

# ネットワーク・データ・モデル – Oracle Databaseの空間機能

2021年2月

Copyright © 2021, Oracle and/or its affiliates

公開

## 本書の目的

本書では、Oracle Databaseに含まれるネットワーク・データ・モデル機能の概要が説明されています。本書は、本機能を使用することで得られるビジネス上の利点の評価と、ITプロジェクトの計画立案を支援することのみを目的としています。

## 免責事項

本文書には、ソフトウェアや印刷物など、いかなる形式のものも含め、オラクルの独占的な所有物である占有情報が含まれます。この機密文書へのアクセスと使用は、締結および遵守に同意したOracle Software License and Service Agreementの諸条件に従うものとします。本文書と本文書に含まれる情報は、オラクルの事前の書面による同意なしに、公開、複製、再作成、またはオラクルの外部に配布することはできません。本文書は、ライセンス契約の一部ではありません。また、オラクル、オラクルの子会社または関連会社との契約に組み込むことはできません。

本書は情報提供のみを目的としており、記載した製品機能の実装およびアップグレードの計画を支援することのみを意図しています。マテリアルやコード、機能の提供をコミットメント（確約）するものではなく、購買を決定する際の判断材料になさらないでください。本書に記載されている機能の開発、リリース、および時期については、弊社の裁量により決定されます。製品アーキテクチャの性質上、コードが大幅に不安定化するリスクなしに、本書に記載されているすべての機能を安全に含めることができない場合があります。

## 目次

---

|                                   |    |
|-----------------------------------|----|
| 本書の目的 .....                       | 2  |
| 免責事項 .....                        | 2  |
| はじめに .....                        | 4  |
| データ・モデル .....                     | 4  |
| データベースに格納されるネットワーク・データ・モデル .....  | 4  |
| 縮約階層モデル .....                     | 5  |
| アーキテクチャ .....                     | 5  |
| ロードオンデマンドのアーキテクチャ .....           | 5  |
| 縮約階層のアーキテクチャ .....                | 6  |
| ネットワーク・データ・モデルの機能 .....           | 7  |
| ネットワーク制約 .....                    | 7  |
| ユーザー・データ .....                    | 7  |
| コスト計算機能 .....                     | 7  |
| 動的なデータセット .....                   | 8  |
| フィーチャのモデリングと分析 .....              | 8  |
| 大規模な運転時間/距離の分析 .....              | 8  |
| 時間のモデリングと分析 .....                 | 8  |
| NDM API .....                     | 9  |
| PL/SQL API .....                  | 9  |
| ロードオンデマンドJava API .....           | 9  |
| 縮約階層REST API .....                | 10 |
| LOD Java APIとCH REST APIの比較 ..... | 11 |
| NDMを使用したルーティング・エンジン機能の拡張 .....    | 12 |
| トラックのルーティング .....                 | 12 |
| 右左折の禁止 .....                      | 12 |
| 交通パターンとタイムゾーン .....               | 12 |
| まとめ .....                         | 13 |

## はじめに

ネットワーク・データ・モデル・グラフ（NDM）は、Oracle Databaseの空間機能であり、この機能を使用すると、輸送、公益事業、通信などの業界で使用される物理ネットワークと論理ネットワークを保存し、分析できます。NDMにより、ネットワークの接続性がデータベースで永続的に管理され、Java APIとREST APIにより、最短パス、最近傍、コスト内、到達可能性など、高速なインメモリ・グラフ分析が提供されます。21cでは、NDMにネットワークのモデリング、分析、および管理を簡素化するオープンなデータ・モデルが用意されているため、ユーザーはアプリケーション・ロジックに集中できます。NDMでは、アプリケーション固有の情報と接続関係が分離されているため、モデルをさまざまな種類のネットワーク・アプリケーションに適用できます。さらに、制約の仕組みが提供されることで、ユーザーはアプリケーションのルールと属性に基づき分析を導くことができます。

NDMは、Oracle Databaseのスケーラビリティ、セキュリティ、パーティショニング、パラレル化などの機能を、特別な設定なしに最大限に活用できるように設計されています。Oracle Exadata Machineの処理能力と帯域幅を利用できるため、これまでとは桁違いの極めて高い性能が実現します。

NDMは、ロードオンデマンド（LOD）のアプローチを使用して、大規模なネットワークのネットワーク分析を実行します。ネットワーク全体をメモリにロードするのではなく、ネットワークを処理しやすいサブネットワーク（ネットワーク・パーティション）に分割し、その後、ネットワークの調査中に分析が行われたネットワーク・パーティションのみをロードします。ネットワーク・パーティションのロードとアンロードは自動管理されるため、メモリは制限要因ではなくなります。LOD APIは、アプリケーション・ロジックで大幅にカスタマイズできます。21cでは、NDMに縮約階層（CH）のアプローチに基づくREST APIが用意されています。このCH REST APIでは、分析で著しいパフォーマンスを発揮する事前計算されたインメモリのアプローチが使用されます。ユーザーは、カスタマイズやパフォーマンス要件に従って、適切なAPIを選択できます。

ネットワーク・データ・モデルでは、オラクルの空間ルーティング・エンジン用のインフラストラクチャも提供されます。ルーティング・エンジンは、2つ以上のロケーション間の最適なルートを運転案内とともに提供する道路ネットワーク・ルーティング・サービスです。NDM LOD APIを使用して、右左折規制、トラック規制、ルート固有のその他の要件に対処できます。

