

DTCC

2017第八届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2017

# DBA成长沉思录

杨建荣

# 个人介绍- 杨建荣

➤ Oracle ACE



➤ DBAplus联合发起人

➤ YEP成员

➤ Oracle 10g OCP,OCM , MySQL OCP

➤ 对shell , Java有一定的功底

➤ <<Oracle DBA 工作笔记>> 作者

➤ 曾在中国数据库大会,Oracle嘉年华,QCon演讲

➤ 坚持每天写点东西, 已连续坚持1100多天

➤ 微信公众号 (jianrong-notes),



# 分享思路

- 1.管中窥豹，通过数据库参数了解技术变化
- 2.四两拨千斤，如何洞察SQL的解析过程
- 3.多一些思考，多一些收获
- 4.用架构的角度来思考
- 5.性能优化面面观
- 6.总结

# 一.通过数据库参数了解技术变化

- DB-Engines数据
- Oracle最新版本12.2
- MySQL 最新版本5.7

## The most popular database management systems

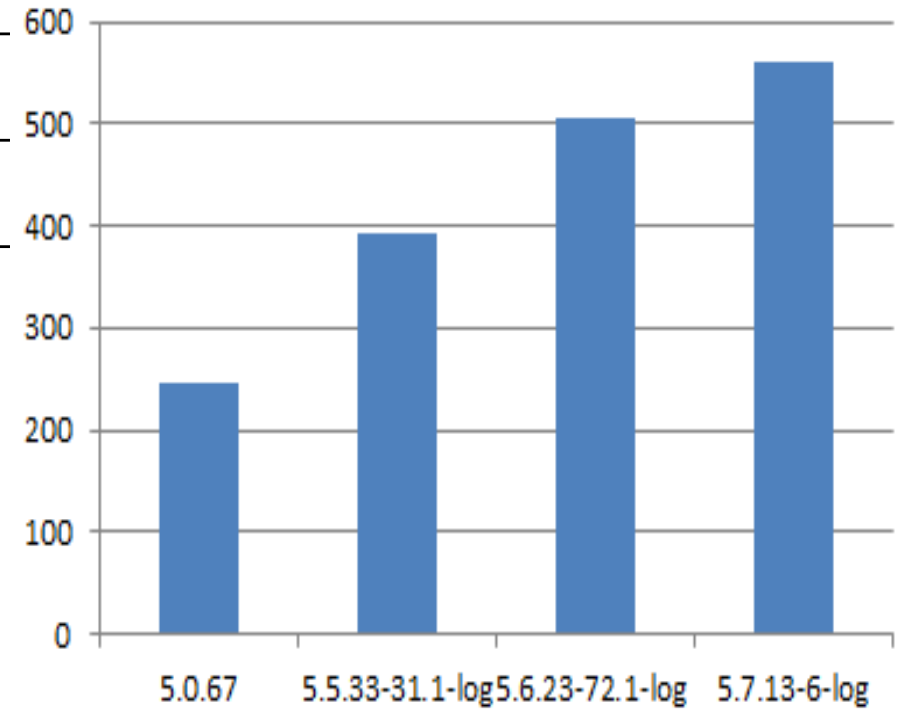
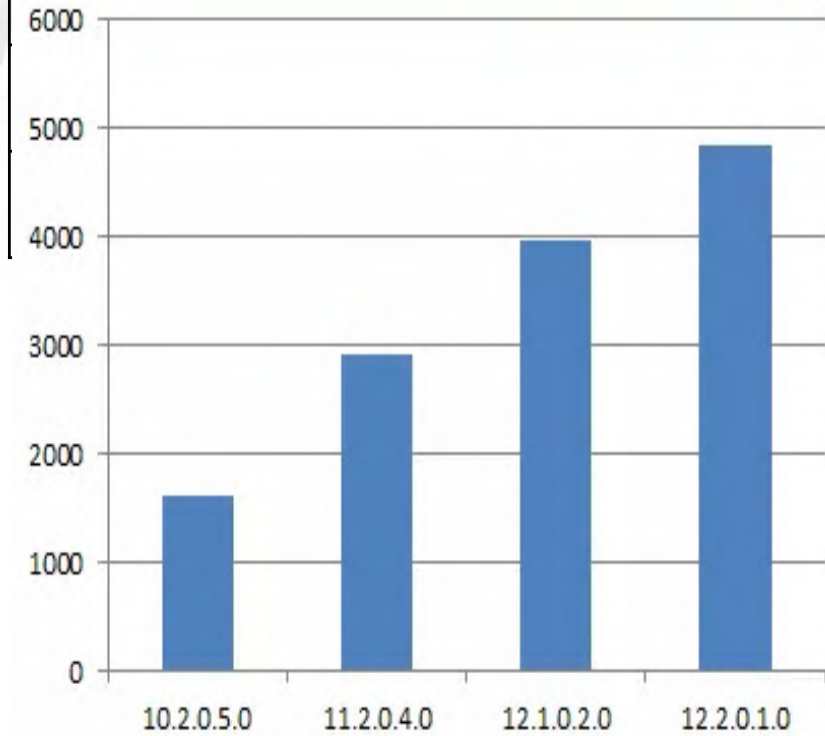
| April 2017                              | Score |
|---|-------|
| 1. <a href="#">Oracle</a>               | 1402  |
| 2. <a href="#">MySQL</a>                | 1365  |
| 3. <a href="#">Microsoft SQL Server</a> | 1205  |
| 4. <a href="#">PostgreSQL</a>           | 362   |
| 5. <a href="#">MongoDB</a>              | 325   |

