SDCC 2017深圳站
大数据技术实战峰会

CSDN

Greenplum应用实践及性能优化实践

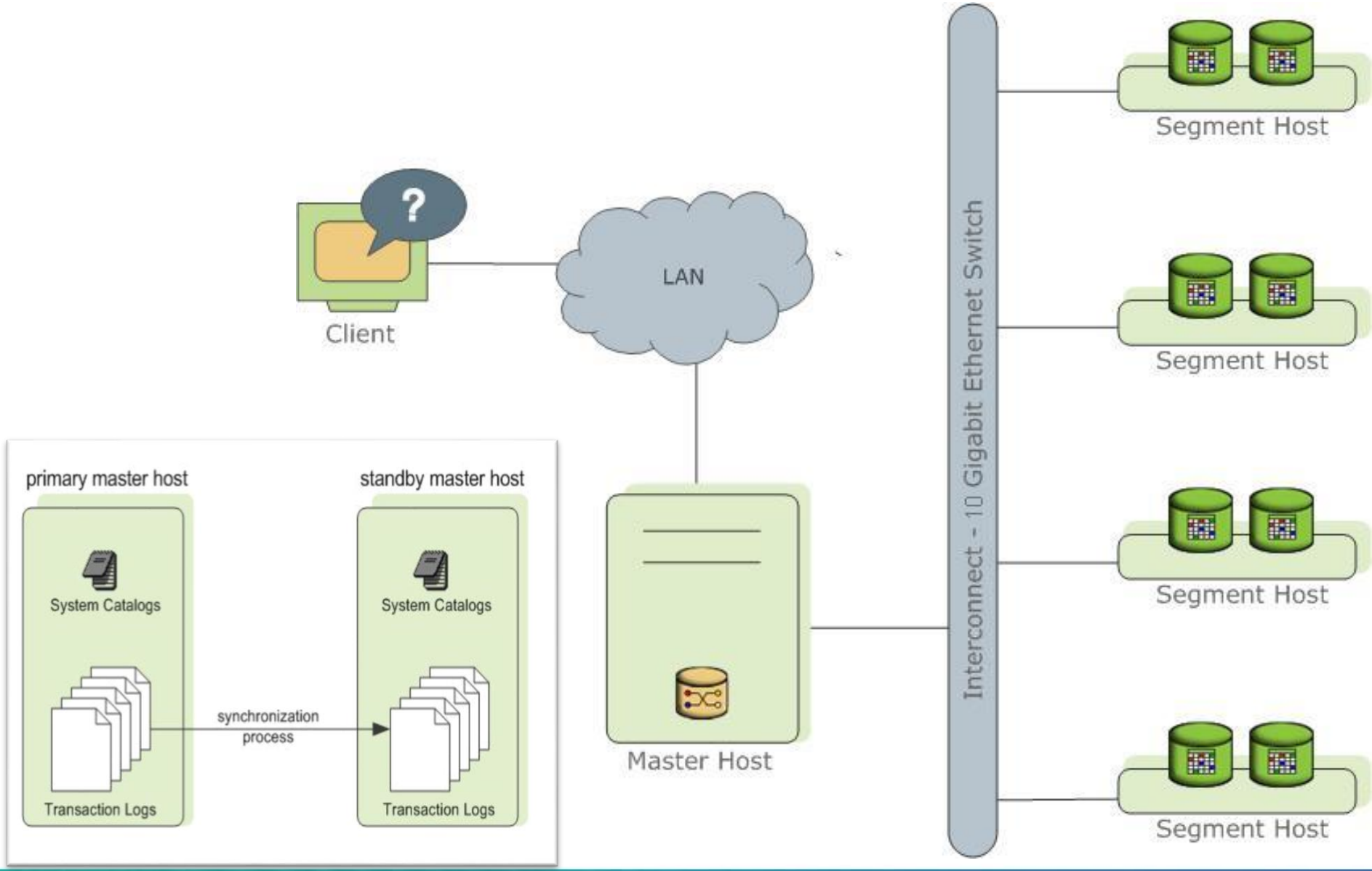
周雷皓

目录

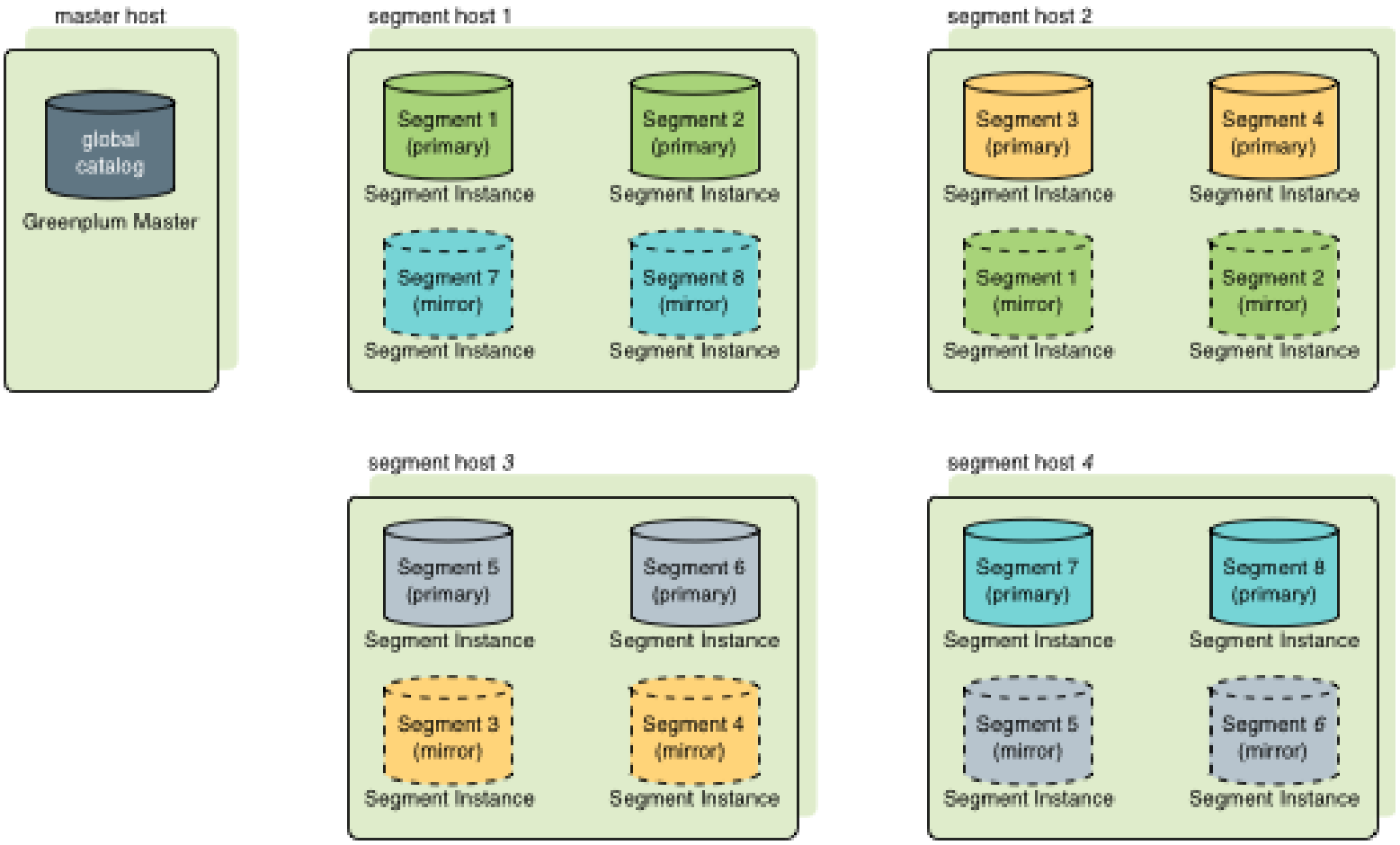
CONTENTS

目录

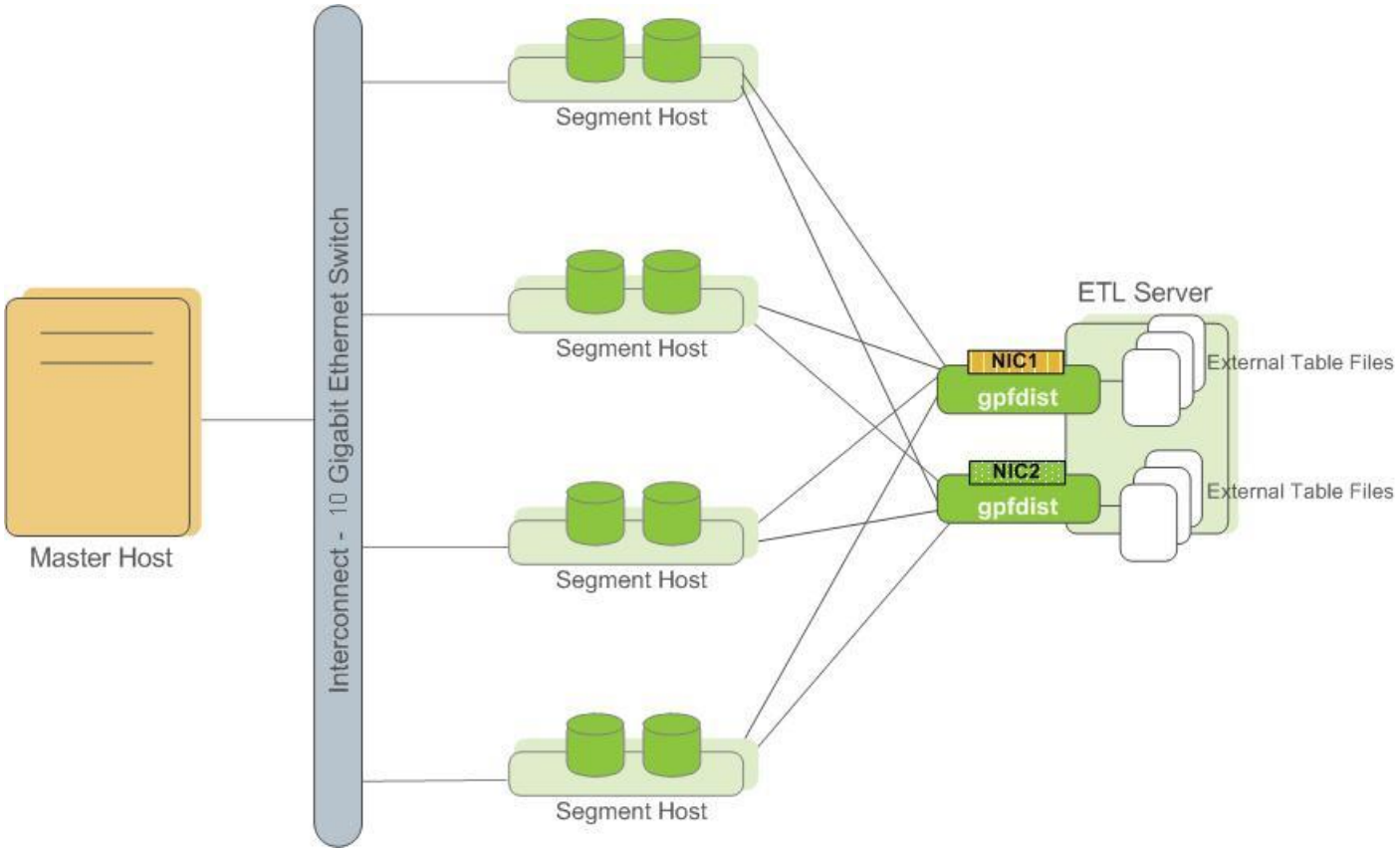
CONTENTS


Client
LAN
Interconnect - 10 Gigabit Ethernet Switch
Segment Host
Segment Host
Segment Host
Segment Host
Master Host
primary master host
standby master host
