Network Data Model – A Spatial Feature of Oracle Database

June 2021

Public

Purpose statement

This document provides an overview of Network Data Model feature included with Oracle Database. It is intended as a technical document to help developers assess the benefits of using this feature and to plan I.T. projects. An understanding of geospatial networks and network analysis is beneficial when reading this paper.

Disclaimer

This document in any form, software, or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement, nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. 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 in this document remain at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Introduction

Network Data Model (NDM) is a spatial feature of Oracle Database that lets you store and analyze networks used in transportation, utilities, and communications. NDM persistently manages the network connectivity in the database.

A Java API and a REST API provide fast in-memory graph analytics for use with spatial networks, including shortest path, nearest neighbors, within cost, and reachability. NDM provides an open data model that simplifies network modeling, analysis, and management so users can focus on application logic. It separates application-specific information from connectivity relationships allowing many network applications to use the same data model. NDM also lets users specify rules and attributes to guide analysis based on the application logic. For example, when designing a transportation application, rules to enforce speed limits, time of day turn restrictions, and one-way traffic can be added to the network.

NDM makes the most of Oracle Database features for scalability, security, partitioning, and parallelism. It supports Oracle Cloud offerings and can exploit the processing power and bandwidth of Oracle Exadata Machine to obtain excellent performance capabilities that are orders of magnitude over what was previously possible.

This technical brief provides a technical overview of the Network Data Model, including the data model, architecture, and other NDM features. It discusses the Load-on-Demand (LOD) Java API and Contraction Hierarchies REST API, describes how and when to use them in a network application.

The Network Data Model contains network information in the database. For different network analysis approaches, additional information needs to be computed and stored. NDM supports two analysis approaches: the Load-On-Demand (LOD) approach and the Contraction Hierarchies (CH) approach. The NDM data model in the database serves as a primary network data source for both these analysis models. In addition, users can generate the additional data directly from the NDM data model in the database.

Network Data Model in Database

The Network Data Model contains the following database objects: network metadata, network tables, user-defined data, and partition tables. Users can create and update an NDM data model with SQL statements.

Network Metadata

Network metadata contains general information about networks. It can also include the geometry metadata information for spatial networks.

Network Tables

A network in Oracle Database contains a node table and a link table. Nodes represent objects of interest and links represent the relationship between nodes. The network data model can handle geometry information. That is, the network data model can represent both logical and spatial network applications. Adding geometric data to a logical network will allow the logical network to be displayed and spatially analyzed. Figure 1 shows the schema for the network data model, which includes these tables.

User-Defined Data

In addition to the connectivity information associated with a network, networks often include related application data. For example, a street network might include the name of the street, the speed limit, and the road segment ID in addition to the information about the road segments to which it is connected. NDM lets users define their application data as part of a network model. Network node and link tables can include user-defined data. The separation of user-defined data and network connectivity information enables NDM to be a generic network modeling and analysis feature for different network applications.