- Oracle的亮点特性 PDB,IMO,sharding
- MySQL的亮点特性 查询优化, MGR

# 1.1 数据库参数的版本变化

| Oracle版本   | 所有参数 | 开放参数 |
|------------|------|------|
| 10.2.0.5.0 | 1618 | 259  |

| MySQL版本         | 参数个数 |
|-----------------|------|
| 5.0.67          | 245  |
| 5.5.33-31.1-log | 397  |

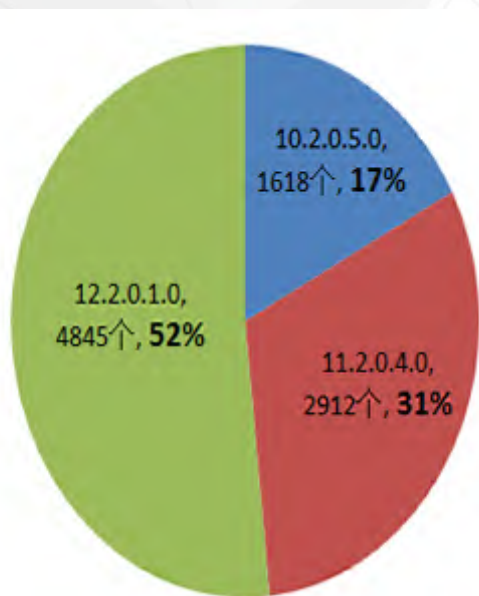




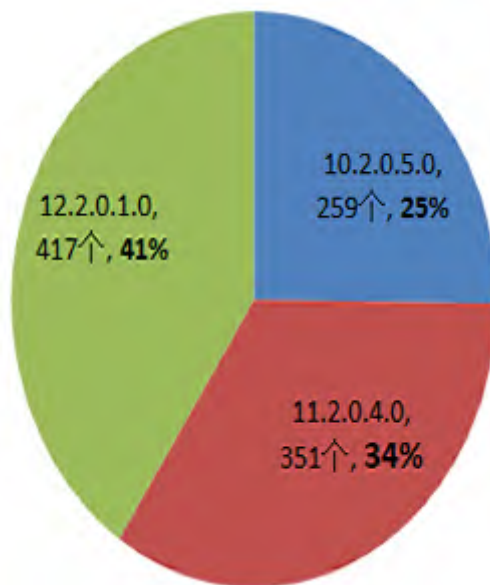
# 1.2 数据库参数的版本变化

## Oracle各版本参数数量分布图

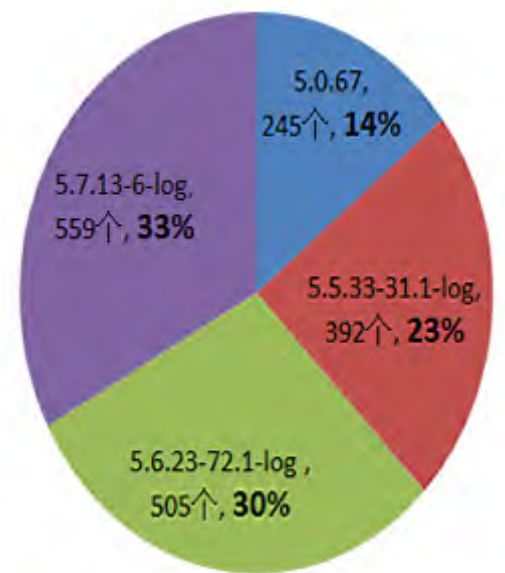
### 所有参数分布图



### 开放参数分布图



## MySQL各版本数量分布图



Oracle R1,R2相隔数年,潜力巨大

MySQL 5.7变化相对平稳, 预计在8.0会有较大的变化

## 二.如何洞察SQL的解析过程

- 准备一个测试表

```
create table test (id number,name varchar2(30));
```

## 2.1 开始推理

- select1 id1 from2 test1 where3 id1='aaa' group by4 id1 order by5 id1;

```
SOL> select1 id1 from2 test1 where3 id1='aaa' group by4 id1 order by5 id1;
```

```
ERROR at line 1:  
ORA-24333: zero iteration count
```



## 2.2 修复语法错误

- select id1 from test1 where3 id1='aaa' group by4 id1 order by5 id1;

```
SQL> select id1 from test1 where3 id1='aaa' group  
select id1 from test1 where3 id1= 'aaa' group by4
```

```
ERROR at line 1:  
ORA-00933: SQL command not properly ended
```

## 2.3 互动部分

- `select id1 from test1 where id1='aaa' group by4 id1 order by5 id1;`

```
SQL> select id1 from test1 where id1='aaa' group  
select id1 from test1 where id1='aaa' group by4  
*
```

```
ERROR at line 1:  
ORA-00924: missing BY keyword
```

