

オラクルのコンバージド・データベース： 開発者とデータの生産性を向上させる方法



ビジネス・プロセスとディジジョン・ポイントのデジタル化を推進するにつれ、企業は、今の開発者の生産性を向上させるか、後でデータの生産性を向上させるかという、不可能に思える選択をするよう迫られます。しかし、オラクルのコンバージド・データベースという根本的に新しいアプローチによってその行き詰りを打破できます。

本書の目的

本書の目的は、単一目的データベースと比較した場合のコンバインド・データベースの利点について、CTO、エンタープライズのアーキテクト、開発管理者にご理解いただくことです。

対象読者

本書の対象読者は、CTO、エンタープライズのアーキテクト、開発管理者など、エンタープライズのコンピューティング・アーキテクチャの将来に関して決定を下すITリーダーです。

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

ジレンマ：開発者の生産性か、データの実産性か	3
インターネット・コンシューマ・サービス：開発者が支配する領域	3
エンタープライズSAAS：データが王様となる領域	4
両立の鍵：オラクルのコンバージド・データベース	4
マルチモデル：単独エンジン、多くの特徴	5
マルチテナント：統合、分離、俊敏性	6
マルチワークロード：一度に様々なジョブを実行	8
Oracle Autonomous Database：サービスとしてのコンバージド・データベース	9
オラクルのコンバージド・データベースによって実現される統一データ層	10
結論	11

ジレンマ：開発者の生産性が、データの生産性が

企業は競争力を維持するために、独自のデータ資産を構築し、それを最大限に活用しなければなりません。しかし、より多くのプロセスや意思決定ポイントをデジタル化するために、より多くのアプリケーション、アナリティクス、AIを活用するようになると、今のアプリケーション開発速度が上がるように最適化するのか、後でデータからの価値創出がしやすくなるように最適化するのかという、難しい選択を迫られます。つまり、開発者の生産性とデータの生産性の二者択一です。

開発者の生産性を優先して最適化する場合、チームは特定のプロジェクト用に複数の単一目的データベースを立ち上げます。各データベースは、その目的に応じた便利なデータモデルとシンプルなAPIセットを提供し、それらを利用した開発を簡単に開始できるようにします。しかし、プロジェクトの規模が大きくなり、追加の単一目的データベースやクラウド・サービスが必要になると、これらのサービス間でデータが断片化してしまいます。データベースごとに独自のツール、セキュリティ方式、運用特性があるため、データの一貫性に欠けるリスク、セキュリティギャップ、重要なレポート作成や分析作業でそのデータを使用する際の難易度が高くなるなどの問題があります。

一方、データの生産性を優先して最適化する場合、チームは企業標準データベース（通常はリレーショナル・データベースまたはリレーショナルベースのマルチモデル・データベース）のインスタンス上に構築します。企業の標準データベースは決められたポリシーに従い、レポートのような予想されるデータの再利用を簡素化しますが、機能が限定されているために、イノベーションが遅れるか、あるいは妨げられる可能性があります。これは、ビジネス全体が競合他社に対して不利な状況に追い込まれるリスクがあります。

このような選択はどちらも受け入れられません。どのようにしてビジネスはこのジレンマに陥ったのでしょうか。より重要な点として、どうしたらそのジレンマから逃れることができるのでしょうか。

インターネット上の消費者向けサービス：開発者が支配する領域

検索、eコマース、ソーシャルメディアなどのインターネット上の消費者向けサービスが飛躍的に普及したとき、これらのサービスは、エンタープライズ・アプリケーションからの新たな要件に直面しました。詳細なビジネス・プロセスではなく、断片的な顧客とのやり取りをサポートしていたため、比較的シンプルなアプリケーション・ロジックを使用するのが一般的でした。また、既存の企業システムには接続せず、ドキュメント、オブジェクト、グラフなどの軽量なデータモデルを好んで使用していました。これらのサービスは、ピーク時には数千万人から数億人のユーザーを抱える前代未聞の変動性に直面していました。そして、これらのユーザーは、サービスがキャプチャするデータの整合性よりも、サービスの全体的な体験をはるかに重視していました。

