PostgreSQL 空间数据业务优化实践

阿里云

digoal

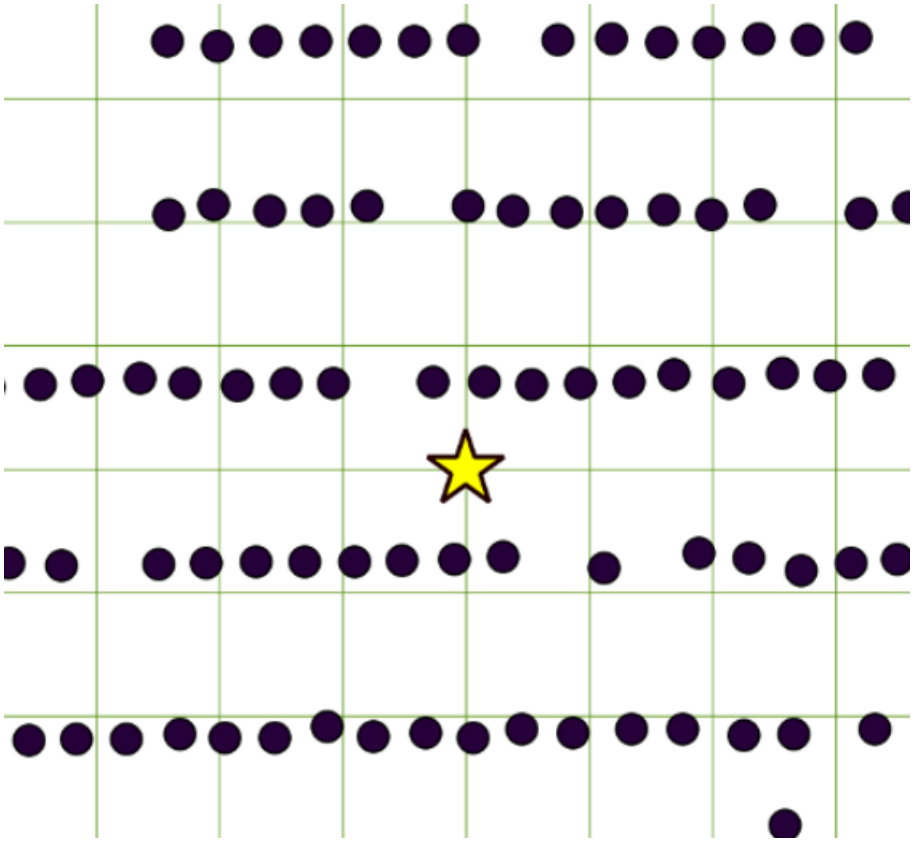
目录

- 空间业务常见需求
- 空间索引
- 优化实践

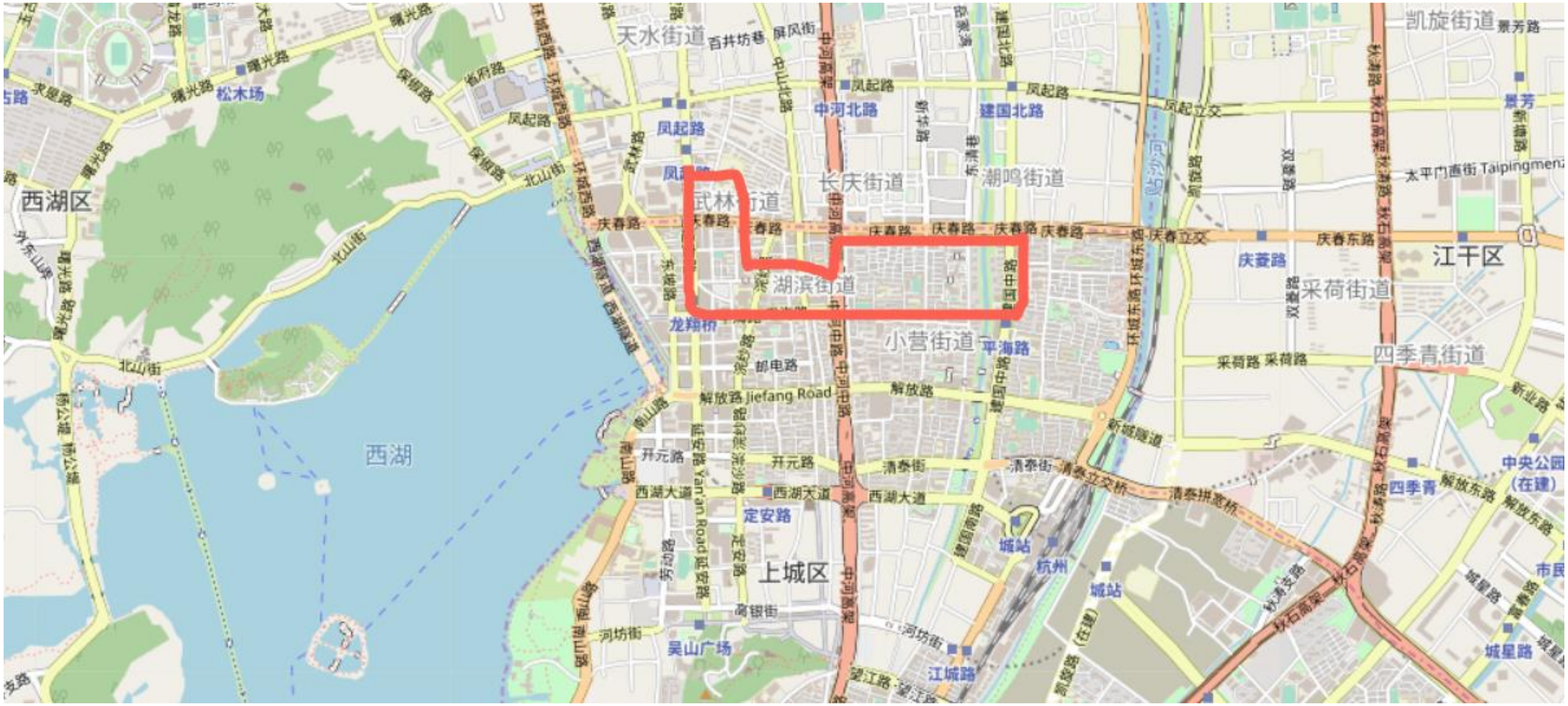
空间业务常见需求

- 近邻搜索、距离排序输出
- 多边形空间圈选
- 辐射式空间圈选
- multi-geom 空间圈选
- 时间、空间、其他属性、多值类 组合圈选
- 栅格，空间可视化分析
- 点云，自动驾驶
- 拓扑，路径规划