本書では、ネットワーク・データ・モデルの概要についてお話しします。NDMのデータ・モデル、アーキテクチャ、機能を説明するとともに、LOD Java APIおよびCH REST APIの詳細と、これらのAPIをネットワーク・アプリケーションでどのように、そしてどのような場合に使用するかについても説明します。さらに、NDMを使用したOracleのルーティング・エンジン機能の拡張についても詳しくお話しします。

## データ・モデル

ネットワーク・データ・モデルは、データベースにネットワーク・モデルを格納します。LOD Java APIは、データベース内のデータ・モデルを直接操作します。一方、CH REST APIは、データベースのデータ・モデルから生成されたCHモデルを操作します。

これらのデータ・モデルについて、以下で説明します。

### データベースに格納されるネットワーク・データ・モデル

ネットワーク・データ・モデルには、ネットワーク・メタデータ、ネットワーク表、ユーザー定義データ、パーティション表といったデータベース・スキーマが含まれます。

## ネットワーク・メタデータ

ネットワーク・メタデータには、ネットワークに関する一般的な情報が含まれます。空間ネットワークのジオメトリ情報も含まれます。

## ネットワーク表

Oracleのネットワークには、少なくともノード表とリンク表が含まれ、必要に応じてパス表も追加できます。図1では、これらの表を含むネットワーク・データ・モデルのスキーマを示しています。

このスキーマは、ネットワークの管理と分析に必要な情報を表します。ユーザー定義データ（アプリケーション属性）をこれらの表に追加できます。ノード・ビューとリンク・ビューもサポートされます。ネットワーク・データ・モデルでは、ジオメトリ情報も処理できます。つまり、ネットワーク・データ・モデルは、論理ネットワーク・アプリケーションと空間ネットワーク・アプリケーションの両方を表すことができます。ジオメトリ・データを論理ネットワークに追加すると、論理ネットワークの表示と空間分析が可能になります。

## ユーザー定義データ

ネットワークに関連する接続情報以外に、アプリケーション・データを保持するのは一般的です。NDMを使用すると、ユーザーは、ネットワーク・データ・モデルの一部としてアプリケーション・データを定義できます。NDMは、ユーザー定義データとネットワーク接続情報を分離することで、さまざまなネットワーク・アプリケーションのための汎用的なネットワーク・モデリングおよび分析機能になることができます。

## パーティション表

パーティション表により、パーティション結果がパーティション・メタデータとパーティションBLOB表に格納されます。NDMでは、大規模なネットワークを複数のネットワーク・パーティションにパーティション化するのに役立つPL/SQL空間パーティション・ユーティリティが提供されます。このノードベースのパーティション化アプローチにより、各ノードにパーティションIDが割り当てられます。小規模なネットワークは、利用可能なメモリに収まればパーティション化する必要はありません。

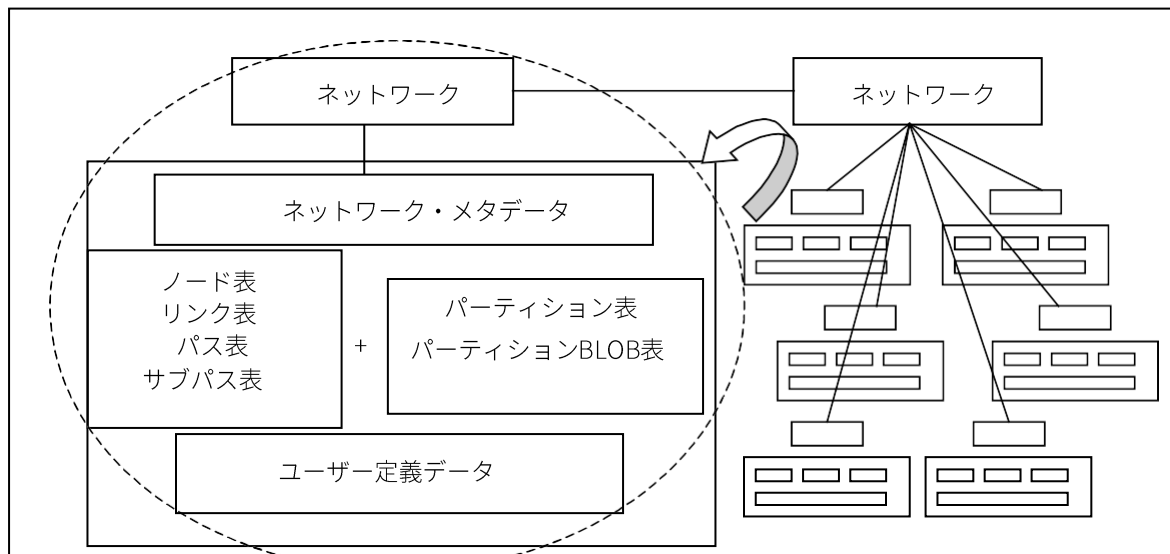


図1: ネットワーク・データ・モデルのスキーマ・ビュー

## 縮約階層モデル

21cでは、ネットワーク・データ・モデルに縮約階層（CH）REST APIが用意されています。CH REST APIを使用すると、ユーザーはCHモデルを作成し、CHモデルでCH分析を実行できます。

