ORACLE

May 22, 2013

Ronald Reagan Building and International Trade Center

Washington, DC USA

Leveraging Advanced Oracle Spatial and Graph Features

Daniel Geringer
Senior Software Development Manager

To

1.
2.
3.
4.
5.
6.
7.
8.
9.

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.

Program Agenda

- Newest LIDAR Data Model
- Newest GeoRaster Functionality
  - In Database Raster Algebra
  - Physical and Virtual Mosaic
  - New GUI For Loading and Unloading Rasters
- Newest Network Graph Functionality

Newest Oracle Spatial and Graph LIDAR Data Model

What is LIDAR Data

Laser Imaging Detection and Ranging (LIDAR) Data

- Optical remote sensors that can collect millions of 3D points per second, along with numeric attributes associated with each point.
- Sensors targeted at the ground generate rich elevation data models, sometimes called point clouds.
- Raw point cloud data can be further refined for:
  - Feature extraction to accurately capture asset locations. Commonly used by departments of transportation, rail companies, agriculture, environmental agencies, etc…
  - Derivative product generation (TINs, DEMs, Contour Lines, etc…)

Oracle's Newest LIDAR Data Model

- "New paradigm" for LIDAR data, optimized for Oracle engineered systems.
- On Exadata and Sparc Supercluster (SSC), leverages Oracle's unique:
  - HCC compression technique, for extremely high compression rates
  - Parallel Enabled Query Offload for extreme performance, including spatial queries.
  - **No indexes**
- Will work on commodity hardware too
  - No HCC compression or Query Offload
  - Requires an index

LIDAR – Ingest and Compression

Proof of Concept

Compression Used	Rows	Size GB	Compression Ratio	Ingest Time (sec)	Ingest Pts/ Sec	Compressed vs NonCompressed Time
No Compression	**2,853,027,995**	**286.80**	--	**240**	**11,887,617**	0 X slower
Query Low	2,853,027,995	37.54	7.64 X	370	7,710,886	1.54 X slower
Query High	2,853,027,995	14.40	19.92 X	486	5,870,428	2.02 X Slower
Archive Low	2,853,027,995	14.36	19.98 X	504	5,660,770	2.1 X Slower
Archive High	2,853,027,995	10.61	27.03 X	1043	2,735,406	4.35X Slower

Oracle's Newest LIDAR Data Model
(continued…)

- "Back to basics… a simple flat relational model
- X,Y,Z and attributes stored as ordinary Oracle NUMBER columns
- Table can contain other columns data types too.
- Works with any point table/view, not just LIDAR data
  - X, Y must be first 2 columns
- Java based loader for:
  - LAS to flat model
  - Flat model to LAS

Code Example – Java, C++, and Other Languages Too


```
DECLARE
  my_cursor SYS_REFCURSOR;
BEGIN
  -- Polygon query, and also Z between 10 and 20 and the 5th column between 100 and 200
  my_cursor :=  SDO_PC_PKG.CLIP_PC_FLAT(
                  geometry         => SDO_GEOMETRY(2003, 8307,NULL,
                                                   SDO_ELEM_INFO_ARRAY(1, 1003, 3),
                                                   SDO_ORDINATE_ARRAY(0,0,1,1)),
                  table_name       => 'TEST_LIDAR',
                  tolerance        => 0.05,
                  other_dim_qry    => SDO_MBR (
                                        SDO_VPOINT_TYPE(10,null,100),
                                        SDO_VPOINT_TYPE(20,null,200)),
                  mask             => 'ANYINTERACT');

FETCH my_cursor BULK COLLECT INTO result_list;
```


LIDAR Query Results

- Queries issued through a Java program with variables assigned
- Queries returned between 100 thousand and 100 million rows

Query ID	Query Low Rows/Sec	Query High Rows/Sec	Achive Low Rows/Sec	Archive High Rows/Sec
1	136,559	143,191	128,505	102,175
2	136,431	139,037	135,704	99,331
3	447,706	448,822	437,729	420,029
4	140,469	128,679	142,332	111,441
5	435,641	432,665	427,734	406,296
6	442,705	427,658	428,184	418,909
7	553,778	560,853	558,951	560,228
8	562,259	573,090	574,527	583,569
9	575,910	587,583	588,732	593,418