System Catalogs
System Catalogs
synchronization process
Transaction Logs
Transaction Logs

Greenplum Segment



特性：并行数据加载


Master Host
Interconnect - 10 Gigabit Ethernet Switch
Segment Host
Segment Host
Segment Host
Segment Host
ETL Server
NIC1
gpfdist
NIC2
gpfdist
External Table Files
External Table Files

目录

CONTENTS

- 极个别节点存储资源不足

```
# 查看数据分布情况
select gp_segment_id,count(*) from gp_test group by gp_segment_id order by 1;
```

- SQL执行效率差
  - 最慢的节点会成为系统的瓶颈
  - 不必要的广播和重分布引起性能问题

一个水桶无论有多高，它盛水的高度取决于其中最低的那块木板。

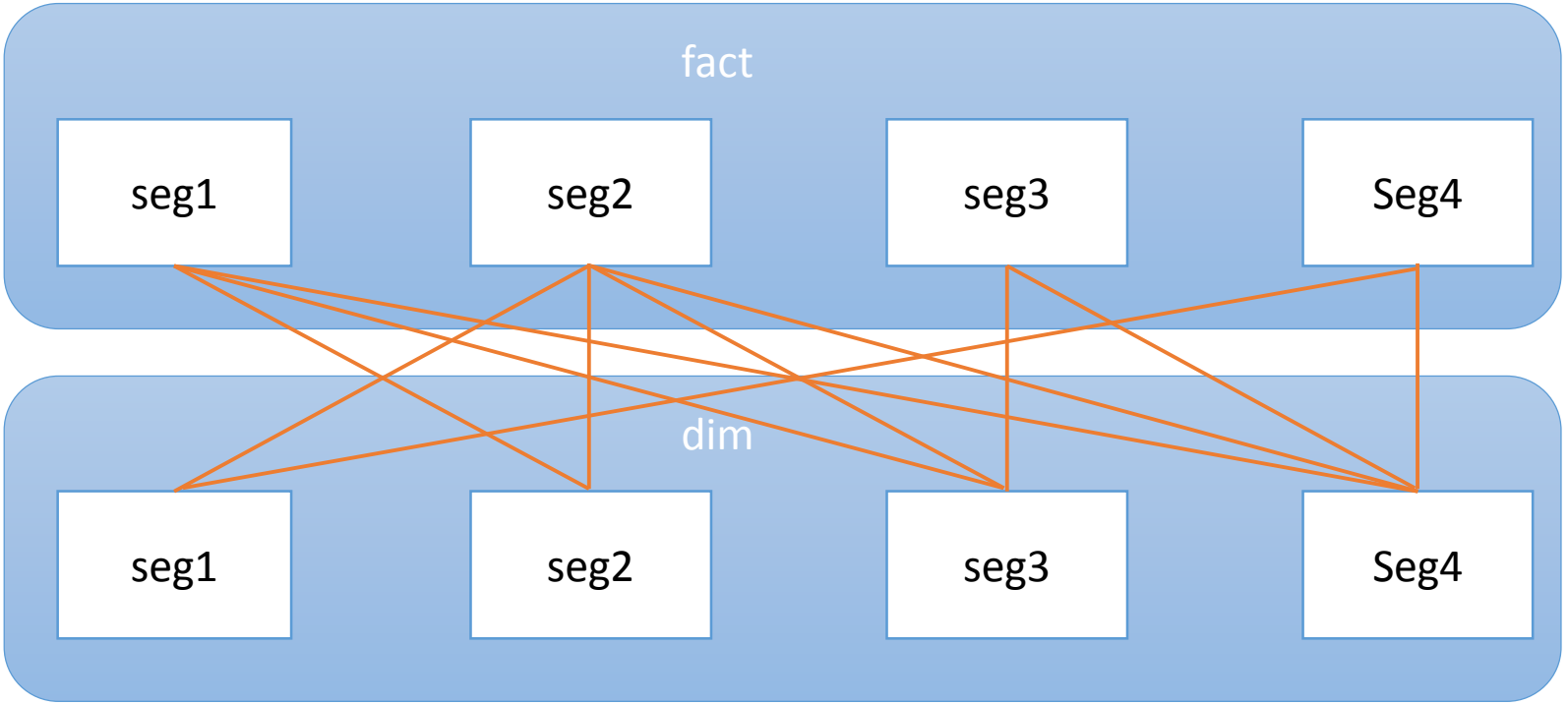
——木桶理论

数据分布：一个简单的例子

```
select count(1) from dim d join fact f on d.shop_id =f.shop_id;
```

fact: distributed by(city_id)
Dim: distributed by(shop_id)

