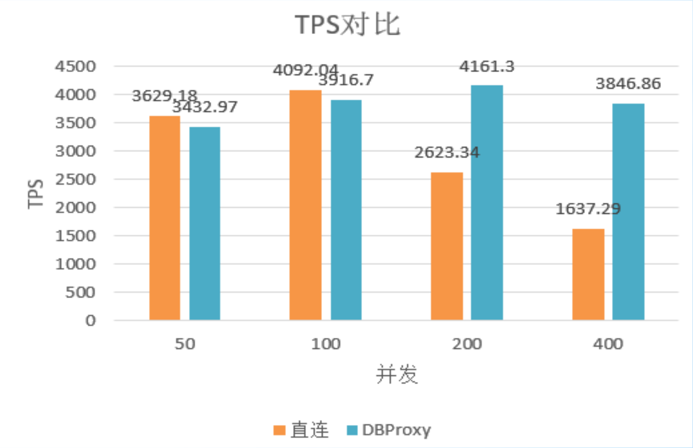
（7）支持连接复用，承载更多前端连接

（8）支持版本化连接池，无缝实现MySQL全局参数的动态变更

连接池收缩



对于单台MySQL实例，我们通常建议的活跃并发连接数为CPU核心数×2，例如对于四路服务器为80核，建议配置活跃并发连接数为160个，可发挥MySQL最佳性能。超过200个连接后，MySQL的性能会迅速下降



mysql优化案例1/11

- InnoDB表必须有主键或唯一索引

```
SELECT t.table_schema, t.table_name FROM information_schema.tables t LEFT JOIN information_schema.table_constraints c
ON (t.table_schema = c.table_schema AND t.table_name = c.table_name AND c.constraint_type IN ('PRIMARY KEY','UNIQUE'))
WHERE t.table_schema NOT IN ('mysql','information_schema',   ‘performance_schema') AND t.engine =   ‘InnoDB' ANDc.table_name IS NULL;
```

- 应用优化(使用自增长字段)

```
UPDATE sequence SET current_value = current_value + increment WHERE name =  NAME_CONST('seq_name',_utf8'VLG_FLOW_NO' COLLATE 'utf8_general_ci')
```

- 避免.大事务(运行行时间长或操作记录多)

```
SELECT a.requesting_trx_id '被阻塞事务ID' ,b.trx_mysql_thread_id '被阻塞线程ID', TIMESTAMPDIFF(SECOND,b.trx_wait_started,NOW())
  ‘被阻塞秒数', b.trx_query '被阻塞的语句句' , a.blocking_trx_id '阻塞事务ID' ,c.trx_mysql_thread_id '阻塞线程ID',d.INFO '阻塞事务信息' FROM
information_schema.INNODB_LOCK_WAITS a
 INNER JOIN information_schema.INNODB_TRX b ON a.requesting_trx_id=b.trx_id
INNER JOIN information_schema.INNODB_TRX c ON a.blocking_trx_id=c.trx_id
 INNER JOIN information_schema.PROCESSLIST d ON c.trx_mysql_thread_id=d.ID ;
```

mysql优化案例2/11

故障现象：

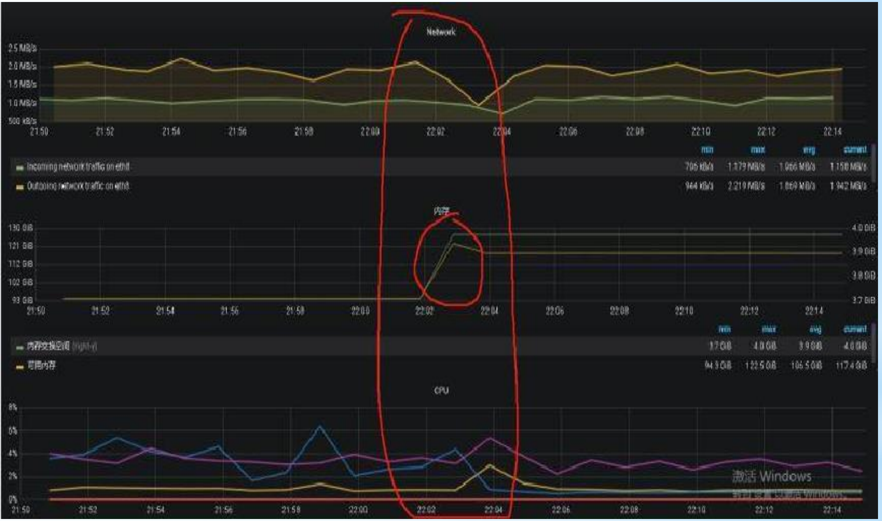
交易连接失败

故障原因：

大量并发连接执行以下语句做绑卡查询：

```
select
ta.ID_SERIAL,ta.ID_USER,ta.CARD_NO,ta.CARD_TYPE,ta.MOBILE,ta.HOLDE
R_NAME,ta.CHANNEL,ta.SEQ,ta.IS_DEF,ta.CRE_TIME,ta.UPD_TIME,ta.ID_C
ARD_AUTH_LOG,ta.DEF_CREDIT,tb.CARD_BIN,tb.CARD_BIN_NAME from
T_CARD_AUTH ta,T_CARD_BIN tb where ta.ID_USER
='2246450297264151557' AND ta.status='1' AND ta.CARD_NO like
concat(tb.CARD_BIN,'%') order by ta.CRE_TIME desc;
```