Newest LIDAR Functionality

Derivative Products

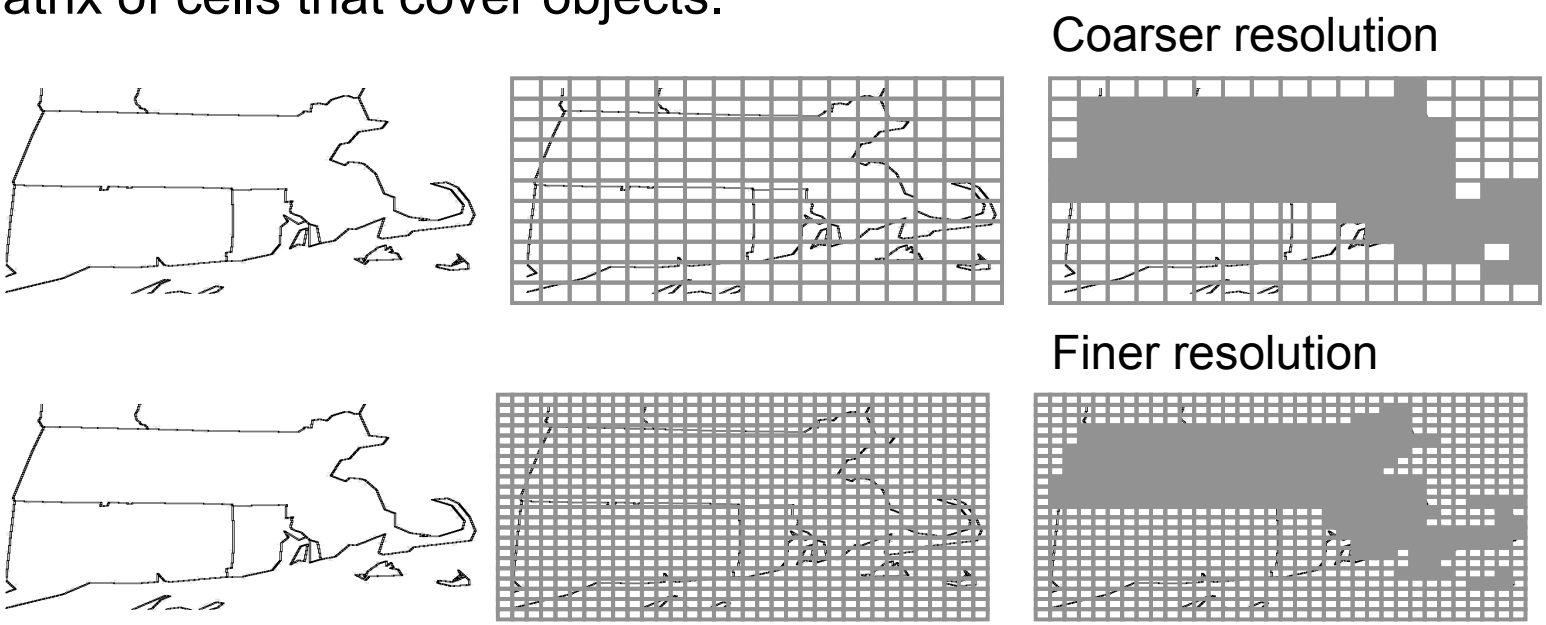
- Digital Elevation Model (DEM) rasters
- Triangulated Irregular Networks (TIN)
- Countour Lines

Newest Oracle Spatial and Graph GeoRaster Features and Functionality

Raster Concepts

What Is Raster Data?

- Raster data is spatial data that is created by assigning values to a matrix of cells that cover objects.



