Network Data Model – A Spatial Feature of Oracle Database

February 2021

Public

Purpose statement

This document provides an overview of Network Data Model feature included with Oracle Database. It is intended solely to help you assess the business benefits of using this feature and to plan your I.T. projects.

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 remains 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.

Table of Contents

Introduction

Network Data Model graph (NDM) is a spatial feature of Oracle Database that lets you store and analyze physical and logical networks used in industries such as transportation, utilities, and communications. NDM persistently manages the network connectivity in the database, while a Java API and a REST API provide fast in-memory graph analytics, including shortest path, nearest neighbors, within cost, and reachability. In 21c, NDM provides an open data model that simplifies network modeling, analysis, and management so that users can focus on application logic. NDM separates application-specific information from connectivity relationships so that the model can apply to many kinds of network applications. NDM further provides a constraint mechanism to let users guide analysis based on application rules and attributes.

NDM is designed to natively make the most of Oracle Database features for scalability, security, partitioning, and parallelism. It can exploit the processing power and bandwidth of Oracle Exadata Machine to obtain extreme performance capabilities that are orders of magnitude over what was previously possible.

NDM performs network analysis on large networks through its load on demand (LOD) approach. Instead of loading the whole network into memory, NDM divides the network into manageable sub-networks (network partitions) and only loads the partitions the analysis has reached while exploring the network. It automatically manages the loading and unloading of network partitions, thus removing memory as a limiting factor. LOD API is highly customizable with application logic. In 21c, NDM provides a REST API based on the contraction hierarchies (CH) approach. This CH REST API uses a pre-computed and in-memory approach with extreme analysis performance for analytics. Users can choose the right API according to their customization or performance requirements.

Network Data Model also provides the infrastructure for the Oracle spatial routing engine. The routing engine is a road network routing service that produces the optimal route with driving directions between two or more locations. With NDM LOD API, the routing engine can handle turn and truck restrictions or other route-specific requirements.

This paper provides an overview of the Network Data Model. It explains the data model, architecture, and features of NDM. It also discusses the LOD Java and CH REST APIs and explains how and when to use them in a network application. Using NDM to extend the Oracle routing engine capabilities is also discussed.

Data Models

Network Data Model stores network models in the database. The LOD Java API directly works on the data models in the database. On the other hand, the CH REST API works on CH models generated from database data models.

We discuss these data models as follows:

Network Data Model in Database

The Network Data Model contains the following database schemas: network metadata, network tables, user-defined data, and partition tables.

Network Metadata

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

Network Tables

An Oracle network contains at least a node table and a link table, and a path table can be added if needed. Figure 1 shows the schema for the network data model, which includes these tables. The schema represents the information necessary for network management and analysis. User-defined data (application attributes) can be added to these tables. Node views and link views are also supported. The network data model can also 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.

User-Defined Data

It is common to have application data other than the connectivity information associated with a network. NDM lets users define their application data as part of a network data model. 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.

Partition Tables

Partition tables store the partition result in the partition metadata and partition BLOB tables. NDM provides a PL/SQL spatial partition utility to help partition large networks into network partitions. This node-based partitioned approach assigns each node with a partition ID. Small networks do not require partitioning if they can fit into available memory.

