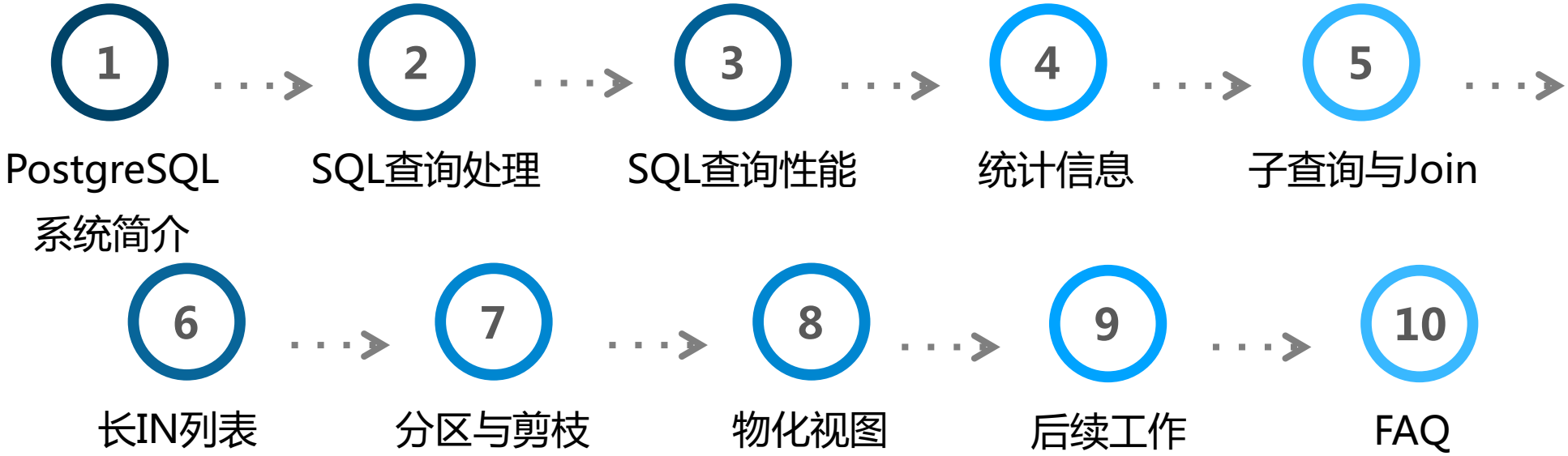
PostgreSQL数据库查询调优分享

孙旭
@tencent

目录

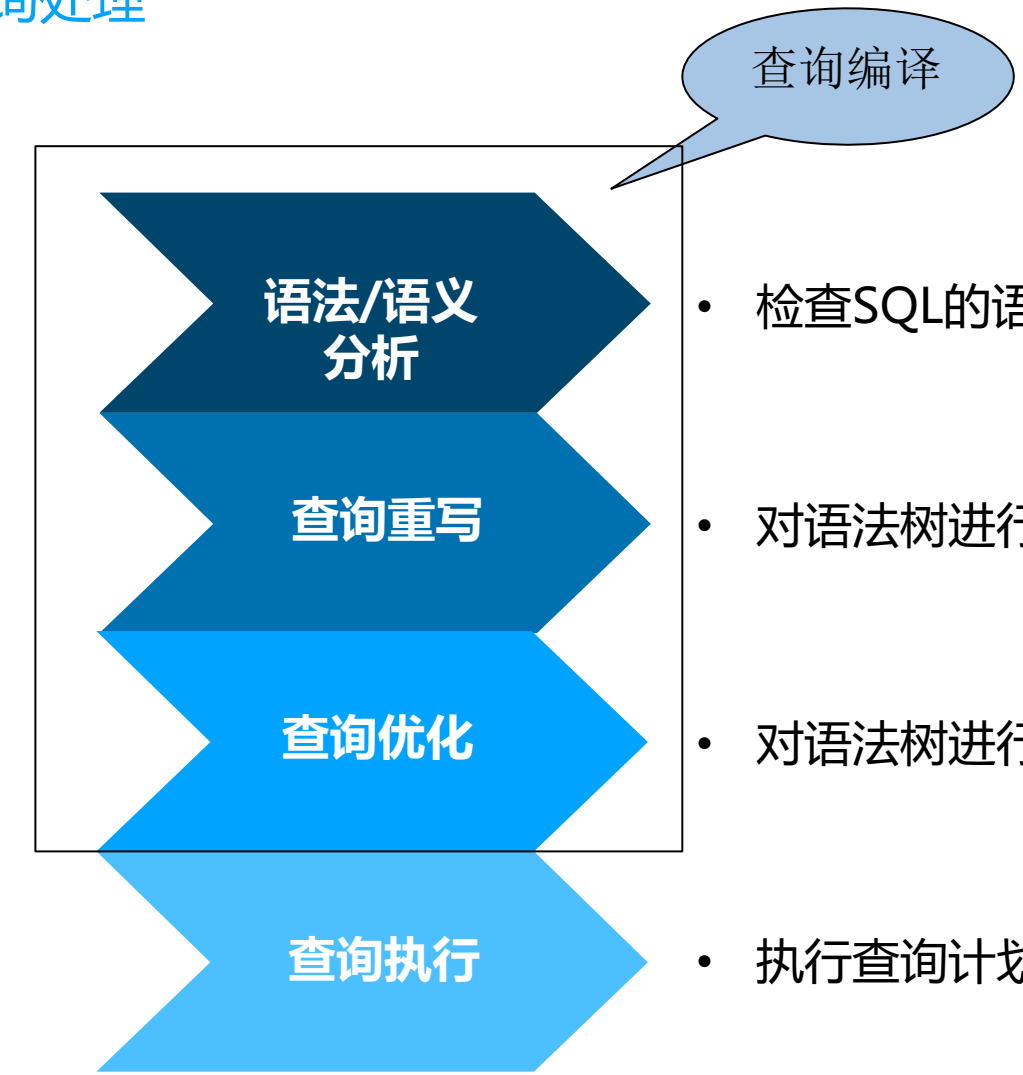

1
PostgreSQL
系统简介
2
SQL查询处理
3
SQL查询性能
4
统计信息
5
子查询与Join
6
长IN列表
7
分区与剪枝
8
物化视图
9
后续工作
10
FAQ

腾讯云

PostgreSQL

- 起始于1986年，目前是世界上先进的开源数据库产品，实现了多种强大并且先进的功能特性。可以在Windows、Linux、Unix平台上运行
- 可以通过插件的方式对数据库的功能进行扩展
- 目前版本演进到10.x

