Oracle Spatial and Graph: Faster, Bigger, Better 2-D and 3-D Spatial Solutions

Siva Ravada
Senior Director of Development
Spatial and Graph & MapViewer
The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract.

It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
Agenda

- Performance Enhancements
- Vector Geometry: New Features
- GeoRaster
- Network Data Model
Planned Oracle 12c
Goals for 12c

- Dramatically improve performance
  - Vector Performance Acceleration
  - Parallel Operations
- Reduce application logic and support real world analysis by moving complex spatial logic into the database
  - Real world application features for Network Data Model Graph
  - Server-based Raster Analytics and Image Processing for multi-terabyte cartographic modeling applications
  - Richer 3D and Point Cloud analysis and visualization; Parametric curves
  - RDF Semantic integration with Spatial Functions
- Ability track usage of different features
Main Issues From Customers

- Optimizer does not pick the right execution plan
- Too many recursive SQL calls to read index and geometry metadata
- Spatial index generates lot of redo for DML operations
- Concurrent DML operations produce locks on the index metadata table
- Make the functions and operators run faster
- Parallel index creation does not always work
- Java API is not easy to use
- Need data loading tools to read data from external data sources
- Web services are hard to deploy and use
Performance Enhancements
SQL Caching

- Kernel level caching for most common SQL queries
  - Geometry metadata
  - Index metadata
  - Coordinate System queries

- Database level caching
  - Sessions can share the cached results
SQL Caching Performance gains

- Substantial Performance gains for DMLs
  - Reduces the total time for each DML operation

- Spatial Operators are faster
  - Optimization especially noticeable in workflows with many fast running queries

- Coordinate System Transformations are faster
  - With EPSG model, we read data from many tables during transformation process
Parallel Index and Functions

- Improved the code so that more tasks can be parallelized
  - Tested with up to 32 CPUs
- Most of the Spatial functions enabled for parallel query
  - Validate_Geometry
  - Relate
  - Union and Intersection
  - Many other functions
- Fixed many bugs with parallel index creation
Oracle Spatial and Graph Extensible Optimizer

Issues with optimizer execution plans

- 11gR2: Relevant statistics about data distribution are not available
  - 12c: Extensive methods for fast, accurate statistics collection
- 11gR2: Incorrect assumptions built into cost models
  - 12c: Provide relevant index and cost functions for different spatial operators
- Statistics collection depends on an existing R-tree index for accessing geometry MBRs or their ancestors in the spatial index
- Execution of dbms_stats invokes spatial statistics collection algorithms and forms the MDXT_%objid%$ table
  - DBMS_STATS.GATHER_INDEX_STATS(...)
  - DBMS_STATS.GATHER_SCHEMA_STATS(...)

Oracle Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Redo Reduction

- Spatial index nodes are stored as LOBs
  - LOBs produce lot of redo when updated
- During DML operations, spatial index nodes get updated many times
- We now use better delayed write methods to reduce the number of times each node is updated on the disk
- Big improvements seen with customer data sets
  - 90% reduction in I/O in some cases
  - Up to 50% improvement in transaction times for most cases
DML Concurrency

- Different models used for single row DML and batch DML operations
- Locks on index metadata table eliminated
- For single-row DML, a bottom up approach is used
  - Locks the index table in row share mode
  - If updates happen in different locations, very unlikely to have locking conflicts for the index nodes
- Batch updates lock the index tables in exclusive mode
  - SQL caching and much smaller redo substantially reduce the duration of the locks
Faster Operators and Functions

1. Use existing picture box, DO NOT delete and
2. Right click on the shape.
3. At the bottom of the submenu select
4. Select "Fill" at the top of the "Format Shape"
5. Select "Picture or Texture fill" from the options.
6. And select "File" under the "Insert from" option.
7. Navigate to the file you want to use and
8. On the "Format" tab, in the Size group, click on

DELETE THIS INSTRUCTION NOTE WHEN
Vector Performance Accelerator
Vector Performance Acceleration

“Turbo-charger” feature for spatial functions and operations

Spatial and Graph option

Performance Improvements

- Union Operations: 5-10x
- DML single insert: 3x
- Coordinate System Transformations: 40-50%
- General DML operations: 30-40%
Vector Performance Acceleration

