neo4j

The World's Leading Graph Database

White Paper

Neo4j と Oracle RDBMS の共存について

Gabe Stanek & Stefan Kolmar

White Paper

目次

イントロダクション 1
Neo4j とは何か? 1
Neo4j と Oracle を併用する利点 2
Neo4j と Oracle の連携 3
結論 6

Neo4j と Oracle RDBMS の共存について

Gabe Stanek & Stefan Kolmar

イントロダクション

Oracle RDBMS の価値を疑う人は誰もいません。Oracle は、常にエンタプライズ・アプリケーションの中心に居るし、居続けるでしょう。しかし、現在の企業 IT 部門は、Oracle や他のデータベースが提供できる能力を超えるものを求めているのは確かです。つまり、組織は、Oracle を捨て去ろうと考えないが、それ以上のものも求めています。

・アプリケーションに新たな機能　・データの多様性
・アジリティー　・速度

そして最も重要と思われる、

・データ固有の関係性を見出し、イノベーションに結び付ける方法

です。

コネクティド・データ（結合したデータ）は、様々な文脈で企業間競争を有利にします。進んだＭＤＭ（マスター・データ・マネジメント）アプローチによりコネクティド・データを取扱うことは、多くの切り離されたシステム間のデータを、より効果的に繋ぎ利用することが可能となります。その結果、増え続ける、変化するデータセットを用いて、文脈を汲んだリアルタイム・リコメンデーションにより状況が好転し、増収に結びつくかもしれません。また、複雑に絡んだコネクションをリアルタイムで分析することは何百万ドルもの金融詐欺を防ぐかもしれません。

全てのビジネスは、データの関係性を、素早く利用することを求めています。グラフ型データベース Neo4j は、Oracle への投資を無駄にすることなく、そのような機能を提供できます。

Neo4j とはなにか?

グラフ型データベースは、Oracle に代表される RDB を理解することから始めると理解しやすいです。RDB は、高度に定型化してデータをストアします。つまり、事前決定されたカラムを持つテーブルに多くの同じタイプの情報を格納します。

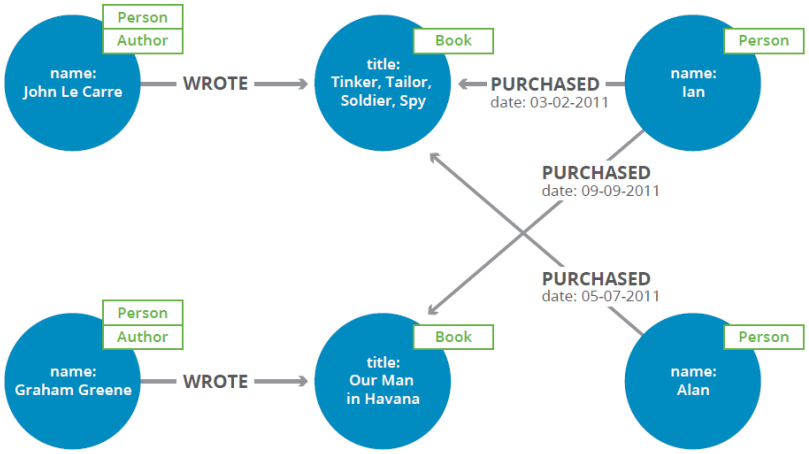
デベロッパーのアプリケーションは、このテーブル・フォーマットに基づき、データを構築しなければならないので、最初からスキーマが決定していて変化しないタイプの仕事には最も適しています。つまり、RDB は、高度に設計され事前決定されたスキーマには完ぺきなツールです。

しかし、このようなデータベースは、変化への対応が不得意であるし、データ間の関係性を辿っていくことも容易ではありません。つまり RDB は、多層の複雑な関係性をリアルタイムに扱うことが苦手です。

名前の意味する所は?