CHモデルには、元のネットワークからのノード順序とショートカット・エッジ情報が含まれます。CH REST APIは、NDMネットワークからCHモデルを作成し、そのモデルを特定のディレクトリにCHファイル・セットとして保管できます。CHモデルには、ジオメトリやユーザー定義データなどの情報を含めることもできます。

NDM CH REST APIは、CHモデルを直接操作することに留意してください。

## アーキテクチャ

ロードオンデマンドと縮約階層のアーキテクチャについて、以下で説明します。

### ロードオンデマンドのアーキテクチャ

NDMのロードオンデマンド・アプリケーションは、通常は3層アーキテクチャを採用しています（図2）。このアーキテクチャでは、データベース・バックエンドにネットワーク・データ・モデルがあり、アプリケーション層にNDM LOD分析サービスがあり、中間層またはデータベースと通信するフロントエンドのユーザー・インタフェースがあります。LOD Java APIは、パーティション・キャッシュのネットワーク・パーティションを効率的に管理するパーティション・キャッシュを保守します。そのため、ネットワーク全体をメモリにロードしなくても、大規模なネットワークのネットワーク分析が可能です。ユーザーは、JavaサブルーレットとしてのLOD Java APIを使用して、J2EEコンテナ、またはTomcatやWebLogicといったアプリケーション・サーバーに分析サービス/アプリケーションを実装します。ロードオンデマンドのアプローチには、JDBCを使用したデータベース接続が必要です。

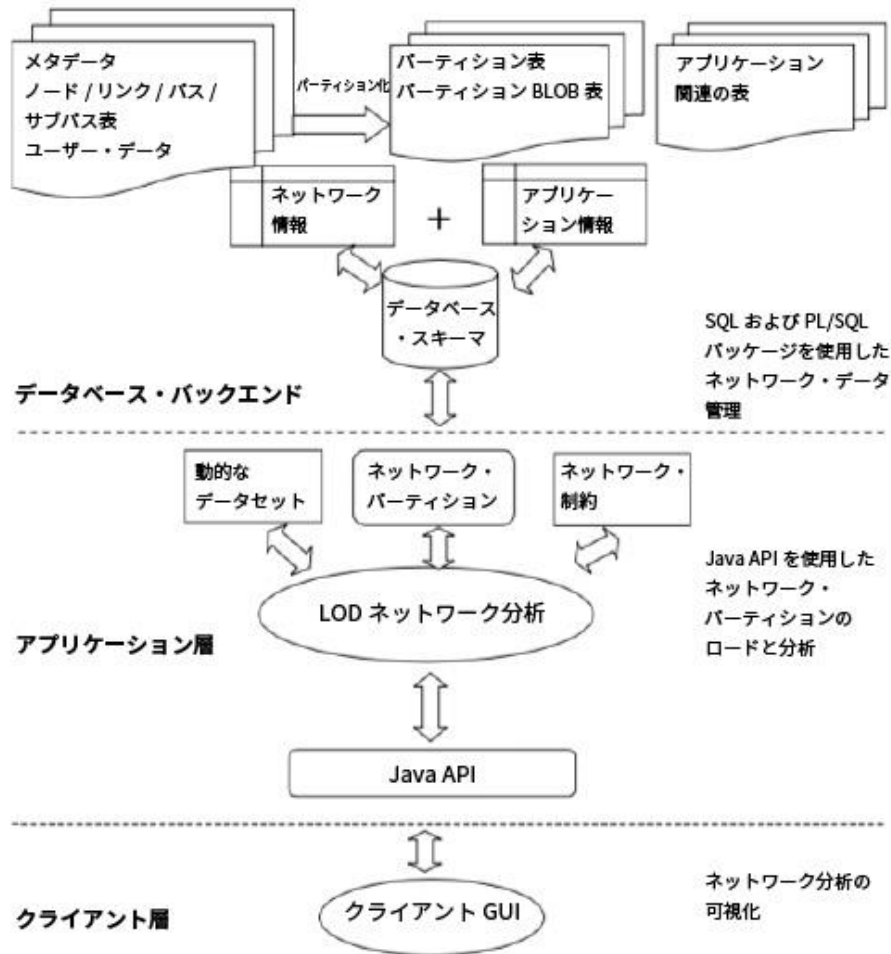


図2：NDMのロードオンデマンド・アーキテクチャ

## 縮約階層のアーキテクチャ

縮約階層REST APIでは、ネットワーク分析に異なるアプローチが使用されます（図3）。データベース内のネットワーク・モデルの縮約階層ノード順序とショートカット・エッジが事前計算されます。NDMは、元のネットワークとともに、縮約階層情報をCHモデルにファイル・セットとして格納します。次にCH RESTサービスが、CHモデル全体をメモリにロードしてCH分析を行います。フロントエンドのWebアプリケーションは、HTTP経由でCH RESTサービスにRESTリクエストを送信し、CH RESTサービスからRESTレスポンスを受信します。

CH REST APIは、LOD Java APIとは異なり、CHモデルを直接操作します。そのため、ファイル・システムでCHモデルが利用できるようになった後は、データベースに接続する必要はありません。

