



# 阿里云HybridDB for PostgreSQL 列存优化



嘉宾：范孝剑

花名：康贤

公司：阿里云





# 阿里云HybridDB for PostgreSQL

- 云数据库HybridDB for PostgreSQL基于Greenplum Database开源数据库项目，并由阿里云深度扩展，支持OSS外部表、JSON数据类型、HyperLogLog预估分析、Madlib、PL/Java等功能特性
- 产品网站: <https://www.aliyun.com/product/gpdb>



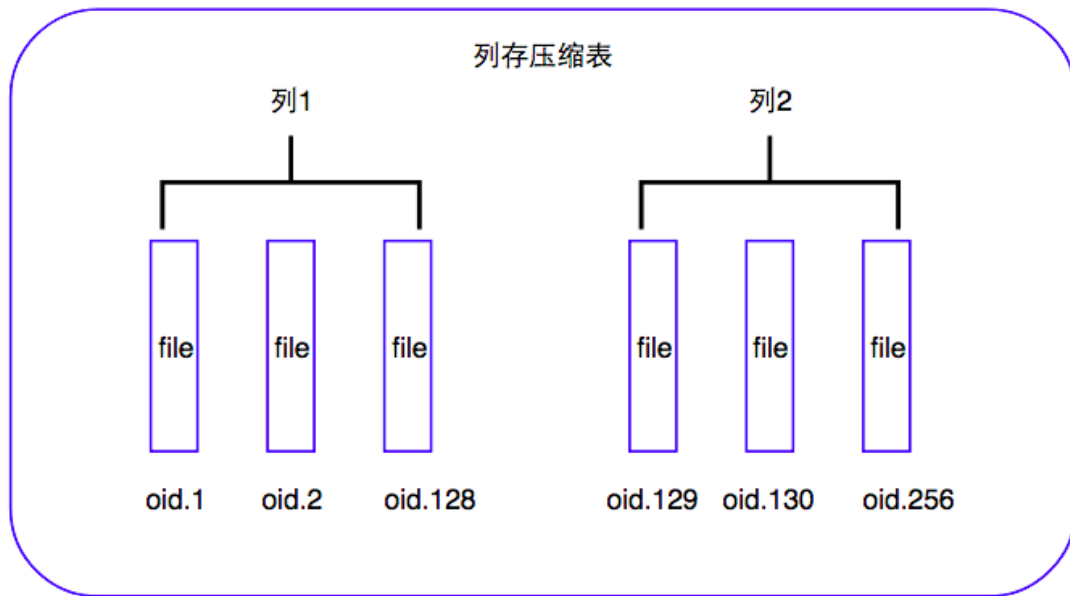
# 列存优化背景

- 存储成本
- 查询性能
- 导入性能



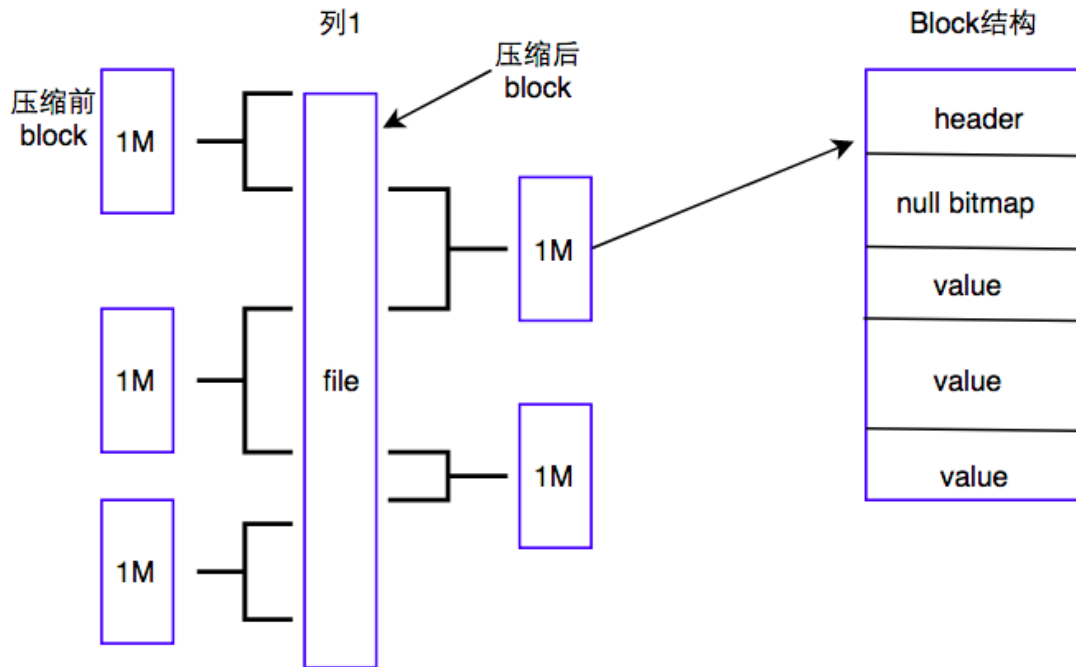
# 列存压缩表原理

- 表组成结构





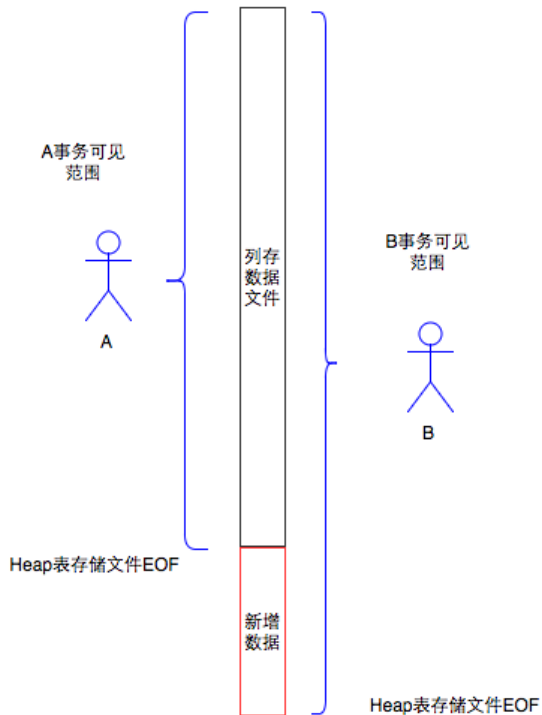
# 文件内部结构





# 事务原理

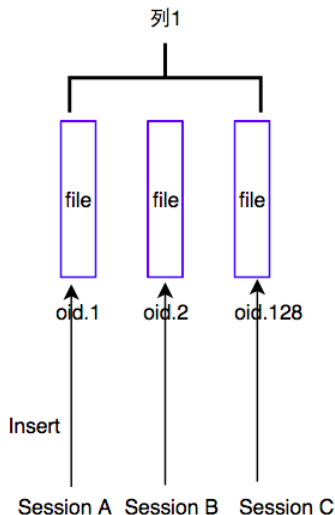