A Raster Is A Matrix Of Numbers

Value Attribute Table

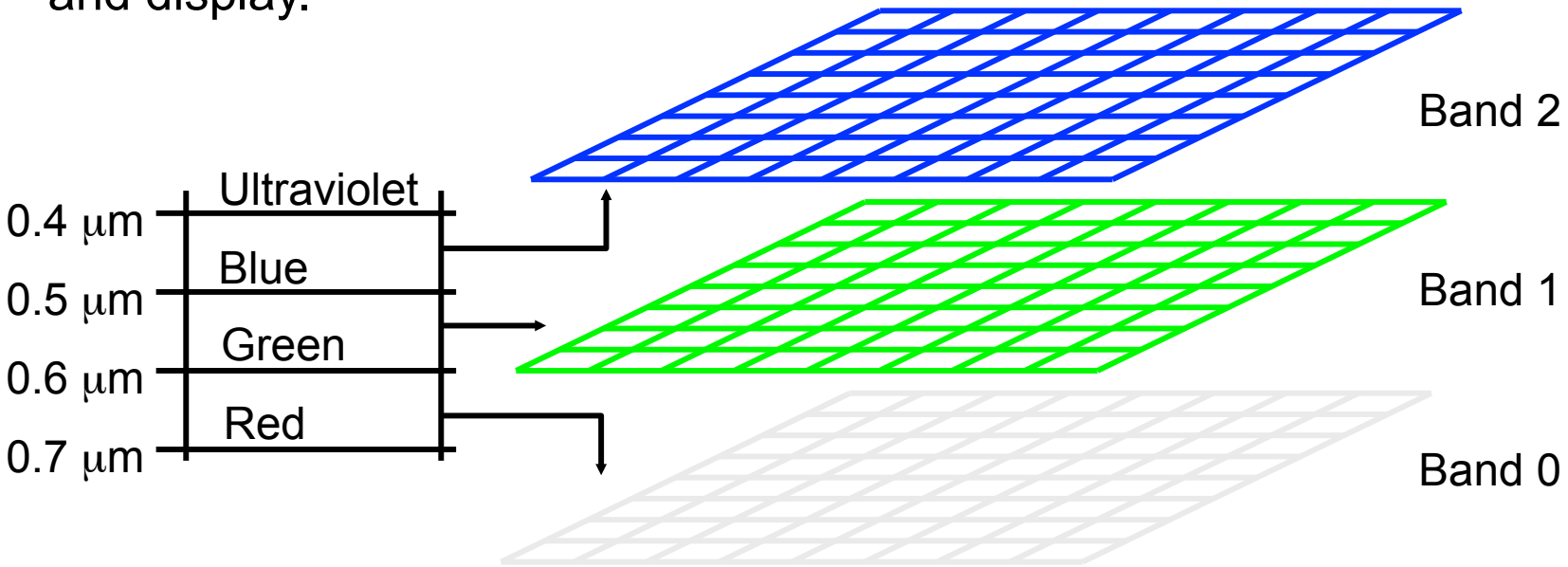
Cell Value	Geological Period
1	Quaternary
2	Tertiary
3	Mesozoic
4	Proterozoic
5	Archaean
6	Gondwana

Matrix of raster cells (or pixels)

2	2	3	5	5	5	5
2	2	3	5	2	2	2
4	3	3	1	3	5	5
4	4	4	6	6	3	5
4	4	4	3	6	4	4
2	2	3	3	3	3	4
2	2	3	5	5	4	4

MultiBand Digital Image: Example

- Each band is collected at a different wavelength for later processing and display.



In Database Raster Algebra

New Raster Algebra and Analytics Engine

- Four Map Algebra cell value based functions:
  - Conditional queries - Searches/masks cells based on booleanExpr
  - Updates - Update cells of a raster-based on booleanExpr
  - Arithmetic operations - ADD, DIVIDE, LOG, etc.
  - Cassification for raster segmentation - Applies arithmeticExpr to cells and then segments the raster
- **Can specify the Parallel Degree for Raster Algebra operations**.
  - Routines leverage native Oracle parallel query
  - Key differentiator