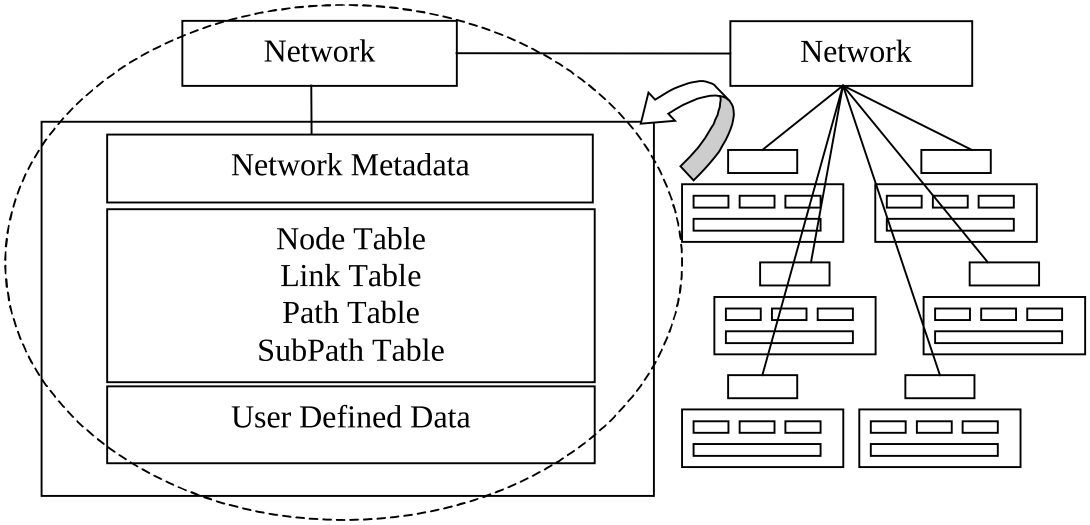

Figure 1 Network Data Model Schematic View

Load-On-Demand

The load-on-demand approach stores the network data model described above and tables with network partitions in the database. To efficiently handle large networks with limited memory resources, NDM lets users partition a large network into network partitions and store them in tables.

Partition Tables

NDM provides a PL/SQL utility to help divide large networks into network partitions. This approach assigns each node with a partition ID. Each network partition is stored as a binary BLOB to further speed up partition loading time from the database. Small networks do not require partitioning if they can fit into available memory.

Contraction Hierarchies

The contraction hierarchies (CH) approach uses precomputed shortest path information to speed up network analysis. By using the node order to represent the importance of nodes in the shortest paths, contraction hierarchies add “shortcut edges” that reduce the number of edges in the shortest path query.

The CH model is generated from the NDM network in the database and is stored in a file system. Because the CH model is a much more compact representation of the network, the entire CH model is loaded into memory during analysis.

An NDM CH model contains the following information:

- Network metadata information such as CH model name, source NDM network, cost information, geometry file information.
- Original network node and link information from the database data model
- Contraction Hierarchies information with node order and shortcut edges
- Node and link geometry information (Optional)
- Other pre-defined user-data information such as turn restrictions and multiple link costs (Optional)

Architecture

In the following sections, we describe the architectures for Load-On-Demand and Contraction Hierarchies.

Load-On-Demand Architecture

NDM Load-On-Demand applications usually adopt a 3-tiered architecture (Figure 2). LOD is designed to enable network analysis on large networks without loading the complete network into memory. In this architecture, we have the network data model in the database backend, with the NDM LOD analysis service in the application tier and a frontend user interface to communicate with the middle tier or the database. LOD Java API maintains a partition cache that efficiently manages network partitions in the partition cache. Users implement the analysis services/application with the LOD Java API as a Java servlet in a JEE container or an application server such as Tomcat or WebLogic. The load-on-demand approach will require a database connection with JDBC.