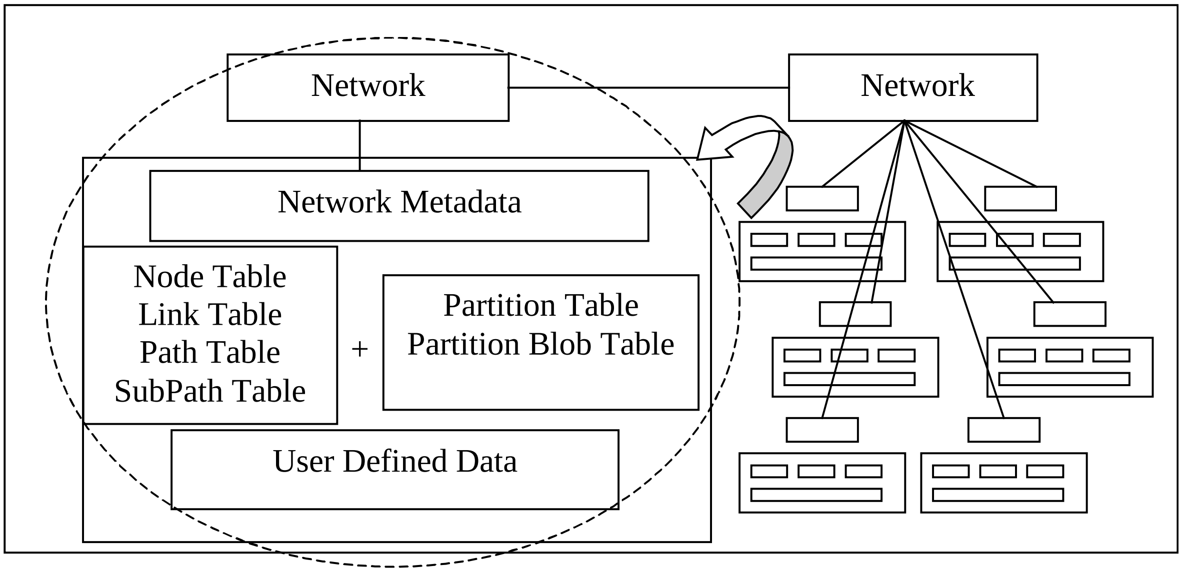

Figure 1 Network Data Model Schematic View

Contraction Hierarchies Model

Network Data Model provides the Contraction Hierarchies (CH) REST API in 21c. The CH REST API lets users create CH models and conduct CH analysis on CH models.

A CH model contains the node order and shortcut edge information from the original network. The CH REST API can create a CH model from an NDM network and store it as a CH file set under a specific directory. A CH model can also include information such as geometry and user-defined data.

Please note that the NDM CH REST API works directly on the CH models.

Architecture

We describe the architectures for Load-On-Demand and Contraction Hierarchies as follows:

Load-On-Demand Architecture

NDM Load-On-Demand applications usually adopt a 3-tiered architecture (Figure 2). 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. As a result, it is possible to conduct network analysis for large networks without loading the complete network into memory. Users implement the analysis services/application with the LOD Java API as a Java servlet in a J2EE container or an application server such as Tomcat or WebLogic. The load-on-demand approach will require a database connection with JDBC.

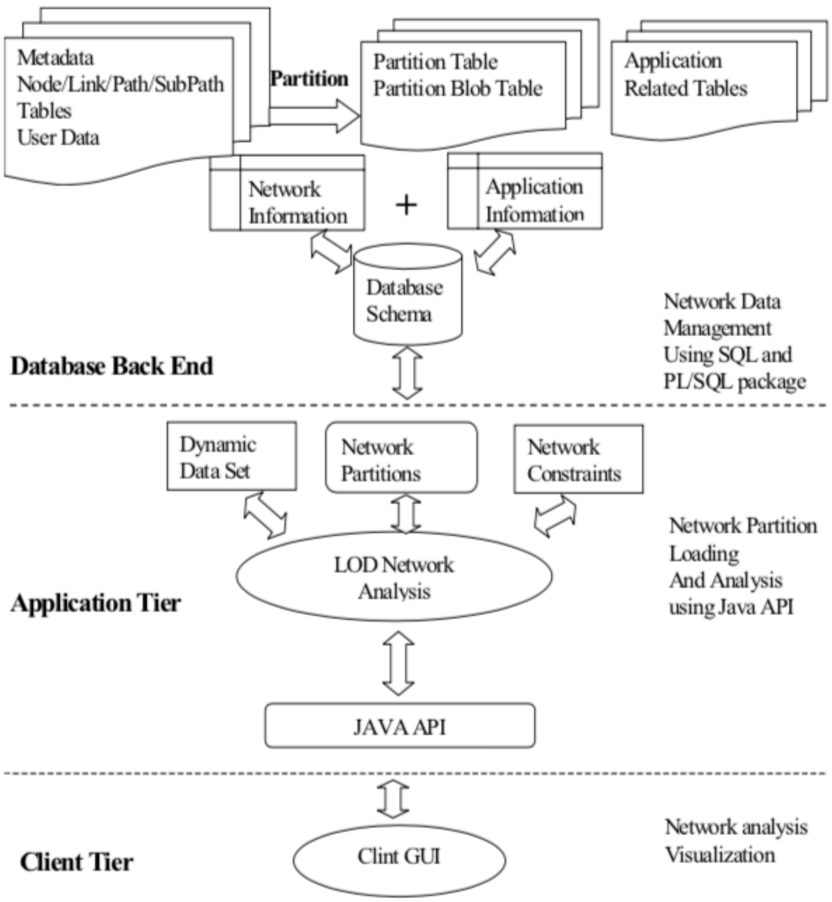

Figure 2 NDM Load-on-Demand Architecture

Contraction Hierarchies Architecture

Contraction Hierarchies REST API uses a different approach for network analysis (Figure 3). It pre-computes the contraction hierarchies node order and shortcut edges of a network model in the database. NDM stores the contraction hierarchies information with the original network in a CH model as a file set. The CH REST service then loads the whole CH model into memory for CH analysis. The frontend web application sends REST requests and receives REST responses to/from the CH REST service over HTTP.