SQL查询处理

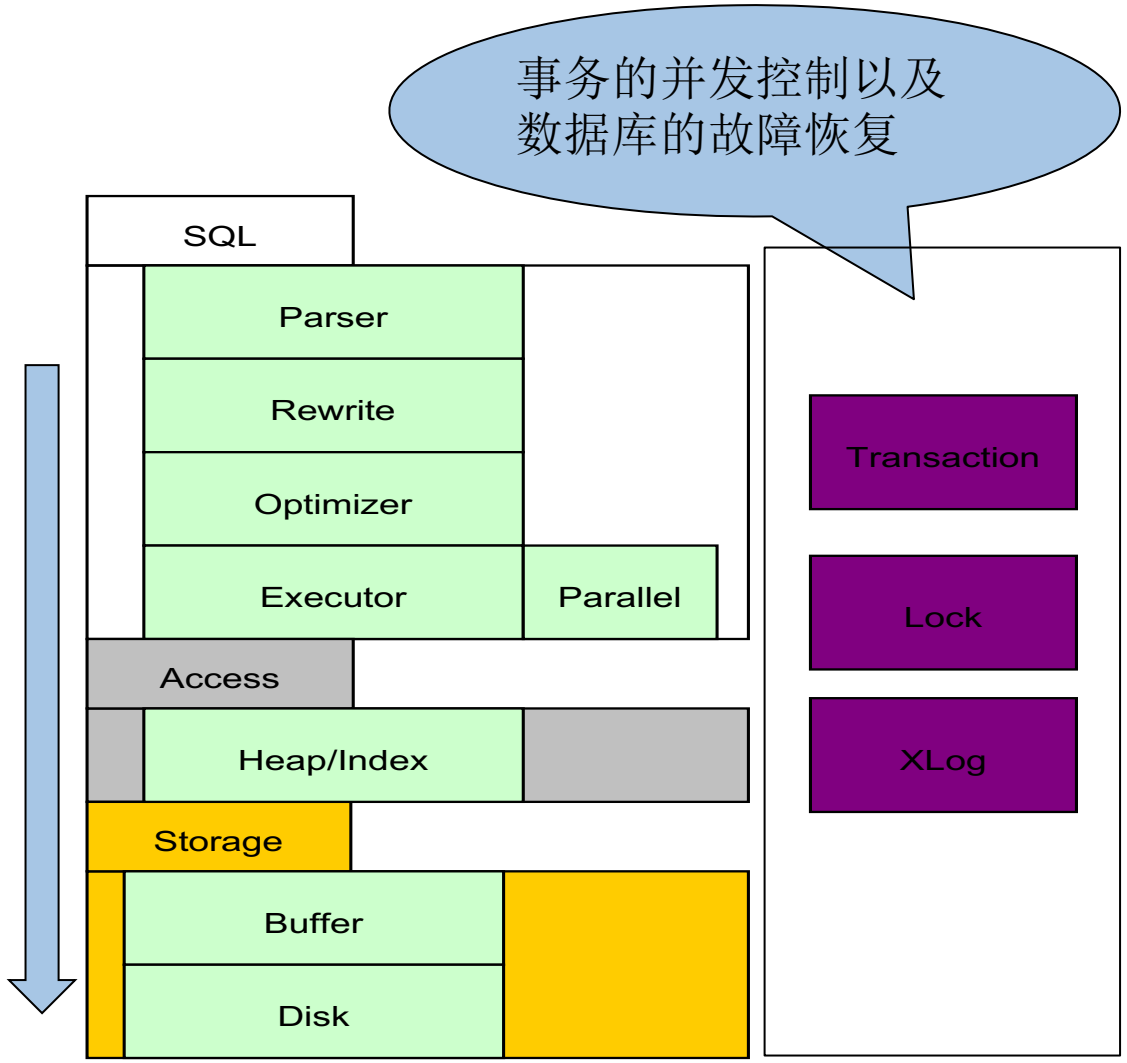


- 检查SQL的语法语义，生成语法树
- 对语法树进行查询变换
- 对语法树进行查询优化，变成查询计划
- 执行查询计划

腾讯云

PostgreSQL查询执行的生命周期

- 启动事务
- 查询进入SQL模块，生成查询计划
- 执行查询计划，扫描数据（加锁，WAL）
- 返回结果



SQL查询性能

腾讯云

数据库性能

- 查询的响应时间 - 用户提交查询，到查询返回结果
- 查询吞吐量 - 每秒（读写）事务数、每秒（读写）查询数

影响性能因素

- CPU - 计算量大，CPU使用率高
- IO - IO量大，随机IO太多等
- 网络 - 通信传输量大

性能调优

- 数据库层面：调整数据库内内核参数，干预查询处理过程
- 操作系统层面：调整OS系统内核参数，保证系统资源充分利用

腾讯云

提升响应时间

- CPU - 减少计算量
  - 数据库层面：选用更好算法，开启并行计算等
- IO - 减少数据读（写）
  - 数据库层面：压缩
  - OS参数设置：磁盘调度算法，磁盘预读
- 网络 - 减少网络流量

提升吞吐量

- 提升响应时间为前提
- 增加并发

PostgreSQL:

- file advise、O_DIRECT
- Huge Page

腾讯云

语法语义

- 语法分析过程是耗费CPU —— 一般不会有性能问题
- 语义分析时会查询系统表 —— 可能会有性能问题

查询重写

- 按照重写规则对查询进行变换
- 保持SQL语义的前提下，进行查询变换

查询优化

- 单表执行路径选择 —— 索引较多
- Join路径选择 —— 表较多

查询执行

- 使用高效的执行算法运行查询计划 - 生成较差的查询计划

腾讯云

生成查询计划阶段 - 性能问题

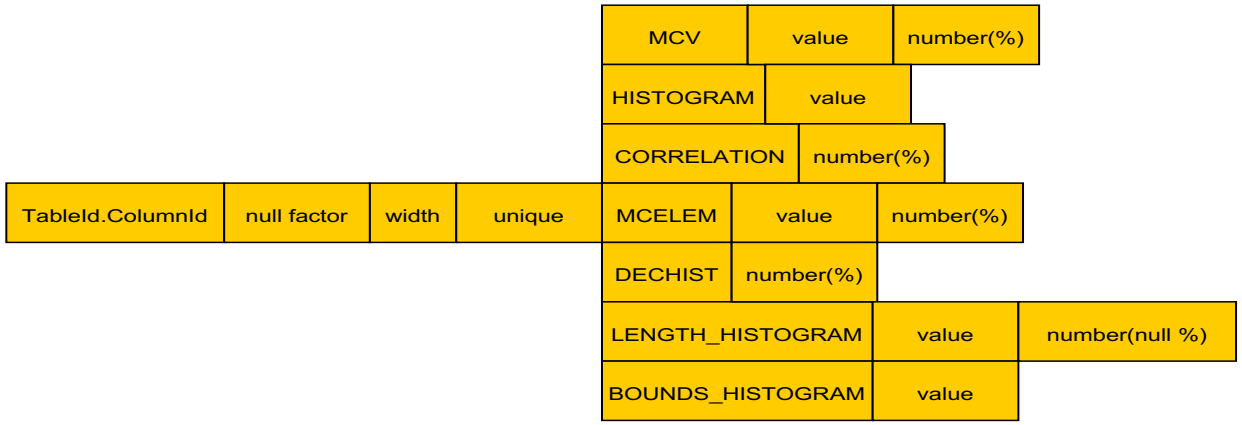
- 查询计划缓存
- 执行计划生成过程中有效的减少路径的搜索空间

查询执行阶段 - 性能问题

- 依赖优化器生成优秀的查询计划，事实并不总是这样：
  - 没有实现高效的执行路径（Path）
  - 估算不准确，没有选择高效的查询计划
- 依赖执行器本身的算法，执行效率

统计信息

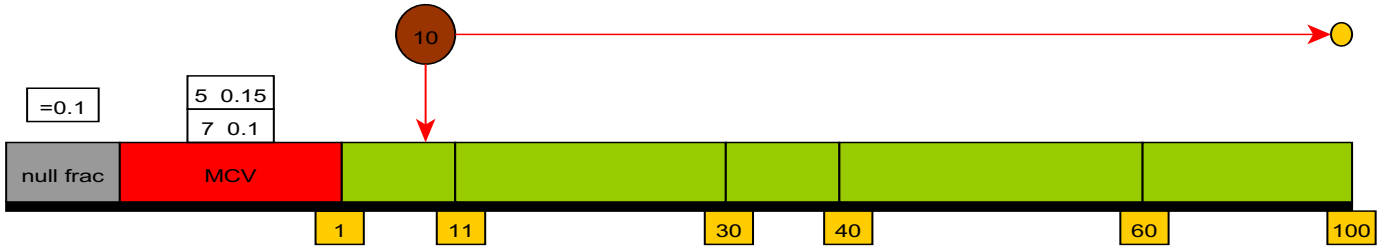
- PostgreSQL优化器使用统计信息计算操作符选择率，并估算查询计划的的代价
- PostgreSQL的统计信息使用的是直方图进行，对每一列存储如下结构：



- 统计信息收集：
  - PostgreSQL对表元组进行取样
  - 取样数=直方图个数（默认100）的300倍（取样数与直方图桶数是弱相关 - In 关系）

统计信息

- 计划算子代价 = CostModel(rows, cpu_cost_unit, io_cost_unit)，其中：
  - rows = table total rows * filter selectivity
- 以a > 10，使用统计信息计算选择率：
  - (1 - 0.1 - (0.15 + 0.1)) * (((11-10)/(11-1) + 4))/5)



- PostgreSQL通过ANALYZE收集统计信息（可以按列收集）。可以调节的参数：
  - default_statistics_target - MCV & histogram
  - ALTER TABLE ... ALTER COLUMN SET STATISTICS/n_distinct