あなたの用いている "リレーショナル（関係性）データベース"は、実は、関係を扱うのには最適化されてません。確かに、JOIN を使用して関係性を検索することは可能ですが、JOIN が深くなるとパフォーマンスが低下します。
実際、「リレーショナルデータベース」という用語は、データ間の関係性を記述することとは全く関係ありません。ここで、「関係」と言っているのは、数学的概念（E.F. Codd's の Relational Algebra の一部）つまり、テーブルへの参照です。

Neo4j のようなグラフ型 DB は、データの関係性に第一順位を置いています。Neo4j は、グラフ型データモデルに対して CRUD（Create, Read, Update and Delete)のトランザクション・データマネジメント・システムを採用し、ACID 特性を備えています。 グラフ型データモデルはモノとその関係性から成り立ち、現実世界のデータの形そのままの如く直観的です。グラフ型データモデルは 2 つの要素：ノードと関係性（訳注：エッジとも呼びます）からなります。ノードはエンティティを表し、関係性はそれぞれのノードがどのように関係しているかを表します。ノードと関係性という単純な構造に抽象化し、結合された構造として表すことにより Neo4j は、対象の主体と密接に関連する洗練されたモデルを構築します。
開発者は、スキーマがオプショナルである（訳注：RDB のような事前決定されたスキーマはなく、制約条件としてスキーマをオプションで課することができるという意味）という事実により、アジャイルな開発が可能となり、開発を速くすることができます。
これは、ビジネス要件が何か、ということに基づいて精巧なデータモデルを事前に固定する必要がないということです。代わりに、この柔軟性は、入力データの取扱に必要な要件を失うことなく、データモデルとアプリケーションの抽象化を可能とします。
Neo4j のデータモデルはホワイトボードに図で描くが如くであり、ビジネスと IT のギャップを埋め、ビジネスの変更要求に柔軟に対応したシステム変更が出来ます。
Neo4j は、誰もが理解できる直感的なデータモデルを使用している為、データベースのバックグラウンドのないチームメンバー、クエリを簡単に理解できます。



A

簡単なデータモデル：書籍、著者、購買者

Neo4j と Oracle を併用することの利点

Neo4j と Oracle RDBMS を併用することにはいくつかの利点があります。まず、Oracle は信頼性が高く確立された技術であるという単純な事実から始めます。Oracle 上で動作する非常に多くのエンタープライズ・アプリケーションがあるため、Oracle インフラストラクチャ、ツール、およびトレーニングへの今までの投資を捨てるべきではありませんし、もちろんそうする必要はありません。すべてのアプリケーションで Oracle RDBMS を引き続き使用します。

あなたの「兵器」に Neo4j を追加することで、何も失う事なしにグラフデータベースの全ての戦略的能力を獲得出来ます。Neo4j と Oracle を併用して、コネクティドデータに関する検索を Oracle からオフロードすることで、アプリケーションのパフォーマンスを向上させることができます。 RDB でも、テーブルを JOIN することによって、コネクティドデータの検索を行うことができますが、多量のメモリと CPU パワーを消費し、データ量の増大と共に、パフォーマンスへの影響が大きくなります。グラフデータベースは、複雑なコネクティドデータをより迅速に分析し、将来の参照の為に保持することができます。

最後に、RDB とグラフデータベースの使い分けは理にかなっています。

Oracle はテーブル形式の構造化されたデータに最適ですが、構造化されていないデータ、あるいは、高度に接続されたデータ、または階層・ネットワーク等のデータは Oracle から Neo4j に移行するのが適切です。そうする事により、機動性が高まります。共存するように設計した場合、Neo4j でグラフに対して高速なトラバーサル方式での検索を実行できる一方で、Oracle からトランザクション・データを取得できるため、アプリケーションのパフォーマンスが向上します。

グラフデータベースと RDB はそれぞれ特定のユースケースに対応しています。したがって、Neo4j は Oracle と競合ではなく、共存します。Neo4j と Oracle を組み合わせることで、開発者やアーキテクトはそれぞれのユースケースに適したタイプのデータベースを使用することができます。