LBS，O2O，快递，打车

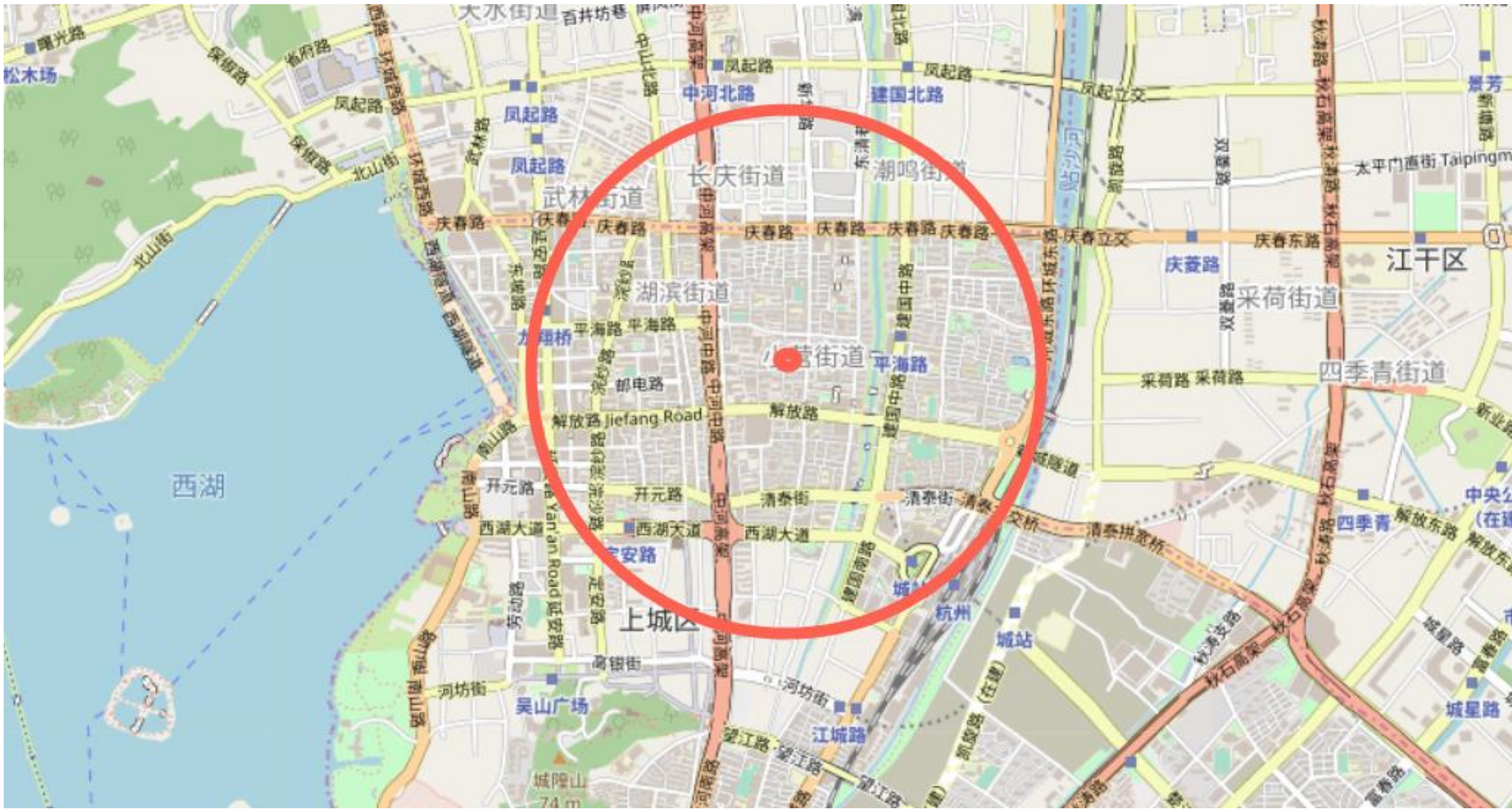


单一商圈分析



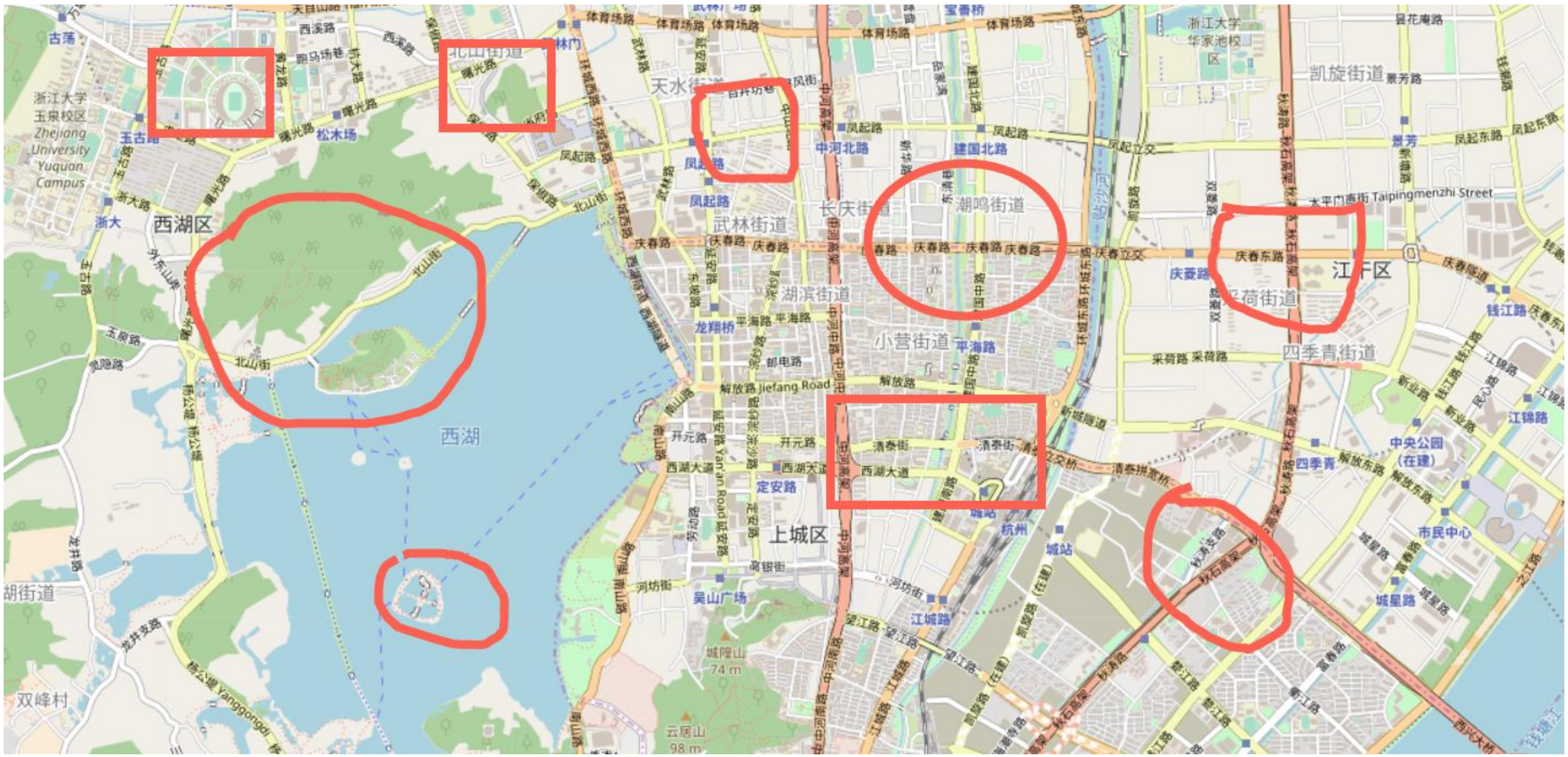
天水街道
百井坊巷
屏风街
凤起路
中河北路
建国北路
凤起路
长庆街道
潮鸣街道
武林街道
庆春路
庆春路
庆春路
庆春路
庆春立交
庆春东路
庆菱路
江干区
采荷街道
湖滨街道
小营街道
平海路
龙翔桥
邮电路
解放路 Jiefang Road
解放路
采荷路
四季青街道
凯旋街道
景芳路
景芳
太平门直街 Taipingmen
西湖区
松木场
曙光路
北山街
西湖
开元路
开元路
清泰街
清泰街
西湖大道
西湖大道
西湖大道
定安路
上城区
高银街
河坊街
吴山广场
城站
杭州
城站
江城路
望江路
中央公园
(在建)
四季青
解放东路
市民
城星路
城星路
新塘路