“Turbo-charger” feature for spatial functions and operators

Spatial and Graph option
Performance Improvements

- Join: 50-100x
- Touch: 50x
- Contains, Overlaps: 50x
- Complex masks: 50x
What is covered under VPA

- These Performance Improvements are only in Spatial Option with VPA
  - SQL Caching
  - RELATE Functions
  - All operators except
    - `ayinteract`, `inside`, `sdo_nn`, `sdo_within_distance`
  - `SDOJOIN`
  - Union operations (`AggregateUnion`, `SDO_UNION`, `AGGRSETUNION`)
- All other improvements are available to Locator
New Features in Locator

- All functions and procedures in SDO_GEOM package
  - Union/intersection, etc. operations are now part of Locator
  - All RELATE functions
  - All aggregate functions (except LRS functions)
  - All spatial operators for 2D vector data

- Java API
12c New Features For Vector Data
# Feature Usage Tracking

A new table to track feature usage

**MDSYS.SDO_FEATURE_USAGE**

<table>
<thead>
<tr>
<th>3D_FUNCTIONS</th>
<th>NDM</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGGREGATES</td>
<td>OPEN_LS</td>
</tr>
<tr>
<td>BUFFER</td>
<td>POINTCLOUD</td>
</tr>
<tr>
<td>CSW</td>
<td>RELATE</td>
</tr>
<tr>
<td>CS_TRANSFORM</td>
<td>ROUTER</td>
</tr>
<tr>
<td>GEOCODER</td>
<td>SAM</td>
</tr>
<tr>
<td>GEORASTER</td>
<td>TIN</td>
</tr>
<tr>
<td>INDEX</td>
<td>TOPOLOGY</td>
</tr>
<tr>
<td>LRS</td>
<td>UNION_INTERSECTION</td>
</tr>
<tr>
<td></td>
<td>WFS</td>
</tr>
</tbody>
</table>
Logical Standby and Replication

- Logical Standby supports SDO_GEOMETRY
  - No special work-around required
  - DDL, DML, geometry and index metadata supported
  - GeoRaster, Topology and NDM planned for next release

- Replication
  - Streams and Advanced replication are in support mode
  - Golden Gate is the supported feature for replication
    - Golden Gate supports all the spatial types
- New XStreams feature in 12c for client-to-database replication
  - Supports SDO_GEOMETRY
Parametric curve support

NURBS Data type support

- Support **Non-Uniform Rational B-Spline** (NURBS) curve geometries
  - NURBS curves represent arbitrary, free-form shapes
  - Control points and knots guide the shape of the curve
  - Enables complex shapes to be represented with little data
    - Highway ramps
- Support for NURBS curves includes the following
  - Functions to convert WKT/WKB and GML representations into SDO_GEOMETRY
  - Validation functions
- Spatial indexing
- Functions and operators approximate NURB curves for computations (implicit densification)
NURBS Data type support

- Implementation conforms to SQL/MM standard
  - SQL/MM standard for NURBS curves represents splines, polynomial splines, cubic splines, B-splines and Bezier curves
- SDO_GEOMETRY object type supports NURBS curves representation
  - 2D and 3D Curves supported
  - SDO_GTYPE 2002 and 2006 (for 2D Curve and MultiCurve)
  - SDO_GTYPE 3002 and 3006 (for 3D Curve and MultiCurve)
New PointInPolygon Function

Fast Point in Polygon without Spatial index

- **SDO_PointInPolygon Function**
  - Arg1: cursor that select a set of points
    - Very flexible as the data can come from a table, or result of another query
    - E.g., select * from point_data where c1 < 10 and c2 > 100 …
  - Arg2: is any Polygon geometry
    - Returns all the points that are inside the polygon

- Useful when large number of points have to be classified based on a set of polygons
- Parallel enabled
- Can easily process 30K points per second in serial case
New Clip PC Function