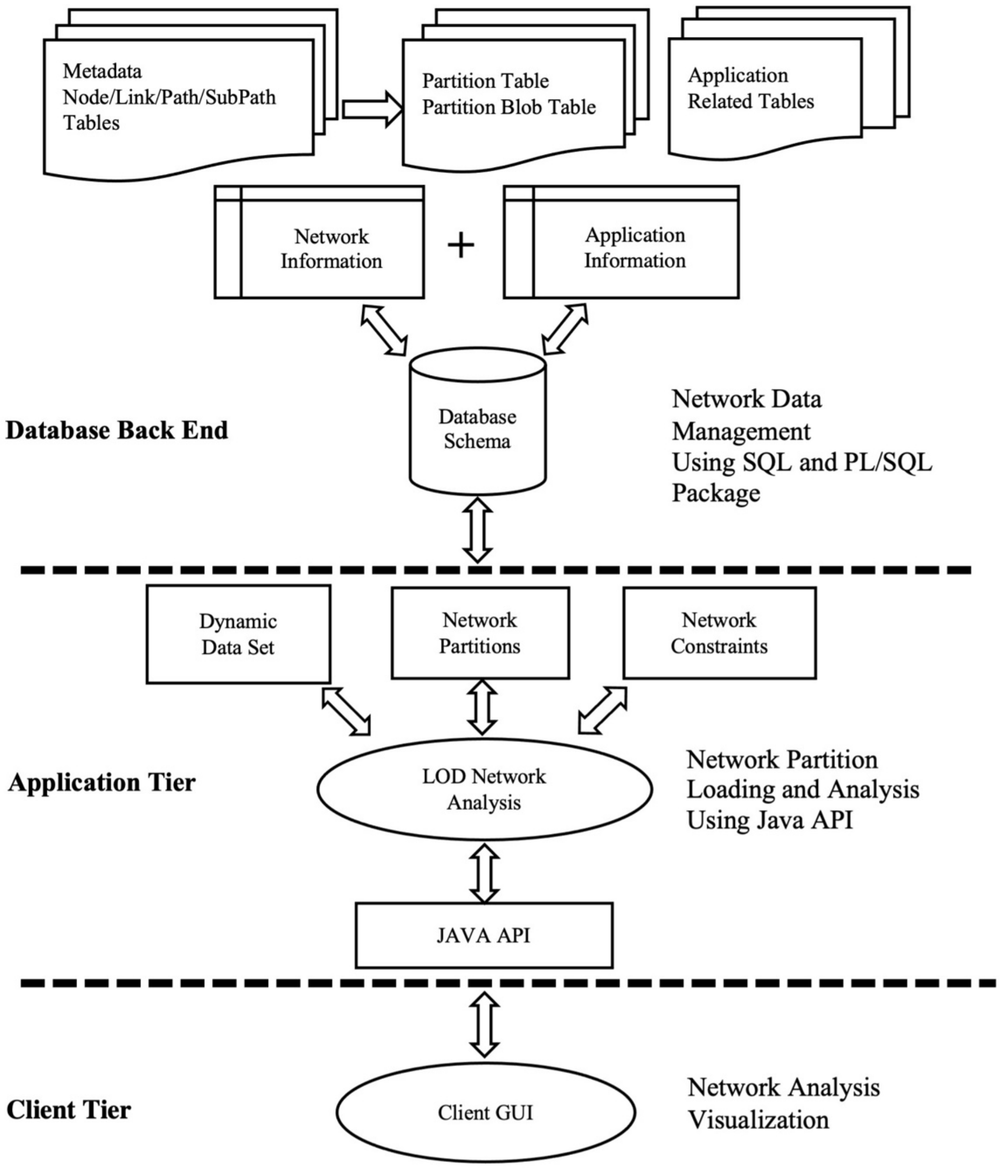

Figure 2 NDM Load-on-Demand Architecture

Contraction Hierarchies Architecture

Contraction Hierarchies applications use a similar 3-tiered architecture for network analysis (Figure 3). It uses the CH model that is precomputed and stored in an application tier file system; no connection to the database is required after the CH model has been generated into the filesystem. Contraction Hierarchies use a REST service deployed in the application tier to load the complete CH model into memory for analysis. The frontend web application sends REST requests and receives REST responses from the CH REST service over HTTP.