腾讯云

PostgreSQL估算模型假设：

- 数据在列上均匀分布
- 条件相互独立（单表过滤条件或者JOIN条件）
- inclusive principle

行数的估算对查询计划的影响很大

多个JOIN下的估算错误会叠加，也可能抵消

腾讯云

收集统计信息策略

- 在数据加载结束，收集统计信息
- 指定部分列收集统计信息。一般来说：过滤条件，连接条件，分组列上收集

统计信息是否必须收集：

- EXPLAIN判断执行计划是合理的查询计划
- EXPLAIN ANALYZE实际行数和估算行数比值（5 到 10倍）

调整配置参数：

- work_mem：排序使用了磁盘
- enable_index/enable_seqscan/enable_nestloop：可以避免收集统计信息

子查询&Join

腾讯云

- PostgreSQL完全支持子查询
  - 过滤条件
  - 结果列
  - 连接条件
  - 在FROM后面作为Derived table

- 子查询分类
  - 非相关子查询
    - SELECT * FROM t WHERE t1 < (SELECT max(f1) FROM f)
  - 相关子查询
    - SELECT * FROM t WHERE EXISTS (SELECT f1 FROM f WHERE f1=t1)
  - 隔层相关
    - SELECT * FROM t WHERE EXISTS (
      SELECT f1 FROM f WHERE f1 IN (
      SELECT m1 FROM m WHERE m2 = t2))

- 相关子查询的执行算法复杂度是m*n，类似Nested Join

子查询&Join

腾讯云

- 子查询消除 - SemiJoin/AntiJoin
  - IN/NOT IN
  - EXISTS/NOT EXISTS

- PostgreSQL可以将SemiJoin自动转换成普通Join。否则可以尝试修改查询，例如：

```
SELECT * FROM t WHERE EXISTS (SEELCT f1 FROM f WHERE f1=t1)
        等价为：
SELECT t.* FROM t JOIN
        (SELECT DISTINCT f1 AS sub_col FROM f) as dt ON dt.sub_col = t.t1
```

- 遇到无法消除的子查询，例如在结果列中。
  - 唯一标识外层查询的行：主键，分组列

```
SELECT t1, (SELECT count(*) FROM f WHERE f1=t1) FROM t;
        等价为：
SELECT t1, sub_col FROM t
    LEFT JOIN
        (SELECT t.t1 AS sub_id, count(*) sub_col
                    FROM f JOIN t ON f1=t.t1 GROUP BY t.t1) AS dt
  ON dt.sub_id=t.t1;
```