重分布或者广播

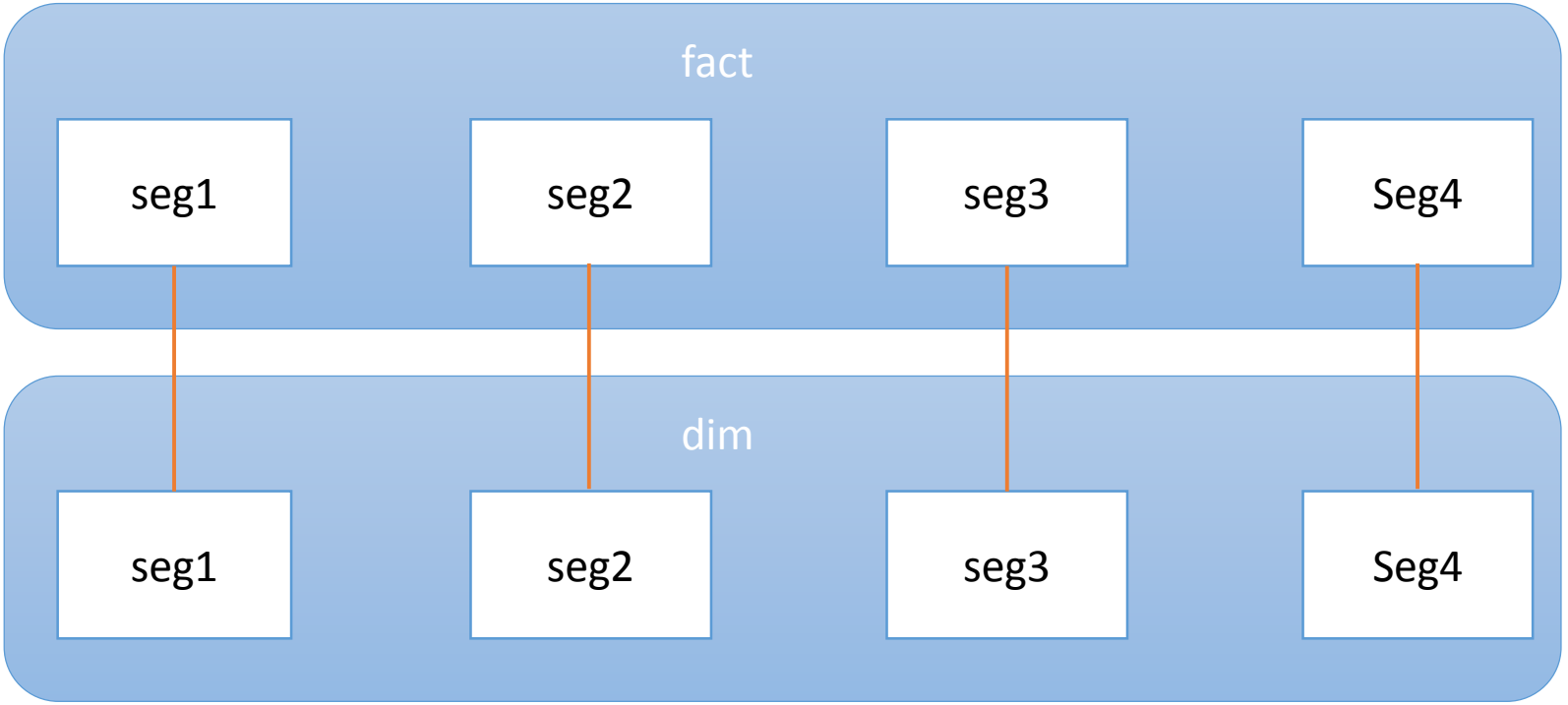


数据分布：一个简单的例子（续）

```
select count(1) from dim d join fact f on d.shop_id =f.shop_id;
```

fact_order: distributed by(shop_id)
Dim_business: distributed by(shop_id)

Segment内部join操作



1. 选择需要Join的列，优先考虑并发高的列
2. 防止数据倾斜
3. 避免条件字段
4. 考虑一下随机分布

- 数据移动
  - Broadcast Motion：每个节点向其它节点广播需要发送的数据
  - Redistribute Motion：利用join列值hash不同，将筛选后的数据在其他segment重新分布
  - Gather Motion：每个节点将join后的数据发到单节点去处理（一般是master）
- 关联操作
  - Hash Join：O(n)
  - Merge Join：O(n*logN)
  - Nested Loop：O(N*N)
- 两种聚合
  - HashAggregate
  - GroupAggregate