Note that, unlike the LOD Java API, the CH REST API works directly on the CH models. Therefore, it does not need to connect to the database once the CH models are available on the file system.

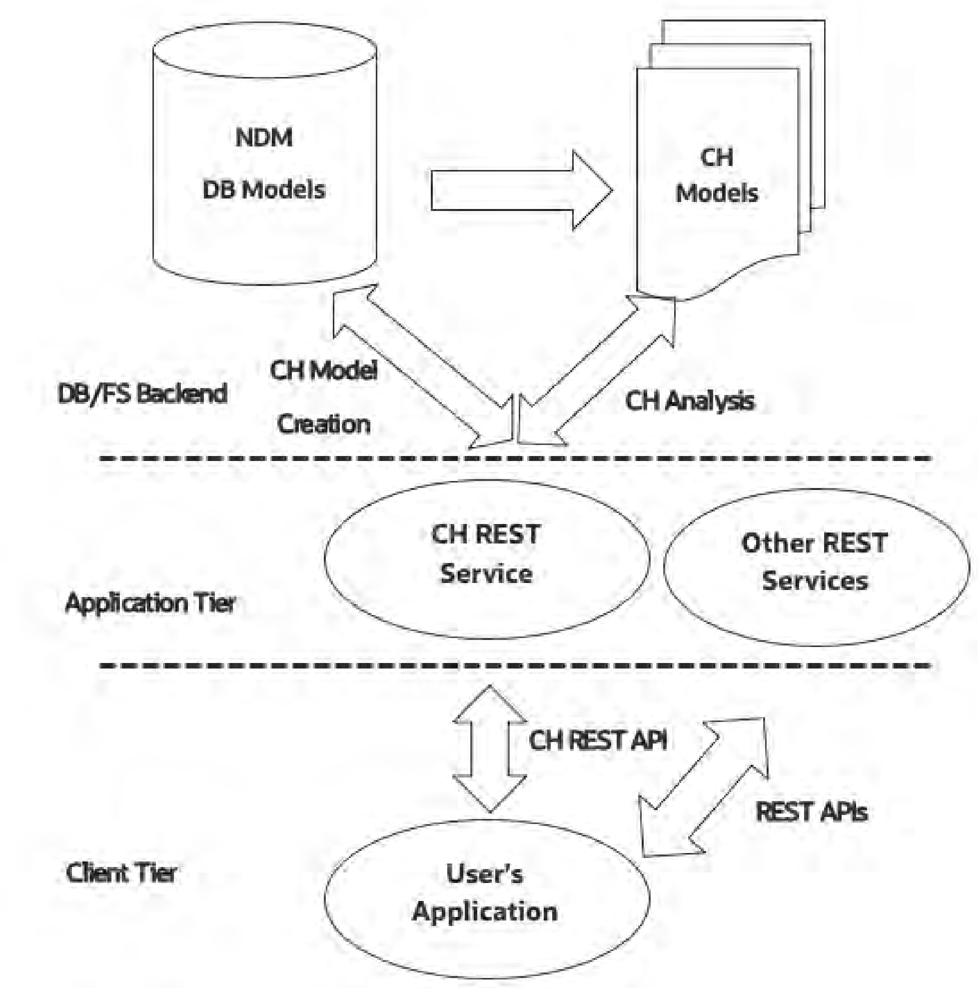

Figure 3 CH REST Architecture

Features of Network Data Model

Network Data Model provides many network modeling and analysis capabilities in the Oracle Database. These unique features make NDM a scalable and customizable network modeling and analysis framework. We explain the NDM features as follows:

Network Constraint

NDM lets users model their business rules and logic as NDM network constraints and plug them into NDM network analysis. With network constraints, you can easily customize your network analysis. For instance, you can implement a network constraint that allows a route to go through a specific type of link or generates a path that avoids a particular area.