辐射分析、搜索



长庆街道
潮鸣街道
武林街道
湖滨街道
采荷街道
四季青街道
江干区
西湖
上城区
凤起路
中河北路
建国北路
庆春路
庆春东路
庆春立交
庆菱路
平海路
邮电路
解放路 Jiefang Road
开元路
清泰街
西湖大道
采荷路
解放东路
河坊街
吴山广场
江城路
望江路
城隍山
杭州
城站
四季青
中央公园（在建）
松木场
曙光路
北山街
太平门直街 Taipingmen

多家实体门店商圈分析

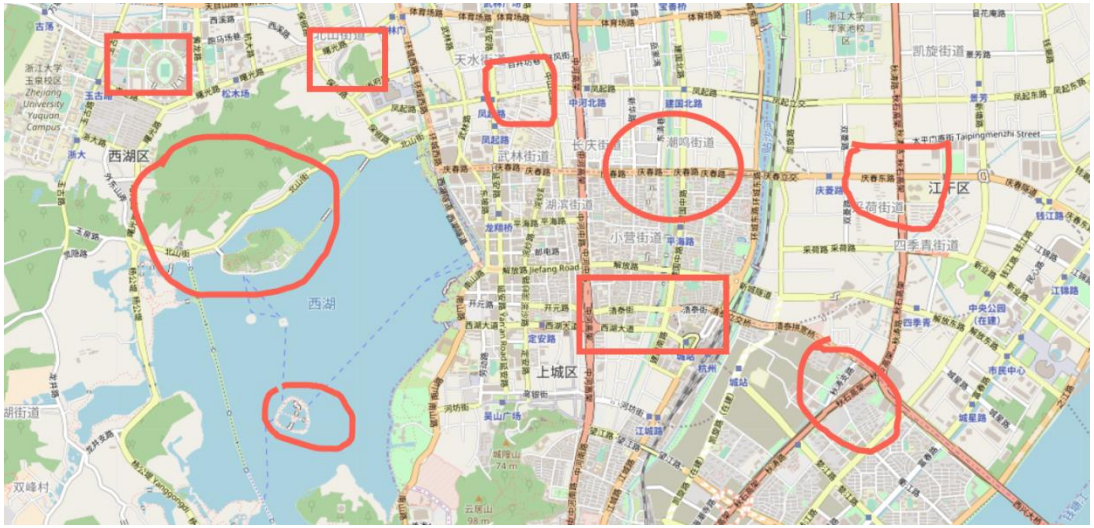


西湖区
西湖
上城区
江干区
武林街道
湖滨街道
小营街道
潮鸣街道
长庆街道
天水街道
北山街道
凯旋街道
四季青街道
采荷街道
浙江大学玉泉校区
Zhejiang University Yuquan Campus
浙江大学华家池校区
体育场路
庆春路
凤起路
解放路 Jiefang Road
开元路
西湖大道
清泰街
北山街
中河北路
建国北路
平海路
太平门直街 Taipingmenzhi Street
城隍山 74 m
云居山 98 m
吴山广场
双峰村

