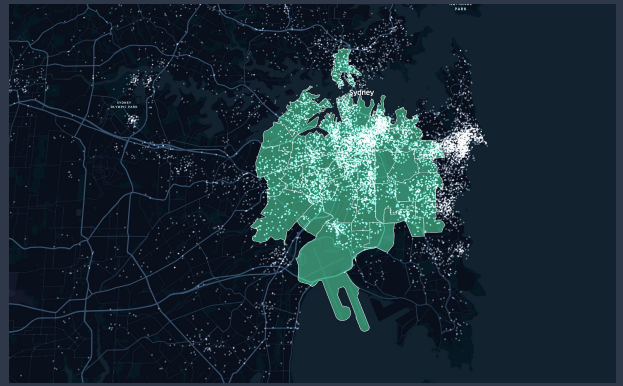
PostGIS & MongoDB

Feature & Performance Comparison for Geospatial Data Model

Introduction & Data

Hypothesis:
Since PostGIS is built specifically for geospatial data, it exceeds MongoDB in performance when using geospatial data.

Primary Datasets used for Testing:
1. Sydney Airbnb Open Data
2. OpenStreetMap Australia



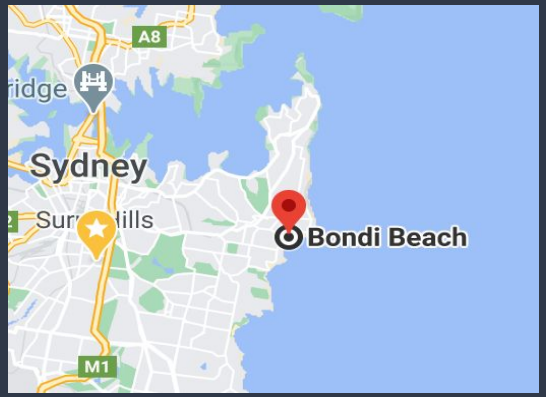
Agenda

- Spherical Geospatial Functions
- Intersects & Within Geospatial Functions
- Dimensionally Extended 9 Intersection Model in PostGIS

Feature 1: $nearSphere in MongoDB vs ST_DWithin in PostGIS

Question:

How many Airbnb listings are near Bondi Beach by setting up with different maximum distance and data size?



Bondi Beach Coordinates:
[151.2767, -33.8915]

Maximum Distance:

- ❑ 5 kilometers
- ❑ 10 kilometers
- ❑ 20 kilometers
- ❑ 30 kilometers

Data Points from Airbnb listings:

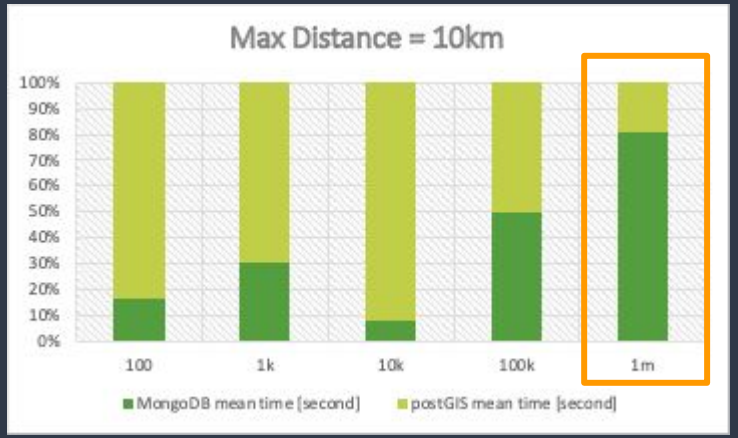
- ❑ 100 Data Points
- ❑ 1000 Data Points
- ❑ 10, 000 Data Points
- ❑ 100,000 Data Potins
- ❑ 1 million Data Points