执行计划：例子

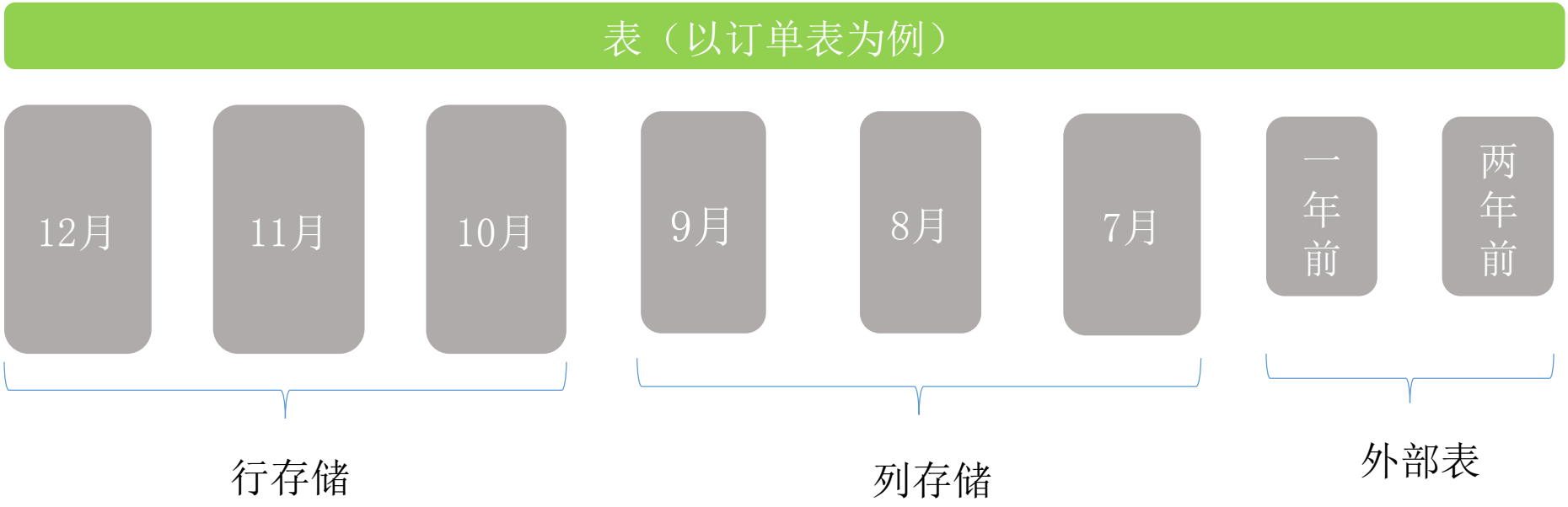

```
Query Plan:
    -> Aggregate (cost=543353.77..543353.78 rows=1 width=8)
          Slice 2
       -> Gather Motion 232:1 (cost=543351.42..543353.76 rows=1 width=8)
             -> Aggregate (cost=543351.42..543351.43 rows=1 width=8)
                   -> Hash Join (cost=155446.02..524212.24 rows=32999 width=0)
                      Hash Cond: f.shop_id = d.shop_id
                         Slice 1
                      -> Redistribute Motion 232:232 (cost=0.00..264111.23 rows=15447 width=8)
                         Hash Key: f.shop_id
                            -> Append (cost=0.00..192438.41 rows=15447 width=9)
                               ->  Append-only Scan on fact_ ' ' . day_key_20170520 f  (cost=0.00..68315.92 rows=8065 wi
                                  -> Seq Scan on fact_ _1_prt_ _' ,_key_20170521 f (cost=0.00..124122.49 rows=7383 width=9)
                      -> Hash (cost=115284.90..115284.90 rows=13849 width=8)
                            -> Seq Scan on dim_' ' d (cost=0.00..115284.90 rows=13849 width=8)
```

```
Query Plan:
    -> Aggregate (cost=471335.92..471335.93 rows=1 width=8)
          Slice 1
       -> Gather Motion 232:1 (cost=471333.56..471335.90 rows=1 width=8)
             -> Aggregate (cost=471333.56..471333.57 rows=1 width=8)
                   -> Hash Join (cost=155446.02..452191.16 rows=33005 width=0)
                      Hash Cond: f.shop_id = d.shop_id
                      -> Append (cost=0.00..192072.50 rows=15450 width=8)
                            -> Append-only Scan on fact_ '- _1_prt_ ' _ ,_key_20170520 f (cost=0.00..67891.92 rows=8065 width=8)
                            -> Seq Scan on fact_ '- _1_prt_ _' ,_key_20170521 f (cost=0.00..124180.58 rows=7386 width=8)
                      -> Hash (cost=115284.90..115284.90 rows=13849 width=8)
                            -> Seq Scan on dim_ d (cost=0.00..115284.90 rows=13849 width=8)
```

- GPDB基于cost base的查询计划，不正确的统计信息导致错误的执行计划
- 使用ANALYZE命令收集统计信息
- 仅对必要的字段执行收集统计信息，不在join,where,order等操作中的字段无需收集
- EXPLAIN [ANALYZE]分析执行计划
  - 分区裁剪
  - 广播或者重分布

- 一般不需要创建索引
- 对于条件筛选性强的SQL，建议增加索引
- 区分度高的建议使用B-Tree索引，区分度低得建议使用Bitmap索引

- 频繁查询的宽表使用列存，其他表使用行存
- 过多的使用列存，会造成小文件过多
- 对于经常更新的表，使用行存
- 分区表支持行列混合存储
- 不建议通过adhoc查询外部表



- 通过资源队列控制集群资源消耗，避免系统超负荷运行
- 基于ROLE体系的资源队列，为所有ROLE分配资源队列
- 设置队列优先级，HIGH > MEDIUM > LOW
  - 在并发争用CPU资源时，高优先级资源队列可以获得更多的CPU资源
  - 资源队列的缺省优先级为MEDIUM
- 动态资源队列
  - 不同时段各队列的资源不同
  - 不同负载各队列的资源不同