Neo4j と Oracle との連携

Neo4j が Oracle（または他の RDBMS）と共に動作する方法はいくつかあります。アプローチは、企業が達成しようとしているゴールによって異なります。

Neo4j にデータのサブセットを移行または同期

現在のアプリケーションがデータの深さ（訳注：JOIN の深さ）と複雑さのために遅いと考える場合、ACID に完全準拠である Neo4j にデータを一部移行することが選択できます。 たとえば、顧客と製品の情報は Oracle に置き、顧客や製品の識別子、およびそれらが相互関係は Neo4j に置きます。 つまり、このアプローチは、RDBMS からのデータの一部を Neo4j に同期させます。以下はノルウェーのテレコム Telenor の場合に用いられた方法です。

Telenor のセルフサービス Web アプリケーションは究極的には顧客を失いかねない重要な課題「ログイン回数の増加」に直面していました。 ログインプロセスでの各ユーザーのリソース認証は、特にユーザー数が多い大規模顧客の場合には、大変な時間がかかっていました。.

当初は、Telenor はストアドプロシージャを使用し、夜間バッチで最大顧客のリソース認証を事前計算の上、結果をキャッシュしようとしました。これにより、最大顧客のログイン認証は早くなりました。しかし、このアプローチにはいくつかの欠点がありました。キャッシュされたデータは最大 24 時間前である可能性があり、新しいユーザーの追加や終了した従業員の削除などの変更には最大 1 日かかるということ。小さい顧客の問題を解決していないということ。 さらに先を見据えると、Telenor は、事前計算アプローチは成長とともにスケールしないと考えました。 また、ストアドプロシージャは非常に複雑で、約 1500 行の SQL コードが必要でした。

Telenor は RDBMS を補完する為、NoSQL を用いるアプローチを検討しました。検索用途には Solr / Lucene とデータを同期し行い、リソース認証用には Neo4j と同期することにより行うことに決めました。 Neo4j を使用することで、Telenor は事前計算するのではなくリアルタイムでリソース認証を実行しつつ、ミリ秒単位のログイン時間を提供し、変更もほぼリアルタイムで反映されるようになります。

Neo4j へのデータ同期は、Oracle GoldenGate、Monsanto が開発した Kafka アダプタ他、GitHub で利用可能なオープンソースのアダプタなどのミドルウェアを介して処理されます。Neo4j へのデータの移行は、CSV ファイルで処理されます。 Oracle からテーブルをエクスポートし、Cypher 言語（訳注：グラフクエリ言語）のコマンドを使用して Neo4j に CSV ファイルをインポートします。 これは手動で実行することも、ETL ツールを使用して移行を自動化することもできます。

Oracle と Neo4j のデータの完全同期

より複雑なユースケースでは、Oracle と Neo4j 間のすべてのデータを同期させます。

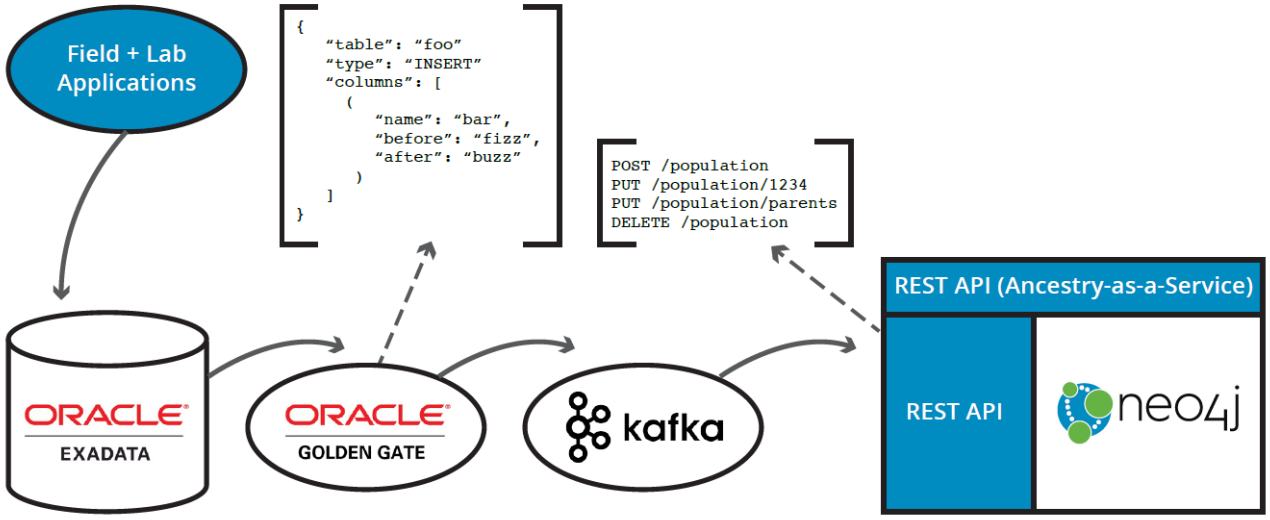
複数のデータソースからのデータを統合するアプリケーションは、完全同期の一例です。完全同期のもう 1 つの使用例は、既存のアプリケーションが Oracle を使用しているが、それらのアプリケーションの変更にはコストがかかりすぎる場合です。データの付加価値をあげる為に、Oracle RDBMS が対応不能な部分に新しいテクノロジを導入する必要があります。 これがモンサントの場合に当てはまりました。