内存溢出



Zabbix监控

mysql优化案例3/11

MYSQL内存计算及优化

```
select
(@@key_buffer_size + @@query_cache_size + @@tmp_table_size
   + @@innodb_buffer_pool_size +
@@innodb_additional_mem_pool_size
   + @@innodb_log_buffer_size
   + @@max_connections * (
      @@read_buffer_size + @@read_rnd_buffer_size
      + @@sort_buffer_size+ @@join_buffer_size
      + @@binlog_cache_size + @@thread_stack
   )
)/1024/1024/1024;
```

SQL语句的优化，避免不必要排序：

```
# User@Host: xlife[xlife] @  [10.32.4.172]  Id: 184943908
# Query_time: 5.554402  Lock_time: 3.566839 Rows_sent:
0  Rows_examined: 0
select
ta.ID_SERIAL,ta.ID_USER,ta.CARD_NO,ta.CARD_TYPE,ta.
MOBILE,ta.HOLDER_NAME,ta.CHANNEL,ta.SEQ,ta.IS_DEF,
ta.CRE_TIME,ta.UPD_TIME,ta.ID_CARD_AUTH_LOG,ta.DEF
_CREDIT,tb.CARD_BIN,tb.CARD_BIN_NAME from
T_CARD_AUTH ta,T_CARD_BIN tb where ta.ID_USER
='2246450297264151557' AND ta.status='1' AND
ta.CARD_NO like concat(tb.CARD_BIN,'%') order by
ta.CRE_TIME desc;
```

mysql优化案例4/11

故障现象：

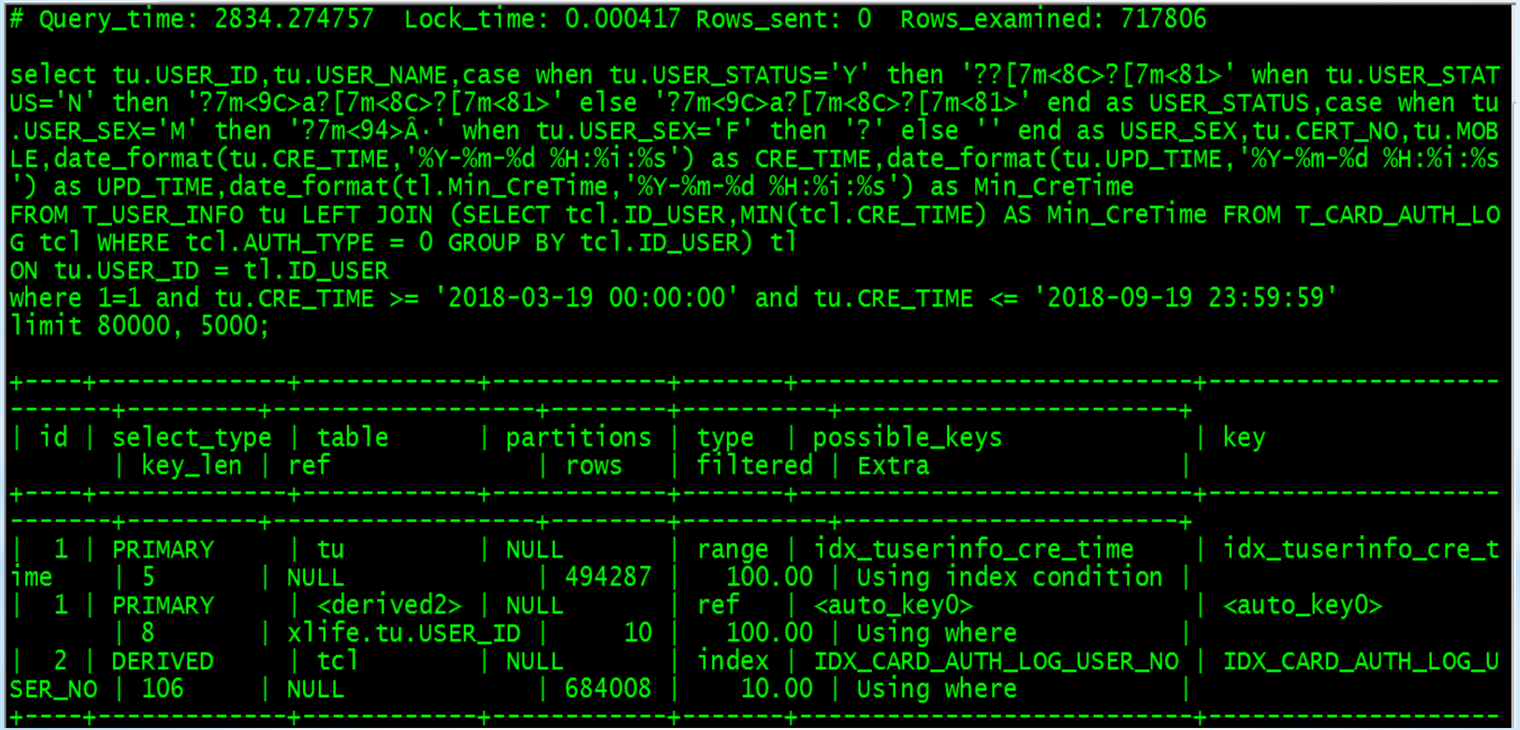
数据库连接过高，导致高可用管理用户不能够连接，因此高可用监控进程判断数据库主库状态异常，从而停止半同步复制连接，关闭应用连接，切换到主库