## 2.4 互动部分

- select id1 from test1 where id1='aaa' group by id1  
order by id1;

```
SQL> select id1 from test1 where id1="aaa"  
select id1 from test1 where id1='aaa' gro  
*
```

```
ERROR at line 1:  
ORA-00942: table or view does not exist
```

## 2.5 互动部分

- select id1 from test where id1='aaa' group by id1 order by id1;

```
SQL> select id1 from test where id1='aaa' group by  
select id1 from test where id1='aaa' group by id1
```

```
ERROR at line 1:  
ORA-00904: "ID1": invalid identifier
```

## 2.6 互动部分

- select id1 from test where id1='aaa' group by id order by id1;

```
SQL> select id1 from test where id1='aaa' g  
select id1 from test where id1='aaa' group  
*
```

```
ERROR at line 1:  
ORA-00904: "ID1": invalid identifier
```

## 三.多一些思考，多一些收获

- UPDATE payment\_data rr  
SET rr.penalty\_date = '2017-4-12'  
where rr.id =  
(SELECT min(r.id)  
FROM payment\_data r  
where data\_no =  
(SELECT data\_no  
FROM user\_debt  
WHERE out\_trade\_no =  
'bestpay\_order\_no1491812746329')));
- ERROR 1093 (HY000): You can't specify target table 'rr' for update in FROM clause



## 3.1 WA

- UPDATE payment\_data rr  
SET rr.penalty\_date = '2017-4-12'  
where rr.id =  
(SELECT min(t.id)  
FROM (select id,data\_no from payment\_data  
r) t  
where t.data\_no =  
(SELECT data\_no  
FROM user\_debt  
WHERE out\_trade\_no =  
'bestpay\_order\_no1491812746329')));

## 3.2 MySQL和Oracle解析的异同

- UPDATE payment\_data rr, (SELECT min(r.id) id  
FROM payment\_data r  
where data\_no =  
(SELECT data\_no  
FROM user\_debt  
WHERE out\_trade\_no =  
'bestpay\_order\_no1491812746329')) t  
SET rr.penalty\_date = '2017-4-12'  
where rr.id =t.id;
- MySQL:能够正确执行
- Oracle:解析出错