User Data

NDM lets users store business or application data (user-defined data) in the NDM data model in addition to network connectivity information. Each 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 the specific cost modeling via cost calculators. Users can provide their cost calculators 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. You can also temporarily disable some nodes or links and see how the change will affect your 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 is the integration of 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. Based on travel distance/time from customers or competitors, business decisions are made to determine the best store locations and most effective product promotions for target customers. This kind of analysis usually requires huge amounts of route computation and could take days for large numbers of customers.

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

NDM introduces the network buffer representation that captures the coverage and cost information on top of network representation. In contrast to the spatial buffer approach, the cost information of network buffers is accurate and efficient. A user can pre-compute and persist network buffers in the database. He or she 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 exhibit a time-dependence. The ability to include a temporal dimension to the NDM model enhances the kinds of analysis often required by geospatial information systems (GIS) 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 useful and accurate. For example, a user would be able to 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?" We explain with examples of routing with traffic patterns and real-time traffic information. Both instances treat traffic information (historical or real-time), reflecting 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.

It is possible to compute optimal routes with real-time traffic information if such information is available. NDM will use the real-time traffic information as actual link costs to compute an optimal route with a start time.

NDM APIS

This section discusses the NDM PL/SQL API, the load-on-demand Java API, and the Contraction Hierarchies REST API. We also explain how to use and when to utilize these APIs and their functionality.

PL/SQL API

NDM provides 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

There are four major steps of using the load-on-demand Java API:

1. Create an NDM Network in the database
2. Partition the NDM network if necessary
3. Configure Load-On-Demand analysis settings
4. Analyze the NDM network with LOD Java API

The LOD Java API supports the following analysis functions:

- Shortest Path
- Traveling Salesman Problems (TSP)
- Reachability Analysis
- Drive Time and Drive Distance Polygons
- Network Buffers

We show a LOD demo (Figure 4), which helps users visualize load-on-demand analysis results. This LOD demo is available for download on the Oracle Spatial and Graph website (https://www.oracle.com/downloads/samplecode/spatial-samplecode-downloads.html).