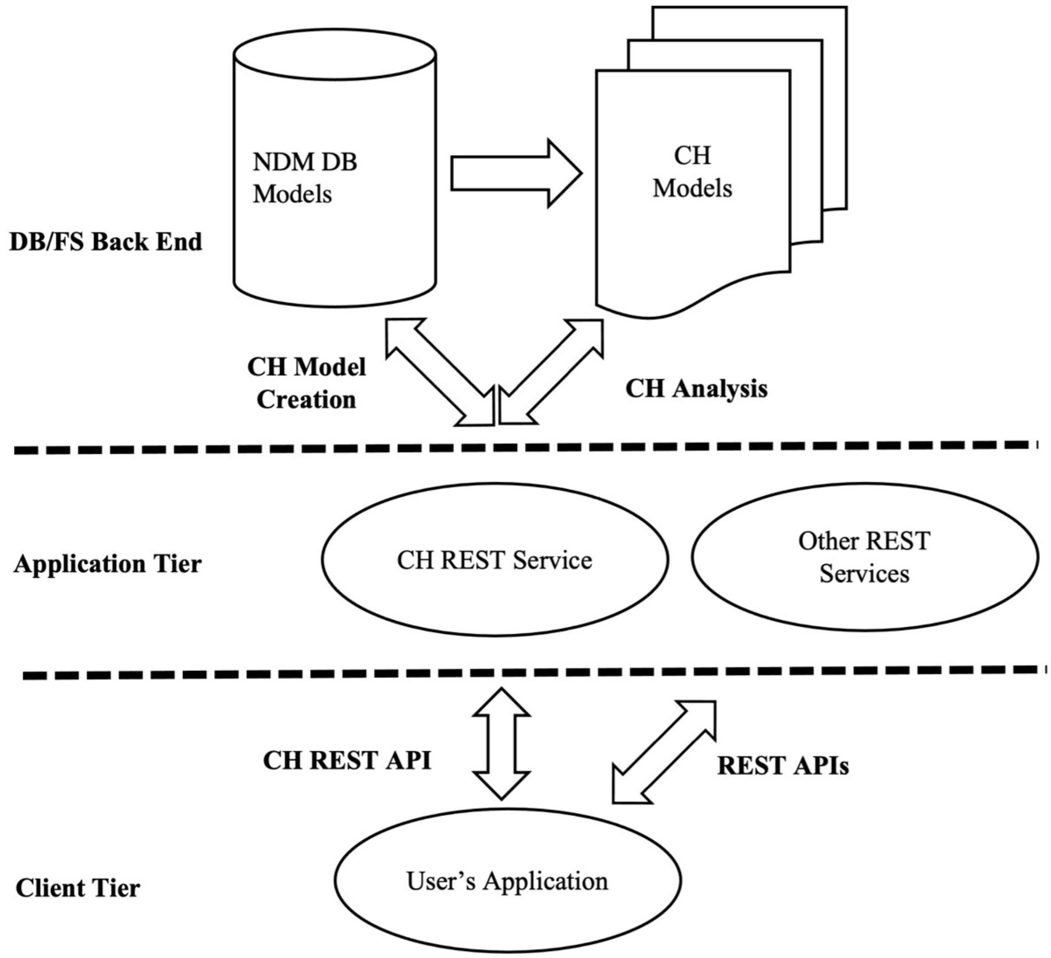

Figure 3 CH REST Architecture

Features of Network Data Model

Network Data Model provides many network modeling and network analysis capabilities in the Oracle Spatial database. These unique features make NDM a scalable and customizable network modeling and analysis framework. NDM is made up of the following components:

Network Constraint

Users model their business rules and logic as NDM network constraints and can plug them into NDM network analysis. With network constraints, a user can easily customize his/her network analysis. For instance, users can implement a network constraint that allows a route to go through a specific type of link (a carpool lane, for example) or generates a path that avoids a particular area (like a road that does not permit commercial vehicles).

User Data

Users often store business or application data (user-defined data) in the NDM data model in addition to network connectivity information. For example, a street network may contain speed limits, road names, toll information, and segment ID in addition to information about the road segments to which they are connected. This user data is associated with a node or a link table or included in a CH model, and the user data is available during analysis. The use of user data is relatively common in a network constraint or a cost calculator implementation.

Cost Calculator

NDM supports specific cost modeling via cost calculators. Users can provide their cost calculator implementation instead of the default cost in the analysis. For instance, the link cost can be the travel distance or the travel time of a road segment. Users can also include penalties like traffic congestion in a link cost calculator.

Dynamic Data Set

NDM lets users temporarily modify in-memory network data, such as adding or deleting nodes or links, without changing the network data model. A user can also temporarily disable some nodes or links and see how the change will affect the user's analysis in a typical what-if design or analysis scenario. An analysis's dynamic data set is only visible to that specific analysis and will not change the shared network in memory.

Feature Modeling and Analysis

A feature is an object of interest in a network application associated with a node or link. For example, in a transportation network, features include exits and intersections (mapped to nodes) and highways and streets (mapped to links).

Features are the main subjects dealt with by network applications. Network elements such as nodes and links are mathematic representations used for network graph modeling and analysis.