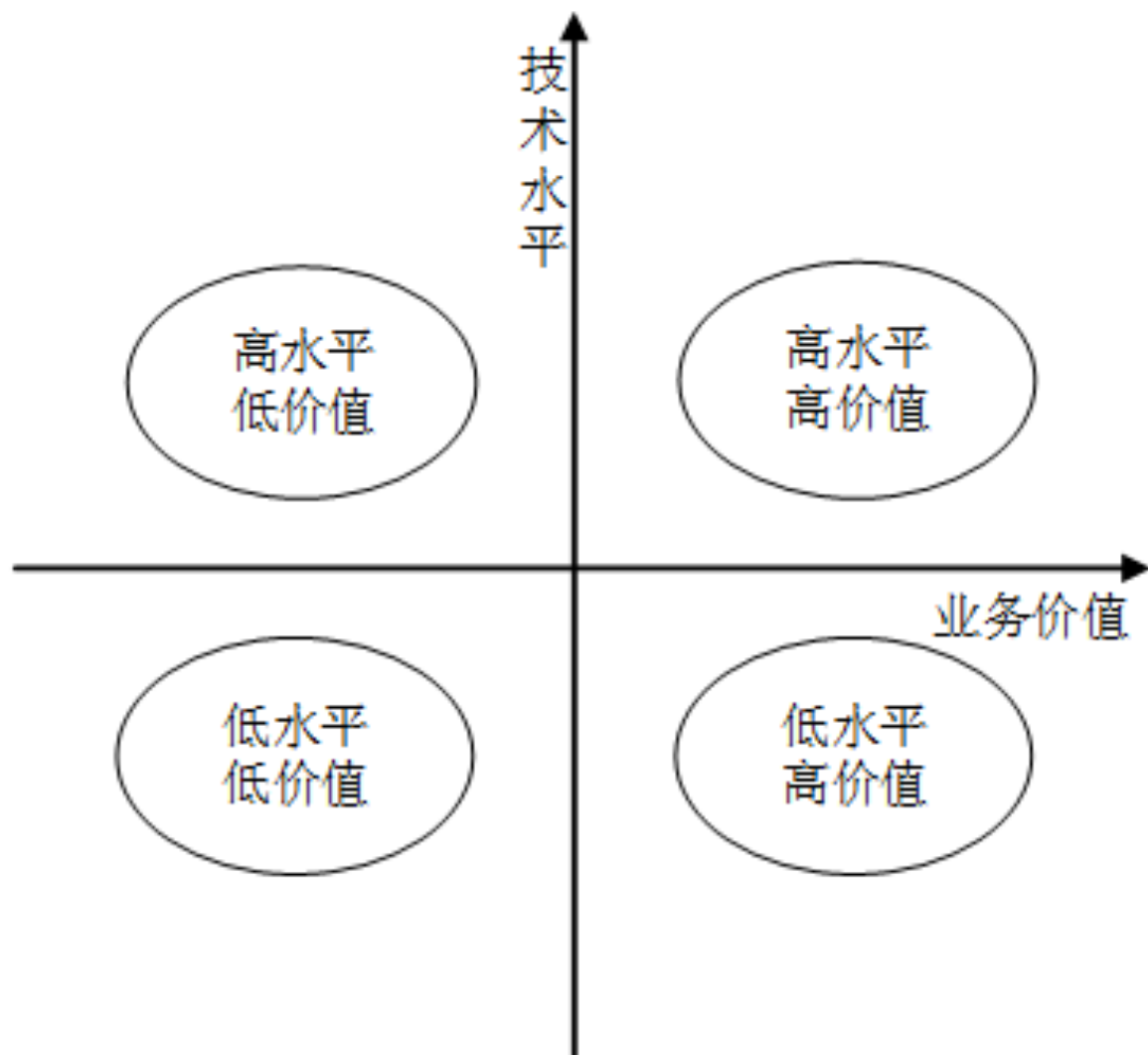
## 3.3 MySQL和Oracle解析的异同

- UPDATE payment\_data rr  
SET rr.penalty\_date = '2017-4-12'  
where rr.id =  
(SELECT min(r.id)  
FROM payment\_data r  
inner join user\_debt b on r.data\_no =  
b.data\_no  
WHERE b.out\_trade\_no =  
'bestpay\_order\_no1491812746329');
- MySQL:执行失败
- Oracle:执行正常

## 3.4 MySQL Derived tables

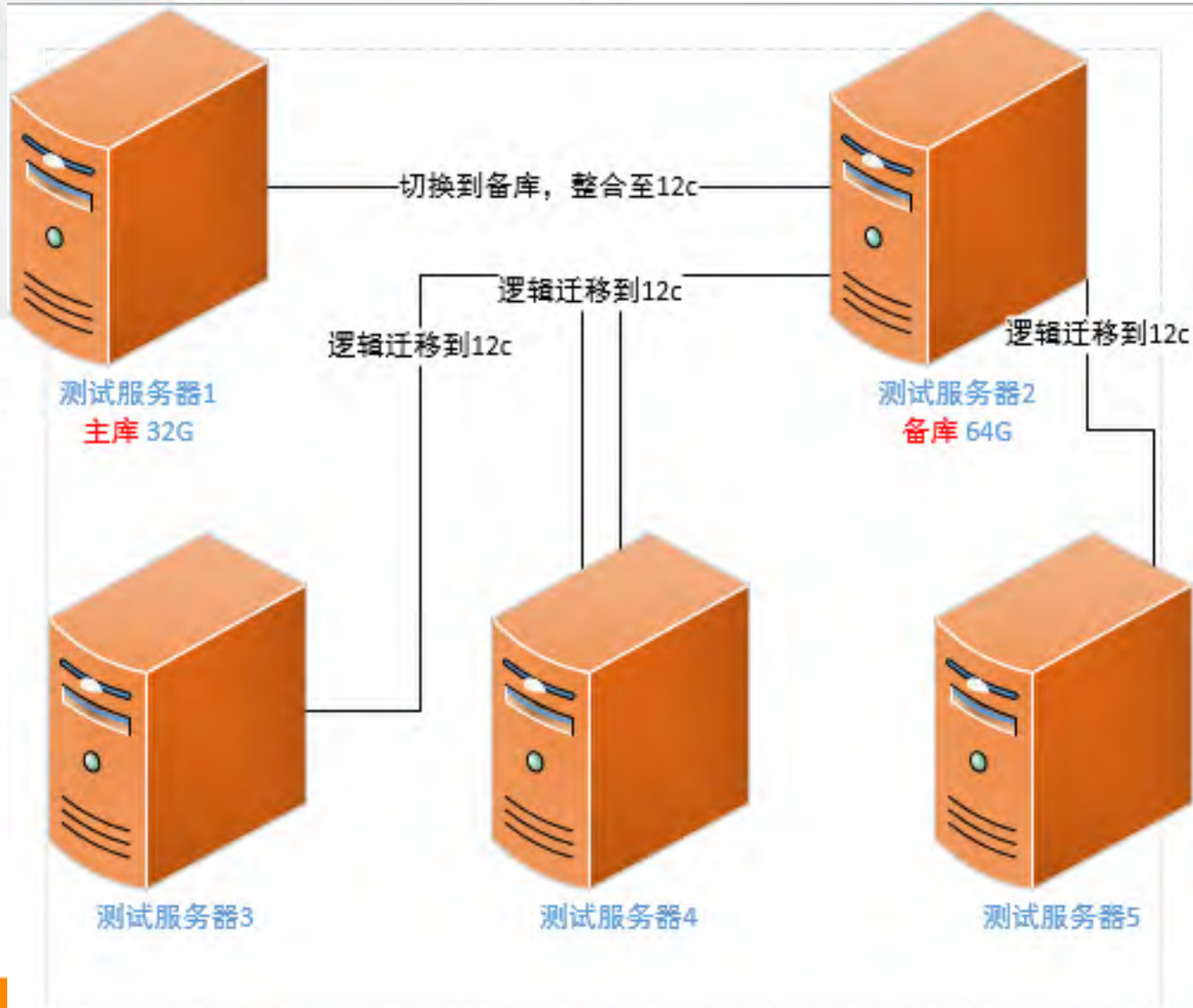
- > select \* from t\_fund\_info where id=138031;  
1 row in set (0.01 sec)
- select \* from (select id from t\_fund\_info) t where  
t.id=138031;  
1 row in set (1.12 sec)
- select \* from (select \* from t\_fund\_info) t where  
t.id=138031;  
ERROR 126 (HY000): Incorrect key file for table  
'/tmp/#sql\_3e34\_0.MYI'; try to repair it --2G+
- create view test\_view as select \* from t\_fund\_info;
- select \* from test\_view where id=138031;  
1 row in set (0.01 sec)