モンサントは、農薬や農業バイオテクノロジーの多国籍企業です。 Monsanto は、Neo4j 採用の前、コア遺伝子・祖先データの管理は 96CPU の Oracle Exadata に依存していました。そこでは多数のストアド・プロシージャ、JOIN テーブル、再帰クエリ、およびデュアルインデックスを使用してパフォーマンスを最適化していました。Monsanto チームは、Oracle チューニングと最適化に精通しており、Oracle RDBMS を 30 年以上にわたってチューニングしてきました。 しかし、Exadata のインスタンスは、遺伝子・祖先データのリアルタイム処理（これはチームが新しいゲノムテスト技術を使用して、市場投入までの時間を 1 年に短縮することの前提条件）に頻繁にフェイルするようになりました。

チームがリアルタイムの結果を生成する最初の試みは、巨大なインメモリグラフを構築し、解析することでした。しかし、一度クエリが完了すると、グラフは消えてしまいます。 チームは長期的にグラフデータを保持する方法を模索し、Neo4j を発見しました。Neo4j に出会って 1 日で、チームは小さなデータセットでのプロトタイプを作成しました。 1 ヵ月後、チームは、遺伝子・祖先データセット全体をベータリリースアプリケーションのため Neo4j に移行しました。 しかし、Neo4j がフル・プロダクションとなった後も、何十ものアプリケーションが引き続き Exadata 環境でデータの読み書きを行っています。チームはこれらのデータベース接続をオフにするのではなく、Exadata と Neo4j との間で情報を同期させるカスタム API を構築したのです。