NDM provides feature modeling and analysis support, enabling users to model and analyze networks in the terminology with which they are most familiar.

Large Scale Drive Time/Distance Analysis

Geo-marketing integrates geographical intelligence into various aspects of marketing, including sales, distribution, and customer relationships.

It is often used in the process of planning and implementing marketing activities. Typical geo-marketing examples are site selection and targeted marketing based on location. It is critical to determine the best store locations and most effective product promotions for target customers based on travel distance/time from customers or competitors. This kind of analysis usually requires vast amounts of route computation and could take days for many customers.

The traditional approach to performing drive time and drive distance analysis is to generate spatial buffers around locations as a primary filter and then compute the optimal route (the shortest path query) for each customer to its nearest store (the nearest neighbor query).

NDM offers a “network buffer” representation that captures the coverage and cost information on top of the network representation. In contrast to the spatial buffer approach, users can precompute and persist network buffers in the database to perform time/distance analysis at scale. They can ask questions of coverage, cost, and shortest path on one or multiple network buffers with simple SQL statements, without any spatial operations or network analysis.

Temporal Modeling and Analysis

Most networks have a time dependence. Adding a temporal dimension to the NDM model enhances the kinds of analysis often required by geospatial and other applications. Travel times on roads vary depending on the time of day. Incorporating these temporal variations into network analysis makes the model more realistic and accurate.

For example, a user can ask questions such as "Which route would take me fastest to my work location if I start from home at 8 AM?" or "When is the best time to go to the grocery store so that I spend the least time driving?" By using historical traffic patterns and/or real-time traffic information, NDM can more accurately reflect the actual travel time given a current time, as an NDM link cost calculator.

Traffic pattern data sets offer information about the historical traffic speed on roads and highways during pre-defined time intervals. Network analysis using these traffic patterns allows computations to account for varying congestion levels on roads at different times of the day. This consideration essentially makes computations sensitive to the start time. For example, the shortest path from point A to point B could be different for different start times at point A.

NDM APIS

This section discusses the NDM PL/SQL API, the load-on-demand Java API, and the Contraction Hierarchies REST API, and how to use and when to use them.

PL/SQL API

NDM includes a PL/SQL package for managing and validating network data models in the database. It also provides a spatial partitioning utility to help users partition large networks.

Load-on-Demand Java API

The Load-on-Demand Java API enables applications to perform network analysis on the database's network data models. It uses an in-memory partition cache to manage network partitions required for analysis with limited memory. It also provides customization interfaces like network constraints and cost calculators to implement business logic and cost models.

There are four main steps when using the load-on-demand Java API:

- Create an NDM Network in the database
- Partition the NDM network
- Configure Load-On-Demand analysis settings
- Analyze the NDM network with LOD Java API

The LOD Java API supports the following analysis functions:

- Shortest path computes the optimal path between point A to point B
- Traveling Salesman Problem (TSP) returns the optimal path the visit all given points
- Reachability Analysis computes the regions given a distance or time that can reach
- Drive Time and Drive Distance Polygons compute the spatial representation (usually polygons) of reachability analysis
- Network Buffers compute the accessibility analysis in the network representation. This allows users to precompute network buffers, store them in the database, and perform efficient coverage and cost queries with SQL statements.

You can download a Network Data Model load-on-demand demo which helps users visualize load-on-demand analysis results from Oracle.com at:

https://www.oracle.com/downloads/samplecode/spatial-samplecode-downloads.html

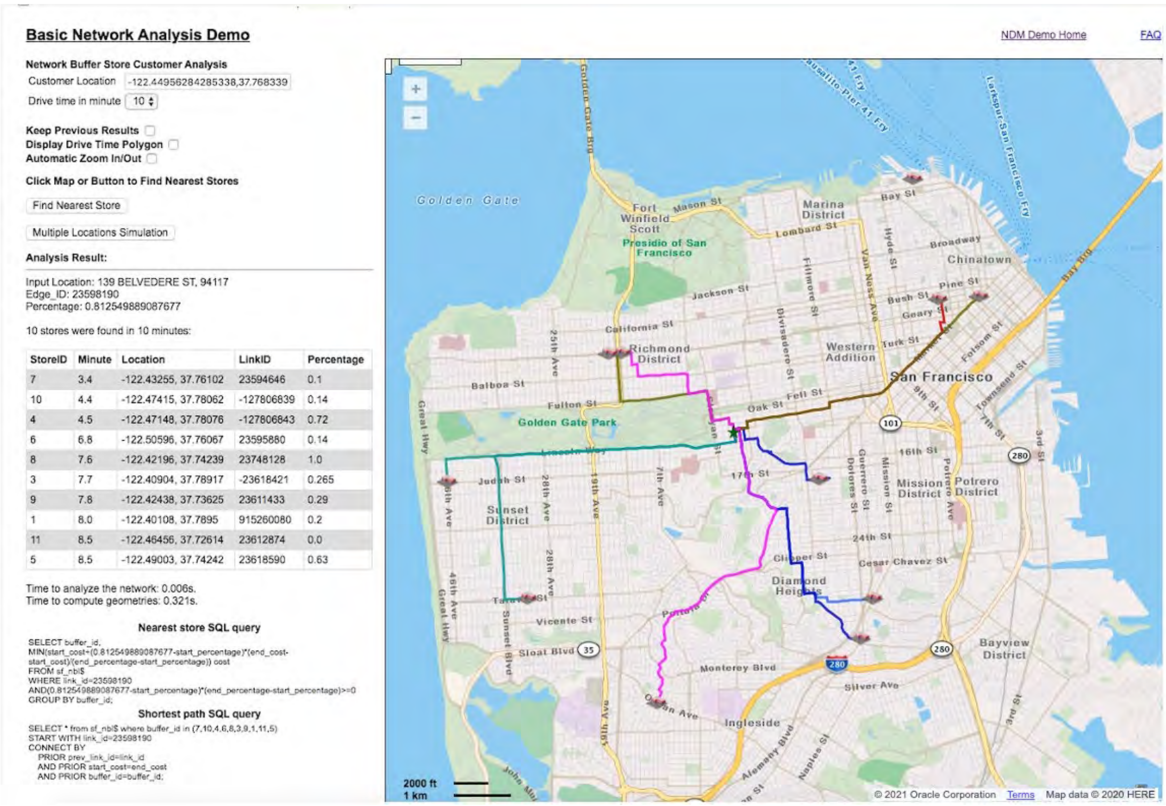

Figure 4   Network Data Model Load-On-Demand Demo

Contraction Hierarchies REST API

Contraction Hierarchies REST API contains a JEE service that handles REST requests and responses in the application tier. This JEE CH service loads the complete CH model into memory for analysis. The CH REST API simplifies network analysis in web service development and supports multiple programming languages with an XML or JSON format.

The main steps of using the NDM CH REST API are:

- Deploy NDM CH REST service in a JEE container or an application server
- Create and store the CH model from an NDM network in the database with the CH REST API
- Load the CH model into memory
- Analyze the CH model with the CH REST API

Users can use the CH REST API to:

- Create and store a CH model from an NDM network in the database
- Load a CH model into memory for analysis
- Query the metadata information of a CH model

The CH REST API supports the following analysis functions:

- Shortest path computes the optimal path between point A to point B
- Traveling Salesman Problem (TSP)) returns the optimal path the visit all given points

- Cost Matrix calculates the optimal costs of multiple originals to multiple destinations
- Alternative Routes return a specific number of routes from the optimal route between point A and point B, based on a given path dissimilarity index.

Figure 5 shows the web console of an NDM CH REST demo. This demo lets you create, configure, and analyze a CH model with the CH REST API. It includes a shortest path request/response example in JSON and displays the result with Mapbox GL JS (Figure 6).