# 四.系统架构演进策略



- 1) 淘汰
- 2) 继承
- 3) 改造
- 4) 集成

# 4.1 数据整合，与时俱进



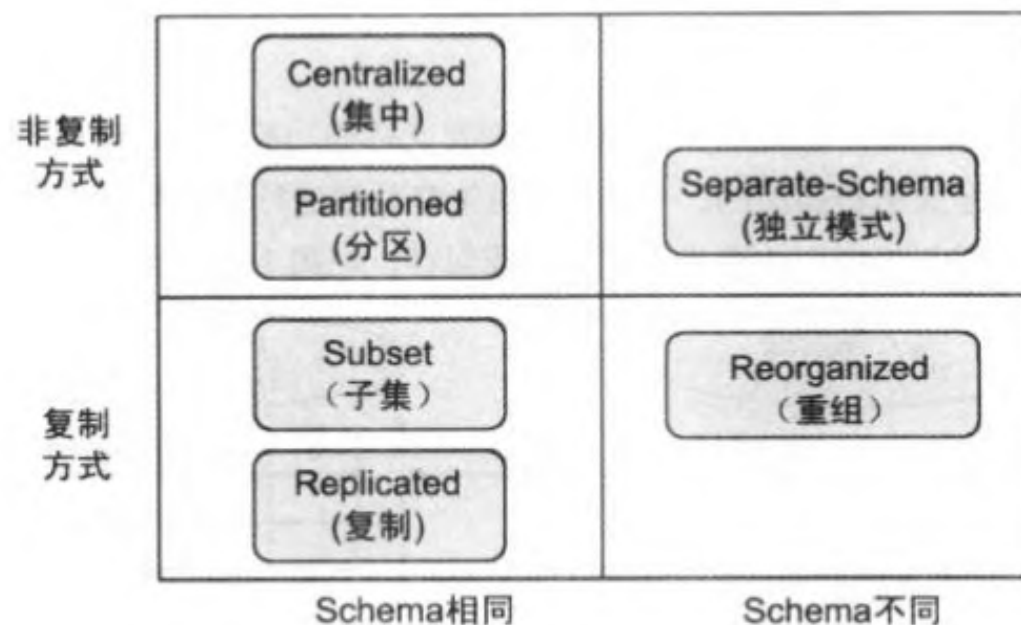


## 4.2 数据架构的难点-数据分布

- 独立Schema(Separate-Schema) 减少系统之间的相互影响，避免将问题复杂化
- 集中(Centralized) 集中存储，分布访问
- 分区 (Partitioned) 不是数据库中的分区表，基于架构层面，水平分区和垂直分区
  - 水平分区 subset (a,b,c,d),
  - 垂直分区 subset(a,b),subset(c,d)
- 复制(Replicated) 实时或者快照级别来保持多个数据副本的数据一致性
- 子集(Subset) 子集数据的一致性，比如网银渠道数据一致性
- 重组 (Reorganized)
  - 统计性重组 交易流水总量
  - 结构性重组，用户的概要信息，用户的详细信息