*each query will be run by ten times

Feature 1: Time Execution Statistics Table

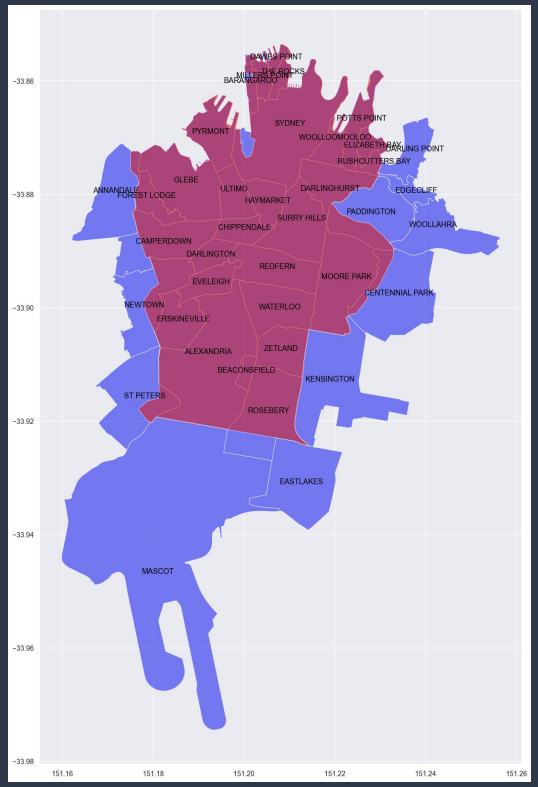
	MongoDB: $nearSphere		PostGIS : ST_Dwithin	
Data Size/max distance (km)	mean time [second]	standard deviation	mean time [second]	standard deviation
100				
5	**0.007187605**	**0.002276653**	0.043274999	0.006481027
10	**0.008389878**	**0.005950478**	0.043648839	0.00233995
20	**0.010199976**	**0.002533084**	0.048395753	0.007962633
30	**0.008744836**	**0.002221548**	0.045145726	0.002509106
1k				
5	**0.025807548**	**0.016310987**	0.065703702	0.022447455
10	**0.035507584**	**0.010341971**	0.081143594	0.027774079
20	**0.040365314**	**0.017878762**	0.082090163	0.015813456
30	**0.03627193**	**0.004936658**	0.103290701	0.031626384
10k				
5	**0.141123176**	**0.075665804**	1.241498327	0.325197246
10	**0.241815591**	**0.029083681**	2.782282567	0.201445733
20	**0.301468277**	**0.03964041**	3.650822353	0.30310949
30	**0.475698066**	**0.444679669**	4.069888043	0.407288217
100k				
5	**1.089699531**	**0.856567986**	1.241498327	0.325197246
10	**2.787020278**	**0.520455389**	2.782282567	0.201445733
20	4.641707802	0.674566761	**3.650822353**	**0.30310949**
30	6.599097824	1.361575328	**4.069888043**	**0.407288217**
1 million				
5	34.98340652	31.24455561	**74.07140317**	**25.09023428**
10	317.8399241	272.6782824	**75.91673875**	**8.069031153**
20	914.9281501	353.0617526	**107.6204711**	**7.79144199**
30	1149.652263	677.2832733	**341.3338144**	**719.1305923**

Feature 1: Comparison of Time Performance



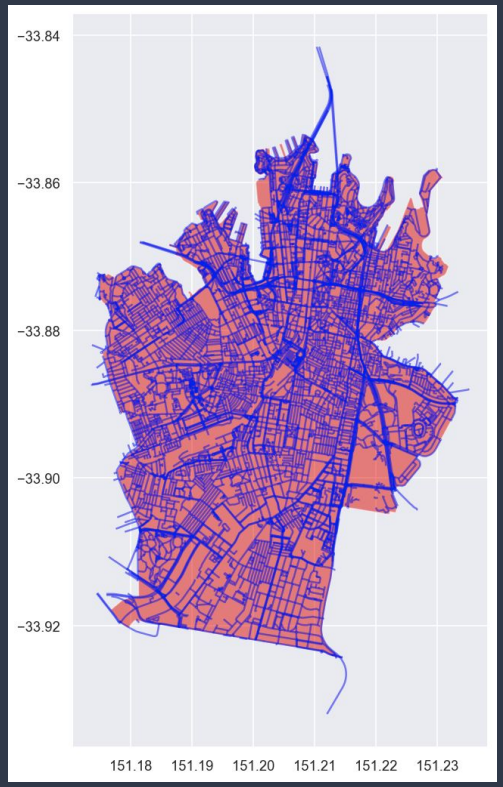
Feature 2

Polygon Filter (red)