现象

- 用户报查询队列满，监控发现资源利用率很低，怀疑SQL被锁

如何查找原因

- gp_toolkit.pg_stat_activity：大量sql处于waiting状态，怀疑被锁
- gp_toolkit.gp_locks_on_relation：查看被锁状况

解决方法

- 查找被哪个会话锁住，等待结束或者cancel掉该会话

锁冲突关系表

锁模式	ACCESS SHARE	ROW SHARE	ROW EXCLUSIVE	SHARE UPDATE EXCLUSIVE	SHARE	SHARE ROW EXCLUSIVE	EXCLUSIVE	ACCESS EXCLUSIVE
ACCESS SHARE							×	×
ROW SHARE							×	×
ROW EXCLUSIVE					×	×	×	×
SHARE UPDATE EXCLUSIVE				×	×	×	×	×
SHARE			×	×	×	×	×	×
SHARE ROW EXCLUSIVE			×	×	×	×	×	×
EXCLUSIVE		×	×	×	×	×	×	×
ACCESS EXCLUSIVE	×	×	×	×	×	×	×	×

锁冲突关系表（续）

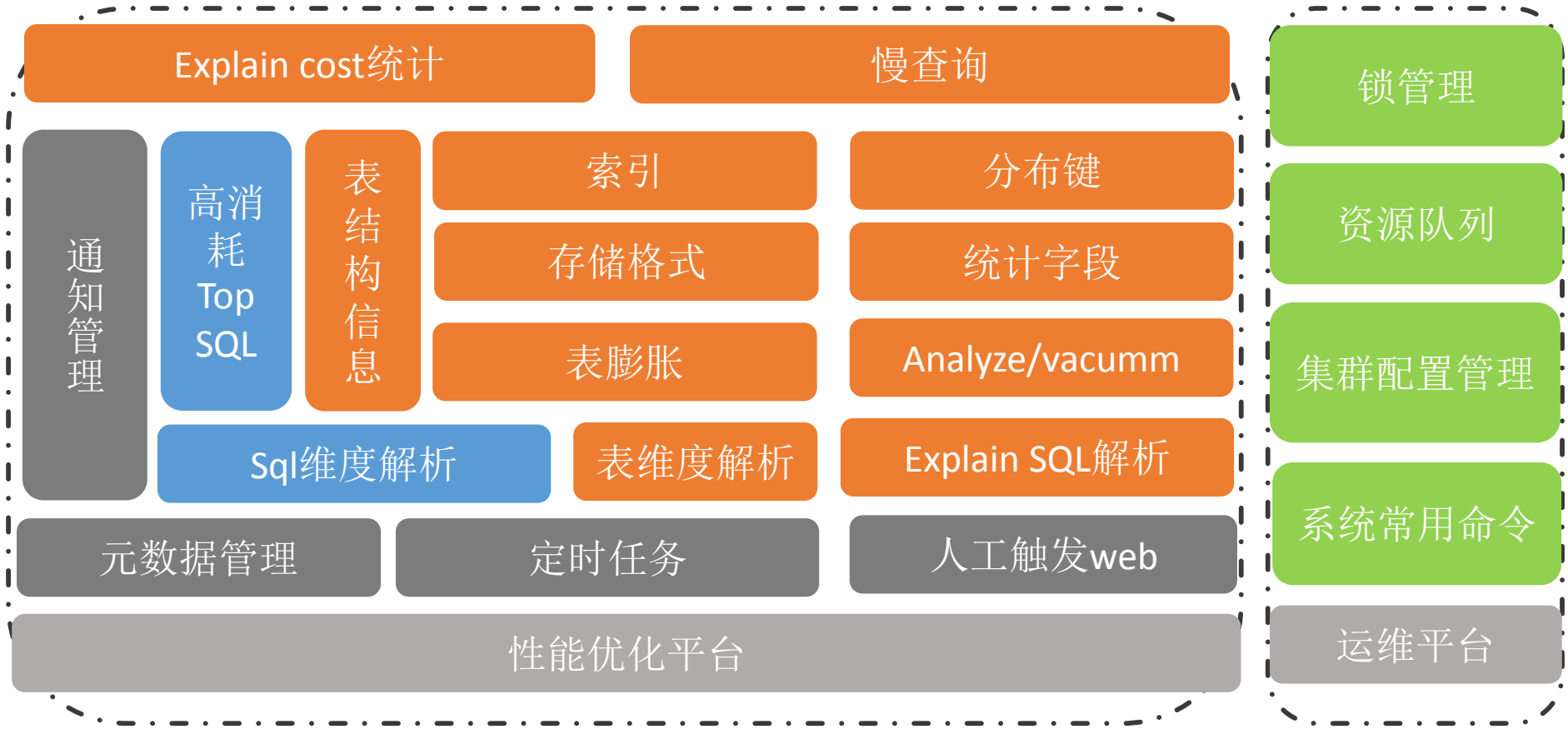
sql语句	select,analyze	select for share	insert, copy	VACUUM (without FULL)	create index	无触发	UPDATE, select for update, DELETE	ALTER TABLE, DROP TABLE, REINDEX, CLUSTER, and VACUUM FULL，Lock命令的默认情况
select,analyze							×	×
select for share							×	×
insert, copy					×	×	×	×
VACUUM (without FULL)				×	×	×	×	×
create index			×	×	×	×	×	×
无触发			×	×	×	×	×	×
UPDATE, select for update, DELETE		×	×	×	×	×	×	×
ALTER TABLE, DROP TABLE, REINDEX, CLUSTER, and VACUUM FULL，Lock命令的默认情况	×	×	×	×	×	×	×	×