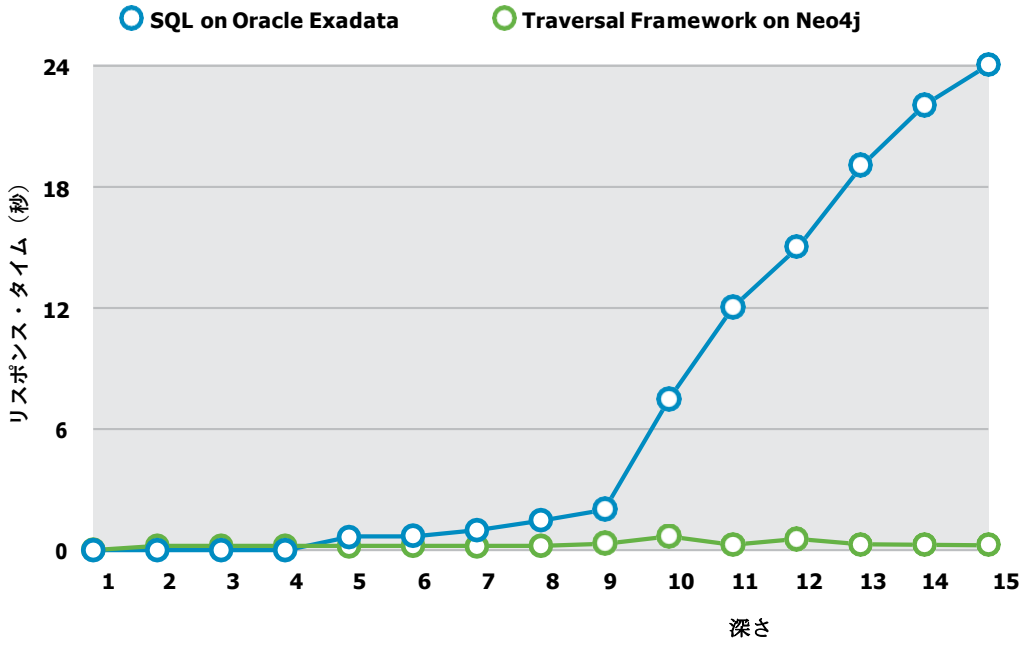
その後、チームは新たなクエリインターフェイスを導入しました。そのインターフェイスは、以前は SQL アルゴリズムでは不可能だったシンプルでキーワードドリブンな方法で、何層にも深い関連性のクエリを実行することを可能にしました。このアーキテクチャーでは、Apache Kafka を分散コミット・ログとして使用して、Neo4j に Oracle のライブ・トランザクション・データを供給します。 チームは Oracle GoldenGate と Kafka コネクタを構築しました。現在、これらはオープンソースで GitHub で利用可能になっています。

How Neo4j Co-exists with Oracle RDBMS



Monsanto における Ancestry(祖先)-as-a-Service

現在は、遺伝子・祖先のソリューションはおよそ 120 の異なるアプリケーションで、データ科学者にサービスを提供し、約 10 億ノードの REST 要求を 6 億件以上処理し、結果はわずか 10 分の数ミリ秒で返されます。



Finding all of a plant's ancestors at Monsanto: Query times for Oracle Exadata versus Neo4j

コネクションの深さとデータ量のクエリ時間に与える影響

Neo4j では、1 秒間に 1 コアあたり数百万の深さのデータを照会することができます。これは、リレーショナルデータベースの数百万もの JOIN 操作を行うことに相当しますが、これは不可能です。 速度には大きな違いがあります。この差異は、関係性がより緊密で、データが多いほど顕著になります。

データが大きくなれば成程、Oracle などのデータベースでは、データ同士の関連付けが遅くなります。 Neo4j を使用すると、何十億ものノードとリレーションを持つデータに対する最短パスのクエリが 1〜2 ミリ秒で出来ます。同等の SQL クエリは、アプリケーションが Oracle などの RDBMS のみを使用していた場合には、何桁も遅く実行されます。[1]

全テータの Neo4j への移行

ケデータベースによっては、既存のアプリケーションがそのアーキテクチャ上およびパフォーマンス上の制限を打破する必要がある場合があります。これは新しいアプリケーションの開発の必要性に直面していることを意味する場合がしばしばあります。データが高度に関係性のあるデータである場合、唯一のデータベースストアとして Oracle ではなく Neo4j を選択することが理にかなっています。その場合、すべてのデータを Neo4j に移行します。

既存のアプリケーションとの置換えに意味のあるシナリオがいくつかあります。例えば、ビジネスユーザーがアプリケーションが非常に遅いと訴えている場合、Neo4j への移行すると、クエリ時間が大幅に短縮される可能性があります（左欄を参照）。

同様に、アプリケーション内の多くのクエリが、様々なデータセットに JOIN され、JOIN の階層が深い場合、また、そのセットが簡単に予測または事前計算できない場合、Neo4j ではデータがより効果的に取り扱える場合があります。そして最終的に、クエリが非常に複雑になり、新しいスタッフを訓練するのに時間がかかりすぎる場合にも、neo4j を選択します。グラフデータベースでは、コネクティド・データのクエリは、初心者が最小限のトレーニングで扱えるくらいに簡単です。