优化切换步骤：

先切换到备库，再关闭故障主库

mysql优化案例5/11

SQL语句优化：



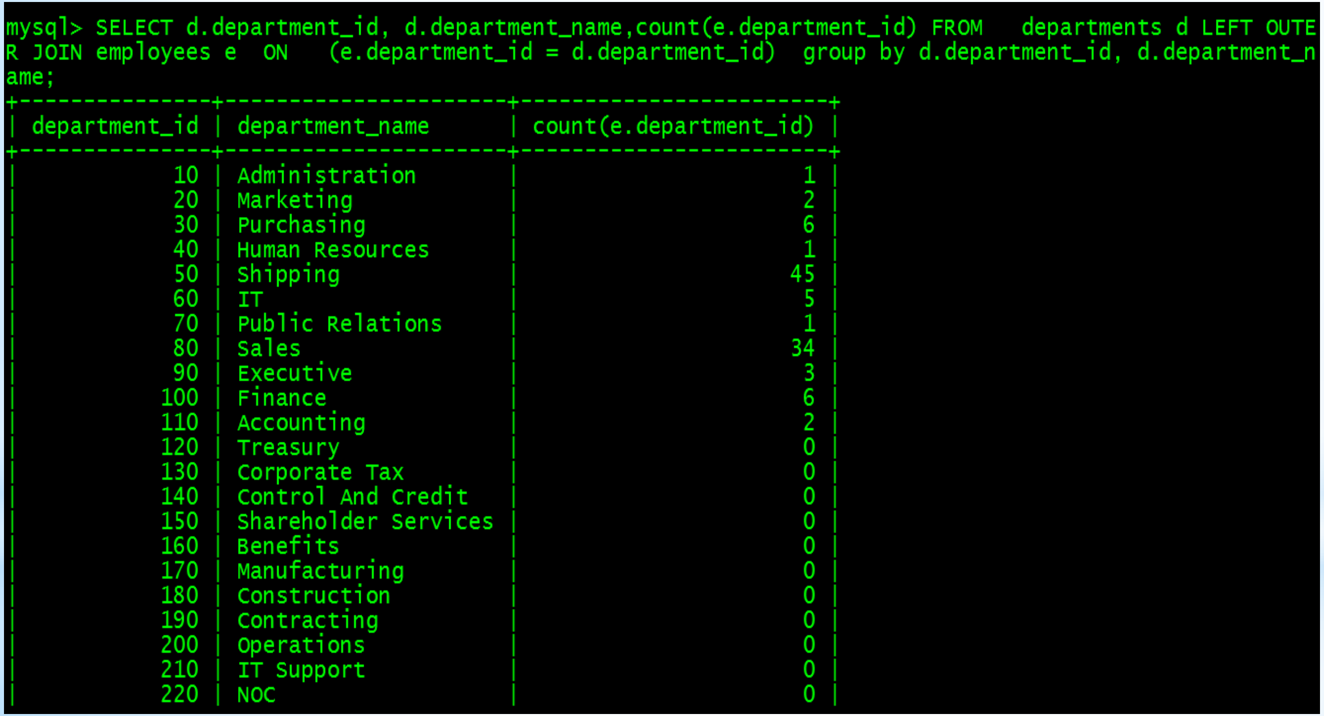
```
# Query_time: 2834.274757  Lock_time: 0.000417 Rows_sent: 0  Rows_examined: 717806

select tu.USER_ID,tu.USER_NAME,case when tu.USER_STATUS='Y' then '??[7m<8C>?[7m<81>' when tu.USER_STAT
US='N' then '?7m<9C>a?[7m<8C>?[7m<81>' else '?7m<9C>a?[7m<8C>?[7m<81>' end as USER_STATUS,case when tu
.USER_SEX='M' then '?7m<94>Â·' when tu.USER_SEX='F' then '?' else '' end as USER_SEX,tu.CERT_NO,tu.MOB
LE,date_format(tu.CRE_TIME,'%Y-%m-%d %H:%i:%s') as CRE_TIME,date_format(tu.UPD_TIME,'%Y-%m-%d %H:%i:%s
') as UPD_TIME,date_format(tl.Min_CreTime,'%Y-%m-%d %H:%i:%s') as Min_CreTime
FROM T_USER_INFO tu LEFT JOIN (SELECT tcl.ID_USER,MIN(tcl.CRE_TIME) AS Min_CreTime FROM T_CARD_AUTH_LO
G tcl WHERE tcl.AUTH_TYPE = 0 GROUP BY tcl.ID_USER) tl
ON tu.USER_ID = tl.ID_USER
where 1=1 and tu.CRE_TIME >= '2018-03-19 00:00:00' and tu.CRE_TIME <= '2018-09-19 23:59:59'
limit 80000, 5000;


+----+-------------+------------+------------+-------+-------------------------+--------------------
-------+---------+-----------------+---------+----------+-----------------------+
| id | select_type | table      | partitions | type  | possible_keys           | key
       | key_len | ref             | rows    | filtered | Extra                 |
+----+-------------+------------+------------+-------+-------------------------+--------------------
-------+---------+-----------------+---------+----------+-----------------------+
|  1 | PRIMARY     | tu         | NULL       | range | idx_tuserinfo_cre_time  | idx_tuserinfo_cre_t
ime    | 5       | NULL            |  494287 |   100.00 | Using index condition |
|  1 | PRIMARY     | <derived2> | NULL       | ref   | <auto_key0>             | <auto_key0>
       | 8       | xlife.tu.USER_ID |     10 |   100.00 | Using where           |
|  2 | DERIVED     | tcl        | NULL       | index | IDX_CARD_AUTH_LOG_USER_NO | IDX_CARD_AUTH_LOG_U
SER_NO | 106     | NULL            |  684008 |    10.00 | Using where           |
+----+-------------+------------+------------+-------+-------------------------+--------------------
```