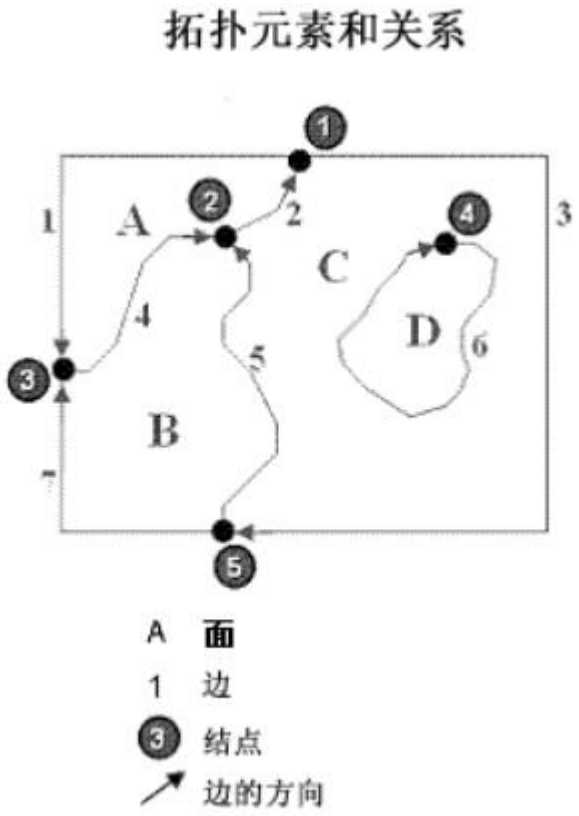
各种属性过滤

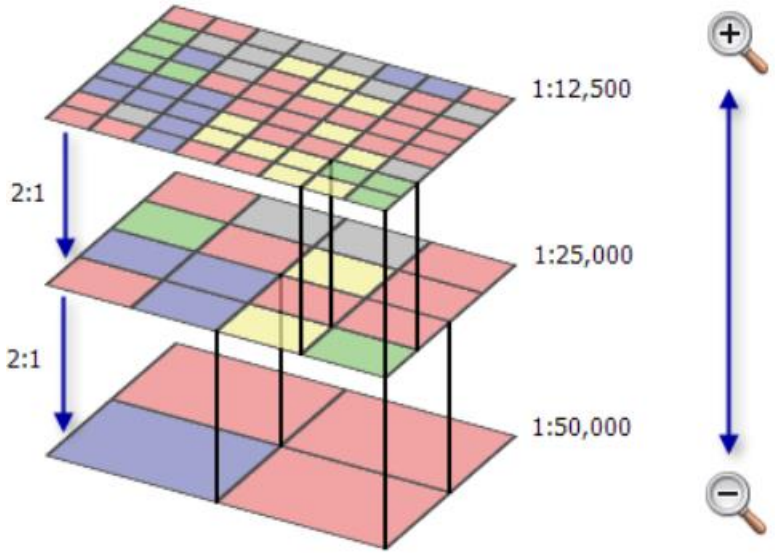
其他条件：

时间范围
多值类型包含、相交
全文检索
组，等值, IN
JOIN
。 。 。 。 。 。

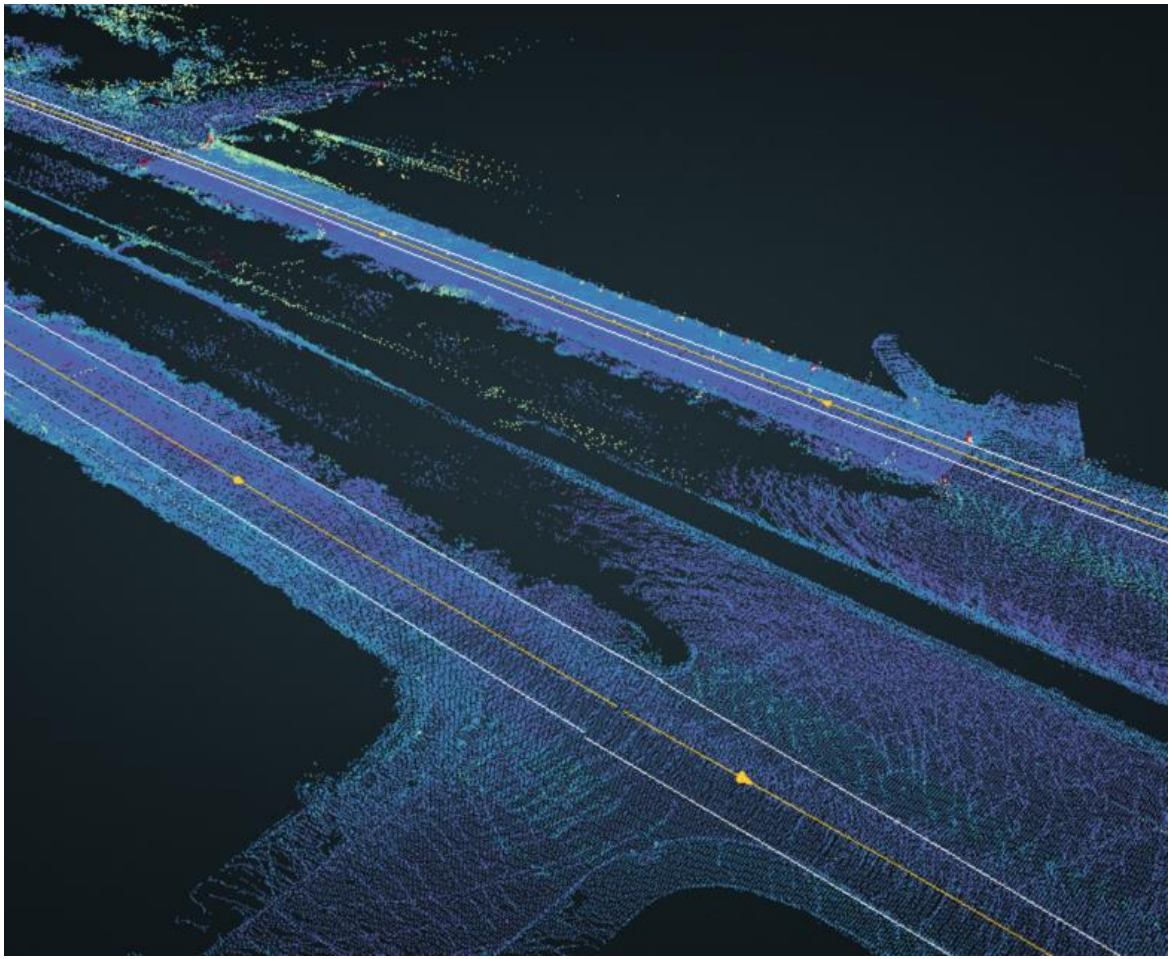


栅格、拓扑





点云



痛点

- 空间大范围查询响应速度问题
- 时空、多维度查询响应速度问题

空间数据索引类别、数据结构

- http://postgis.net/docs/manual-2.4/using_postgis_dbmanagement.html#idm2267
- GiST
  - r-tree on GiST
  - r-tree
    - break up data into rectangles, and sub-rectangles, and sub-sub rectangles
  - GiST
    - break up data into "things to one side", "things which overlap", "things which are inside"
- SP-GiST
- BRIN
  - block bound box
  - BRIN index is to store only the bouding box englobing all the geometries contained in all the rows in a set of table blocks, called a range