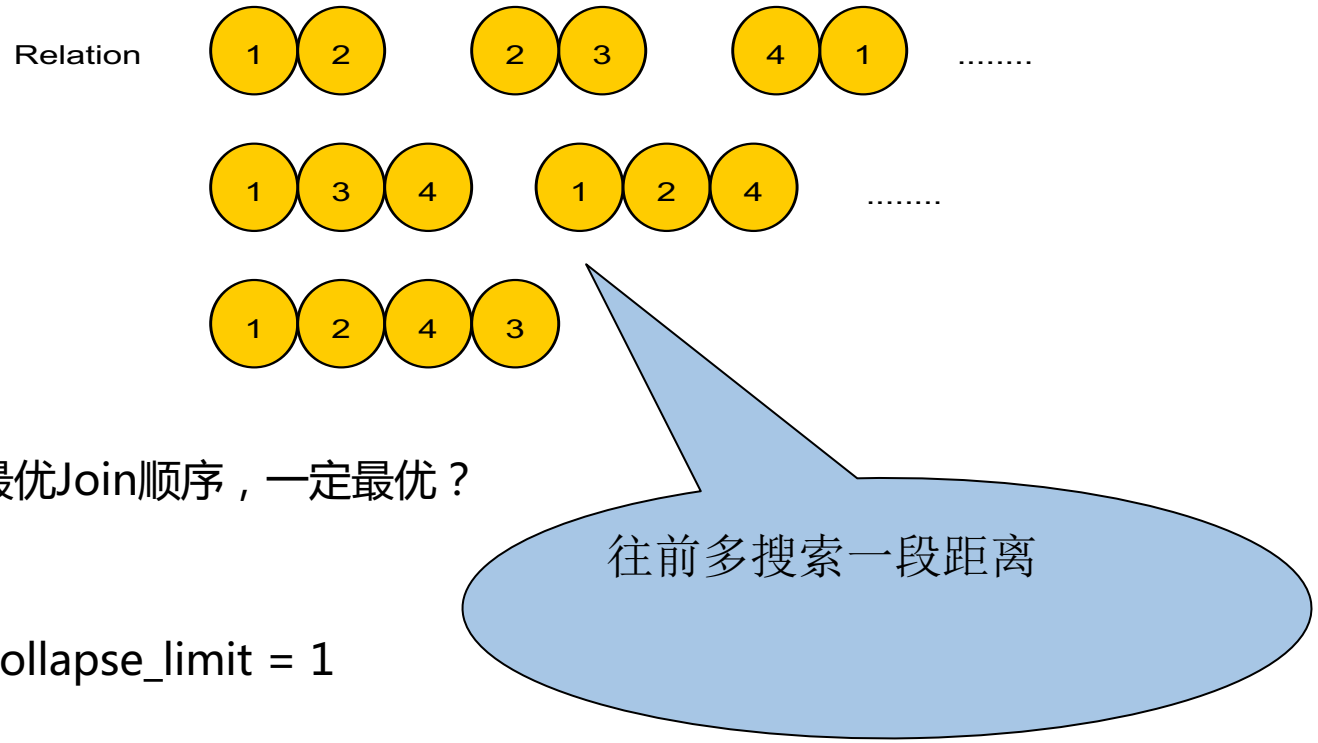
子查询&Join

腾讯云

- 子查询消除的优点
  - 降低算法执行复杂度（CPU，IO），有机会试用更高效的hash join或者merge join完成查询
  - 增加JOIN顺序的搜索空间，有机会选出更好查询计划
- PostgreSQL没有实现对所有子查询的消除
- 查询重写注意需要保证SQL的语义

子查询&Join

- Join顺序搜索空间
  - PostgreSQL提供动态规划算法和遗传算法



- 选择table1和table2作为最优Join顺序，一定最优？

- 手动规定Join顺序。join_collapse_limit = 1

长IN列表

- PostgreSQL对常量IN列表执行是按照顺序查找执行。如下查询:

```
select * from t where t1 in
                    (22, 13, 12, 32, 43, 82, 19, 2, 1, 0, 81, 41, 47, 53)
```

- 将IN转化成Join
  - 优化器自动完成
  - 查询手动改写

```
select * from t,
      (values(22), (13), (12), (32), (43),
            (82), (19), (2), (1), (0), (81), (41), (47), (53)) dt(x)
where t1 = x;
```

腾讯云

分区表

- 按照某列进行分区 - 列表，范围等

剪枝

- 静态剪枝（Static Partition Elimination - SPE）：PostgreSQL支持
  - 分区列与常量比较，如：WHERE p1= 15

```
create table p(p1 int, p2 int) partition by range(p1);
create table p_p1 partition of p for values from (10) to (20);
create table p_p2 partition of p for values from (20) to (30);
create table p_p3 partition of p for values from (30) to (40);
```

- 动态剪枝（Dynamic Partition Elimination - DPE）：PostgreSQL不支持，可以使用PREPARE 语句模拟
  - 分区列与变量比较，如：WHERE p1 = (SELECT max(t1) FROM t)

```
PREPARE stmt as SELECT * FROM p WHERE p1=$1;
execute stmt(15);  -- 假设SELECT max(t1) FROM t返回15
```

腾讯云

Join

- 2个分区表（分区方式相同）在分区列上的连接
  - PostgreSQL的处理是把分区表当成普通来处理
  - Partition Table Join

- 在PostgreSQL数据库，罗列子表分别Join，然后取UNION ALL

```
(select * from p_p1 x join p_p1 y on x.p1=y.p1)
union all
(select * from p_p2 x join p_p2 y on x.p1=y.p1)
union all
(select * from p_p3 x join p_p3 y on x.p1=y.p1)
```

腾讯云

物化视图

- 其行为方式同CREATE TABLE AS，只是多存储了生成数据的查询
- 无法自动刷新（REFRESH MATERIALIZED VIEW ），因此无法对查询进行重写

优点

- 优先计算结果
- 业务只读取物化视图

视图自动刷新

- 像索引一样带来维护代价
- 但是可以支持查询重写

后续工作

腾讯云

将调优方式固化到数据库内核中

优化器更智能

- 提升优化器估算水准
- 在优化器中实现更多的查询重写和变换

执行器更高效

- 实现高效的执行算法，例如DPE，Partition Join

腾讯云

谢谢！

腾讯云
官网：http://cloud.tencent.com
微博：http://weibo.com/qcloud2014