Raster Algebra – Conditional Queries

SDO_GEOR_RA.findCells

- Given a source raster, generate a target raster.
- Target raster is a copy of the source raster, with certain pixels are masked (set to a background color) based on conditions.
- Examples for a three band raster:
  - { 0 } >200 – Mask all pixels where a cell in band 0 is greater than 200
  - ({ 1 }>200 & { 2 }>500) - Mask the pixel if both conditions are true:
    - Band 1 cell value is greater than 200
    - Band 2 cell value is greater than 500
  - ({ 1 }>200 & { 2 }>500) | { 3 }>1000 - "OR" band 3 cell value is > 1000

Raster Algebra – Conditional Updates

SDO_GEOR_RA.rasterUpdate

- Evaluate a user define condition for each pixel.
- If the condition is met, update all the cell values for that pixel.
- Example for a three band raster:
  - conditions = '(abs({0}-{1})=48) & ({2}-{1}=-101)‘
    - ABS (band 0 cell value – band 1 cell value) is 48
    - AND (band 2 cell value – band 1 cell value) is -101
  - If condition is met, set pixel’s cell values to ('123','54','89')
- You can also pick the pyramid level for the operation

Raster Algebra – Arithmetic Operations

SDO_GEOR_RA.rasterMathOp

- Takes two or more rasters as input.
- Performs mathematical operation across cell values in different rasters and generates a new raster.
- You can define your own arithmetic operation {raster id, band}
  - ('{0,0}/2', '{1,2}/2', '{0,2}/2')
  - Output Raster is:
    - First raster band 0 cell values divided by 2
    - Second raster band 2 cell values divided by 2
    - First raster band 2 cell values divided by 2
- Or pick from defined pre-defined operations:
  - OPERATOR_ADDCONST, etc… defined in user's guide.

Raster Algebra – Classification For Segmentation

SDO_GEOR_RA.classify

- Takes bands from one or more rasters as input
- Applies a user defied expression to cell values, for example:
  - {1,0} / {2,1} * 100
  - {Raster 1 band 0 cell value} divided by {Raster 2 band 1 cell value} times 100
- Expression result maps to range array, and a new cell value is assigned
- For example:
  - If expression above evalutates to 75, the new cell value is assigned 80
    - rangeArray:=sdo_number_array(70,80,90,100,110,120,130,140)
    - valueArray:=sdo_number_array(70,80,90,100,110,120,130,140,150)