空间数据索引支持的操作

- 包含
- 相交
- 方位（上下左右）
- 距离排序
- 约束（排他）

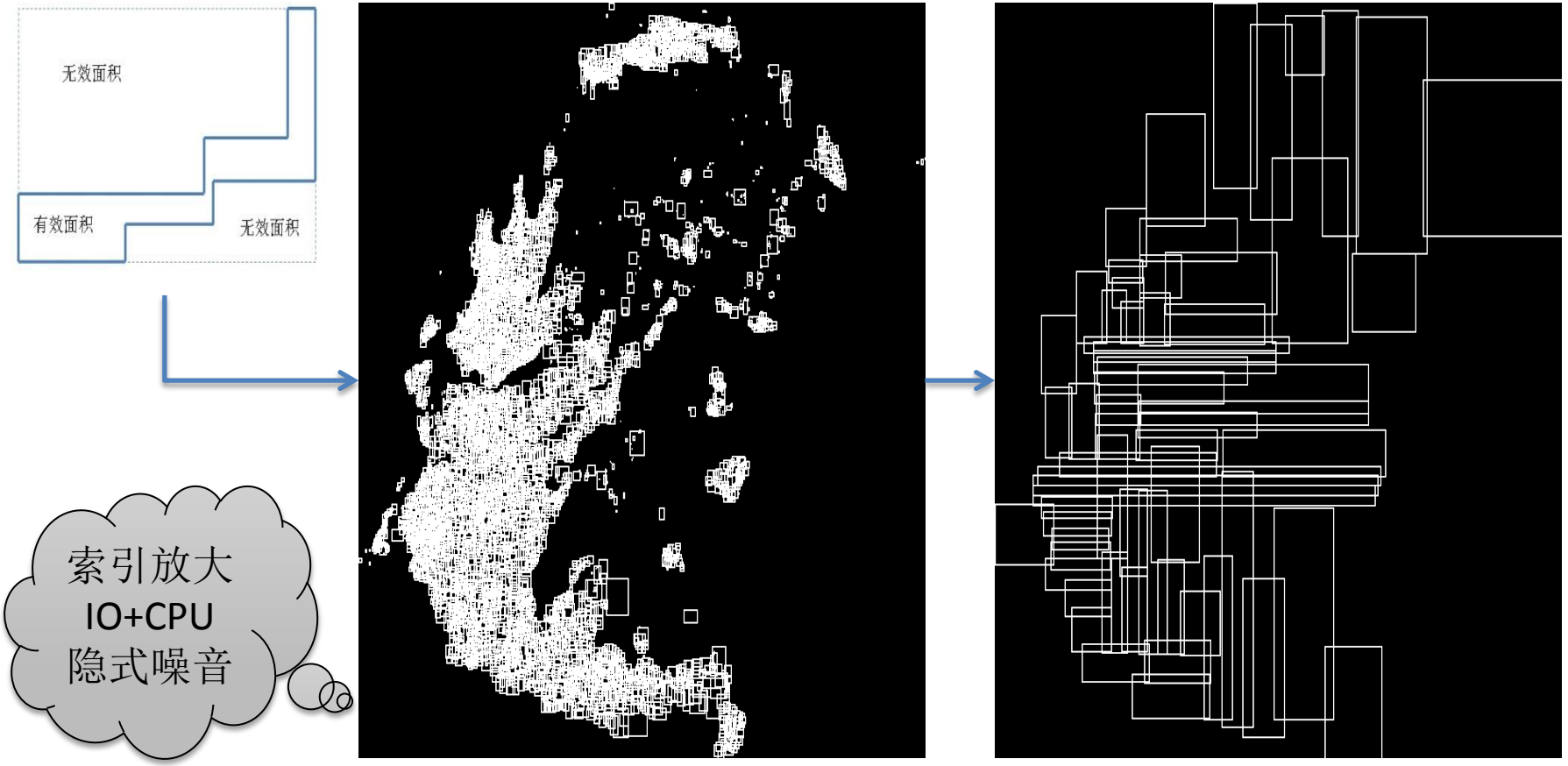
brin vs gist索引对比

- 创建耗时
  - brin 快，GIST慢很多
- 查询效率（KNN、包含、相交等）
  - 空间聚集情况下，效率差不多
- 索引的实时性
  - BRIN等autovacuum才更新
  - GIST实时
- 对DML性能的影响
  - BRIN小
- 空间占用
  - BRIN小很多很多
- 支持的操作类型、适用范围
  - brin，相交、包含
  - Gist，相交、包含、方位、距离排序

痛点背后的原因

- IO\CPU 放大

隐式噪音1 - IO+CPU放大



位图扫描放大

- 一维、多维查询bitmap scan放大(扫描更多的Tuple，引入了cpu filter)

```
postgres=# explain (analyze,verbose,timing,costs,buffers) select count(*) from tbl where id between 1 and 100;
                                                                                  QUERY PLAN
-------------------------------------------------------------------------------------------------------------------------------------------------------------------
 Aggregate  (cost=10000004856.60..10000004856.61 rows=1 width=8) (actual time=4.820..4.820 rows=1 loops=1)
   Output: count(*)
   Buffers: shared hit=203
   ->  Bitmap Heap Scan on public.tbl  (cost=10000000003.03..10000004856.60 rows=1 width=0) (actual time=0.036..4.806 rows=100 loops=1)
         Output: id, gid
         Recheck Cond: ((tbl.id >= 1) AND (tbl.id <= 100))
         Rows Removed by Index Recheck: 45276
         Heap Blocks: lossy=201
         Buffers: shared hit=203
         ->  Bitmap Index Scan on idx_tbl_id  (cost=0.00..3.03 rows=28571 width=0) (actual time=0.025..0.025 rows=2560 loops=1)
               Index Cond: ((tbl.id >= 1) AND (tbl.id <= 100))
               Buffers: shared hit=2
 Planning time: 0.143 ms
 Execution time: 4.884 ms
(14 rows)
```