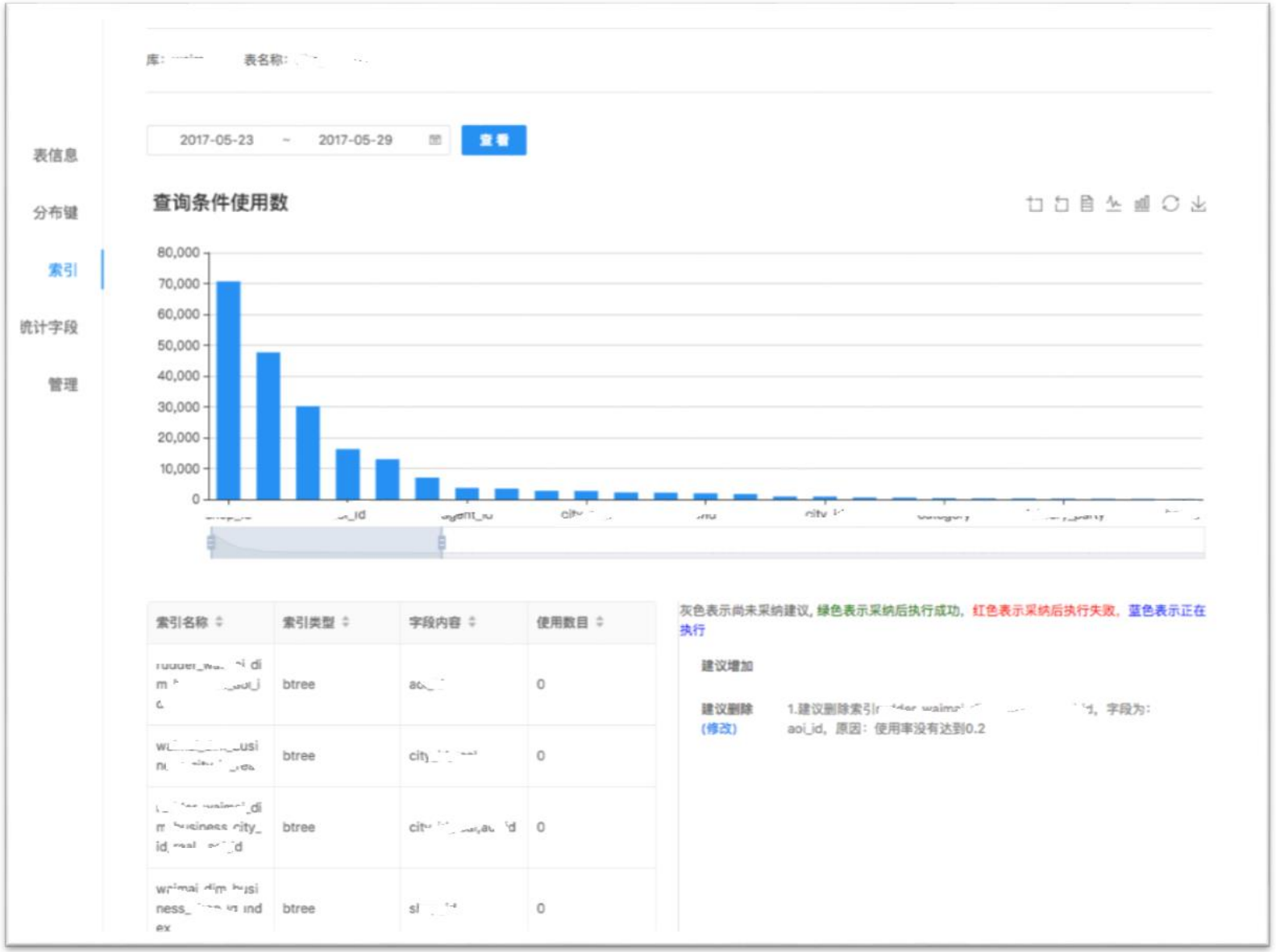
- 系统维护：
  - 对系统表定期做vacumm analyze，对经常更新的表做vacumm
  - Down节点自动拉起，IDLE连接定时查杀
  - 参数调优（深入理解GPDB，系统表）
- SQL优化
  - 分布键（数据均匀分布）
  - 统计信息（正确）
  - 高耗SQL优化
  - 慢查询优化
- 资源队列，合理的资源划分，动态调整