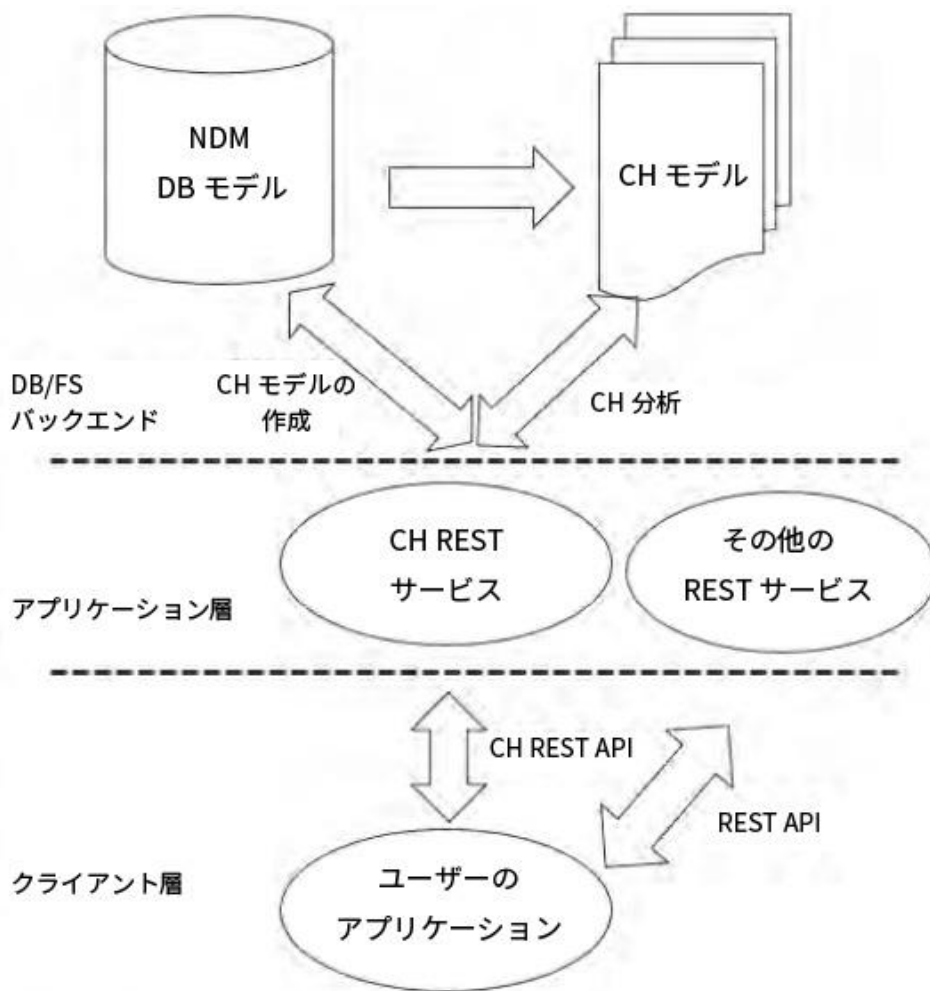


図3：CH RESTのアーキテクチャ

## ネットワーク・データ・モデルの機能

ネットワーク・データ・モデルにより、ネットワークのモデリングと分析を行う多数の機能がOracle Databaseで提供されます。NDMがスケーラブルでカスタマイズ可能なネットワーク・モデリングおよび分析フレームワークであるのは、これらの独自機能のおかげです。NDMの機能について、以下で説明します。

### ネットワーク制約

NDMを使用すると、ユーザーは、ビジネス・ルールとビジネス・ロジックをNDMネットワーク制約としてモデリングし、NDMネットワーク分析に組み込むことができます。ネットワーク制約を使用して、ネットワーク分析を容易にカスタマイズできます。たとえば、特定の種類のリンクを経由するルートを許可したり、特定の地域を避けるパスを生成したりするネットワーク制約を実装できます。

### ユーザー・データ

NDMを使用すると、ネットワーク接続情報に加えて、ビジネス・データやアプリケーション・データ（ユーザー定義データ）をNDMデータ・モデルに格納できます。各ユーザー・データは、ノード表またはリンク表と関連付けるか、CHモデルに含めて、分析中に利用できます。ユーザー・データは、ネットワーク制約やコスト計算機能の実装において比較的良好に使用されます。



## コスト計算機能

NDMでは、コスト計算機能を使用した特定のコストのモデリングがサポートされます。ユーザーは、デフォルトのコスト計算機能ではなく、自身のコスト計算機能を分析に実装できます。たとえば、リンク・コストを道路セグメントの移動距離または移動時間にすることができます。リンク・コスト計算機能に交通渋滞などのペナルティを含めることもできます。

## 動的なデータセット

NDMを使用すると、ユーザーは、ネットワーク・データ・モデルを変更せずに、ノードやリンクを追加または変更するなど、インメモリ・ネットワーク・データを一時的に変更できます。一部のノードやリンクを一時的に無効化し、一般的なwhat-if設計やwhat-if分析のシナリオで変更が分析にどのような影響を及ぼすかを確認することもできます。分析の動的なデータセットは特定の分析でのみ表示され、メモリ内の共有ネットワークは変更されません。

## フィーチャのモデリングと分析

フィーチャとは、ノードまたはリンクと関連付けられたネットワーク・アプリケーション内の対象オブジェクトです。たとえば、輸送ネットワークでは、フィーチャには（ノードにマッピングされる）出口や交差点、（リンクにマッピングされる）高速道路や一般道路などが含まれます。