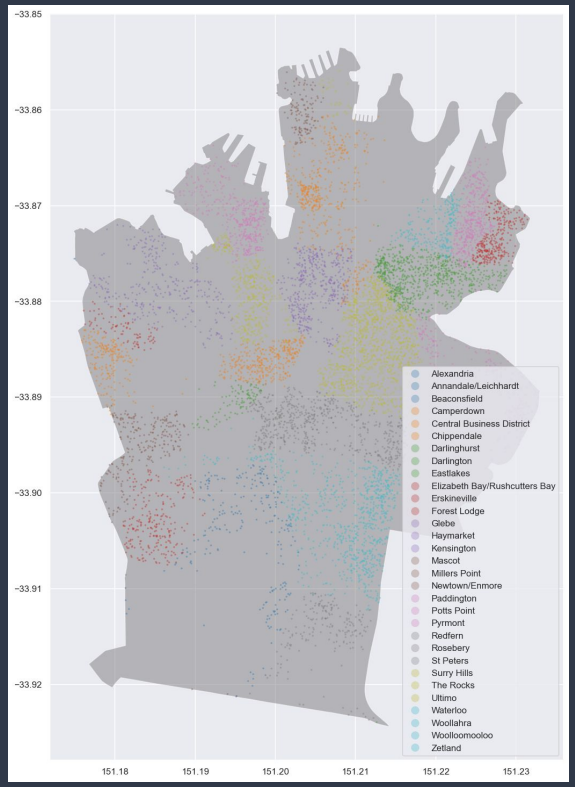


>>>

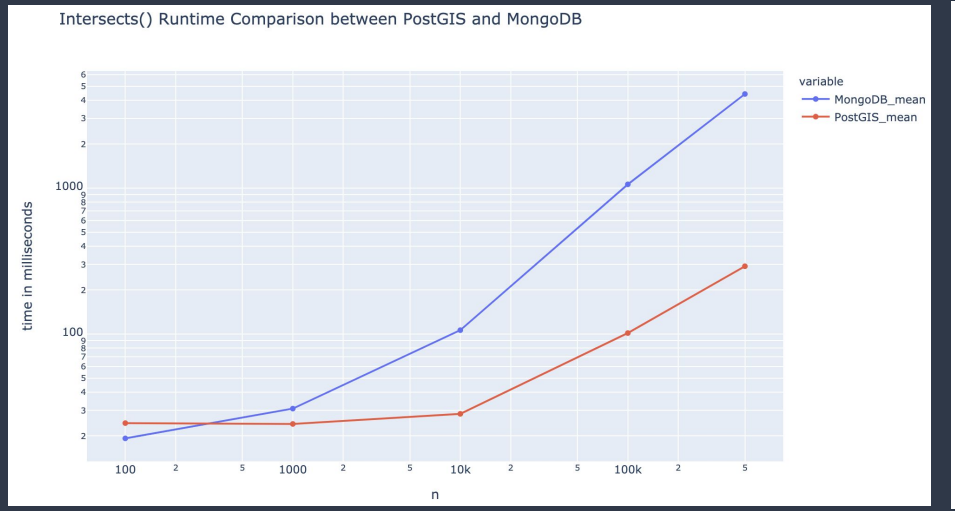
Intersects (with roads)