mysql优化案例6/11

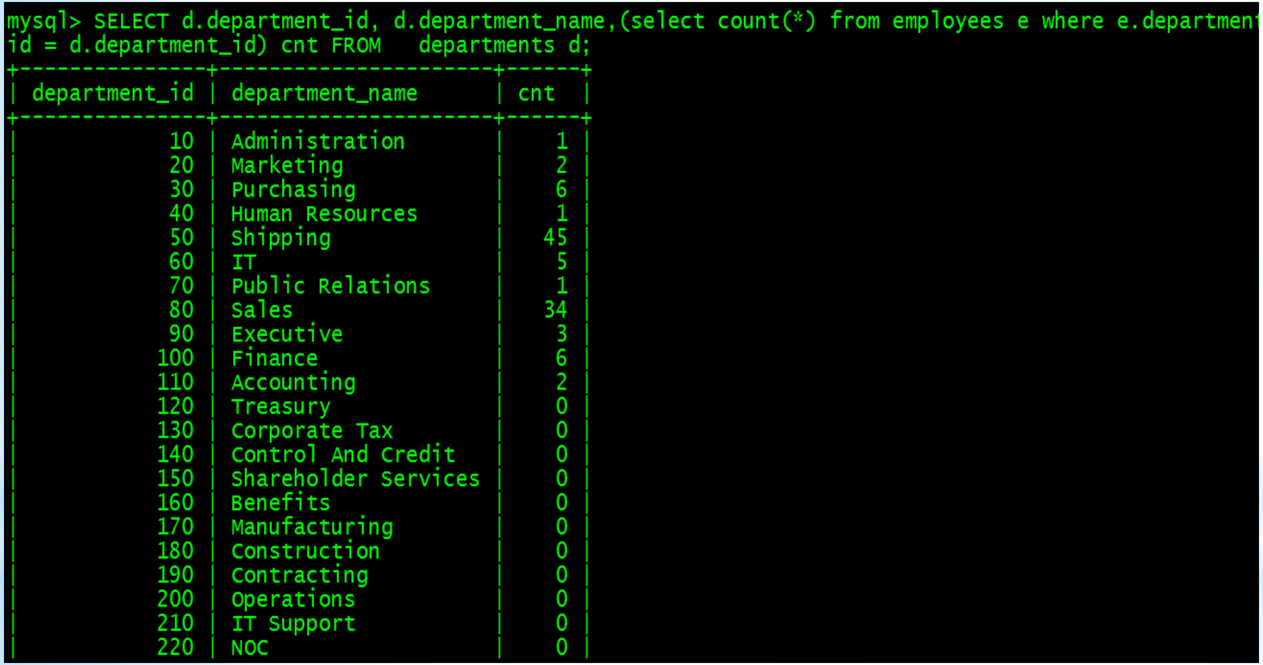
左外连接语句：



```
mysql> SELECT d.department_id, d.department_name,count(e.department_id) FROM   departments d LEFT OUTE
R JOIN employees e  ON   (e.department_id = d.department_id)  group by d.department_id, d.department_n
ame;
+---------------+----------------------+------------------------+
| department_id | department_name      | count(e.department_id) |
+---------------+----------------------+------------------------+
|            10 | Administration       |                      1 |
|            20 | Marketing            |                      2 |
|            30 | Purchasing           |                      6 |
|            40 | Human Resources      |                      1 |
|            50 | Shipping             |                     45 |
|            60 | IT                   |                      5 |
|            70 | Public Relations     |                      1 |
|            80 | Sales                |                     34 |
|            90 | Executive            |                      3 |
|           100 | Finance              |                      6 |
|           110 | Accounting           |                      2 |
|           120 | Treasury             |                      0 |
|           130 | Corporate Tax        |                      0 |
|           140 | Control And Credit   |                      0 |
|           150 | Shareholder Services |                      0 |
|           160 | Benefits             |                      0 |
|           170 | Manufacturing        |                      0 |
|           180 | Construction         |                      0 |
|           190 | Contracting          |                      0 |
|           200 | Operations           |                      0 |
|           210 | IT Support           |                      0 |
|           220 | NOC                  |                      0 |
```