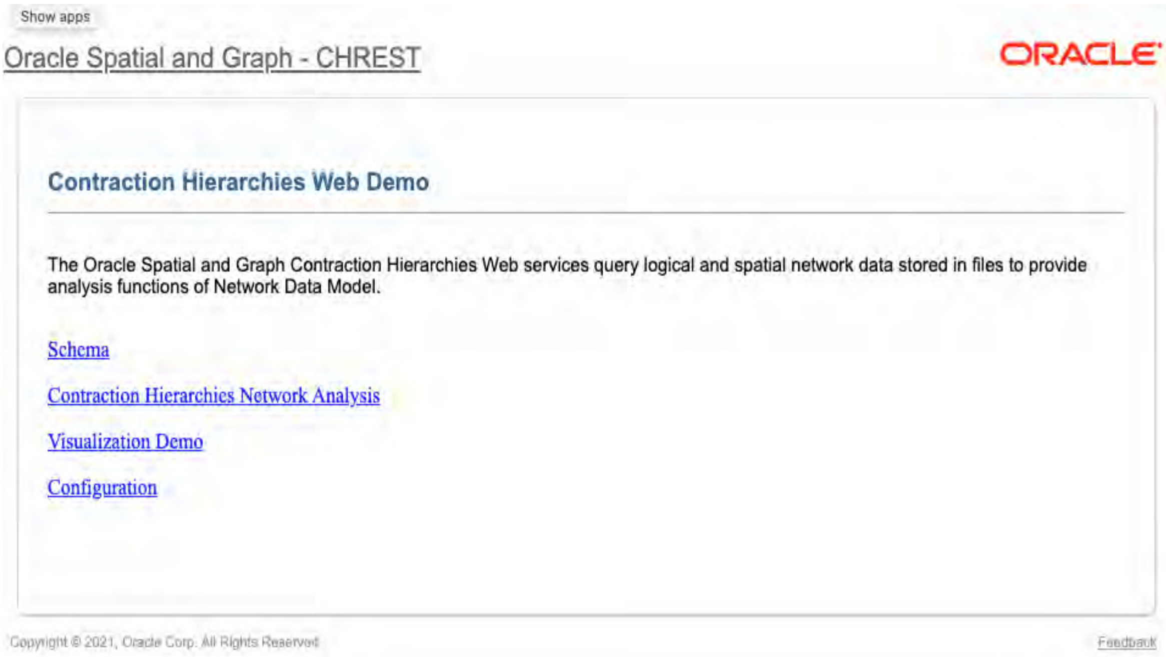

Figure 5 CH REST Demo

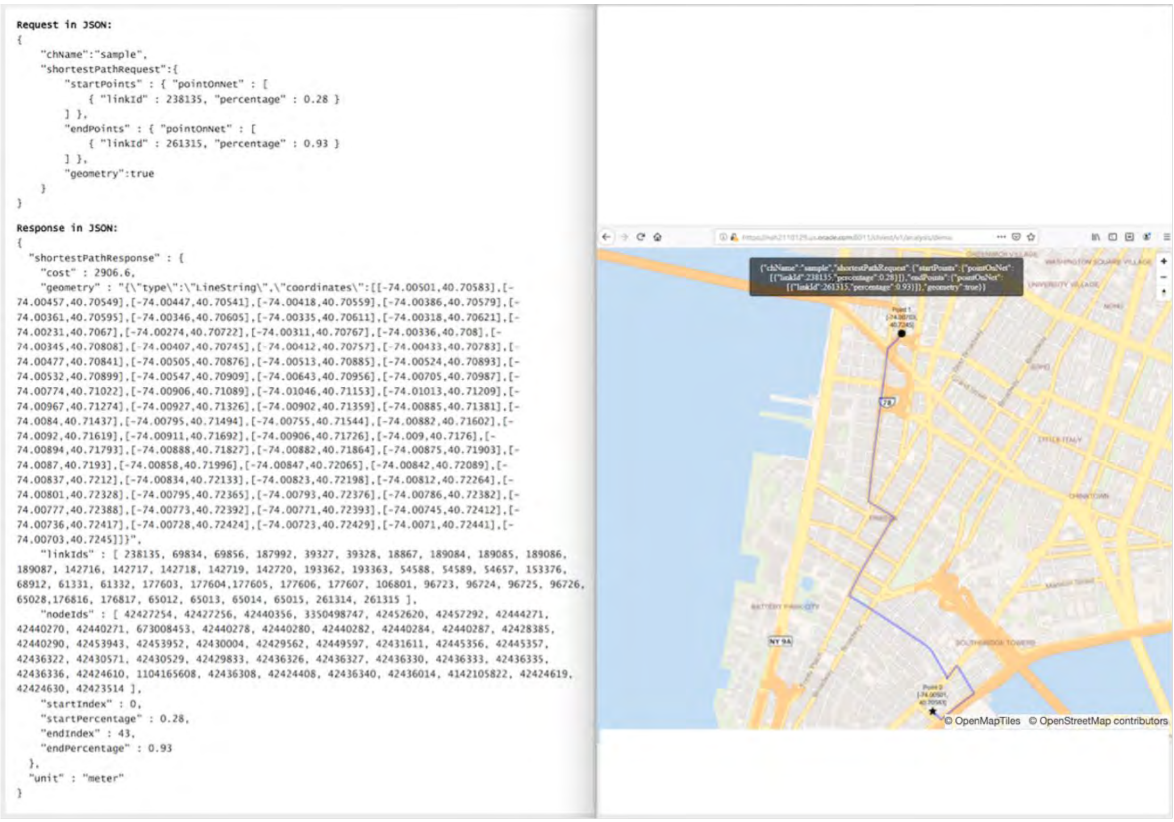

Figure 6 CH REST shortest path request, response, and path result

Comparisons between LOD Java API and CH REST API

The major differences between LOD JAVA API and CH REST API are customization and performance. For a network with dynamic properties or business logic, we recommend LOD Java API. LOD Java API provides network constraints, user data, and cost calculators to help users customize network analysis. CH REST API is a better choice if the application has a static network and requires high-performance analysis.

LOD Java API lets users implement their customization, while the CH REST currently supports limited pre-defined customization, such as turn restrictions and multiple costs.

LOD Java API requires less memory as it only loads network partitions into memory when it needs them. CH REST API requires that the whole CH model is loaded into memory and, as a result, CH REST API requires more memory resources.

As for data models, the LOAD Java API works on networks in the database and requires an active database JDBC connection. The CH REST API works with networks stored in file-based CH models.

The Contraction Hierarchies REST API has excellent performance and can achieve one or two orders of magnitude over the LOD Java API (with A* and Dijkstra algorithms). Users can generally expect more performance improvement on long routes. The performance speedup is due to the precomputed contraction hierarchies information on static topology and cost for shortest path query.