フィーチャは、ネットワーク・アプリケーションによって処理される主要なサブジェクトです。ノードやリンクなどのネットワーク要素は、ネットワーク・グラフのモデリングと分析で使用される数学的な表現です。

NDMでは、フィーチャのモデリングと分析がサポートされるため、ユーザーは、もっとも使い慣れた用語でネットワークをモデリングし、分析できます。

## 大規模な運転時間/距離の分析

ジオマーケティングとは、地理情報を、販売、配布、顧客との関係など、マーケティングのさまざまな側面と統合することです。マーケティング活動を計画および実施するプロセスでよく使用されます。ジオマーケティングの典型的な例は、場所に基づくサイト選択とターゲット・マーケティングです。顧客や競合企業からの移動距離/時間に基づき、店舗の最適な立地や、ターゲット顧客に対するもっとも効果的な商品の販売促進活動が決定されます。この種の分析には、通常はおびただしい量のルート計算が必要であり、顧客の数が膨大であれば、計算には数日かかる可能性があります。

運転時間や運転距離の分析を実行するための従来型のアプローチは、1次フィルタとして店舗の周りに空間バッファを生成し、次に各顧客からもっとも近い店舗（最近傍の問合せ）への最適なルート（最短パスの問合せ）を計算するというものです。

NDMでは、ネットワーク表現に加えて、カバレッジ情報とコスト情報を取得するネットワーク・バッファ表現を導入しています。空間バッファのアプローチと比較して、ネットワーク・バッファのコスト情報は正確で有用です。ユーザーは、ネットワーク・バッファを事前計算し、データベースで永続化できます。そうすることで、空間操作やネットワーク分析をしなくても、簡単なSQL文を使用して、1つまたは複数のネットワーク・バッファのカバレッジ、コスト、最短パスを問い合わせることができます。

## 時間のモデリングと分析

ほとんどのネットワークは時間への依存性があります。NDMモデルに時間的な次元を含めることができれば、地理空間情報システム（GIS）や他のアプリケーションで通常必要な分析の種類を増強できます。道路での移動時間は時間帯に左右されます。このような時間的な変化をネットワーク分析に組み込むことで、モデルがより有用で正確なものになります。たとえば、ユーザーは、「家を午前8時に出発する場合、職場までの最速なルートはどれですか」、「スーパーに行く場合、移動にもっとも時間がかからない最適な時間はいつですか」といった質問ができます。交通パターンとリアルタイムの交通情報を使用したルーティングの例を用いて説明します。どちらのインスタンスも交通情報（過去の交通情報またはリアルタイムの交通情報）を扱うため、NDMリンク・コスト計算機能として、現在の時間を考慮した実際の移動時間が反映されます。



交通パターンのデータセットにより、事前定義された時間間隔内の一般道路と高速道路における過去の走行速度に関する情報が提供されます。これらの交通パターンを使用してネットワーク分析を行うことで、異なる時間帯における道路のさまざまな混雑レベルを計算によって明らかにすることができます。このネットワーク分析では、基本的に計算は出発時間の影響を受けやすくなります。たとえば、A地点からB地点への最短パスは、A地点を出発する時間によって異なります。

リアルタイムの交通情報を利用できれば、そのような情報を使用して最適なルートを計算することが可能です。NDMは、実際のリンク・コストとしてリアルタイムの交通情報を使用することで、任意の出発時間における最適なルートを計算します。

## NDM API

このセクションでは、NDMのPL/SQL API、ロードオンデマンドJava API、および縮約階層REST APIについて説明します。これらのAPIをどのように、どのような場合に使用するか、そしてそれぞれの機能についても説明します。

### PL/SQL API

NDMには、データベースのネットワーク・データ・モデルを管理および検証するためのPL/SQLパッケージが用意されています。大規模なネットワークのパーティション化に役立つ空間パーティション化ユーティリティも提供されます。

### ロードオンデマンドJava API

ロードオンデマンドJava APIを使用するための4つのおもな手順は以下のとおりです。

1. データベースにNDMネットワークを作成する
2. 必要に応じてNDMネットワークをパーティション化する
3. ロードオンデマンド分析を設定する
4. LOD Java APIを使用してNDMネットワークを分析する

LOD Java APIは、以下の分析機能をサポートしています。

- 最短パス
- 巡回セールスマン問題（TSP）
- 到達可能性分析
- 運転時間と運転距離のポリゴン
- ネットワーク・バッファ

ロードオンデマンド分析の結果を可視化できるLODデモを図4に示します。このLODデモは、Oracle Spatial and GraphのWebサイト（<https://www.oracle.com/downloads/samplecode/spatial-samplecode-downloads.html>）からダウンロードできます。

## Basic Network Analysis Demo

[NDM Demo Home](#)

[FAQ](#)

### Network Buffer Store Customer Analysis

Customer Location: -122.44956284285338, 37.768339

Drive time in minute: 10

Keep Previous Results ☐  
 Display Drive Time Polygon ☐  
 Automatic Zoom In/Out ☐