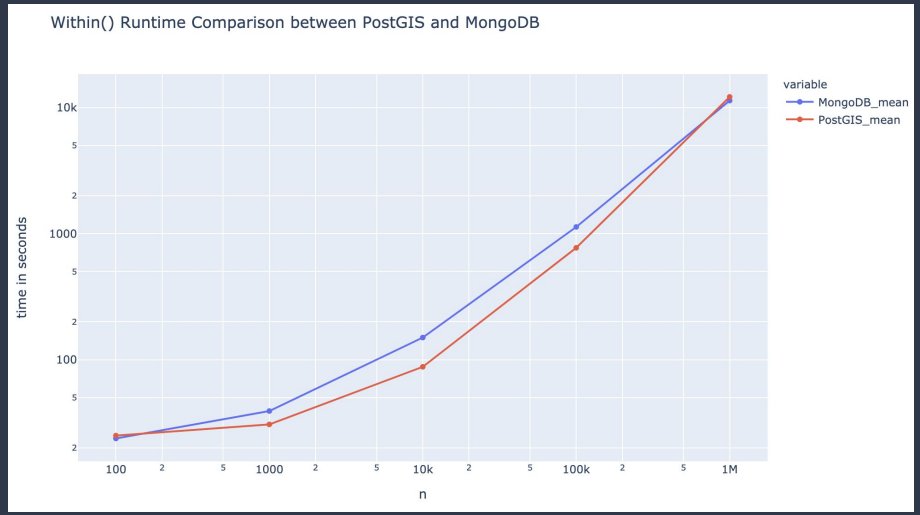


Within (Airbnb listings)



Performance Comparison





Execution stats

Intersects

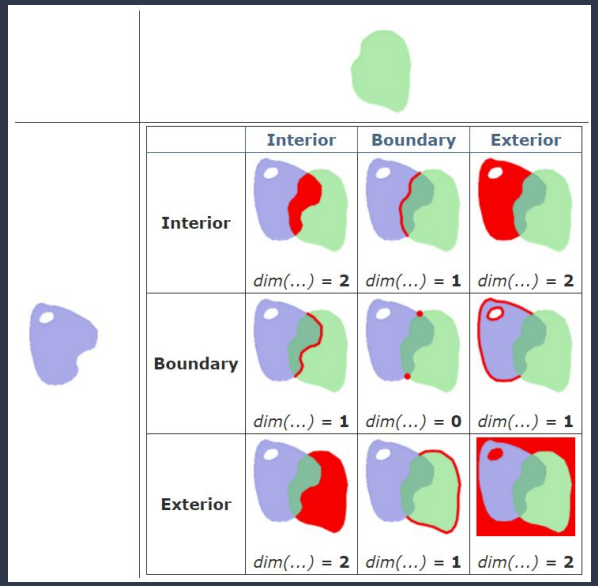
	time in milliseconds			
system	PostGIS		MongoDB	
	mean	std	mean	std
n				
100	24.532204	3.141809	19.28	7.039365
1000	24.205370	2.090392	30.84	7.220605
10000	28.385696	3.764740	106.34	34.110630
100000	101.556478	20.511965	1061.42	106.424274
500000	291.452546	69.281944	4414.42	1021.952714

Within

	time in milliseconds			
system	PostGIS		MongoDB	
	mean	std	mean	std
n				
100	25.001583	2.027436	23.72	4.965349
1000	30.667968	2.160910	39.16	5.056255
10000	87.840109	19.978261	149.86	45.633862
100000	772.345309	334.494025	1130.92	441.380711
1000000	12163.742747	8471.295160	11361.42	2825.621074

Feature 3: Dimensionally Extended 9 Intersection Model

	Interior	Boundary	Exterior
Interior	*dim(I(a) ∩ I(b))*	*dim(I(a) ∩ B(b))*	*dim(I(a) ∩ E(b))*
Boundary	*dim(B(a) ∩ I(b))*	*dim(B(a) ∩ B(b))*	*dim(B(a) ∩ E(b))*
Exterior	*dim(E(a) ∩ I(b))*	*dim(E(a) ∩ B(b))*	*dim(E(a) ∩ E(b))*

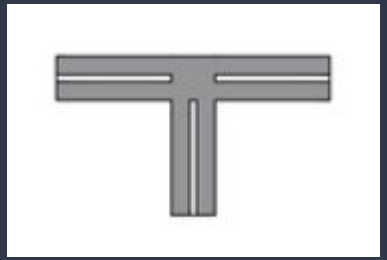


- Extensive framework that describes how two spatial objects interact.
- Matrix shows intersections of features.
- Number in cell corresponds to the dimension of the intersection eg. dimension of a surface intersection is 2.