Table 1 (below) summarizes the usages and advantages of the Load-On-Demand Java API and Contraction Hierarchies REST API:

	Network Type	**Language**	**Customization**	**Approach**	**Data Model**	**Performance**
LOD Java API	Static/Dynamic	Java	High (user-defined)	Load on demand With Cache	Database	X
CH REST API	Static	REST (JSON/XML)	Low (pre-defined)	Pre-computed In Memory	Files	10X

Table 1 Comparison between NDM LOD API and CH REST API

Conclusion

Network Data Model is a spatial feature of Oracle Database. It is specifically designed to address the network modeling and analysis requirements of geospatial networks. The NDM network schema is open and accessible via SQL. It includes a PL/SQL API for database data management, a load-on-demand Java API for customizable network analysis, and a REST API for high-performance network analysis.

NDM offers a scalable and customizable solution for static and dynamic network applications with the load-on-demand Java API and a REST API based on the contraction hierarchies approach for static applications that demand fast analysis response times. The load-on-demand Java API allows users to consider business logic and data in their analytics and address dynamic changes in network topology and costs. The CH REST API offers performance that is up to two orders of magnitude better for traditional algorithms like Dijkstra or A* on static networks. This CH REST API also simplifies web service and application development. Depending on the network type and requirements, NDM users can choose the appropriate API for their network applications.

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

blogs.oracle.com facebook.com/oracle twitter.com/oracle