Fast clip PC when data is stored as Relational table

- SDO_PC_PKG.Clip_PC_Flat
  - Use a flat relational table for ingesting large volumes of data
  - Since there is no index overhead, the ingest is very fast
  - Typically used for data collection
- Clip data based on any polygon geometry
  - Other filter conditions can be passed in as parameters
    - Filter based on time, intensity, or any of the other attributes
- Mainly designed for Exadata systems
  - It will still work very well on cluster systems and can use very high degree of parallelism
Schema based storage for Point Clouds

- XML schema based storage for SDO_PC data
  - This is similar to the metadata schema used for GeoRaster
  - First three fields should be doubles to store X,Y,Z
  - The optional fields can have one of the following data types
    - 8, 16, 32, and 64 bit signed and unsigned integer
    - Double, float
    - Strings can be stored as unsigned 8 bit integer (1 byte)
- Useful for publishing PC data for end user applications
- PDAL is an open source loading tool that supports SDO_PC
3D Geodetic Support

- DISTANCE, ANYINTERACT and INSIDE operations take height into account
- LRS functions also support 3D geodetic data
- Height for geodetic data is usually supplied as meters
  - Or it can be any other UNIT type defined by a coordinate system
Topology Data Model

- Performance Improvements
  - Get_Geometry: orders of magnitude faster
  - Edit operations
- Robust addFeature support
  - Feature snapping
- Workspace Manager support for Topology
  - Support many common use cases for collaborative editing of topologies
Java API, Tools and Web Services
Simplified Java API for 2D and 3D

- SDOAPI contains an in memory R-Tree for primary filter operations
- New SDOAPI functions, enhanced to support the following in-memory operations
  - Distance
  - Inside
  - AnyInteract
- New in memory functions support the following geometries types
  - Projected 2D and 3D
  - Geodetic 2D and 3D
  - Cricular and NURB curve types
Web Feature Server 1.1

Web-based Administrative console

- Menu driven GUI simplifies registration of spatial layers with Oracle Spatial and Graph's WFS
  - Browse existing spatial layers
  - Eliminates the need for DBA to run PL/SQL scripts to publish spatial layers
- Includes tutorial on how to configure and use WFS
- Provide sample request and response pages for WFS queries
- Can also be used as a client to other WFS servers
  - In this mode, only browsing options are enabled
SQL Developer

- Fully integrated support for SDO_GEOMETRY
- Metadata data management via the GUI
- Visualization of Geometry
- Next version in development
  - Better support for overlays of geometries
  - More options for visualization styles
- What is the future?
  - Support GeoRaster, Topology and other types?
  - Need feedback
GeoRaster
GeoRaster 12c New features

- **Advanced Queries and Manipulations**
  - From general spatial queries to content and context based raster analytical queries and image aggregations

- **Massive In-Database Processing**
  - From a focus of database management capabilities to in-database image processing and raster analysis engines

- **High Performance Computing**
  - From serial processing to parallel processing
Support Raster Algebra

• **Raster Algebra (Map Algebra)** is a set-based algebra for manipulating geographic raster data
  • It’s widely used for cartographic modeling and spatial analysis
  • It includes *a set of primitive algebraic operators* applied on one or more raster layers of similar dimensions to produce one or more new raster layers or other values
  • There are *4 types of operations*: local, focal, zonal and global
  • It provides *a procedural or scripting language* enabling very complex operations
GeoRaster Raster Algebra and Analytics Engine

- Raster Algebra Language is provided as an extension to the PL/SQL language
- Supports mainly local raster algebra in this release
- Includes an expression language for raster algebra operators
  - general arithmetic, casting, logical and relational operators
- Includes raster algebra functions
  - cell value based conditional queries
  - cell value based conditional updates
  - arithmetic operations
  - raster classification/segmentation
Historical Temperature Analysis

- A collection of 33 years of global temperature data for each month. In total, 396 layers stored in 1 georaster object:

1. The values are in Kelvin, which is converted to Fahrenheit
2. Generate the Average (mean) temperature for each month
3. Compute the Mean Absolute Deviation per Month (the mean difference from the average across 33 years for each month).

- use PL/SQL program to generate all expressions on the left and execute the rasterMathOp operations.