Click Map or Button to Find Nearest Stores

Find Nearest Store

Multiple Locations Simulation

### Analysis Result:

Input Location: 139 BELVEDERE ST, 94117

Edge\_ID: 23598190

Percentage: 0.812549889087677

10 stores were found in 10 minutes:

| StoreID | Minute | Location             | LinkID     | Percentage |
|---------|--------|----------------------|------------|------------|
| 7       | 3.4    | -122.43255, 37.76102 | 23594646   | 0.1        |
| 10      | 4.4    | -122.47415, 37.78062 | -127806839 | 0.14       |
| 4       | 4.5    | -122.47148, 37.78076 | -127806843 | 0.72       |
| 6       | 6.8    | -122.50596, 37.76067 | 23595880   | 0.14       |
| 8       | 7.6    | -122.42196, 37.74239 | 23748128   | 1.0        |
| 3       | 7.7    | -122.40904, 37.78917 | -23618421  | 0.265      |
| 9       | 7.8    | -122.42438, 37.73625 | 23611433   | 0.29       |
| 1       | 8.0    | -122.40108, 37.7895  | 915260080  | 0.2        |
| 11      | 8.5    | -122.46456, 37.72614 | 23612874   | 0.0        |
| 5       | 8.5    | -122.49003, 37.74242 | 23618590   | 0.83       |

Time to analyze the network: 0.006s.

Time to compute geometries: 0.321s.

### Nearest store SQL query

```
SELECT buffer_id,
MIN(start_cost*(0.812549889087677*start_percentage)/(end_cost-
start_cost)/(end_percentage-start_percentage)) cost
FROM sf_nbs
WHERE link_id=23598190
AND (0.812549889087677*start_percentage)/(end_percentage-start_percentage)>=0
GROUP BY buffer_id;
```

### Shortest path SQL query

```
SELECT * from sf_nbs where buffer_id in (7,10,4,6,8,3,9,1,11,5)
START WITH link_id=23598190
CONNECT BY
PRIOR prev_link_id=link_id
AND PRIOR start_cost=end_cost
AND PRIOR buffer_id=buffer_id;
```

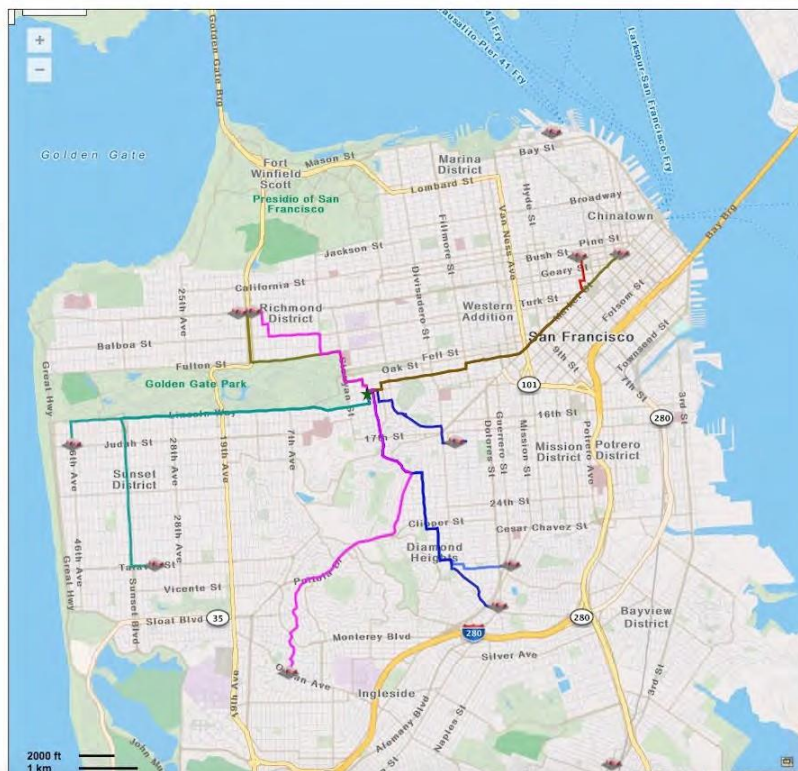


図4：LODデモ

## 縮約階層REST API

NDM CH REST APIを使用するためのおもな手順は以下のとおりです。

1. NDM CH RESTサービスをJEEコンテナまたはアプリケーション・サーバーにデプロイする
2. CH REST APIを使用してNDMネットワークからCHモデルを作成し、データベースに格納する
3. CHモデルをメモリにロードする
4. CH REST APIを使用してCHモデルを分析する

CH REST APIは、以下の分析機能をサポートしています。

- 最短パス
- 巡回セールスマン問題（TSP）
- コスト・マトリックス
- 代替ルート

図5に、NDM CH RESTデモのWebコンソールを示します。このデモで、CHモデルの作成、設定、分析を行うことができます。また、図6には、JSONでの最短パスのリクエストとレスポンス、およびその結果を示します。

