

Comparing Oracle Spatial and Postgres PostGIS

Deepika Shukla, Chirag Shivnani, Darshit Shah

Comp. Science and Engineering Department, Institute of Technology, Nirma University

deepika.shukla@nirmauni.ac.in

Abstract: Spatial data is the data or information that identifies the geometry information of an object or location in space thus alternatively referred as geospatial data. More commonly it is description of objects in space. A spatial database is a database with additional capabilities for handling spatial data. This document talks about Oracle Spatial, which is a set of features provided by Oracle to access and manipulate spatial data. The main contribution of the work is that, it compares experimentally, the performance of Oracle Spatial from Oracle and Postgres PostGIS, an open source database software used to handle spatial data. Performance is measured and conclusions are drawn upon. A comparison between traditional DBMS and spatial database is also done and reasons are provided as to why a traditional DBMS is inefficient to store spatial data.

Keywords: Oracle Spatial, PostGIS, Spatial Databases, R-Tree, R-Tree Indexing, Relational Database management system (RDBMS), Minimum bounding rectangle (MBR).

Introduction

Spatial data is the data or information that identifies the geometry information of an object or location in space thus alternatively referred as geospatial data. More commonly it is description of objects in space. For example, spatial data can be used to represent the location and size of a building or a lake or any other such object on Earth. It can even represent even the shop in the building or near a lake. While spatial data would contain data like census data, weather and climate data, Location data of places, Satellite data, medical imaging or other such data, represented using numerical values; non-spatial data examples include names, phone numbers or e-mail addresses of people etc. whereas queries or commands posed on spatial data are called spatial queries. Therefore, a spatial database is a database with additional capabilities for handling spatial data. It offers spatial data types in its model and spatial query language [1]. Although there are specialized Geographic Information Systems (GIS) software like GRASS, ENVI, Mapnik etc; those are used to visualize and manipulate geospatial data; two very popular and widely used generic spatial databases are Oracle spatial; which is a set of features provided by Oracle to access and manipulate spatial data is a proprietor extension provided by Oracle. Also, Postgres PostGIS, open source database software used to handle spatial data.

The rest of this paper is organized as follows. Introduction section is followed by the section which essentially establishes the need of spatial databases in clearer way. Section after that compares spatial databases with traditional databases. The paper also well discusses the characteristics of Oracle spatial in the section after that wherein we have we have focused spatial data types and Object relational model supported by Oracle spatial and highlighted query model and indexing model supported by Oracle spatial. The study can be well utilized for introducing the readers to spatial databases and gives an introduction to Oracle Spatial but the main focus of the paper is to compare performance of Oracle Spatial and Postgres PostGIS experimentally using various queries. The final section concludes the study and also presents certain future directions of the work. We have used a same dataset for the comparison process on both the databases and compared using various types of queries and provided performance results.

Need of Spatial databases

Currently, human kind is experiencing a boom in generation of images, videos and Geospatial data because of reduced cost and easy availability of related hardware component in form of camera and other sensor devices. With the advent of sophisticated devices, the information and processing demands of users have also increased and as a consequence the natures of applications have also changed dramatically. Today we need to deal with geometric, geographic or spatial data in almost every field. For e.g. a tourist with a mobile phone would be interested in querying nearest food joint in the periphery of 2 kms. A doctor would be willing to draw the references to earlier case treated by seeing the MRI reports of a patient or a town planner would want to know that where and how the existing road network should be expanded in the benefit of the citizens of the town [3]. Traditional DBMS systems due to their very basic concept are designed to manage, store and retrieve textual and numeric type of data. They do not provide support for managing spatial objects and performing spatial operations. So a system is needed which provide database capabilities for spatial data. , Spatial databases allow the user to model, store, retrieve and query the spatial objects along with their geometry with reference to their reference frame. Most spatial databases like Oracle spatial, Postgres PostGIS, IBM DB2, etc. , in addition to provide simple querying such as select statement, they provide support for variety of spatial operation. For e.g; calculate the length of a line, area of a polygonal shape, and distance between two spatial objects, facilitate to create new geometries, spatial relationship between two geometries etc.