目录

CONTENTS

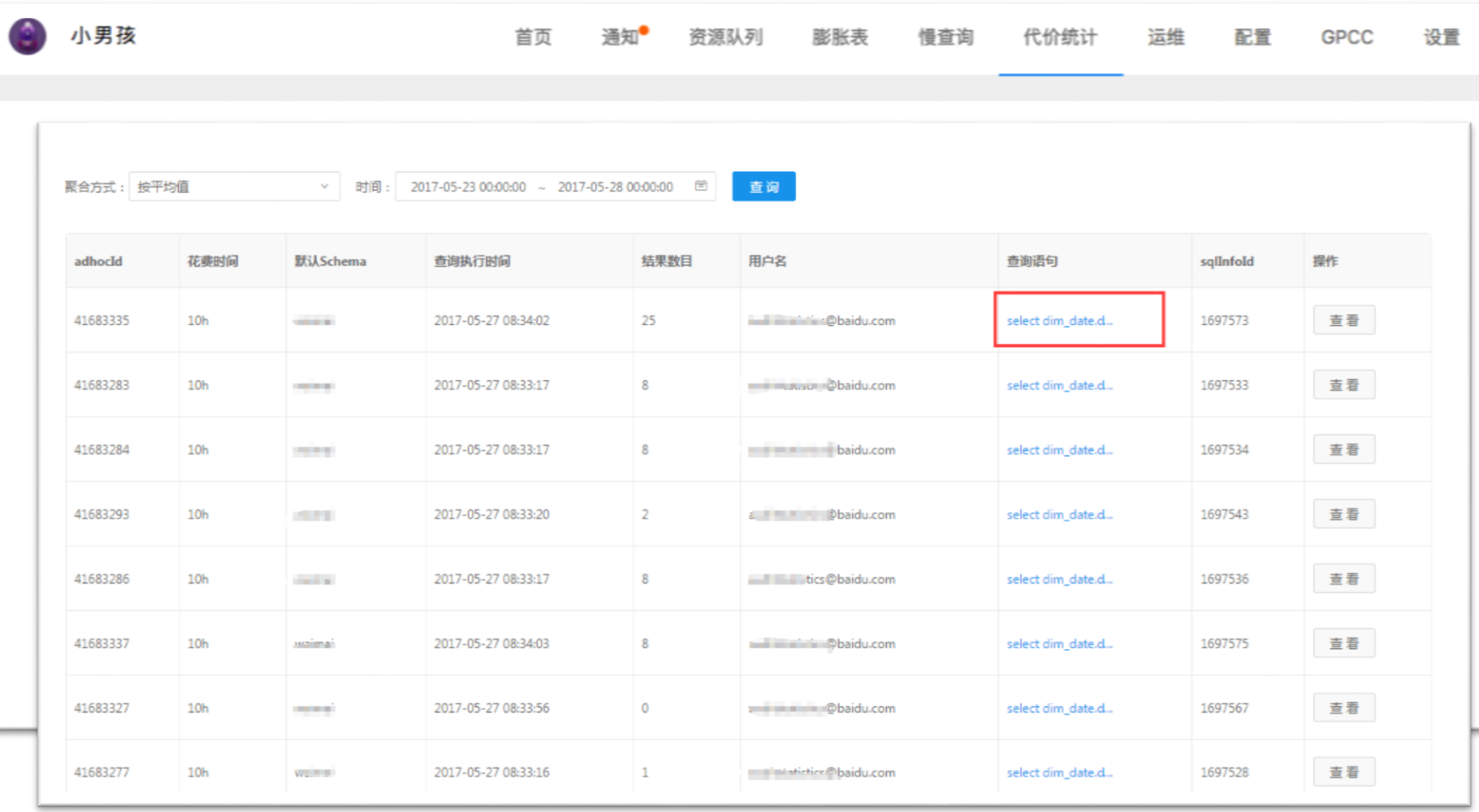
平台化：小男孩架构


Explain cost统计
慢查询
锁管理
通知管理
高消耗Top SQL
表结构信息
索引
分布键
资源队列
存储格式
统计字段
表膨胀
Analyze/vacumm
集群配置管理
Sql维度解析
表维度解析
Explain SQL解析
系统常用命令
元数据管理
定时任务
人工触发web
性能优化平台
运维平台


表信息
分布键
索引
统计字段
管理
2017-05-23
2017-05-29
查看
查询条件使用数
80,000
70,000
60,000
50,000
40,000
30,000
20,000
10,000
0
索引名称
索引类型
字段内容
使用数目
btree
0
灰色表示尚未采纳建议，绿色表示采纳后执行成功，红色表示采纳后执行失败，蓝色表示正在执行
建议增加
建议删除
(修改)
1.建议删除索引
，字段为:
aoi_id，原因：使用率没有达到0.2


详情信息
库名
表名
锁模式
进程id
用户名
始时间
segmentId
查看
操作
AccessExclusiveLock
6409
-1
被本锁阻塞的锁
导致本锁阻塞的锁
删除SessionID
常用命令
命令日志
当前锁信息
用户名：
库名：
表名：
查询
用户图表
grantedCount
waitingCount
sumCount
表信息图表

小男孩：代价统计/慢查询


小男孩
首页
通知
资源队列
膨胀表
慢查询
代价统计
运维
配置
GPCC
设置
聚合方式：
按平均值
时间：
2017-05-23 00:00:00 ~ 2017-05-28 00:00:00
查询
adhocId
花费时间
默认Schema
查询执行时间
结果数目
用户名
查询语句
sqlInfoId
操作
41683335 10h 2017-05-27 08:34:02 25 @baidu.com select dim_date.d... 1697573 查看
41683283 10h 2017-05-27 08:33:17 8 @baidu.com select dim_date.d... 1697533 查看
41683284 10h 2017-05-27 08:33:17 8 baidu.com select dim_date.d... 1697534 查看
41683293 10h 2017-05-27 08:33:20 2 @baidu.com select dim_date.d... 1697543 查看
41683286 10h 2017-05-27 08:33:17 8 tics@baidu.com select dim_date.d... 1697536 查看
41683337 10h 2017-05-27 08:34:03 8 @baidu.com select dim_date.d... 1697575 查看
41683327 10h 2017-05-27 08:33:56 0 @baidu.com select dim_date.d... 1697567 查看
41683277 10h 2017-05-27 08:33:16 1 @baidu.com select dim_date.d... 1697528 查看

SDCC 2017深圳站
大数据技术实战峰会

CSDN

THANKS

http://waimai.baidu.com