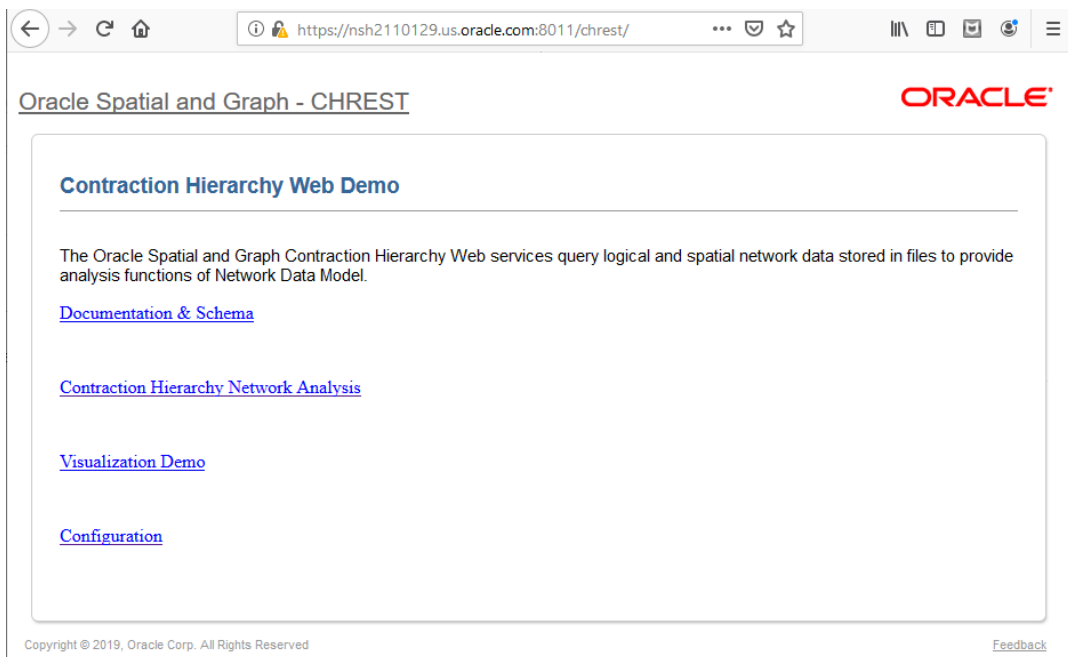


図5：CH RESTデモ

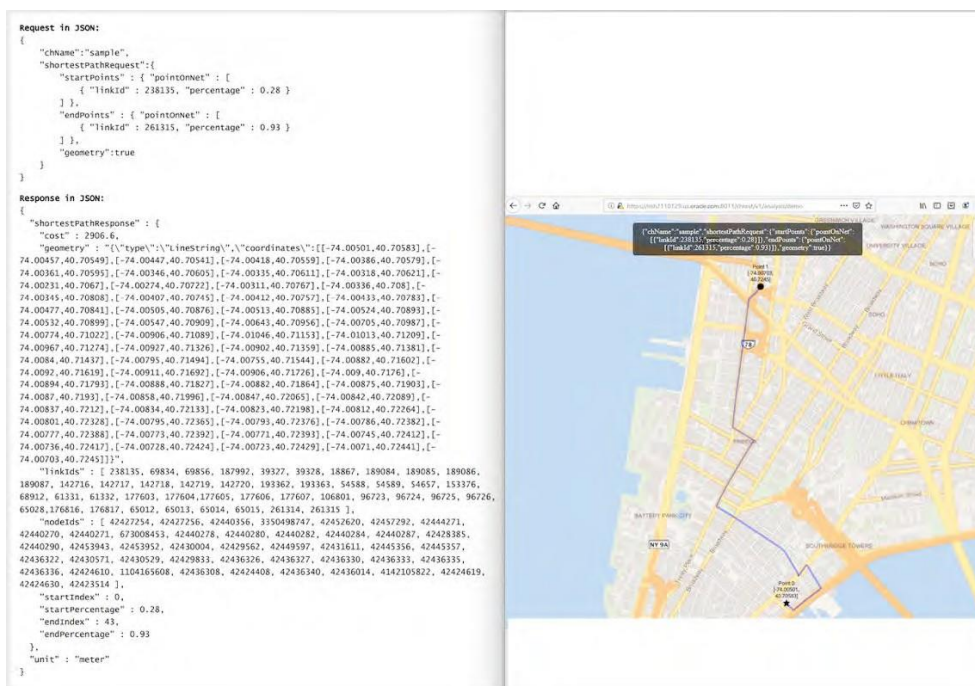


図6：CH RESTの最短パスのリクエスト、レスポンス、パス結果

## LOD Java APIとCHREST APIの比較

ロードオンデマンドJava APIと縮約階層REST APIを比較し、以下の表（表1）にそれぞれの使用法と利点をまとめました。

|              | ネットワーク・タイプ | 言語                 | カスタマイズ        | パフォーマンス | アプローチ               |
|--------------|------------|--------------------|---------------|---------|---------------------|
| LOD Java API | 静的/動的      | Java               | 高<br>(ユーザー定義) | 良い      | キャッシュを使用したロードオンデマンド |
| CH REST API  | 静的         | REST<br>(JSON/XML) | 低<br>(事前定義)   | 非常に良い   | メモリでの事前計算           |

表1：NDM LOD APIとCH REST APIの比較

LOD JAVA APIとCH REST APIのおもな違いは、カスタマイズとパフォーマンスです。動的なプロパティやビジネス・ルール/ロジックがあるネットワークでは、オラクルはLOD Java APIを使用することを推奨します。LOD Java APIでは、ネットワーク分析のカスタマイズを支援するネットワーク制約、ユーザー・データ、コスト計算機能が提供されます。CH REST APIは、ネットワークが静的で、高性能の分析を求める場合に適した選択肢です。