## 4.3 数据架构策略的优缺点

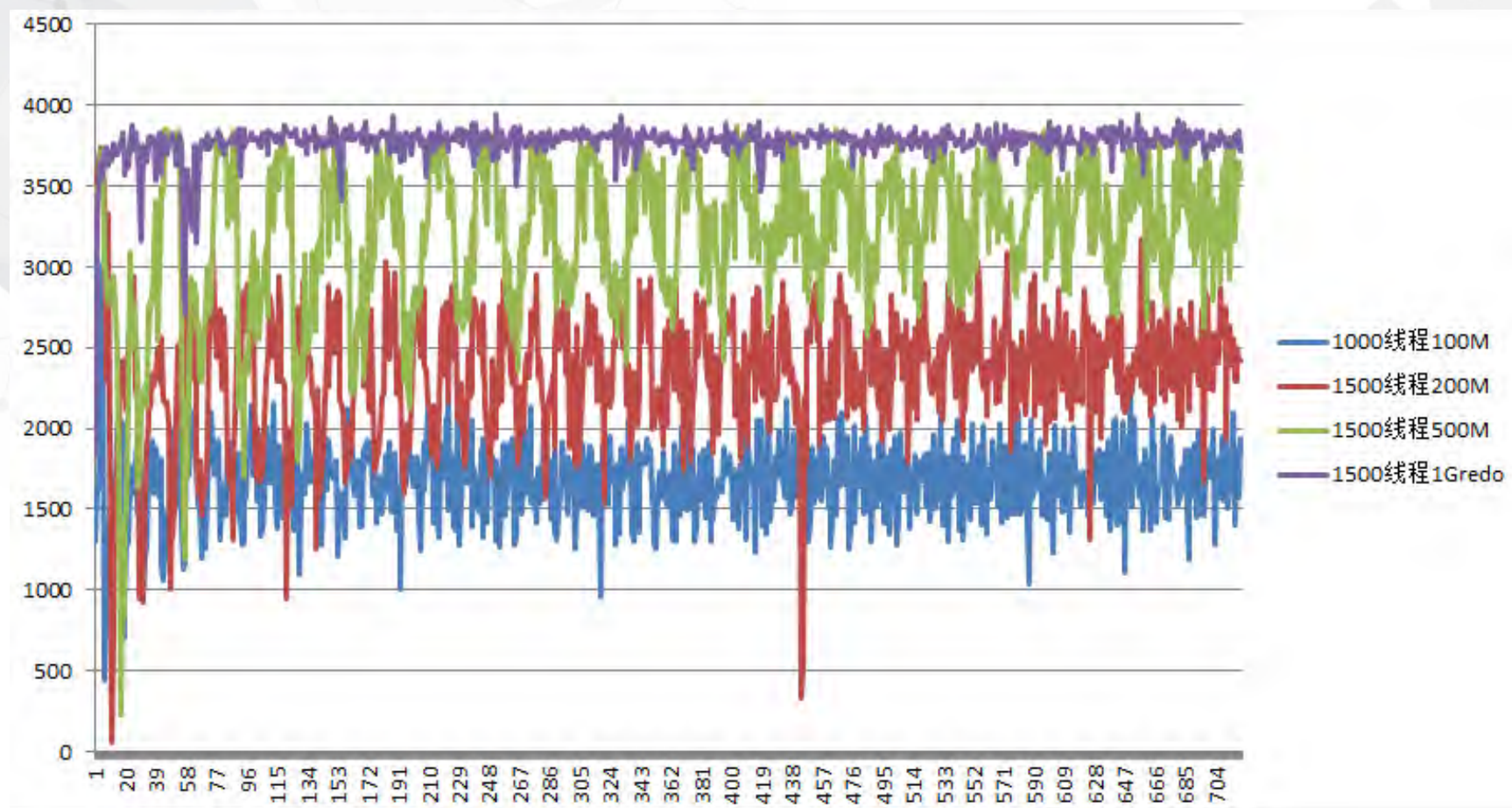
- 策略的质量属性
  - 集中: 可管理性, 数据一致性
  - 复制: 可靠性
  - 分区: 可伸缩性



