



IBM Cloudant®

NoSQL を導入すべき理由とは

新たな非リレーショナルな世界で活用可能なデータベース・オプション

目次

新たな種類のアプリケーションが新たな種類のデータを生み出す	3
NoSQL の簡単な歴史	3
オープン・ソースから生まれた NoSQL	3
NoSQL データベースの種類	4
キー・バリュー・ストア	4
グラフ・ストア	4
カラム・ストア	4
ドキュメント・ストア	4
NoSQL を検討すべき理由	5
柔軟性	5
スケーラビリティ	5
可用性	5
運用コストの削減	5
特殊機能	5
まとめ	6
IBM Cloudant の使用を開始するには	6
詳細情報	6

新たな種類のアプリケーションが新たな種類のデータを生み出す

オンラインでの消費者の行動が変化し、新たな種類のデータが出現したことに加えて、Web アプリケーション、モバイル・アプリケーション、IoT アプリケーションの増大が止まらないため、開発者はデータの保存と管理の方法を見直すようになってきました。現代のニーズによって、グローバルのユーザー・ベースとの間で膨大なデータの流れを効果的かつ安全に管理できるスケラブルで柔軟なソリューションを提供するデータベースが求められています。

開発者も IT 部門も、動的なスケールアップによってユーザーが求めるパフォーマンス・レベルを維持しつつ、このデータのすべてを迅速にリレーショナル・モデルに統合することは困難であり、場合によっては不可能であるとさえ考えています。このため多くのユーザーは柔軟性を求めて NoSQL データベースを検討するようになり、この状況から見て、グローバルの SQL 市場は 2020 年にはほぼ 2 倍に成長し、34 億米ドルに達するものと考えられています¹。

NoSQL の簡単な歴史

「NoSQL」とは「SQL だけではない」すなわち「リレーショナルではない」という意味であり、ドットコム・ブームによって高まったデータ型、データ・アクセス、データの可用性のニーズに特に対応するために生まれました。

20 年ほど前にさかのぼると、アプリケーション・アーキテクトや開発者がアプリケーションのためにデータ・ストアを必要とする場合、さまざまなリレーショナル・データベースを選択していました。実際、リレーショナル・データベースは、各大学でコンピューター・サイエンスを学んだ開発者とツールと予測可能な運用特性を熟知していたインフラ・チームの双方にとって利用可能な唯一のオプションであったため、1970 年代以降、標準の選択肢となっています。よく使用されるリレーショナル・データベースとしては、Oracle、MySQL、SQL Server、DB2 が挙げられます。

しかし、1990 年代後半から 2000 年代初めにかけてインターネットを活用するアプリケーションと企業が爆発的に増加し、アプリケーションのユーザーは数千人の社内ユーザーから公開されたインターネット上の数百万人へと増加しました。このようなアプリケーションにとっては、パフォーマンスと可用性が何よりも重要です。大規模に高可用性を実現しなければならない新たな課題によって、Google、Facebook、Amazon といった企業は新たなテクノロジーの開発を迫られました。幸いなことに、このような企業は自社の取り組みを文書化し、ホワイト・ペーパーを発表し、テクノロジーをオープン・ソース化したため、インターネット・コミュニティはこれらのリソースを活用することができました。2000 年代後半になると、いくつかの非リレーショナル・データベース・テクノロジーが出現し、NoSQL がこのようなテクノロジーを表す名称となりました。

オープン・ソースから生まれた NoSQL

多くの NoSQL データベースは、オープン・ソースのコミュニティから生まれています。NoSQL の人気と活用の増大がとどまるところを知らないのは、この歴史的背景によるものです。このテクノロジーのサービスとサポートの付いた有償版のデータベースを提供すると同時に、オープン・ソース版のコミュニティに参加している企業もよく見かけます。このような関係を示す例としては、Datastax と Apache Cassandra™、IBM Cloudant と Apache CouchDB™、さらには MongoDB とオープン・ソース版の MongoDB ソフトウェアが挙げられます。

¹ <http://www.marketresearchmedia.com/?p=568>

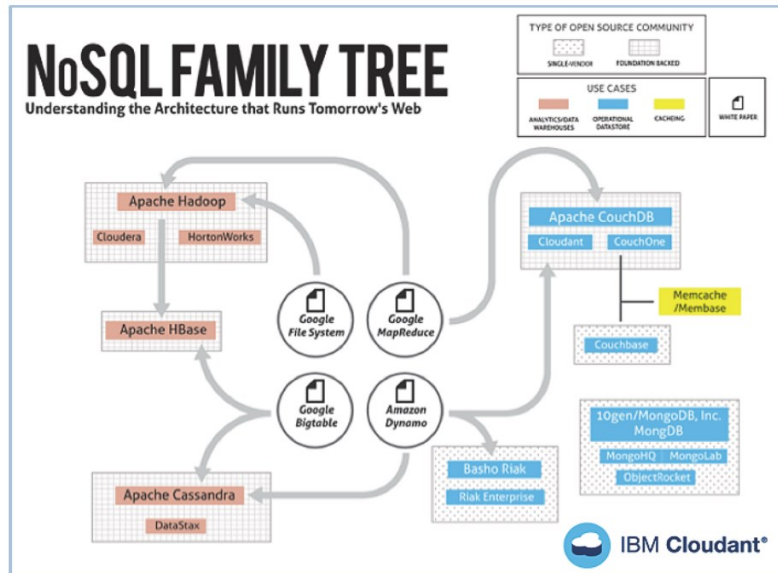
NoSQL データベースの種類

NoSQL とは、リレーショナルではないすべてのデータベースを表すための言葉に過ぎません。テクノロジー、データ型、ユース・ケースがデータベース間で大幅に異なるにもかかわらず、以下の 4 種類の NoSQL データベースが存在すると通常考えられています。