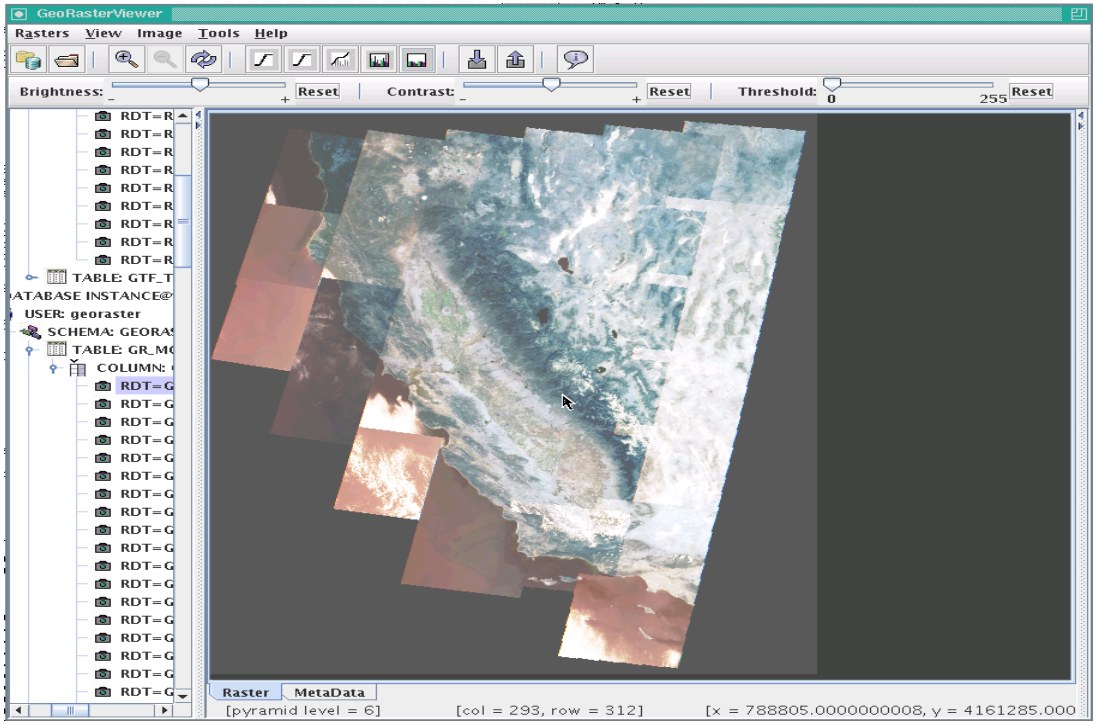
In Database Raster Algebra

Summary

- Algebraic functions applied to raster data to derive new results
  - Generate new raster layer from two or more raster layers
  - Raster algebra operations to implement sophisticated analytical algorithms, such as a Normalized Difference Vegetation Index (NDVI) , and TCT (Tasseled Cap Transformation)
  - NDVI helps classify the amount of vegetation in a region, from none to rainforest
- Raster Algebra operations can leverage Oracle Parallel Query
- Optimized for parallel hardware architectures

GeoRaster
Physical and Virtual Mosaic Support

Mosaic of Lansat Images



Newest Mosaic Functionality

- Mosaic can be persisted (physical), or virtual (on the fly)
- Rasters can be different coordinate systems and resolution (resampling)
- Pyramids can be mosaiced
- Virtual mosaic defined by a SQL statement or view
- User defined priority for overlapping regions (Date or SQL ORDER BY)

Newest Mosaic Functionality

- Supports gaps, no data, and overlapping regions
- More advanced mosaicking support:
  - georeferenced raw images
  - internal reproject/rectification
  - common point rules (max value, min value, etc...)
  - large-scale image append

GeoRaster
Parallel Load and Unload
New Graphical User Interface

GeoRaster – Parallel Load and Unload

New Graphical User Interface (GUI)

- Oracle developed GUI based on GDAL
- GDAL… most widely accepted “open source” ETL software for raster data.
- GDAL can transform to/from hundreds of raster formats
- GUI designed to parallel process batch loading and unloading of raster data
- Download here:
  - Oracle Spatial and Graph page on the Oracle Technology Network (OTN)
  - Click on “Downloads” tab
  - Click on “Spatial and Features Sample Code” link
  - **GDAL-based GeoRaster Concurrent Batch Loading and Exporting Wizard**

Building A Batch With The GUI

- GUI builds XML files (see next slide)
- Each XML file is considered a BATCH job
- Each batch job contains one or more GDAL based COMMANDS to load / unload rasters
- GUI executes BATCH jobs in parallel