mysql优化案例7/11

标量子查询语句改写外连接及适用场景：



```
mysql> SELECT d.department_id, d.department_name,(select count(*) from employees e where e.departmen
id = d.department_id) cnt FROM   departments d;
+---------------+----------------------+-------+
| department_id | department_name      | cnt   |
+---------------+----------------------+-------+
|            10 | Administration       |     1 |
|            20 | Marketing            |     2 |
|            30 | Purchasing           |     6 |
|            40 | Human Resources      |     1 |
|            50 | Shipping             |    45 |
|            60 | IT                   |     5 |
|            70 | Public Relations     |     1 |
|            80 | Sales                |    34 |
|            90 | Executive            |     3 |
|           100 | Finance              |     6 |
|           110 | Accounting           |     2 |
|           120 | Treasury             |     0 |
|           130 | Corporate Tax        |     0 |
|           140 | Control And Credit   |     0 |
|           150 | Shareholder Services |     0 |
|           160 | Benefits             |     0 |
|           170 | Manufacturing        |     0 |
|           180 | Construction         |     0 |
|           190 | Contracting          |     0 |
|           200 | Operations           |     0 |
|           210 | IT Support           |     0 |
|           220 | NOC                  |     0 |
```

mysql优化案例8/11

问题描述：

1、消费信贷多渠道业务系统，数据量2T以上

2、一个实例，一套数据库，不同渠道通过表名yewu1_ 、yewu2_ 类似区分

3、每天有跑批量

4、批量延迟大，过万秒

5、主备数据库

优化方法：

1、硬件优化、固态盘

2、数据库参数调整：innodb_flush_log_at_trx_commit、sync_binlog