Oracle RDBMS を Neo4j に置き換えることは、データのサブセットを移行することに似ています。すべてのデータを移行するだけです。マイグレーションには、データベースとインターラクトするアプリケーションコードの編集してそれらのクエリーを書き換えるアプリケーションコードの編集とクエリの変更が含まれます。

多くの企業が、サプライチェーンマネジメントのような従来のシステムを Neo4j に移行しました。 ドイツ最大の玩具メーカーの Schleich GmbH は、RDBMS を搭載した製品データ管理（PDM）ソリューションが、サプライチェーンに必要な柔軟性、パフォーマンス、および運用効率を提供していないことを発見しました。 Schleich は Neo4j で次世代 PDM システムを構築することを決めました。

同社の技術パートナーStructr は、6 ヶ月で、以前の PDM システムの全データを Neo4j 用いるオープンソースのグラフアプリケーションプログラムに移行しました。 バリューチェーン全体のトレーサビリティとコンプライアンスは、玩具業界の重要な要素であり、新しい PDM システムは、第三者の可視化が可能となるなど、必要な要件をすべて満たしています。 現在ではグラフベースの PDM システムは SAP と統合され、会社の主要分野に導入されています。

1. *Graph Databases*, published by O'Reilly Media

Neo4j の効果

パフォーマンス

- コネクティド・データに対して非常に高速

柔軟性と機動性

- データモデルをいつでもリスク無く変更可能

理解のし易さ

- ホワイトボードに図で描くが如きデータモデルの直観性

コーディングのし易さ

- 50～ 200 行の SQL が、 3～7 行となる Cypher 言語（グラフ・クエリ言語）

Neo4j にすべてのデータを移行させたもう 1 つの使用例は、買収した企業の多数のシステムを一つのアプリケーションに統合することでした。 1939 年に創業した印刷会社 Billes は、テクノロジーパートナーNetConsult と共に、Neo4j に基づいた企画、発注、配送、請求システムを構築しました。 さまざまな合併や買収により、ツールと並列 IT システムのパッチワークが生まれました。 これらのシステムのすべての関連部分は、Poff と呼ばれる Neo4j ベースのシステムにシームレスに統合されました。Neo4j にはいつでも変更できる柔軟なデータモデルがあるため、システム統合の労力が大幅に削減されました。 合計で約 40 台の外部システムが注文を受け取り、Poff を通じてそれらを処理します。 システムの成功と容易な管理により、同社は消費者からのオンライン注文を含む新しいビジネス分野を追加することができました。

結論

データベースの真実を知るときです。

リレーショナルデータベースでは、必要なすべてを行うことはできません。

さらに重要なことは、Oracle RDBMS への投資を守ろうとグラフ・データベース、用いない選択をする必要はありませんし、グラフ・データベースを用いたから Oracle RDBMS への投資が無駄になる訳でもありません。

Neo4j を使用しながらも、グラフデータベースであろうとリレーショナルデータベースであろうと、アプリケーションに最適のテクノロジーを選ぶべきです。 Neo4j は、ネイティブグラフストレージでネイティブグラフプロセッシングで構築された世界初のグラフデータベースとして、巨大なデータに対してもパフォーマンスが落ちません。また、Neo4j エコシステムへのコントリビュートするグラフデータベース愛好家の大規模コミュニティのおかげで、Neo4j は日々進化します。

Neo4j を環境に追加する方法については、Neo4j エキスパートに連絡ください。

Neo Technology is the creator of Neo4j, world's leading graph database. Neo4j is a highly scalable native graph database that leverages data relationships as first-class entities to help companies build intelligent applications that meet today's evolving connected data challenges including fraud detection, real-time recommendations, master data management, network security and IT operations.

Enterprises like Walmart, UBS, Cisco, HP, adidas and Lufthansa and hot startups like Medium, Musimap and Glowbl rely on Neo4j to harness the connections in their data.

Questions about Neo4j?
Contact us:
株式会社ＵＴＩ（正規代理店）
092-715-1010
hosoi@uti-inc.com

© 2016 Neo4j. All rights reserved.