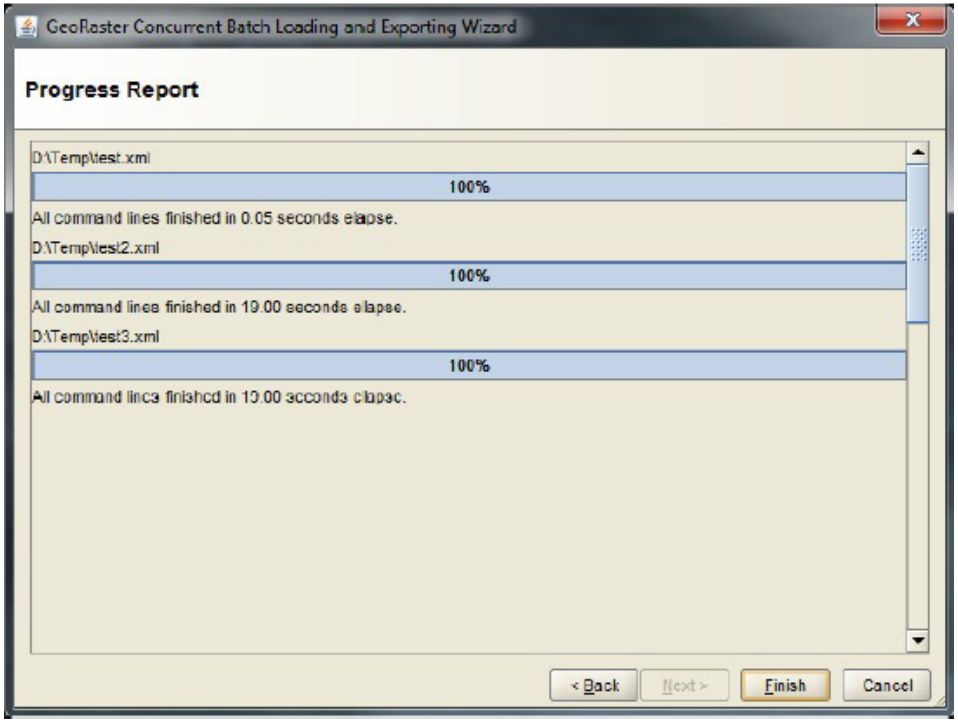
Batch Can Contain Many Commands

- This batch (XML file) only has one command… loads one raster.
- Multiple commands in a batch are not run in parallel, but batches are run in parallel.

```
<Batch>
 <Command>
    <Source>D:\RASTER_DATA\raster.tif</Source>
    <Format>GEORASTER</Format>
    <Target>tableName,rasterColumn</Target>
   <CreationOptionList>
       <BLOCKYSIZE>512</BLOCKYSIZE>
       <INSERT>(SOURCE_FILE,GEORASTER,GEORID) VALUES
                  (‘raster.tif',sdo_geor.init('RDT_01‘,10),10) </INSERT>
       <BLOCKXSIZE>512</BLOCKXSIZE>
    </CreationOptionList>
 </Command>
</Batch>
```

Batches (or XML Files) Executed In Parallel - Progress

Three batches executed in parallel



Oracle Spatial and Graph
Network Graph
New Features

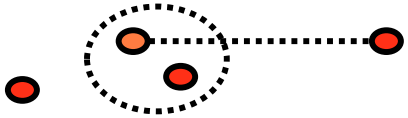
Network Graph Concepts

Spatial Analysis Versus Network Analysis

- Oracle Locator and Oracle Spatial solve spatial proximity problems.
- Another type of analysis that is required by users and applications is network analysis.
- Network applications deal with the connectivity of features. Spatial data is optional.

Spatial Closest feature

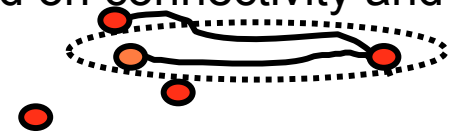
(based on distance)



Verses

Network Closest feature

(based on connectivity and cost)



What Is Oracle Spatial & Graph Network Graph?

- An open data data model to store and analyze network data.
- Connectivity is determined using nodes and links:
  - Each link has a start node and an end node.
  - Links and/or nodes can have costs
  - Links can be one way or bi-directed

What Is Oracle Spatial & Graph Network Graph?

- Analysis is based on connectivity and optionally cost information.
- Network analyses includes:
  - Shortest path analysis
  - Nearest neighbor analysis
  - Within cost analysis
  - Minimum cost spanning tree
  - Traveling salesman problem
  - Reachable/Reaching nodes
  - K-shortest paths analysis

Network Graph Demo

- Demo available on OTN Spatial Page
- San Francisco data sample supplied by Navteq

Feature Modeling/Analysis

Model networks with application features instead of nodes and links

- Data model to manage node and link “features” with their associated “network elements”
  - 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
- Model is similar to Topology but not limited to geometry features
  - Supports for logical network and physical network application features