3、业务优化，监管数据到大数据平台

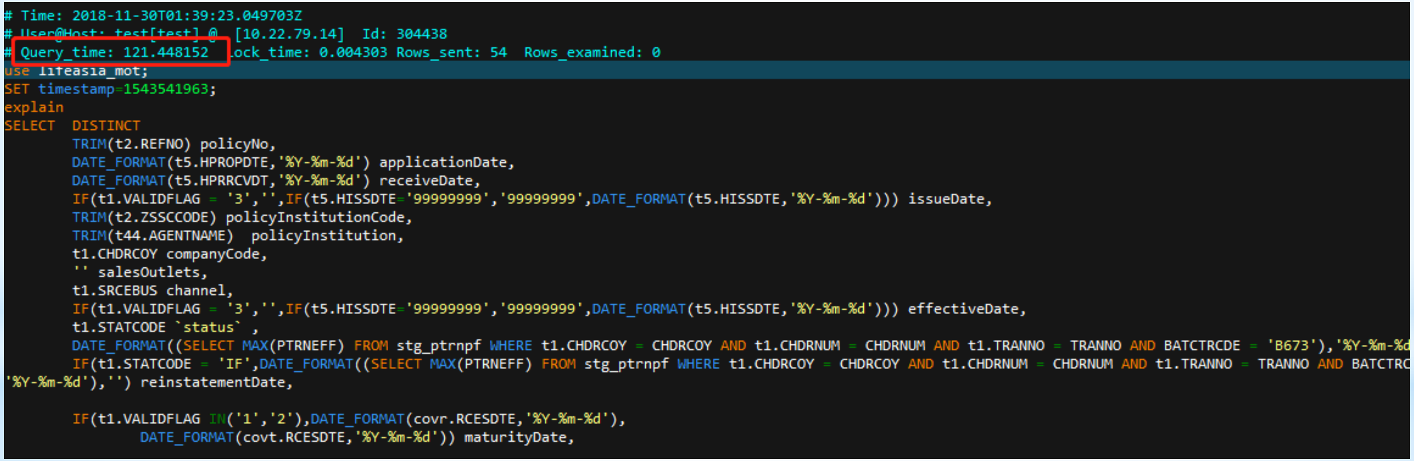
4、分库，不同业务分库处理

5、主从复制参数调整等

mysql优化案例9/11

故障现象：

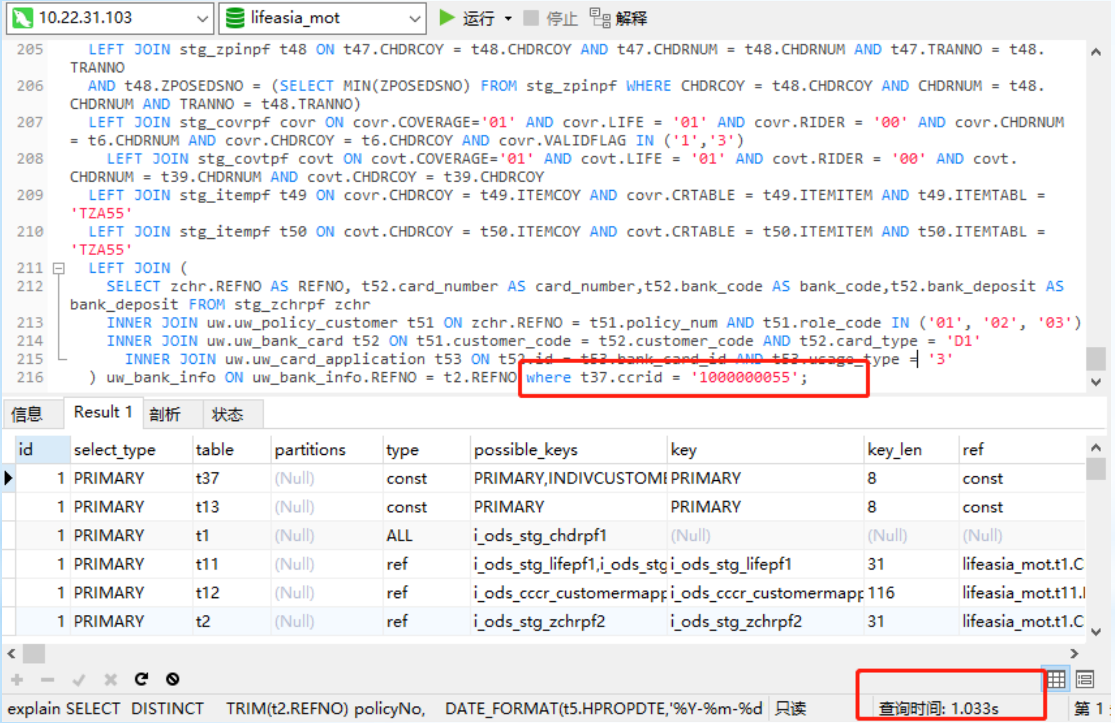
开发同事反映xxx测试环境执行存在一条SQL(SQL请查看附件)查询效率很低(单次查询耗时400s)，在配置类似情况下，相同SQL在UAT环境执行很快。



```
# Time: 2018-11-30T01:39:23.049703Z
# User@Host: test[test] @  [10.22.79.14]  Id: 304438
# Query_time: 121.448152  Lock_time: 0.004303 Rows_sent: 54  Rows_examined: 0
use lifeasia_mot;
SET timestamp=1543541963;
explain
SELECT  DISTINCT
        TRIM(t2.REFNO) policyNo,
        DATE_FORMAT(t5.HPROPDTE,'%Y-%m-%d') applicationDate,
        DATE_FORMAT(t5.HPRRCVDT,'%Y-%m-%d') receiveDate,
        IF(t1.VALIDFLAG = '3','',IF(t5.HISSDTE='99999999','99999999',DATE_FORMAT(t5.HISSDTE,'%Y-%m-%d'))) issueDate,
        TRIM(t2.ZSSCCODE) policyInstitutionCode,
        TRIM(t44.AGENTNAME)  policyInstitution,
        t1.CHDRCOY companyCode,
        '' salesOutlets,
        t1.SRCEBUS channel,
        IF(t1.VALIDFLAG = '3','',IF(t5.HISSDTE='99999999','99999999',DATE_FORMAT(t5.HISSDTE,'%Y-%m-%d'))) effectiveDate,
        t1.STATCODE `status` ,
        DATE_FORMAT((SELECT MAX(PTRNEFF) FROM stg_ptrnpf WHERE t1.CHDRCOY = CHDRCOY AND t1.CHDRNUM = CHDRNUM AND t1.TRANNO = TRANNO AND BATCTRCDE = 'B673'),'%Y-%m-%d
        IF(t1.STATCODE = 'IF',DATE_FORMAT((SELECT MAX(PTRNEFF) FROM stg_ptrnpf WHERE t1.CHDRCOY = CHDRCOY AND t1.CHDRNUM = CHDRNUM AND t1.TRANNO = TRANNO AND BATCTRC
'%Y-%m-%d'),'') reinstatementDate,

        IF(t1.VALIDFLAG IN('1','2'),DATE_FORMAT(covr.RCESDTE,'%Y-%m-%d'),
                DATE_FORMAT(covt.RCESDTE,'%Y-%m-%d')) maturityDate,
```