Traditional DBMS Vs. Spatial database

Spatial database is a database system that is optimized for a single objective – to store and query data that is contained within a geometric space. We can use spatial databases in order to store points, lines, polygons, etc. These geometric shapes can be used to represent any object like a building or a lake on Earth and many more objects on Earth. Examples of spatial databases include Oracle Spatial, MySQL, Postgres, IBM DB2 etc.

Traditional DBMS allow concurrent access to data and are persistent across failures, scale well when it comes to very large datasets which can't be stored inside a computer's main memory but they are not an efficient solution when it comes to spatial queries or even storing spatial data.

Non-spatial queries include fetching the list of names of all the bookstores that have a certain amount of books while spatial queries would return information about the names of all the bookstores within a certain distance.

A. Representation and storage of data compared

Consider that we require to store the given polygon representing a census block. If this information is to be presented in form of RDBMS, then one would store attributes like Block-ID, Area of the census block, its population and boundary/polygon information of the census block.

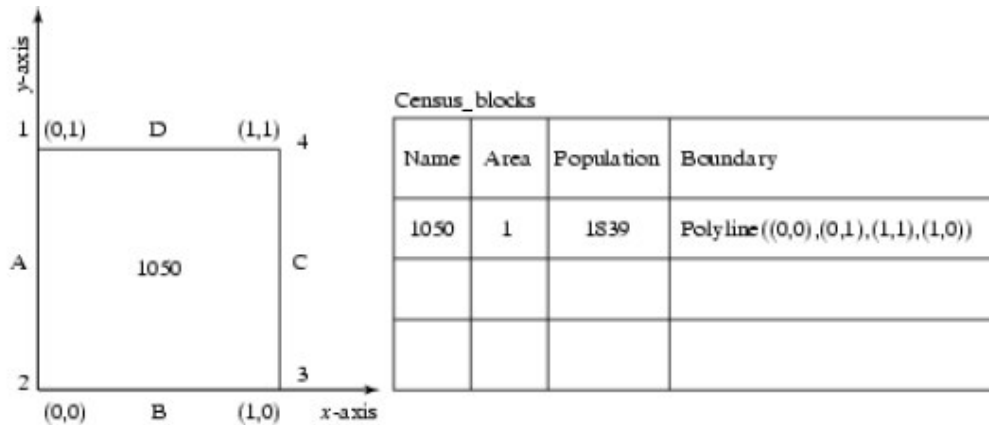


Figure 1: Census Block represented using a polygon[3]

A traditional relational DBMS supports simple data types like strings, numbers, date, etc. and has no support for complex geometric shapes like a line, arc or a polygon.

In order to store the above mentioned polygon in a traditional database, we would need to create three separate tables namely polygon, edge and points.

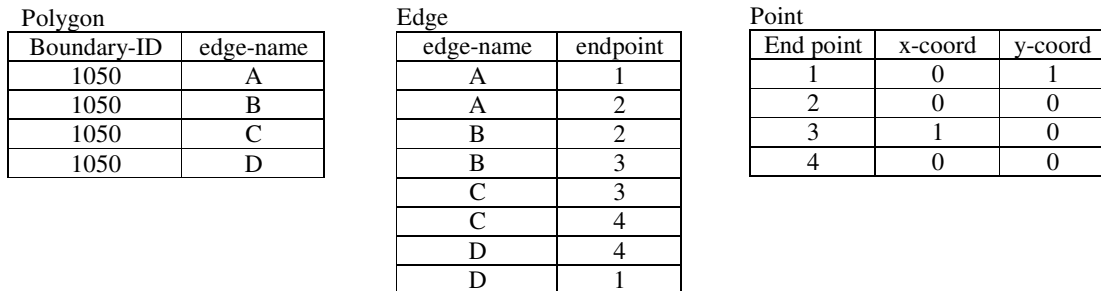


Figure 2: Traditional DBMS representation of polygon shown in Figure 1

A simple unit square of the polygon would span over 16 rows across three tables. Additionally, In order to fetch the required information about the polygon like area, we would need to execute complex join queries which would be time consuming and inefficient. Storing spatial data in a traditional database also consumes lots of storage space which isn't worth it considering the slow information retrieval time.