キー・バリュー・ストア

このデータベースは、コンピューター・プログラム内のハッシュ・テーブルのようにキーと値をペアで連携させます。キー・バリュー・ストアはよくファイルシステムに例えられます。パスがキーの役割を果たし、コンテンツがファイルの役割を果たします。更新対象となる「フィールド」は存在しないことが多く、変更を行う場合は、キー以外の値全体を更新する必要があります。このようなシンプルな構造によって効率

的にスケールアップを行うことができるものの、クエリーの複雑度やキー・バリュー・ストアで利用可能なその他の先進機能で制限が発生することがあります。純粋なキー・バリュー・ストアの例としては、*MemcacheD*、*REDIS*、*Riak* があります。



グラフ・ストア

グラフ・データ・ストアは相互接続するデータの処理に優れています。グラフ・データベースはノード間の接続 (グラフ・データベースの用語では「エッジ」と言います) で構成されています。ノードとエッジの双方が追加のプロパティ (キーと値のペアなど) を保存できます。グラフ・データベースのメリットとは、ノード間の接続をスキャンできることです。しかし、通常グラフ・データベースではすべてのデータが 1 台のマシンに収まる必要があり、スケーラビリティに限りがあります。グラフ・ストアの例としては、*Neo4j* と *Sesame* があります。

カラム・ストア

リレーショナル・データベースはオンディスクで特定のテーブルの行にすべてのデータを保存するため、特定の行の検索をスピードアップすることができます。カラム・データベースは通常特定の列のすべての値をオンディスクでシリアライズするため、特定の属性を持つ大量のデータをスピーディーに抽出できます。このアプローチは、特定のフィールドに対して一連のクエリーを実行する可能性があるクエリーとアナリティクスの手法を統合する処理に適しています。カラム・ストアの例としては、*HBase* と *Cassandra* があります。

ドキュメント・ストア

ドキュメント・データベースはレコードを「ドキュメント」として保存します。ドキュメントとは通常キーと値のペアをグループ化したものを指します。キーは必ず文字列であり、値は文字列、数値、ブール値、配列、その他のネストされたキーと値のペアのいずれかです。値は任意の深度でネストできます。ドキュメント・データベースでよく使用される構文には、XML と JavaScript Object Notation (JSON) があります。ドキュメント・ストアの例としては、*MongoDB*、*CouchDB*、*Cloudant*、*MarkLogic* があります。

NoSQL を検討すべき理由

現在利用可能なさまざまな NoSQL データベースはそれぞれ内容が大きく異なるものの、柔軟性、スケーラビリティ、可用性、低コスト、特殊機能といったそれぞれに共通する要素があります。

柔軟性

スキーマの柔軟性と直感的なデータ構造はアジャイルな開発サイクルで作業する開発者にとって魅力的な主要機能であり、ほとんどの NoSQL データベースはこの特性に対応しています。通常、リレーショナル・データベースのように、スキーマの更新とそれに伴うダウンタイムは必要ありません。NoSQL の柔軟性によってコードベースの変更に伴うデータベース・スキーマの変更の複雑性を排除できるため、アジャイルな開発作業を実現するためによく使用されます。

スケーラビリティ

ここで言う「スケール」とは、データのサイズだけでなくデータに対して処理を行う同時ユーザーの数のことも指します。通常、NoSQL データベースはリレーショナル・データベースよりもスケールアップを伴うさまざまなユース・ケースに適しており、関連するアプリケーション機能の開発がより簡単になっています。

多くの NoSQL データベースはスケールアウト機能を備えており、リレーショナル・データベースよりも簡単にサーバーのクラスターに対してデータを分散すること（シャーディング）ができます。クエリーが複数のシャードに対して JOIN を実行すると、リレーショナル・データベースのパフォーマンスが悪化するものの、NoSQL データベースでは全く JOIN を実行する必要がなく、高いパフォーマンスを維持できます。