mysql优化案例10/11



```sql
  LEFT JOIN stg_zpinpf t48 ON t47.CHDRCOY = t48.CHDRCOY AND t47.CHDRNUM = t48.CHDRNUM AND t47.TRANNO = t48.TRANNO
  AND t48.ZPOSEDSNO = (SELECT MIN(ZPOSEDSNO) FROM stg_zpinpf WHERE CHDRCOY = t48.CHDRCOY AND CHDRNUM = t48.CHDRNUM AND TRANNO = t48.TRANNO)
  LEFT JOIN stg_covrpf covr ON covr.COVERAGE='01' AND covr.LIFE = '01' AND covr.RIDER = '00' AND covr.CHDRNUM = t6.CHDRNUM AND covr.CHDRCOY = t6.CHDRCOY AND covr.VALIDFLAG IN ('1','3')
    LEFT JOIN stg_covtpf covt ON covt.COVERAGE='01' AND covt.LIFE = '01' AND covt.RIDER = '00' AND covt.CHDRNUM = t39.CHDRNUM AND covt.CHDRCOY = t39.CHDRCOY
  LEFT JOIN stg_itempf t49 ON covr.CHDRCOY = t49.ITEMCOY AND covr.CRTABLE = t49.ITEMITEM AND t49.ITEMTABL = 'TZA55'
  LEFT JOIN stg_itempf t50 ON covt.CHDRCOY = t50.ITEMCOY AND covt.CRTABLE = t50.ITEMITEM AND t50.ITEMTABL = 'TZA55'
  LEFT JOIN (
    SELECT zchr.REFNO AS REFNO, t52.card_number AS card_number,t52.bank_code AS bank_code,t52.bank_deposit AS bank_deposit FROM stg_zchrpf zchr
    INNER JOIN uw.uw_policy_customer t51 ON zchr.REFNO = t51.policy_num AND t51.role_code IN ('01', '02', '03')
    INNER JOIN uw.uw_bank_card t52 ON t51.customer_code = t52.customer_code AND t52.card_type = 'D1'
      INNER JOIN uw.uw_card_application t53 ON t52.id = t53.bank_card_id AND t53.usage_type = '3'
  ) uw_bank_info ON uw_bank_info.REFNO = t2.REFNO where t37.ccrid = '1000000055';
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref
1	PRIMARY	t37	(Null)	const	PRIMARY,INDIVCUSTOMI	PRIMARY	8	const
1	PRIMARY	t13	(Null)	const	PRIMARY	PRIMARY	8	const
1	PRIMARY	t1	(Null)	ALL	i_ods_stg_chdrpf1	(Null)	(Null)	(Null)
1	PRIMARY	t11	(Null)	ref	i_ods_stg_lifepf1,i_ods_stg	i_ods_stg_lifepf1	31	lifeasia_mot.t1.C
1	PRIMARY	t12	(Null)	ref	i_ods_cccr_customermapp	i_ods_cccr_customermapp	116	lifeasia_mot.t11.I
1	PRIMARY	t2	(Null)	ref	i_ods_stg_zchrpf2	i_ods_stg_zchrpf2	31	lifeasia_mot.t1.C