Network Constraint Example

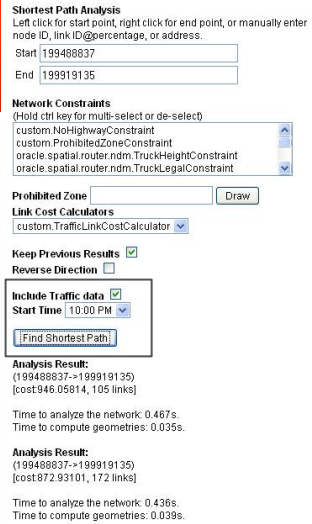
Customize Oracle's Algorithms

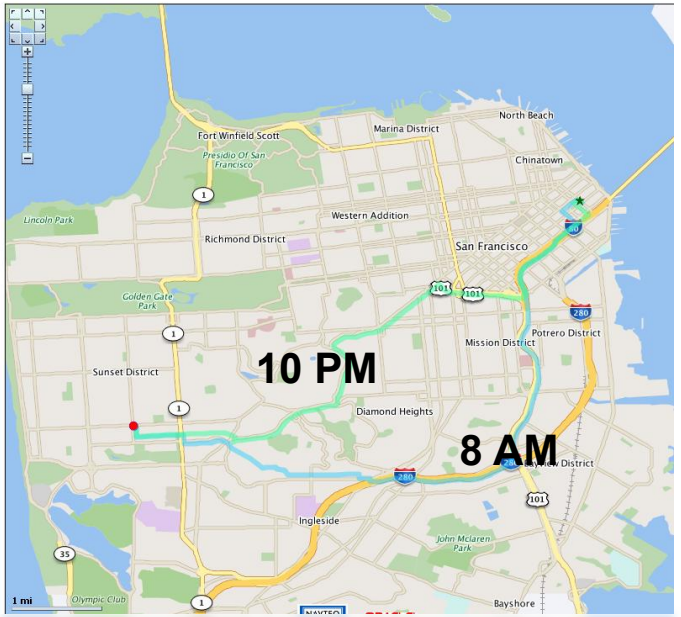

```
public class NoHighwayConstraint  implements LODNetworkConstraint
{
  public  NoHighwayConstraint(){}

 public boolean isSatisfied(LODAnalysisInfo info)
 {
   LogicalLink link = info.getNextLink();
   if (link==null || link.getLevel() == 1 )
     return true;
   else
     return false;
 }
```

Oracle Spatial and Graph - Network Data Model Best Route Using Traffic Pattern Information

Temporal Modeling/Analysis

- Traffic Patterns
  - Record historical travel patterns for different classes of roads
  - Data collected based on time of day and day of the week
- Use traffic patterns to compute shortest paths
  - Find shorted path from point A to B with start time of 8 AM
  - Find shortest path from point A to B and reach destination at 5.30PM
- Support NAVTEQ Traffic Patterns format out of the box





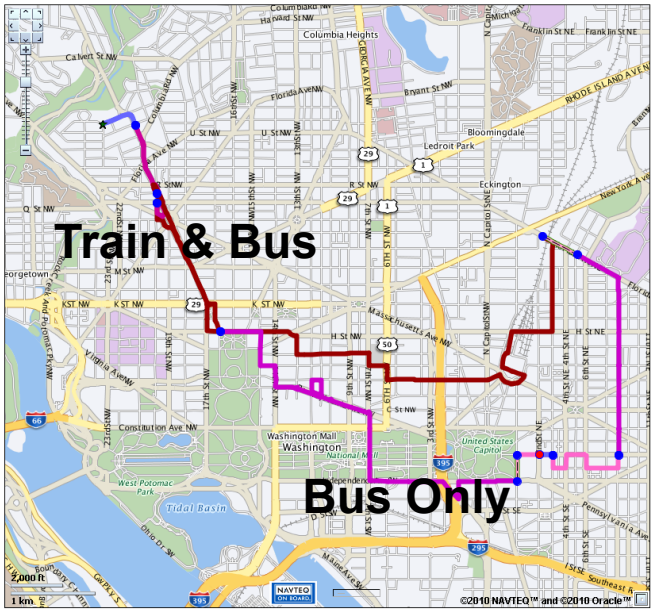
 | |

Oracle Spatial and Graph - Network Graph Routes with Preferences (walk/car/bus/subway)

Multi-Modal Routing

- Each mode (car, bus, rail, bike, etc) modeled as a separate network
- Single logical network represents all modes of transportation
- Transition nodes where networks meet
- NDM APIs can specify the modes to consider
- Out of the box support for transit data published by transit authorities
  - GTFS (General Transit Feed Spec) supported