可用性

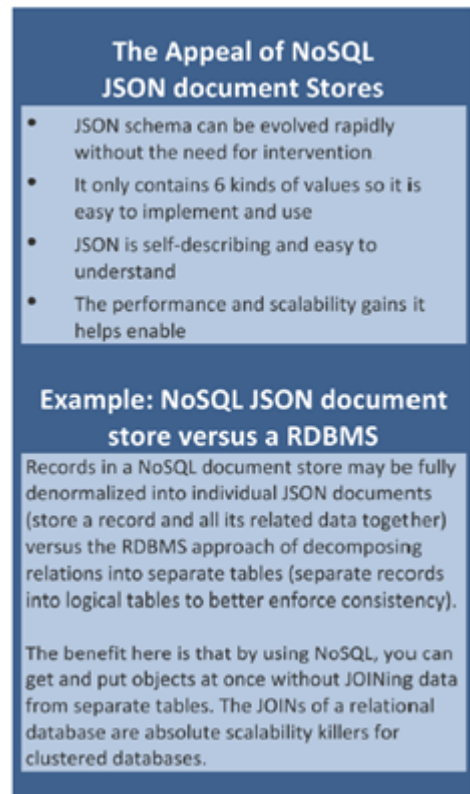
ダウンタイムが発生すると売上と顧客が失われ、ユーザーの満足度が低下します。幸いなことに、一部の NoSQL データベースは新規またはさまざまなレプリケーション・アーキテクチャーを提供するため、さまざまな種類の NoSQL データベースは読み取り処理だけでなく書き込み処理においても可用性を向上させることができます。つまり、1 つ以上のデータベース・サーバーすなわち「ノード」がダウンしても、データの損失を伴うことなくシステムに含まれる他のノードが業務を継続できるのです。

運用コストの削減

オープン・ソースから生まれた NoSQL の明らかなメリットは簡単に導入できることであり、NoSQL の導入企業が既存のデータベースと同等以上のパフォーマンスと機能を維持しながら、大幅なコスト削減を実現していることはよくあることです。これまで、大規模なリレーショナル・データベースは高価なマシンやメインフレーム上で稼働してきました。NoSQL データベースは分散処理の特性を持つため、クラウド・アーキテクチャー上のサーバー・クラスター上で実装と運用が可能です。

特殊機能

すべての NoSQL データベースが同じ機能を提供するわけではありません。インセンティブと追加の統合機能をユーザーに提供することを目的として、NoSQL データベースの提供企業の多くはさまざまな特殊機能を提供しています。その例としては、地理空間データに対する特殊なインデックス機能と検索機能、検索機能に組み込まれたフルテキスト・インデックス機能、データのレプリケーションと同期の自動化機能、アプリケーションに対して透過的な RESTful な Web API などが挙げられます。



The Appeal of NoSQL JSON document Stores

- JSON schema can be evolved rapidly without the need for intervention
- It only contains 6 kinds of values so it is easy to implement and use
- JSON is self-describing and easy to understand
- The performance and scalability gains it helps enable

Example: NoSQL JSON document store versus a RDBMS

Records in a NoSQL document store may be fully denormalized into individual JSON documents (store a record and all its related data together) versus the RDBMS approach of decomposing relations into separate tables (separate records into logical tables to better enforce consistency).

The benefit here is that by using NoSQL, you can get and put objects at once without JOINing data from separate tables. The JOINs of a relational database are absolute scalability killers for clustered databases.

まとめ

今後のアプリケーションのためのデータベースの選択が今ほど重要な意味を持つことはありません。幸い、データベースを 1 つだけ選択する必要はありません。現代のアプリケーションはノンストップで稼働することが求められ、その際増大の一途をたどる複数構造を持つデータを効率的に管理しなければなりません。この結果、NoSQL はもはや単なる流行語ではなくなり、小規模な店舗から大企業まで、あらゆるデータベースを検討する際の重要な要素となりました。

NoSQL データベースにはいくつかの共通する特性（リレーショナルでないことや通常簡単にスケールアップできることなど）がありますが、お客様の独自のデータ・ニーズに対応する適切な NoSQL ソリューションを決定する際には、検討すべき独自の差別化要因が数多く存在します。適切な NoSQL データベースはリレーショナル・データベースの有効な代替ソリューションとして機能し、既存のシステムとともに補完的に活用できます。

将来においては効率的なデータ・デリバリーを実現する必要があり、お客様がスケーラビリティ、さまざまな種類のデータ、大量の潜在的なユーザーのニーズがあるアプリケーション開発プロジェクトのプランを策定する際には、NoSQL テクノロジーの検討が必須となります。

IBM Cloudant の使用を開始するには

Cloudant は完全に管理された NoSQL ベースの Database-as-a-Service (DBaaS) です。そのため、迅速かつ簡単にプロビジョニングが完了し、データ管理を簡略化できます。また、自社専用のデータ・センターで Cloudant プラットフォームのパワーを活用できる Cloudant Local としても導入できます。さらに、Cloudant Local のデータベースと Cloudant Managed の DBaaS データベースを統合することによってハイブリッド・クラウド・データベースを構築し、クラウド・コスト、リーチ、パフォーマンス、コンプライアンス管理のバランスを最適化することもできます。<https://cloudant.com> で無償のアカウントを登録するだけで、Cloudant の使用を開始することができます。

詳細情報

詳細情報が必要な場合は、IBM の営業担当員か IBM のビジネス・パートナーにお問い合わせをいただくか、cloudant.com または ibm.com/cloudant の Web サイトをご参照ください。