```
explain SELECT  DISTINCT    TRIM(t2.REFNO) policyNo,    DATE_FORMAT(t5.HPROPDTE,'%Y-%m-%d  只读    查询时间: 1.033s    第 1
```

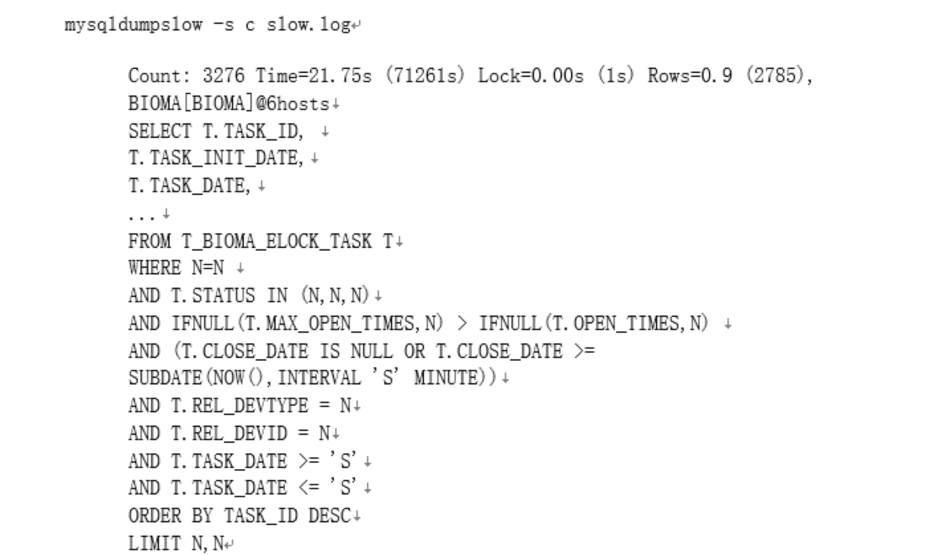
故障原因：

该环境操作系统时间在前段时间进行过修改，曾经从2018年改至2020年进行某项测试，后续由于测试完毕，系统时间回调至2018年，由于MySQL依赖操作系统时间，导致在这端期间内，MySQL统计信息收集到的信息为2020年的统计信息状态值，从而当时间回调后，MySQL依据了错误的统计信息(未来时间的统计信息)，导致执行计划失效，执行效率低下。

通过以下操作进行修复：

- 生成对该实例下的所有库表的统计信息手动收集语句并在业务低峰期执行
- 统计当前数据库数据量判断收集统计信息的时间成本消耗
- 运行手动收集统计信息语句并后续观察类似SQL执行效率

mysql优化案例11/11



```
mysqldumpslow -s c slow.log↵

    Count: 3276 Time=21.75s (71261s) Lock=0.00s (1s) Rows=0.9 (2785),
    BIOMA[BIOMA]@6hosts↓
    SELECT T.TASK_ID, ↓
    T.TASK_INIT_DATE,↓
    T.TASK_DATE,↓
    ...↓
    FROM T_BIOMA_ELOCK_TASK T↓
    WHERE N=N ↓
    AND T.STATUS IN (N,N,N)↓
    AND IFNULL(T.MAX_OPEN_TIMES,N) > IFNULL(T.OPEN_TIMES,N) ↓
    AND (T.CLOSE_DATE IS NULL OR T.CLOSE_DATE >=
    SUBDATE(NOW(),INTERVAL 'S' MINUTE))↓
    AND T.REL_DEVTYPE = N↓
    AND T.REL_DEVID = N↓
    AND T.TASK_DATE >= 'S'↓
    AND T.TASK_DATE <= 'S'↓
    ORDER BY TASK_ID DESC↓
    LIMIT N,N↵
```

故障原因：

从大屏监控来看，1月2号和1月11号上午九点有一个服务器CPU升高到100%，怀疑是业务再九点会有压力下发，按执行次数统计slow log排查到有一条执行次数最多的sql。

优化建议：

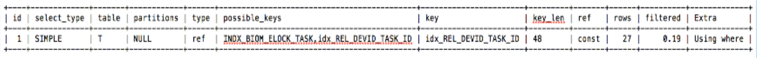
1.添加 REL_DEVID，TASK_ID 组合索引，测试sql 性能：

alter table T_BIOMA_ELOCK_TASK add index idx_REL_DEVID_TASK_ID(REL_DEVID,TASK_ID);

2.添加索引后执行计划：

注意REL_DEVID 字段数据类型为 varchar，需要在sql 中加引号：

AND T.REL_DEVID = 000000025180033 》》 AND T.REL_DEVID = '000000025180033’



```
+----+-------------+-------+------------+------+------------------------------------------+-----------------------+---------+-------+------+----------+-------------+
| id | select_type | table | partitions | type | possible_keys                            | key                   | key_len | ref   | rows | filtered | Extra       |
+----+-------------+-------+------------+------+------------------------------------------+-----------------------+---------+-------+------+----------+-------------+
|  1 | SIMPLE      | T     | NULL       | ref  | INDX_BIOM_ELOCK_TASK,idx_REL_DEVID_TASK_ID | idx_REL_DEVID_TASK_ID | 48      | const |   27 |     0.19 | Using where |
+----+-------------+-------+------------+------+------------------------------------------+-----------------------+---------+-------+------+----------+-------------+
```

3.执行时间从 10s+ 降到毫秒级别

Thank You!