Feature 3: Dimensionally Extended 9 Intersection Model

- Finding specific types of intersecting roads:
  - Cross intersections: 0F1FF0F02

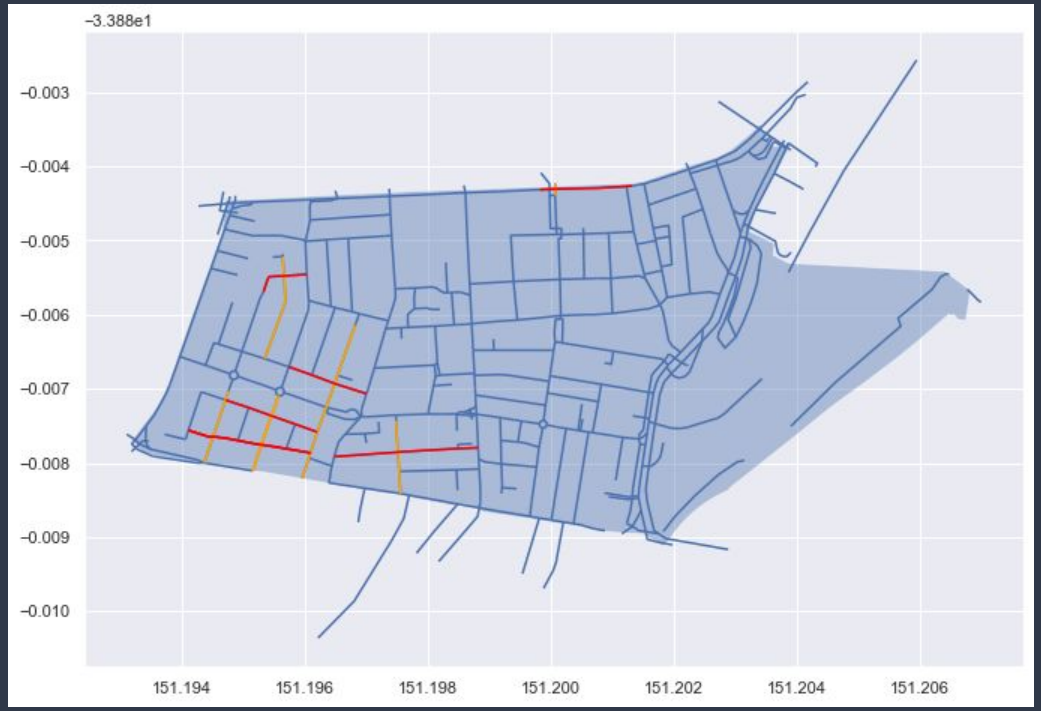
		Road Y X----------X		
		Interior	Boundary	Exterior
Road X X | | X	Interior	0	F	1
	Boundary	F	F	0
	Exterior	1	0	2



  - T intersections: F010F0102

		Road Y X----------X		
		Interior	Boundary	Exterior
Road X X | | X	Interior	F	0	1
	Boundary	0	F	0
	Exterior	1	0	2

Feature 3: Dimensionally Extended 9 Intersection Model



Cross Intersections in Chippendale

- Small number of cross intersections found and highlighted

Feature 3: Dimensionally Extended 9 Intersection Model

Cross Intersections in Alexandria

- None found in Chippendale
- 2 were found in Alexandria

Feature 3: Dimensionally Extended 9 Intersection Model

- Roads composed of multiple segments/records.
- Feature works as intended.
- Effectiveness determined by data model structure.
- Recommend pre-processing dataset for increasing number of results.

	osm_id	code	fclass	name	ref	oneway	maxspeed	layer	bridge	tunnel	geometry
7	411192715	5113	primary	Cleveland Street	None	B	50	0	F	F	LINESTRING (151.19981 -33.88874, 151.20001 -33...
12	411017206	5113	primary	Cleveland Street	None	F	50	0	F	F	LINESTRING (151.19349 -33.88781, 151.19426 -33...
19	411192716	5113	primary	Cleveland Street	None	B	50	0	F	F	LINESTRING (151.20037 -33.88882, 151.20049 -33...
23	595889192	5113	primary	Cleveland Street	None	B	50	0	F	F	LINESTRING (151.19822 -33.88851, 151.19849 -33...
40	175969863	5113	primary	Cleveland Street	None	B	50	0	F	F	LINESTRING (151.19516 -33.88811, 151.19507 -33...