痛点优化例子

- 选择索引
- 空间包含查询优化
- 聚集优化
- KNN查询优化
- 时空、多维查询优化

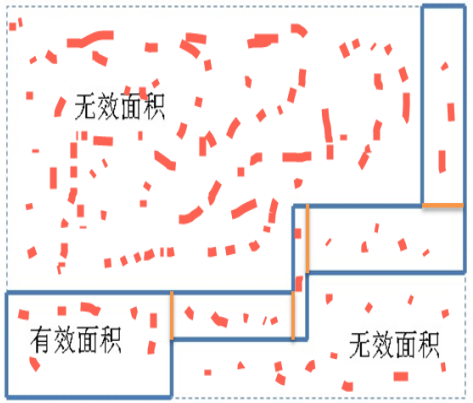
选择索引

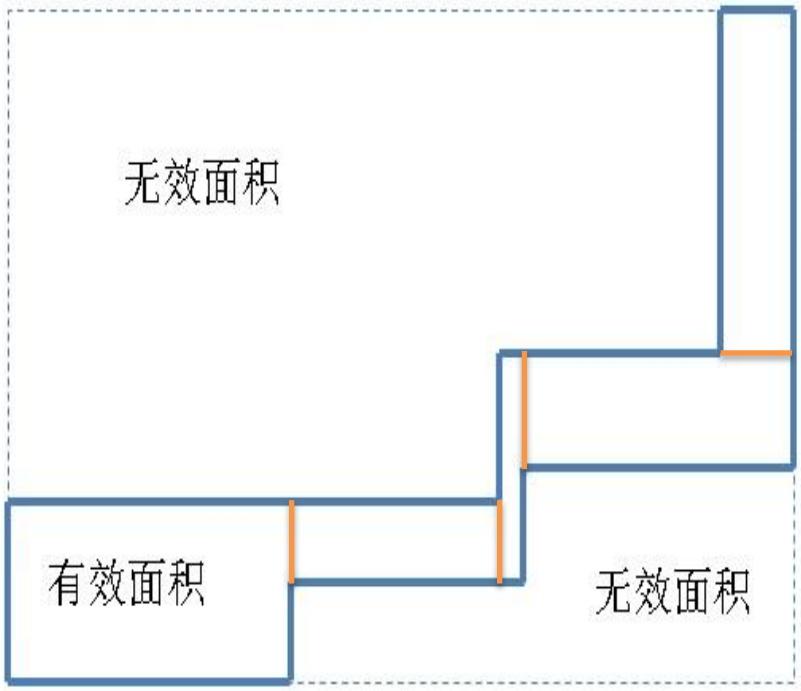
- BRIN
  - 数据按 st_geohash(geometry) 聚集
  - 包含、相交查询需求
- GiST
  - 数据不需要聚集
  - 包含、相交、距离排序、方位等查询需求

空间包含查询优化

ST_subdivide()

自动切割，写成UDF





切割效果：
100%或接近100%
有效面积。

https://github.com/digoal/blog/blob/master/201710/20171005_01.md

聚集优化

- BRIN索引
  - 按st_geohash的顺序聚集
  - heap page的bound box更清晰，使用BRIN索引扫描访问的HEAP PAGE更少
- GIST索引
  - CLUSTER my_geom_index ON my_table;
  - 数据按GIST索引结构存放，空间范围搜索扫描的HEAP BLOCK更少
  - 数据更密集，更容易内存命中，消耗的shared buffer更少

KNN查询优化

- SELECT the_geom FROM geom_table WHERE ST_Distance(the_geom, ST_GeomFromText('POINT(100000 200000)', 312)) < 100;
  - 无法使用空间索引，全表扫描
- SELECT the_geom FROM geom_table WHERE ST_DWithin(the_geom, ST_MakeEnvelope(90900, 190900, 100100, 200100,312), 100);
  - 可以使用空间索引
- UDF
  - for x in SELECT the_geom FROM geom_table order by the_geom <-> ST_GeomFromText('POINT(100000 200000)', 312) loop
  - if ST_Distance(the_geom,x) >= 100
  - then return;
  - else return next x;
  - end if;
  - end loop;

多维查询优化

- 多值字段、单值字段、空间字段、时间字段
  - 多值包含、相交
  - 单值范围、等值
  - 空间包含、相交、距离
  - 时间范围
- 数据分区、过滤条件字段分区
  - RANGE、LIST、HASH
  - 多级分区
- btree_gist，复合索引，等值在前，其他字段在后
- btree_gin，复合索引，等值在前，其他字段在后
- 索引分区
  - partial index
  - create index idx1 on tbl using gist(geo) where class='玩具';
  - ……

谢谢

- 参考文档
- https://github.com/digoal/blog/blob/master/201711/20171122_03.md
- https://github.com/digoal/blog/blob/master/201710/20171005_01.md
- https://github.com/digoal/blog/blob/master/201710/20171004_01.md
- https://github.com/digoal/blog/blob/master/201708/20170820_01.md
- https://github.com/digoal/blog/blob/master/201708/20170809_01.md
- https://github.com/digoal/blog/blob/master/201707/20170722_01.md
- https://github.com/digoal/blog/blob/master/201706/20170620_01.md
- https://github.com/digoal/blog/blob/master/201704/20170413_02.md
- https://github.com/digoal/blog/blob/master/201703/20170328_04.md
- https://github.com/digoal/blog/blob/master/201708/20170827_02.md
- https://github.com/digoal/blog/blob/master/201706/20170629_01.md
- https://github.com/digoal/blog/blob/master/201801/20180129_01.md