- 辅助heap表





# DML实现

- insert如图所示



- delete和update不会在数据文件中直接更新或者删除，而是使用一张heap辅助表visimap



# 当前机制缺点

- 查询性能低
  - 无索引，每次查询都需要扫描全表，块解压缩消耗大量资源
  - 有索引，严重影响导入性能，数据膨胀





# 优化方案

元信息表

- 元数据搜集
  - 导入性能下降10%内
  - 字符串特殊优化

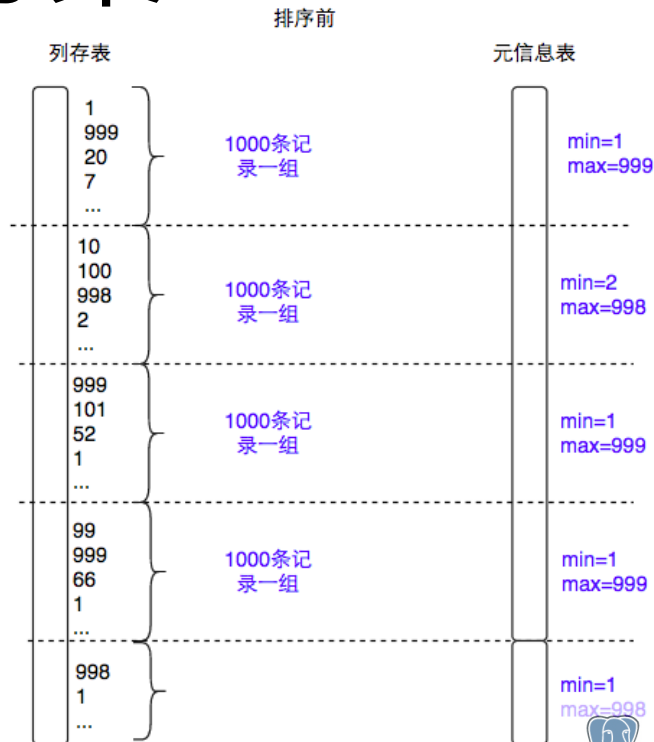
firstGroupNum	metaInfo
0	{min:x1, max:y1}
1000	{min:x2, max:y2}
2000	{min:x3, max:y3}
3000	{min:x4, max:y4}





# 优化方案

- 如果数据平分比较均匀，绝大部分Group的min,max区间很接近并且范围很大，过滤率差，如果解决这个问题呢

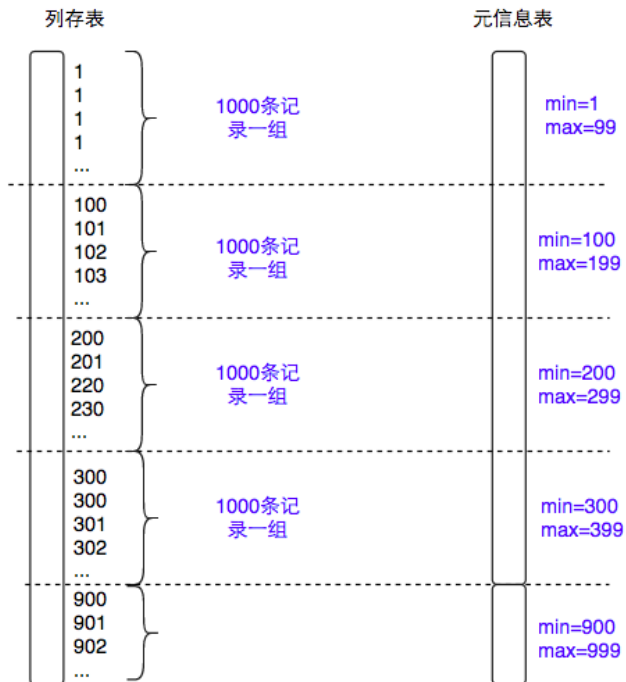




# 优化方案

排序后

- 排序





# 排序键

- 排序键可以让用户将数据按照顺序存储在磁盘文件中
- 两大优势:
  - 加速列存优化，收集的min、max元信息很少重叠，过滤性很好
  - 对于含有order by和group by等需要排序的SQL可以避免再次排序，直接从磁盘中读取出来就是满足条件的有序数据



# 排序键

- 组合排序键
  - order by a,b,c
  
- 交错排序键
  - 每个列都有较好的过滤性



# TPCH测试

- 数据量及规格
  - TPCH 1T 数据量
  - 16个计算节点，每个计算节点16CPU,128G内存,128000 IOPS
- 测试对象
  - TPCH Q6,Q14,Q15



# TPCH测试结果

TPCH	优化前	优化后	提升效果
Q6	4213.940 ms	1286.342 ms	3.27
Q14	6210.617 ms	1726.453 ms	3.59
Q15	13186.588 ms	5351.904 ms	2.46