Conclusion

MongoDB:

- Ease of setup (or the lack thereof)
- Tends to outperform PostGIS when using spherical filters (near) if maximum radius is less than 100,000km
- May outperform PostGIS when using spatial filters (Within) if two conditions are met
  - Only dealing with points
  - Small dataset (< 1000 data points)
- Equipped with only essential functions
  - Near, Within, Intersects
- Does not support
  - Coordinate reprojection (projected to non-projected, vice versa)
  - Geometry transformation (to_centroid, buffer, unary_union)
  - Distance calculation
- Due to limitation of geospatial features, often requires use of programming language such as Python to perform necessary transformations
  - Converting all geometries to the same coordinate system
  - Geometric manipulations

PostGIS:

- Ease of setup
- Tends to outperform MongoDB when
  - Dealing with multiple geometries, using lines and polygons
  - Dataset is large (> 1000 data points)
- Better performance consistency - standard deviation of execution time is usually very small
- Comprehensive and powerful support for geospatial data
  - Coordinate reprojection
  - Geometric manipulation / transformation
    - Buffer, centroid, distance calculation, unary union, aggregation and merge of geometry
  - Syntactic difference makes it easy to chain geospatial functions compared to MongoDB
  - DE-9IM for specific intersection projection or filtering
- Usually comes with GDAL and OGR when installing PostGIS
  - GDAL for raster data import / export / conversion
  - OGR for vector data import / export / conversion

Thank you!

Sample Queries: $nearSphere vs ST_DWithin

```
from bson.son import SON # required for $maxDistance
def distance_to_radians(km):
    return km/6378.137

#bondi beach coordinates
lat = -33.8915
lon = 151.2767
point= [lon, lat]

km=5
result= db.airbnb_sydney_100.find(
    {
        "geometry": SON([
            ("$nearSphere", point),
            ("$maxDistance", distance_to_radians(km))
        ])
    }
)
results= [item for item in result]
print(len(results))
```

MongoDB : SnearSphere + $maxdistance

```
sql='''
SELECT *
FROM airbnb_sydney_100
WHERE ST_DWithin(airbnb_sydney_100.geometry,
ST_SetSRID(ST_MakePoint(151.2767,-33.8915),4326),0.05)'''

result =gpd.read_postgis(sql, engine, geom_col='geometry')
results = [item for item in result]
print(result.shape)
```

PostGIS : ST_DWithin

Sample Queries

Intersects

PostGIS

```
sql = f"""
SELECT *
FROM roads_sydney_500k as roads
WHERE ST_Intersects(roads.geometry,
ST_GeometryFromText('{district_polygon.wkt}', 4326));
"""
start = time.time()
df = gpd.read_postgis(sql, engine, geom_col='geometry')
end = time.time()
time_in_ms = (end - start) * 1000
```

MongoDB

```
cursor = db['roads_sydney_500k'].find(
    {
        'geometry': {
            '$geoIntersects': {'$geometry': js['features'][0]['geometry']}
        }
    }
)
print(len([item for item in cursor]), 'documents')
stats = cursor.explain()
time_in_ms = stats['executionStats']['executionTimeMillis']
```

Within

PostGIS

```
sql = f"""
SELECT *
FROM airbnb
WHERE ST_Within(airbnb.geometry,
ST_GeometryFromText('{district_polygon.wkt}', 4326));
"""
start = time.time()
df = gpd.read_postgis(sql, engine, geom_col='geometry')
end = time.time()
time_in_ms = (end - start) * 1000
```

MongoDB

```
cursor = db['airbnb_sydney_1m'].find(
    {
        'geometry': {
            '$geoWithin': {'$geometry': js['features'][0]['geometry']}
        }
    }
)
print(len([item for item in cursor]), 'documents')
stats = cursor.explain()
time_in_ms = stats['executionStats']['executionTimeMillis']
```

Sample Queries: DE-9IM

PostGIS

MongoDB Nil