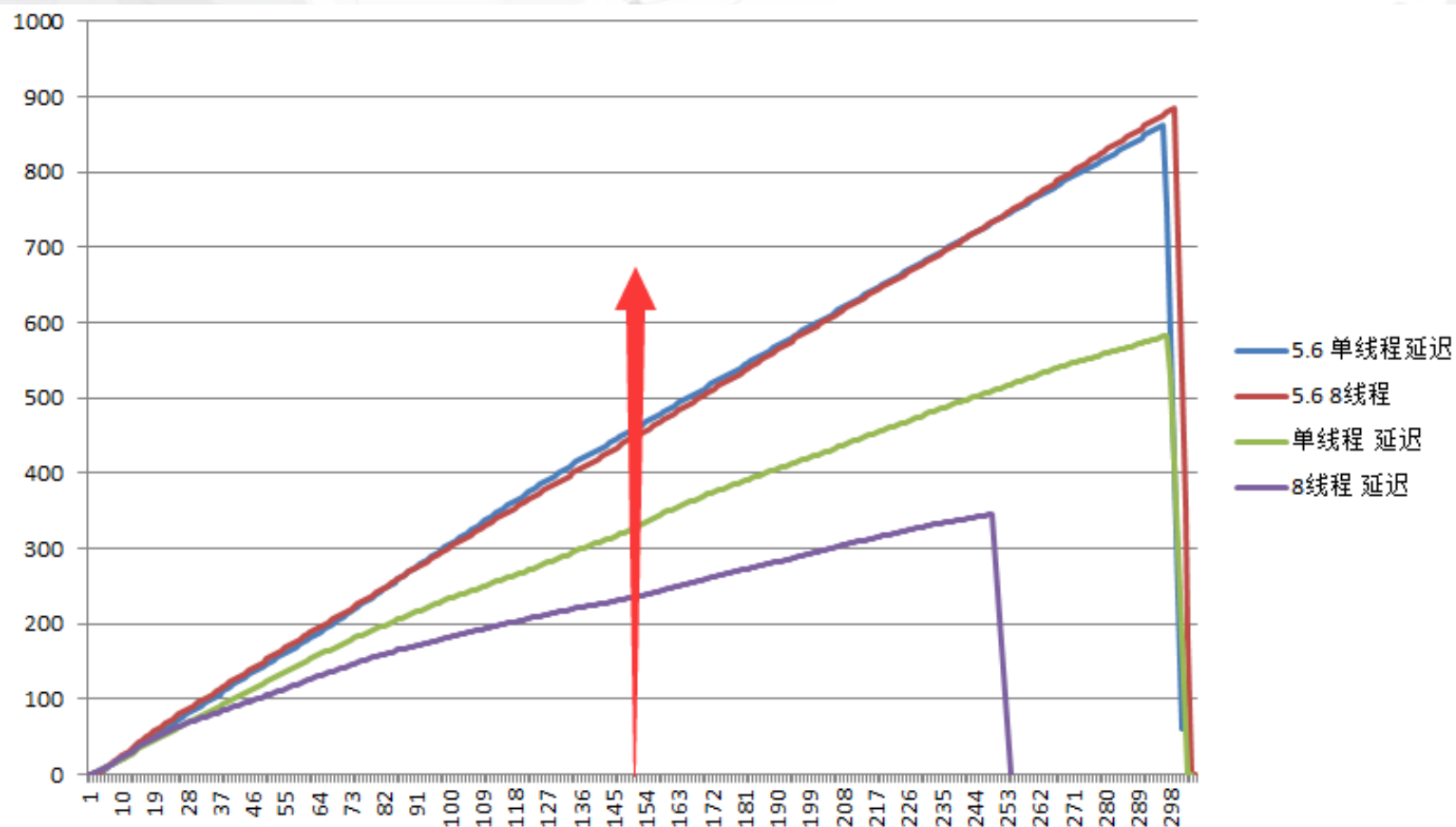
# 五. 性能优化面面观

- MySQL压力测试
- Oracle压缩归档
- SQL优化

# 5.1 MySQL选择合适的Redo大小



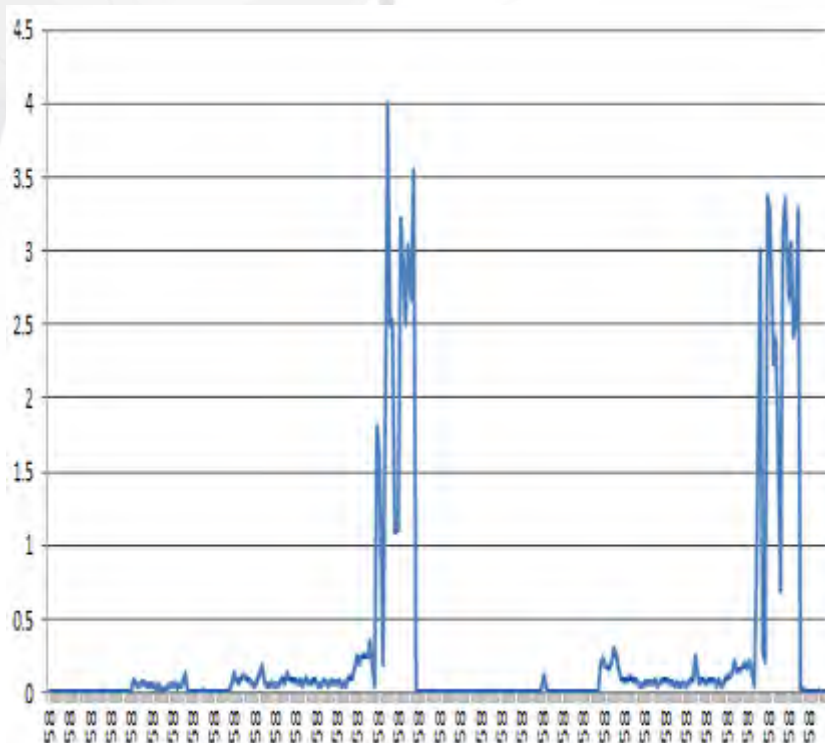
# 5.2 MySQL:主从延迟的测试（延迟回落）



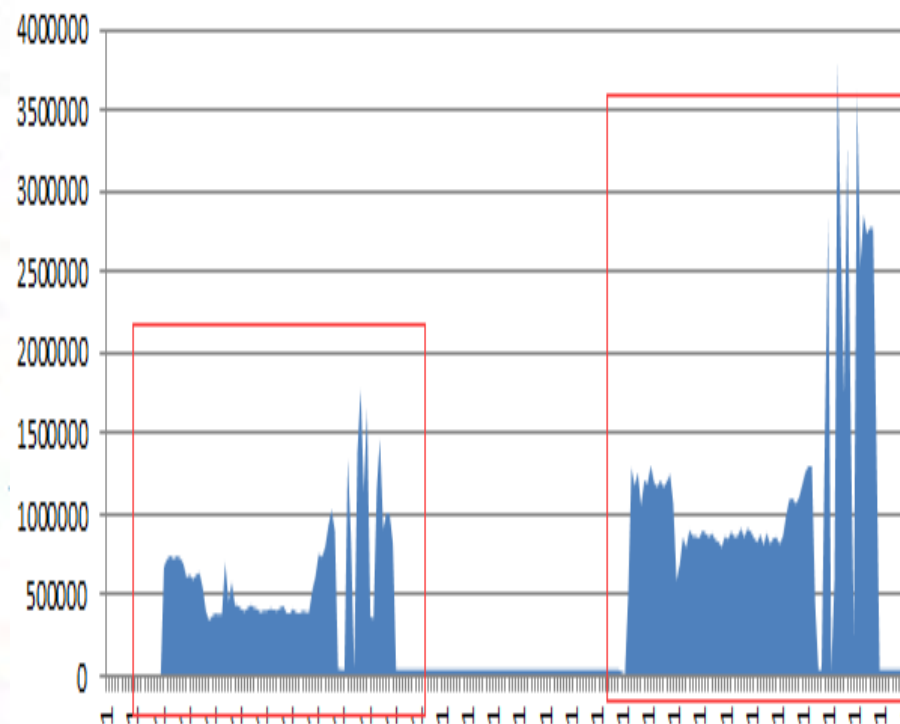


# 5.3 Oracle压缩归档

## CPU使用率对比



## 压缩前后带宽流量对比





## 5.4 MySQL优化案例

- select account from **t\_fund\_info**  
where money >=300 and **account not in**  
(select distinct(login\_account) from  
t\_user\_login\_record where login\_time >='2016-06-01')  
into outfile '/tmp/data.txt';
- 执行时间: **61213s**
- 表**t\_fund\_info**数据量近200万, 存在一个主键在  
**id**列, 唯一性索引在**account**上。