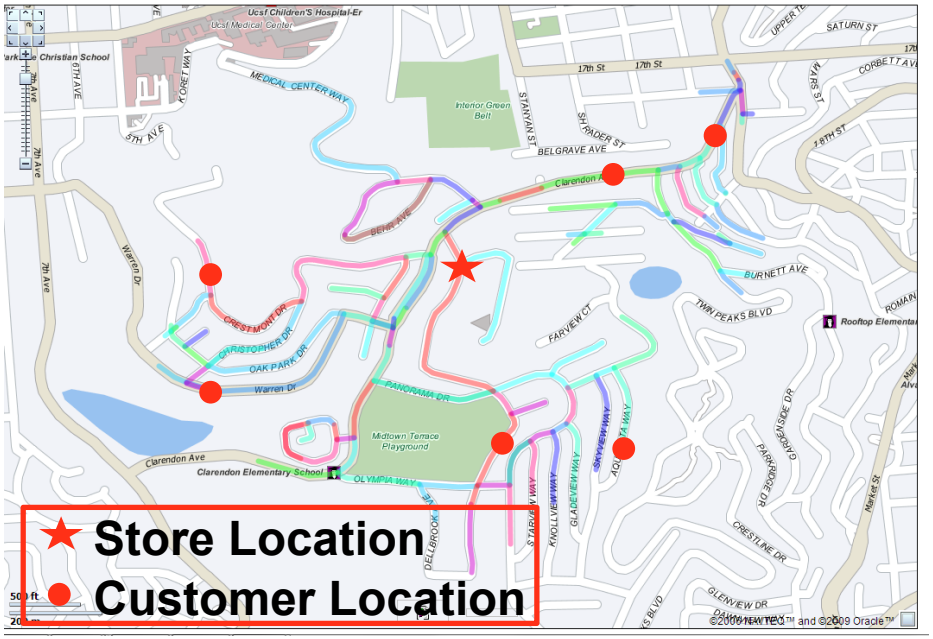
```
Analysis Result:
From: 575456205
To:   575481535

Drive/Walk to
'CONNECTICUT AV and WYOMING AV'
 (31 meters).

[1]
Board Route 227 (Inbound)
At 'CONNECTICUT AV and WYOMING AV'
Dep. Time : 10:10:42

Get down at 'NW CONNECTICUT AV and NW 20TH ST';

[2]
Transfer to Route 86
Board Route 86 (Outbound)
At 'NW CONNECTICUT AV and NW 20TH ST'
Dep. Time : 10:21:00

Get down at 'NW H ST and NW JACKSON PL';

[3]
Transfer to Route 75
Board Route 75 (Inbound)
At 'NW H ST and NW JACKSON PL'
Dep. Time : 10:32:42

Get down at 'SE INDEPENDENCE AV and SE 1ST ST';

[4]
Transfer to Route 131
Board Route 131 (Outbound)
At 'E CAPITOL ST and SE 1ST ST'
Dep. Time : 11:01:06

Get down at 'E CAPITOL ST and SE 3RD ST'
At 11:02:00

Drive/Walk from
'E CAPITOL ST and SE 3RD ST'
 (0 meters) to destination.

Trip Travel Time:  51 minutes.

Number of Bus Routes=4
Number of Train Routes=0

Time to analyze the network: 0.914s.
```



Large Scale Drive Time/Distance Analysis

For millions of customers, find closest store within a specified drive time

- Same underlying data for geocoder and road network
- Customers geocode as link id and percentage (instead of longitude/latitude)
- 5 mile Network Buffer generates all possible paths
- Each persisted path includes:
  - Covered link IDs, nodes ID, and associated costs
- Single database query to find closest store and drive time/distance for each customer (join on link_id)



 | |

Enhanced NDM XML API (Callable From PL/SQL)

- 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
- New Driving Direction API
  - For NDM users with data in Routing Schema
  - Generate step by step driving directions format for final path
  - Oracle Route Server does not have to be deployed

Q&A

ORACLE

May 22, 2013

Ronald Reagan Building and International Trade Center

Washington, DC USA