A spatial database on the other hand provides a geometry object that stores the geometry features like coordinates, length and other features in a single row of a single column which makes querying data easily and data is retrieved very quickly without the need of any join query.

Oracle Spatial

Oracle Spatial is not separate software but an in-built feature of Oracle DBMS and is commonly referred as Spatial. It contains supports for storage, retrieval and manipulation of spatial data and features. It provides SQL schema and different functions to perform spatial operations.

A. Oracle Spatial Features

- As mentioned earlier, Oracle Spatial provides a schema known as MDSYS. The schema dictates necessary syntax and semantics of data types supported by Oracle spatial.
- A spatial indexing mechanism is also provided in order to facilitate faster retrieval of data.
- A variety of operators and functions are provided for performing region-of-interest queries and other spatial analysis operations.
- It also provides something known as spatial component which is the geometric representation of a shape in some co-ordinate space. This is known as the spatial feature's geometry.

B. Spatial Data Types

The main data type we are going to take about is SDO_GEOMETRY. It is composed of:

- SDO_GTYPE
- SDO_SRID
- SDO_POINT
- SDO_ELEM_INFO
- SDO_ORDINATES

▪ SDO_GTYPE

This table is used to define the geometry type that is stored in the SDO_GEOMETRY object. The table contains the fields: name, 2D, 3D and 4D. The name field can contain various geometric shapes like point, polygon, multipoint, etc.

▪ SDO_POINT

It is used to store point data. For example, if we need to store a point on an application that makes use of maps, we would use this type. SDO_POINT would be ignored if SDO_ELEM_INFO and SDO_ORDINATE have some value. It is of type SDO_POINT_TYPE and contains X, Y and Z values.

▪ SDO_ELEM_INFO

As the name suggests, it stores information regarding a particular element. The table consists of three fields: number- which stores the starting position of the coordinates, element_type- which store the type of the element (point, line string, polygon, etc.) and interpretation – the manner in which the element type is to be interpreted.

▪ SDO_ORDINATES

It stores the list of an object's co-ordinates and is stored as SDO_ORDINATE_ARRAY.

C. Object Relational Model

Oracle Spatial provides support for object-relational model which is used to represent geometries. It makes use of a single column known as MDSYS.SDO_GEOMETRY.

▪ Benefits of Object-Relational Model

Oracle spatial provides support for many geometry types like arcs, circles, line strings and optimized rectangles. The model provides ease of use in creating and maintaining indexes and also in performing spatial queries. Unlike traditional DBMS, the geometries are modelled in a single row and a single column due which helps to gain optimal performance from the database.

▪ Data Model

Data model in Oracle spatial is hierarchical. Spatial data of the application is represented in terms of elements, geometries and layers. One Layer is composed of multiple geometries and one geometry is formed from multiple elements. For example a point might represent a building location, a line string might represent a road or flight path, and a polygon might represent a state, city, zoning district, or city block.

1) Elements: An element can be said to be the basic building block of geometry. Points, line strings and polygons can said to be the example of elements. Elements can be used to model star constellations, roads and county boundaries. Elements consist of coordinates and each coordinate in an element is stored as an (X, Y) pair. Point data will consist of only one coordinate while line data would consist of two.