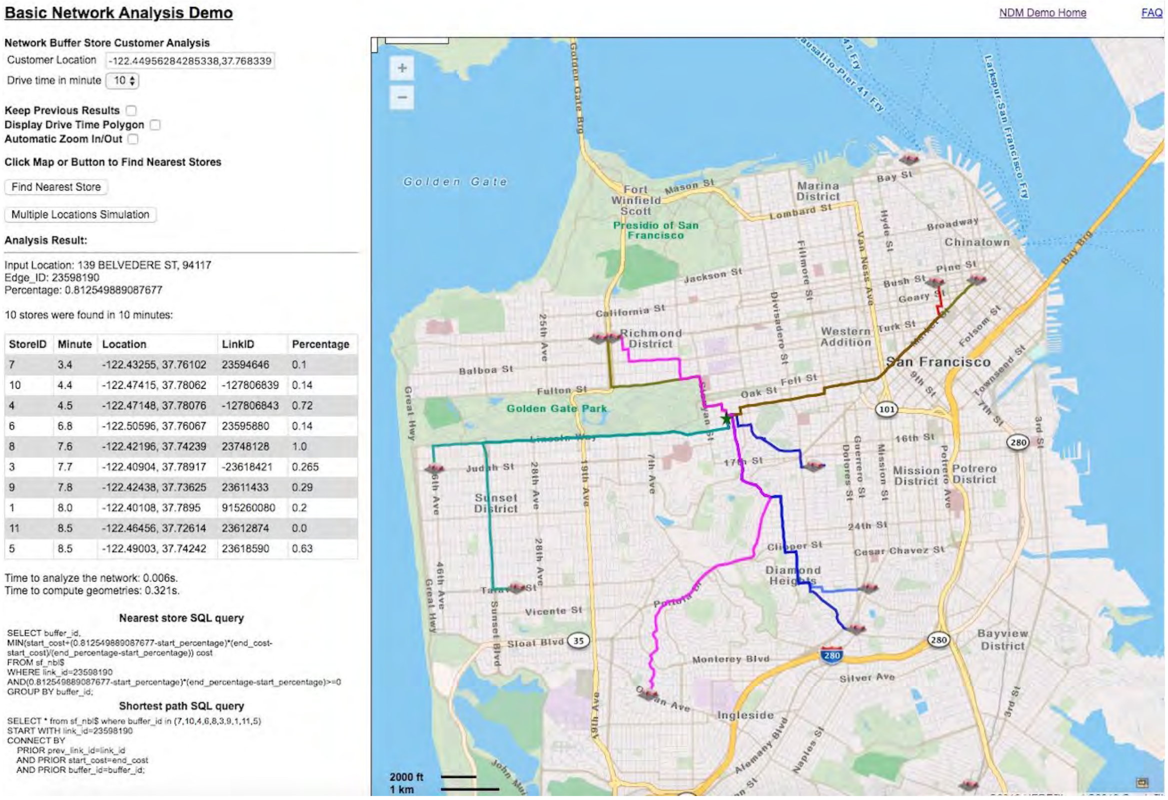

Figure 4 LOD Demo

Contraction Hierarchies REST API

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

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

The CH REST API supports the following analysis functions:

- Shortest Path
- Traveling Salesman Problems (TSP)
- Cost Matrix
- Alternative Routes

Figure 5 shows the web console of an NDM CH REST demo. This demo lets users create, configure, and analyze a CH model. We present a shortest path request/response in JSON and display the result (Figure 6).