# 列存优化与索引比较

- 我们在TPCH lineitem表中l\_shipdate列上创建一个btree索引

- 导入性能

不含索引	包含索引
149680.927 ms	476795.551 ms



# 列存优化与索引比较

- 查询性能

TPCH	列存优化	索引查询
Q6	1286.342 ms	1806.992 ms
Q14	1726.453 ms	1689.487 ms
Q15	5351.904 ms	5998.706 ms



# 列存优化与索引比较

- 不同过滤率比较

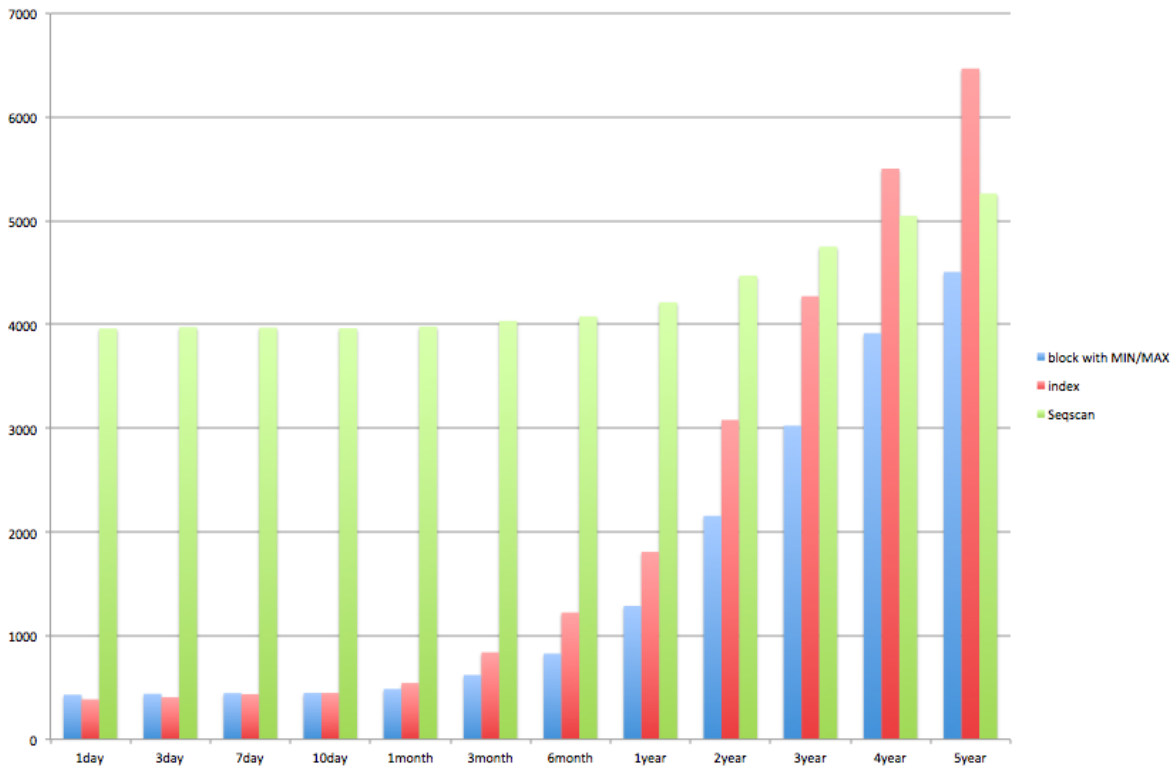
- 每次查询调整Q6  
l\_shipdate的范围  
(1day, 3day, 7day,  
10day,.....)

```
select
    sum(l_extendedprice * l_discount) as revenue
from
    lineitem
where
    l_shipdate >= date '1994-01-01'
    and l_shipdate < date '1994-01-01' + interval '1' year
    and l_discount between 0.09 - 0.01 and 0.09 + 0.01
    and l_quantity < 24
LIMIT 1;
```





# 列存优化与索引比较





Thanks!