- 2) Geometry: It is the representation or visualization of a spatial feature, modelled as an ordered set of primitive elements. It can consist of a single element or a collection of elements. The single element can be an instance of the primitive types that are supported by Oracle Spatial or a homogeneous or heterogeneous collection of elements. A heterogeneous collection consists of multiple elements of different types, for example, a point and a polygon.
- 3) Layer: It is a collection of geometries that have the same attribute set. Every layer has a set of geometries and an associate spatial index. Both are stored in the database in standard tables.
- 4) Coordinate System: It is a means of assigning coordinates to a particular location and building relationships between sets of coordinates. It helps us to visualize a set of coordinates as a representation of position in real world space. Spatial data might or might have a coordinate system associated with it. If it does then it can be geo-referenced which means that it is related to a specific representation of the Earth) or not geo-referenced.
- 5) Tolerance: It is used to associate a level of precision with spatial data. The restriction is that it must be a non-negative number greater than zero. What the tolerance value must be and its significance depends on whether the spatial data is associated with a geodetic coordinate system or not. Depending on the association, if it is geodetic data then the tolerance value is a number of meters else the tolerance value can be up to 1 which refers to the decimal fraction of the distance unit in use [7].

D. Oracle Spatial Query Execution Model

Oracle Spatial uses a query execution model which is two-tiered. It is used to resolve spatial queries and spatial joins. We can say that two separate operations are executed in order to resolve queries. The output after the second operation would provide an exact result.

- Primary Filter: It permits quick selection that might be good candidates and the selection is passed to secondary filter. It compares geometry approximations to reduce computation complexity and is considered a lower-cost filter. As it compares geometric approximations, it returns a superset of the exact result set.
- Secondary Filter: As shown in figure 3, secondary and is applied to the results of primary filter and not the entire database. In secondary filter same computations are applied to geometries as are applied in primary filter. This filter is considered as more expensive as it involves the execution of procedures, but yields an accurate result when a spatial query is fired.

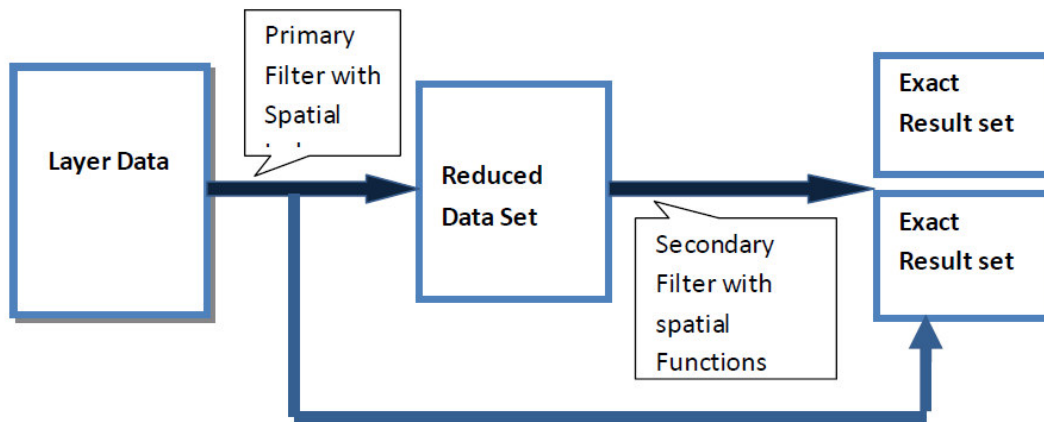


Figure 3: Query Model of Oracle Spatial

E. Indexing Mechanism of Oracle Spatial

Oracle Spatial makes use of R-Tree data structure for indexing information that span across multiple dimensions like geographical coordinates, polygons or rectangles. The concept of this data structure is to group nearby objects and represent them with their MBR (Minimum Bounded Rectangle). The leaf nodes represent a single object while the higher levels store the aggregation of an increasing number of objects.

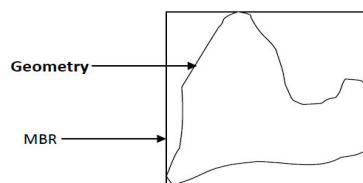


Figure 6: Minimum Bounding Rectangle - R-Tree Indexing

Oracle Spatial vs. PostGIS: Experiment and Results

For performance comparison same data set was used.