-- converting Kelvin to Fahrenheit
'({0}-273.15)*9/5+32'

-- compute average temperature for a month (January)
'((0)+{12}+{24}+{36}+{48}+{60}+{72}+{84}+{96}+{108}+{120}+{132}+{144}+ {156}+{168}+{180}+{192}+{204}+{216}+{228}+{240}+{252}+{264}+{276}+{288}+{300}+{312}+{324}+{336}+{348}+{360}+{372}+{384})/33'

-- compute mean absolute deviation for a month (January)
'((abs({1,0}-{0,0})+abs({1,0}-{0,12})+abs({1,0}-{0,24})+abs({1,0}-{0,36})+abs({1,0}-{0,48})+abs({1,0}-{0,60})+abs({1,0}-{0,72})+abs({1,0}-{0,84})+abs({1,0}-{0,96})+abs({1,0}-{0,108})+abs({1,0}-{0,120})+abs({1,0}-{0,132})+abs({1,0}-{0,144})+abs({1,0}-{0,156})+abs({1,0}-{0,168})+abs({1,0}-{0,180})+abs({1,0}-{0,192})+abs({1,0}-{0,204})+abs({1,0}-{0,216})+abs({1,0}-{0,228})+abs({1,0}-{0,240})+abs({1,0}-{0,252})+abs({1,0}-{0,264})+abs({1,0}-{0,276})+abs({1,0}-{0,288})+abs({1,0}-{0,300})+abs({1,0}-{0,312})+abs({1,0}-{0,324})+abs({1,0}-{0,336})+abs({1,0}-{0,348})+abs({1,0}-{0,360})+abs({1,0}-{0,372})+abs({1,0}-{0,384}))/33'}
Historical Temperature Analysis: Average

June Average Temperature over 33 Years
(Source Temperature Data Courtesy of Remote Sensing Systems - http://www.ssmi.com/)
New and Advanced Large-Scale Mosaicking

- `sdo_geor_aggr.mosaicSubset` to create persistent physical mosaic
- A complete new implementation, parallelized and scalable
- Source images can be tables, views or a SQL statement
- Source images can be rectified or unrectified or in different CS
- Support internal resampling, reprojection or rectification
- Supports gaps, no data, and overlapping regions
- Support 8 common point rules (max, min, avg, LATEST, OLDEST ...)
- User defined priority for overlapping regions (Date or SQL ORDER BY)
- Simple color balancing (linear stretch and normalization)
Large-Scale Appending

- **sdo_geor_aggr.append** to update existing image with new ones
- Generally used to append or update a large **physical mosaic** using newly acquired smaller images
- The source image can be on **ANY SIDE** of the target image!
- Support the same functionality as mosaicking:
  - Source images can be rectified or unrectified or in different CS
  - Support internal resampling, reprojection or rectification
  - Supports gaps, no data, and overlapping regions
  - Support 8 common point rules (max, min, avg, LATEST, OLDEST, etc...)
  - Simple color balancing (linear stretch and normalization)
Virtual Mosaic - An Advanced Image Query and Image Serving Engine

- A virtual mosaic is defined as any large collection of georeferenced GeoRaster objects (images) that is treated as if it is a single GeoRaster object (physical mosaic).

- Three ways to define a virtual mosaic:
  - a list of GeoRaster tables
  - a database view with a GeoRaster column
  - a SQL query statement (i.e., a CURSOR)

- A virtual mosaic can contain unlimited number of images of any size

- There is no need to define a description file for the virtual mosaic
Sample Virtual Mosaic Use Cases

- DOQs can be stored as is but used as seamless mosaic (as fast as physical mosaics)
- Users may not want to mosaic DEM’s
- Store large volume of imagery without making too many extra copies (save disk space)
- New images coming in and immediately displayed on the mosaic (dynamic updates)
- The same images can be displayed or removed in different virtual mosaics (flexible model)
Image Rectification and Orthorectification

- **Rectification** of georeferenced raw images
- **Orthorectification** of georeferenced raw image using DEM
- Both support on-the-fly cropping queries or for persistent storage

(Image Courtesy of Digital Globe)
New and Enhanced Tools and Java API

- **New Concurrent Batch Loading and Exporting Tool**
  - GUI to create GDAL-based batch loading and exporting description files (XML)
  - GUI to load batches of files concurrently
  - All GDAL supported file formats