# SQL优化案例1-MySQL反连接

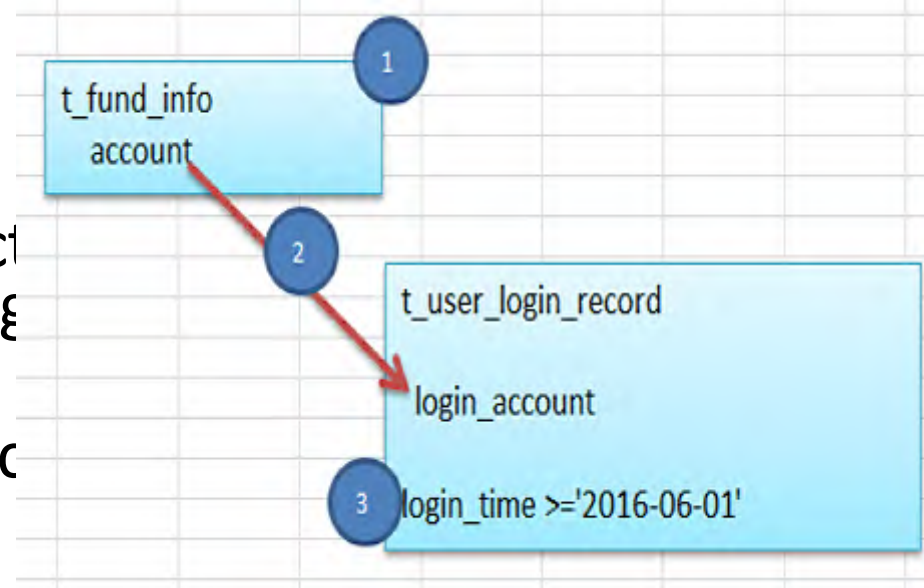
- > select count(\*)from t\_fund\_info where money >=300;
- 执行时间1秒内
- select distinct(login\_account) from t\_user\_login\_record where login\_time >='2016-06-01';  
执行时间1分钟。
- explain extended
- Show warnings;

# SQL优化案例1-MySQL反连接

```
select test.t_fund_info.account AS account
  from test.t_fund_info
 where (
    (test.t_fund_info.money >= 300)
    and
    (not (
      (test.t_fund_info.account,
        (select distinct 1
         from test.t_user_login_record
        where ((test.t_user_login_record.login_time >= '2016-06-01') and
              (
                ((test.t_fund_info.account) = test.t_user_login_record.login_account)
                or
                isnull(test.t_user_login_record.login_account)
              )
            )
        )
      having(test.t_user_login_record.login_account)
    )
  )
)
```

# SQL优化案例1-MySQL反连接

- 改进方案:
- 1.创建临时表
- create table test\_tab as select login\_account from t\_user\_log where login\_time >='2016-06-01';
- create table test\_tab1 as select t\_fund\_info from test\_tab where money >=300;
- 2.创建索引
- create index ind\_tmp\_login\_account on test\_tab(login\_account);
- 11364 rows in set (2.52 sec)



## 5.5 引申半连接，反连接

- in,exists
- 在不同版本有着不同的变化
- 5.6+, exists会转换为semi join的方式
  
- Not in ,not exists
- not in内部转换为not exists的处理方式，而not exists的方式在存在索引和不存在，执行计划差别很大

# 六.小结

## 没有主见，就没有远见

- 做了一年工单,其实太能忍受,不是一件好事
- 两类复杂问题
  - 简单问题复杂化,
  - 问题本身就比较复杂
- “笑林广记”里面写的笑话:一个聋子看别人放炮仗
- 解决方法: 抛开现实的实现, 去挖掘问题的本质。
- 不是 实现简单=成功, 而是: 逻辑清晰=成功。



# 情怀和沟通

- 功利化、实用化和过分细节化了
  - 中美教育的一些差距 Deversity和“根红苗正”
  - 人文情怀
- 
- 沟通
  - 你肯定知道你自己一路走来的艰辛，可是不要指望任何人能脑补出更多的细节。

# 别错把平台当成你的本事

- 常期对话大咖带来的虚无自信心应当克制。衬托他人光芒只是锦上添花，并不能照亮自己前路。
- 聪明之人，清醒地明白，哪些是自己的能力，哪些只是自己所在的平台带来的福利
- 在大平台拿来的资源，其实没什么好炫耀的。毕竟，离开了这个平台，你还剩下的东西，才是你真正的本事。
- <http://mp.weixin.qq.com/s/T08mSYw9KPIfmM-k5ocBtQ>



# THANKS

SequeMedia  
盛拓传媒

IT168.com

ITPUB

ChinaUnix