To load the data in Oracle Spatial, following command was issued:

```
java -classpath
C:\app\CHIRAG\product\11.2.0\dbhome_2\jdbc\lib\ojdbc5.jar;C:\app\CHIRAG\product\11.2.0\dbhome_2\md\jlib\sdoutl.jar;C:\app\CHIRAG\product\11.2.0\dbhome_2\md\jlib\sdoapi.jar
oracle.spatial.util.SampleShapefileToJGeomFeature -h localhost(host) -p 1521(port) -s orcl2(database) -u
system(username) -d <password> -t IND_roads(table name) -f C:\Users\CHIRAG\Desktop\oracle
spatial\IND_roads(.shp file location) -r 0 -g geom
```

Whereas, Postgres PostGIS supported GUI was used to load the data.

Query 1: select * from nyc_census_blocks;

SDBMS	Time (Seconds)	Cost (units)
Oracle Spatial	1.51	1068
Postgres PostGIS	0.00467	2064

The query returns 38794 rows and Oracle Spatial took 1.51 seconds and consumed 1068 units of computer resources. Postgres PostGIS took a mere 4.467ms and consumed 2064 units of computer resources. This means that PostGIS is almost 300% faster when it comes to select queries.

Query 2: select sdo_geom.sdo_length(geom,0.005) from nyc_streets where name='Columbus Cir';

SDBMS	Time (milliseconds)	Cost (units)
Oracle Spatial	5	516
Postgres PostGIS	2.431	764

This query is used to find the length of the road Columbus Cir. Oracle Spatial took 5ms to retrieve the results and used 516 units of computer resources. PostGIS took 2.431ms and consumed about 764 units of computer resources. That makes Postgres PostGIS almost 2 times faster for this case.

Query 3: select sum(sdo_geom.sdo_area(geom,0.005))/4047 from nyc_neighborhoods where boroname='Manhattan';

SDBMS	Time (milliseconds)	Cost (units)
Oracle Spatial	30	14
Postgres PostGIS	0.667	16

This query is used to find the area of the city of Manhattan. Oracle Spatial consumed about 14 units of computer resources and took 30ms. PostGIS consumed 16 units of computer resources and took a mere 0.0667ms. In this case PostGIS is almost 450% faster than Oracle Spatial.

Conclusion/Future Work

After executing queries on a similar data set, we concluded that Oracle Spatial is comparatively slower than PostGIS when it comes to accessing and querying data. There was a minimum of 50% difference between Oracle Spatial and PostGIS in all the queries. In the best case, PostGIS was almost 450% faster in executing a query as compared to Spatial whereas if computer resources are the parameter for comparison then query execution is costlier on Postgres PostGIS. Future work may involve incorporating more complex queries like containing joins between multiple tables or how both the databases behave when complex spatial operations are given to them through queries.

References

- [1] Güting, Ralf Hartmut. "An introduction to spatial database systems." *The VLDB Journal—The International Journal on Very Large Data Bases* 3.4 (1994): 357-399.
- [2] Shekhar, Shashi, et al. "Spatial databases-accomplishments and research needs." *Knowledge and Data Engineering, IEEE Transactions on* 11.1 (1999): 45-55.
- [3] <http://www.spatial.cs.umn.edu/Book/sdb-chap1.pdf>
- [4] https://en.wikipedia.org/wiki/Spatial_database
- [5] Paton, Norman W., et al. "VESPA: A benchmark for vector spatial databases." *Advances in Databases*. Springer Berlin Heidelberg, 2000. 81-101.
- [6] Stonebraker M., Frew J., Gardels K., and Meredith J.1993. The SEQUOIA 2000 Storage Benchmark, Proc. ACM SIGMOD Int. Conf. on Management of Data, Washington, DC, 1993, pp. 2-11.
- [7] http://docs.oracle.com/cd/E11882_01/appdev.112/e11830.pdf