LOD Java APIを使用すると、ユーザーはカスタマイズを実装できます。一方、CH REST APIでは、現在は右左折規制といった一部の事前定義済みのカスタマイズのみがサポートされます。縮約階層などの事前計算のアプローチで動的な変更を考慮するのは困難なためです。

LOD Java APIは、必要な場合にのみネットワーク・パーティションをロードするため、必要とされるメモリが少なくてすみます。CH REST APIはインメモリ分析のアプローチを使用しており、CHモデル全体をメモリにロードする必要があります。そのため、より多くのメモリ・リソースが必要です。LOD Java APIを使用するには、ユーザーは、Javaを使用してネットワーク・アプリケーションやサービスを実装する必要があります。一方、CH REST APIでは、他のAPIとともに、HTTP経由でREST APIを使用できるため、分析と開発が簡素化されます。

## NDMを使用したルーティング・エンジン機能の拡張

Oracle Databaseの空間ルーティング・エンジンは、2つ以上のロケーション間の最適なルート、運転案内、およびルート・ジオメトリを提供するWebルーティング・サービスです。このルーティング・エンジンではNDM LOD Java APIが使用されます。ルーティング・エンジンは、トラックのルーティングや右左折規制に対処したり、交通パターンやタイムゾーン情報を分析に含めたりするまでに機能が拡張されます。これらの機能は、NDMのユーザー・データ、ネットワーク制約、コスト計算機能とともに実装されます。

### トラックのルーティング

Oracleのルーティング・エンジンは、道路、重量、高さ、時間帯、商用アプリケーションや物流アプリケーションに適用されるその他の条件、および論理的な右左折規制に基づいた、トラック固有のルーティングを提供します。一般車両の速度制限とは通常異なるトラックの速度制限に基づき、運転時間を計算できます。また、ルート上にある重量検査所やトラック運転者用施設など、トラック向けサービスも提供できます。

### 右左折の禁止

ルーティング・エンジンは、ルート・ジオメトリ内で2方向以上の点に関わる論理的な右左折規制にも対処できます。これらの機能強化によって、物流アプリケーションや車両ルーティング・アプリケーションで、より正確な結果が得られるようになります。

### 交通パターンとタイムゾーン

交通パターンは、特定の時間帯における過去の移動時間を反映しています。NDMは、コンテンツ・プロバイダの交通パターン・データを使用して、与えられた出発日と出発時間を基にルートを計算できます。また、現地時間情報を反映させるために、タイムゾーン情報を道路セグメント（ネットワーク・リンク）に追加し、ルート・レスポンスに含めることができます。

## まとめ

Oracle Databaseの空間機能であるネットワーク・データ・モデルでは、ネットワークのモデリングと分析を行う機能が提供されます。NDMネットワーク・スキーマはオープンで、SQL経由でアクセスできます。データベース・データ管理向けのPL/SQL API、カスタマイズ可能なネットワーク分析向けのロードオンデマンドJava API、そして21cでは高性能のネットワーク分析向けのREST APIが用意されています。

21c以前は、NDMでは、ロードオンデマンドJava APIを使用して、スケーラブルでカスタマイズ可能なソリューションを多くのネットワーク・アプリケーションに提供することに重点が置かれていました。21c以降は、縮約階層のアプローチに基づく高性能のREST APIが提供されています。CH REST APIは、静的ネットワークのネットワーク分析で著しいパフォーマンスを実現します。さらに、Webサービスとアプリケーション開発を簡素化します。NDMユーザーは、ご自身のネットワーク・アプリケーションに適したAPIを、ネットワークの種類と要件に応じて選択できます。

---

## Connect with us

+1.800.ORACLE1までご連絡いただくか、[oracle.com](http://oracle.com)をご覧ください。北米以外の地域では、[oracle.com/contact](http://oracle.com/contact)で最寄りの営業所をご確認いただけます。

 [blogs.oracle.com](http://blogs.oracle.com)

 [facebook.com/oracle](https://facebook.com/oracle)

 [twitter.com/oracle](https://twitter.com/oracle)

---

Copyright © 2021, Oracle and/or its affiliates. All rights reserved. 本文書は情報提供のみを目的として提供されており、ここに記載されている内容は予告なく変更されることがあります。本文書は、その内容に誤りがないことを保証するものではなく、また、口頭による明示的保証や法律による黙示的保証を含め、商品性ないし特定目的適合性に関する黙示的保証および条件などのいかなる保証および条件も提供するものではありません。オラクルは本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクルの書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

本デバイスは、連邦通信委員会のルールに基づいた認可を未取得です。認可を受けるまでは、このデバイスの販売またはリースを提案することも、このデバイスを販売またはリースすることはありません。

---

OracleおよびJavaはOracleおよびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。

IntelおよびIntel XeonはIntel Corporationの商標または登録商標です。すべてのSPARC商標はライセンスに基づいて使用されるSPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMDロゴおよびAMD Opteronロゴは、Advanced Micro Devicesの商標または登録商標です。UNIXは、The Open Groupの登録商標です。0120

免責事項：データシートにこの免責事項の記載が必要かどうか分からない場合は、収益認識方針を参照してください。本書の内容と免責事項の要件についてさらに質問がある場合は、[REVREC\\_US@oracle.com](mailto:REVREC_US@oracle.com)宛てに電子メールでご連絡ください。