Spatial Indexes in Oracle Spatial

Indexes in Oracle

- R-tree (default), or
- linear quadtree

both implemented using extensible framework

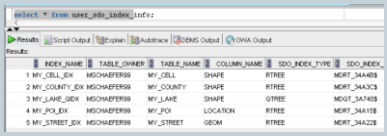
- linear quadtree based on B-tree indexes
- R-tree: using tables/recursive SQL

Indexes in Oracle

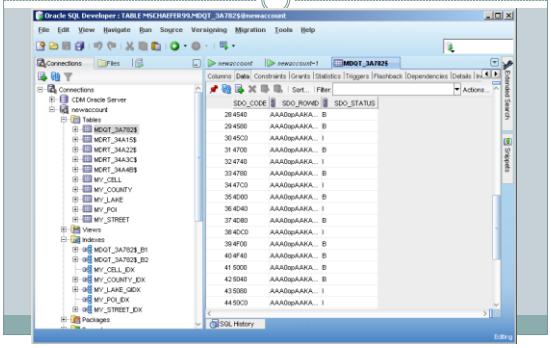
select * from user_sdo_index_info;

Quadtree in Oracle

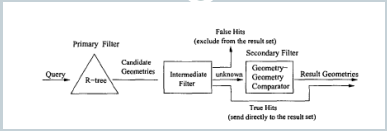
```
create index my_lake_qidx on my_lake(shape)
indextype is MDSYS.SPATIAL_INDEX
parameters ('sdo_level = 5');
```



Quadtree in Oracle

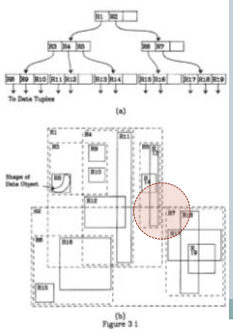


Query Processing in Oracle Spatial

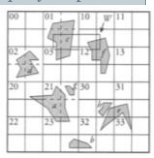


- primary filter (exterior, e.g. R-tree: mbb)
- intermediate filter (interior)
- geometry engine (computational geometry algorithms)

Primary Filter



- R-tree:
  - all shapes whose mbb overlaps mbb of query shape
  - implemented using recursive traversal of R-tree
- B-tree:
  - range query on query shape
  - need to create query shape tiling at same level as index

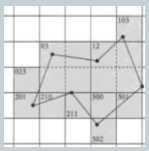


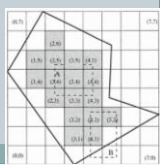
Intermediate Filter

uses interior approximation

- quadtree:
  - distinguish interior and boundary tiles
  - what would an "unknown" case look like ?
- R-tree:
  - not for data, just for query shape
  - depends on whether shape is
    - convex: split into 4 pieces, find largest rectangle in each piece
    - concave: level-4 (quadtree)-tiling of query and mbbs

SDO_CODE	SDO_ROWID	SDO_ST
28 4540	AAA0opAAKA...	B
29 4580	AAA0opAAKA...	B
30 45C0	AAA0opAAKA...	I
31 4700	AAA0opAAKA...	B
32 4740	AAA0opAAKA...	I
33 4780	AAA0opAAKA...	B





Experimental Results: quadtree from Kothuri, Ravada, Abugov)

Tiling Level	Avg. # of tiles per geometry	Indexing time(s)
2	1	21
4	1	21
6	1.01	21
8	1.03	21
10	1.16	24
12	1.59	26
14	4.35	46
16	29.81	259
18	359.01	3031

quadtree indexing (23982 geometries)

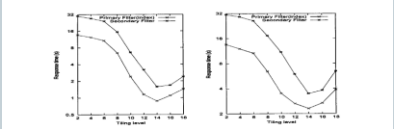


Figure 2: Anyinteract Query Performance for different tiling levels for (a) 5-mile radius, and (b) 10-mile radius windows: Both primary and secondary filter performance decreases as tiling level increases from 2 to 14 and starts to increase again beyond tiling level 14. The initial decrease is due to better approximations in the index and fewer resulting candidates from index. The subsequent increase is due to increase in the number of tiles per geometry and associated searching costs.

Experimental Results: comparison from Kothuri, Ravada, Abugov)

Performance Criterion	Quadtree	R-tree
ABI Index Creation	4203s	20406s
USBG Index Creation	8330s	454s
10K-point inserts (ABI)	67s	131s
20K-small-polygon inserts (USBG)	183s	330s
10K-large-polygon inserts (USBG)	3600s	166s
Storage(ABI)	909MB	824MB
Storage(USBG)	728MB	24MB

Table 3: Comparison of Quadtree and R-tree create and update times and storage characteristics: Quad-tree is faster for ABI (point) data in index creation/update. R-tree is faster for large-polygon USBG data. R-tree needs less space in both cases.

ABI: 10M of data (point data)
USBG: 230K of data (polygons)

Query mask	Quadtree(s)	Rtree(s)
anyinteract	0.81	0.49
inside	0.80	0.28
contains	0.85	0.04
touch	1.52	1.13
coveredby	0.88	0.66
covers	0.44	0.05
equal	1.53	0.04
overlapbydisjoint	1.77	1.41
overlapbyintersect	1.53	1.41

Table 4: Comparison of Average Query times for Quadtree and R-tree indexes on USBG data: Queries are from counties dataset. R-tree is faster for all masks — anyinteract, R-tree faster by 35%; inside by 65%.

Sources

- Kothuri, Godfrind, Beinat, Pro Oracle Spatial for OracelDatabase 11g, Apress, 2007.
- Garcia-Molina, Ullman, Widom, Database Systems; the complete book, Pearson, 2009.
- Kothuri, Ravada, Abugov; Quadtree and R-tree Indexes in Oracle Spatial: A comparison using GIS Data, ACM SIGMOD, 2002.