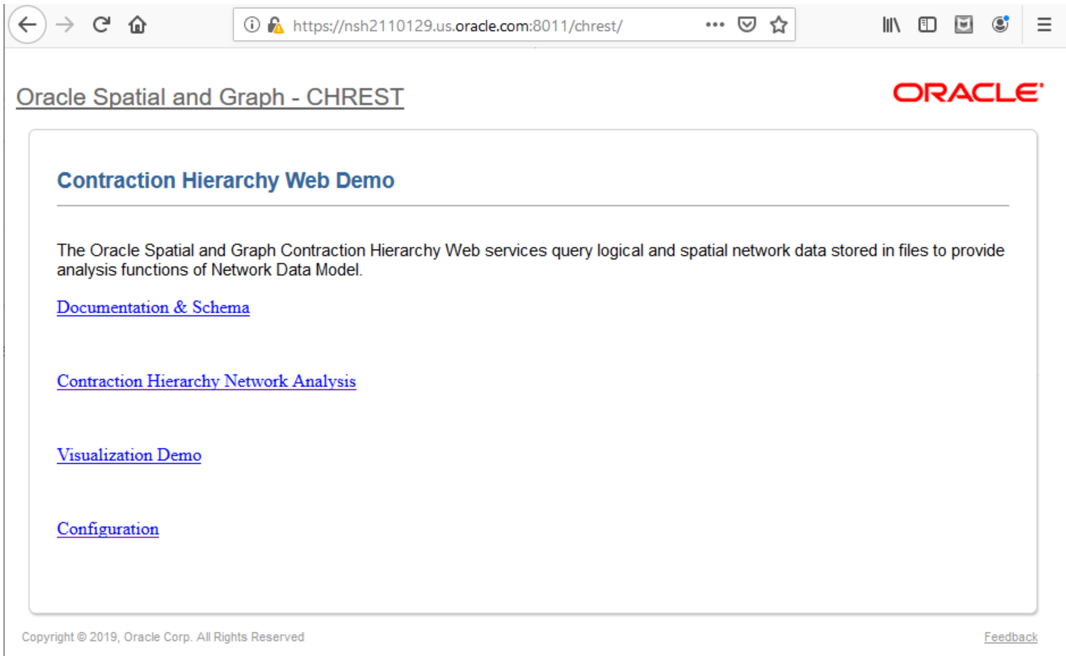

Figure 5 CH REST Demo

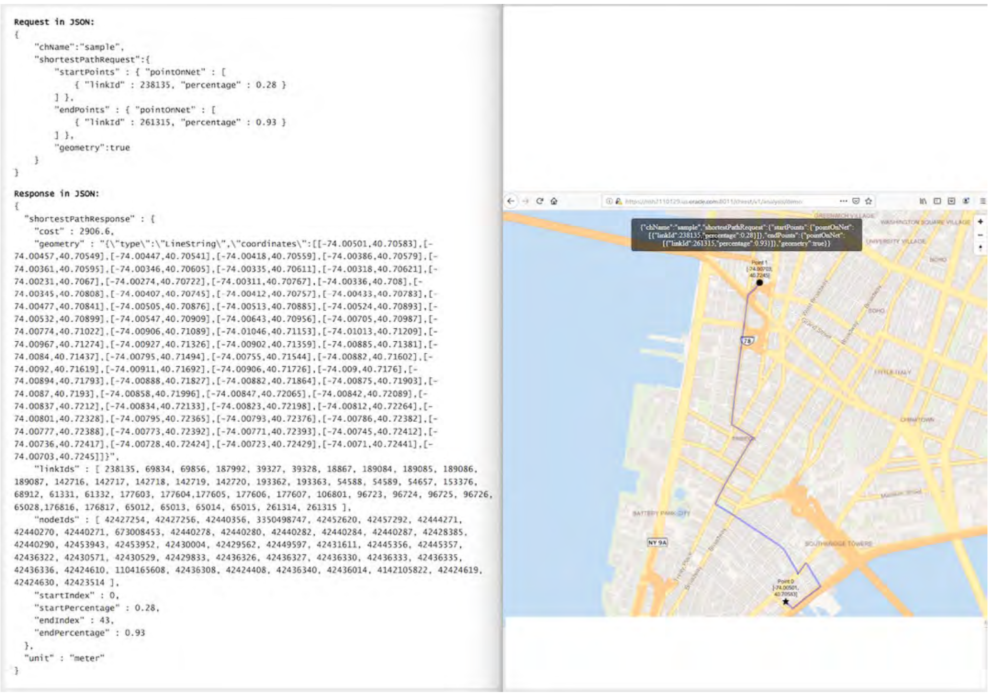

Figure 6 CH REST shortest path request, response, and path result

Comparisons between LOD Java API and CH REST API

We compare the Load-On-Demand Java API and Contraction Hierarchies REST API. The following table summarizes their usages and advantages (Table 1):

	Network Type	Language	Customization	Performance	Approach
LOD Java API	Static/Dynamic	Java	High (user-defined)	Good	Load on demand With Cache
CH REST API	Static	REST (JSON/XML)	Low (pre-defined)	Excellent	Pre-computed In Memory

Table 1 Comparison between NDM LOD API and CH REST API

The key differences between LOD JAVA API and CH REST API are customization and performance. For a network with dynamic properties or business rules/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 you have static networks and want high-performance analysis.

LOD Java API lets users implement their customization, while the CH REST currently only supports some pre-defined customization, such as turn restrictions. This is because it is challenging to consider dynamic changes in a pre-computed approach like contraction hierarchies.

LOD Java API requires less memory as it only loads network partitions when it needs them. CH REST API uses an in-memory analysis approach, which requires loading the whole CH model into memory. As a result, CH REST API requires more memory resources. To use LOD Java API, users need to use Java to implement their network applications or services. But for CH REST API, users can use the REST API over HTTP with other APIs, thus simplify the analysis and development.

Extending Routing Engine Capabilities with NDM

The Spatial routing engine in Oracle Database is a web routing service that provides the optimal route with driving directions and route geometry between two or more locations. This routing engine uses NDM LOD Java API. The routing engine extends its capabilities to handle truck routing and turn restrictions and include traffic patterns and time zone information in the analysis. These capabilities are implemented with NDM user data, network constraints, and cost calculators.

Truck Routing

The Oracle routing engine provides truck-specific routing based on roads, weight, height, time of day, and other conditions applied to commercial and logistics applications and logical turn restrictions. It can compute the drive times based on truck speed limits, which often differ from car speed limits. It can also provide truck services, such as weigh stations and truck stops along a route.

Prohibited Turns

Routing Engine can handle logical turn restrictions involving more than two-way points in the route geometry. These enhancements yield more accurate results for logistics and car and truck routing applications.

Traffic Patterns and Time Zones

Traffic patterns reflect the historical travel time at a specific time of day. With traffic patterns data from content providers, NDM can compute a route based on a given start day and start time. Also, NDM can add time zones information to road segments (network links) and include it in the route response to reflect the local time information.

Conclusion

Network Data Model, a spatial feature of Oracle Database, provides network modeling and analysis capabilities. The NDM network schema is open and accessible via SQL. It provides a PL/SQL API for database data management, a load-on-demand Java API for customizable network analysis, and a REST API in 21c for high-performance network analysis.

Before 21c, NDM focused on providing a scalable and customizable solution to many network applications with the load-on-demand Java API. Starting from 21c, NDM provides a high-performance REST API based on the contraction hierarchies approach. The CH REST API offers extreme performance for network analysis on static networks. The REST API also simplifies web service and application development. Depending on their network type and requirements, NDM users can choose the right 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