- **GeoRaster Java API Enhancement**
  - Support for ground control point (GCP) storage and manipulation, GCP georeferencing, reprojection, grid interpolations, and getCellValue

- **GeoRaster viewer enhancement**
  - Display a virtual mosaic defined as one or a list of GeoRaster tables or views
  - Zoom-in, zoom-out and pan
New Concurrent Batch ETL Tool

- Oracle® Spatial and Graph GeoRaster Batch ETL User Guide
  ($ORACLE_HOME/md/demo/georaster/etl/georaster_etl_user_guide.pdf)
Network Data Model
Feature Modeling/Analysis

Model networks with application features

- Intuitive representation for applications
- Similar to Topology Data Model but not limited to geometry based features
  - Supports logical networks and physical networks in application representation
- Data model to manage node and link “features” with their associated “network elements” (Nodes/Links)
  - Node features: transformers, sub-stations, etc.
  - Link features: power lines, transit routes, etc.
- Consistency between network features and network elements automatically maintained
- Feature level analysis
  - Find the shortest path between two transformers
  - Find the shortest path between two transformers, but use only a certain wire type
Feature Modeling / Analysis

Example

- Create a POI feature layer (Hotels) on top of a road network
- Create a POI feature layer (Customers) on top of a road network
- Add Hotel and Customers feature layers to the road network
  - Hotels/Customers are point on link features (LINK_ID, Percentage)
- Create a transformer feature on a utility network
- Define rules for feature snapping
- Queries based on features
  - Find the shortest distance hotel from a customer
  - Find the 3 nearest hotels from a customer
Temporal Modeling/Analysis

NDM Traffic Patterns

- Traffic Patterns
  - Traffic patterns record historical travel patterns for different classes of roads
  - Data is collected based on time of day and day of the week
  - Each road segment can have hundreds of travel patterns
- NDM can use this data while computing shortest paths
  - Analysis function APIs are extended with the concept of time
  - Find the shorted path from point A to point B with a start time of 9AM
  - Find the shortest path from point A to point B so that the destination is reached at 5.30PM
- Support NAVTEQ Traffic Patterns format out of the box
Best Routes Using Traffic Pattern Information

Shortest Path Analysis
Left click for start point, right click for end point, or manually enter node ID, link ID, or address.
Start: 199408837
End: 199919125

Network Constraints
- Field entry for multi-select or de-select
  - Custom: No Highways Constraint
  - Custom: Prohibited Zone Constraint
  - Custom: Spatial Mode Filter

Prohibited Zones

Link Cost Calculators
- Custom Traffic Link Cost Calculator

Keep Previous Results
Reverse Direction
Include Traffic Data
Start Time: 10:00 PM

Find Shortest Path

Analysis Result:
104940227->199919125
Time to analyze the network: 0.407s
Time to compute geometries: 0.055s

Analysis Result:
195448387->199919125
Time to analyze the network: 0.407s
Time to compute geometries: 0.055s
Multi-Modal Routing

Modeling

- Each service network is modeled as an NDM network
  - Bus, Train, Subway, Ferry, Flight
- Each service network is accessible from a base network
  - Road network/Pedestrian network
- The multi-modal network is an aggregate network of all service networks and the base network
- Transfers nodes and links that travel across the same/different networks need to be specified
- Schedule information is stored at node level
- Geometry (schematic) information is optional for visualization purposes
Enhanced NDM XML API

Web Service Enabled

- Provides a full fledged web services framework for network analysis
- Enhanced XML API with Network Constraint and Cost Calculator Support
- Integrated with Oracle Spatial and Graph Web Service Framework
- A Simple PL/SQL Wrapper (http put/get) on top of the XML API for Database Users
- A web service demo based on Oracle Spatial Web Service Framework is available in ndmdemo download kit
- New Driving Direction API
  - For NDM users with data in Routing Schema
  - Generate step by step driving directions format for final path
Oracle Spatial and Graph - Routing Engine

Enhancements

- Enhanced Navigation Support
  - Multi-link turn restrictions – Highway has entry point, and left
  - Restrictions where there are no physical barriers

Enter highway with no turn restriction on upcoming left hand exit.

Left lane only exit
Q&A