これらの新しい要件に対処するために、インターネットのパイオニアたちは、個々のサービスから独自のデータ階層を考案し、それぞれが特定のデータモデル、アクセス方法、ワークロードのニーズをサポートしました。こうしたデータ層のクラウド・サービスは、インターネットの消費者向けサービスにとって意味のある機能を具現化しましたが、次のようなトレードオフの受け入れが必要でした。

1. 最大限の規模での可用性を整合性より優先する。サービスがダウンしては意味がないため、たとえアプリケーション側で不整合データを管理することが必要になるとしても、残ったサービスの可用性を確保することが最優先された。そのため、最大限の規模でのパフォーマンスを改善する目的で、トランザクションのサポートとデータの整合性の要件は緩和された。
2. データベース管理システム (DBMS) よりもBitbucketを優先する。JSON、グラフ、分析などの特定のデータモデルやワークロードをサポートする、インターネットプロトコルに準拠した基本的なリクエストを提供するベアボーン単目的APIが好まれた。これらの最小限のデータベースは複雑なエンタープライズのプロセスにはかなり原始的だが、消費者向けインターネットアプリの個々の機能の開発をサポートすることはできた。
3. モノリスよりもマイクロサービスを優先する。アプリケーションがより多くの機能と複雑さを必要とする場合、複数の「素」の単一目的サービスを一緒に利用して、個々のサービスの障害に対する耐性を維持しながら、よりリッチなアプリケーションを構築することができた。マイクロサービスをベースにしたこのアプローチは、開発チームがお互いに独立して動作することを可能にし、俊敏性と革新性を高めることが可能だった。

柔軟性に優れるが、運用上の複雑さが増し、リスクが高まる

特定のデータ・タイプやワークロードを対象とした単一目的データベースからなる新しいクラウド・データ層を構築することで、早期に採用した企業は、シンプルな消費者向けクラウド・アプリケーションの開発期間を短縮することができました。

しかし、複数の単一目的データベースを統合して完全にセキュアな高可用性エンタープライズ・ソリューションを構築しようとする、すぐに複雑になる可能性があります。複数のサービスにまたがって断片化されたデータを管理するために、多くのカスタム・コードとハードなトレードオフが必要になるかもしれません。いくつもある単一目的データベースの上に構築されたアプリを大規模に協調可能にするために必要となる統合作業の負担を企業自身が負いつづけることになり、半永久的に再統合、再テスト、再調整やトラブルシューティングに振り回されかねません。

単一目的データベースごとに運用に精通した人材が必要です。セキュリティ・ポリシーはすべてのデータベースで再実装する必要があり、アプリケーションは、あるデータベースから別のデータベースへデータを伝播させるため、より複雑になります。皮肉なことに、「ベスト・オブ・ブリード」の単一目的データベースを組み合わせると、セキュリティ、スケーラビリティ、レプリケーションなどのエンタープライズ・アプリケーションを実行するために必要な機能を共有することになり、そのため「もっとも脆弱な」システムが出来上がります。残念ながら、もっとも脆弱なリンクに基づいて全体のレベルが決まります。

エンタープライズSaaS：データが王様となる領域

消費者向けインターネットアプリとほぼ同時期に、エンタープライズ・クラウド・アプリケーション、通称SaaS (Software-as-a-Service) は、インターネット上で配信されるエンタープライズ・アプリケーションとして構築されました。消費者向けのインターネット・サービスとは異なり、企業向けSaaSアプリケーションは、重要なビジネス・プロセスをサポートするために比較的複雑なアプリケーション・ロジックを採用していました。多くの場合、既存のエンタープライズ・システムに接続する必要がありました。加えて、SaaSアプリは非常に高い拡張性を必要とします。また、SaaSアプリを使用するビジネスでは、アプリを通して作成されたデータを、アプリ自体と全く同じように重視しました。厳格なトランザクションの一貫性とデータの整合性の制約は譲れませんでした。

その結果、企業向けSaaSベンダーはリレーショナル・データベースに固執し、それを構築したOracleなどのベンダーは、以下のような要求を満たすように新たなレベルに押し上げられました。

1. 非常に大規模な場合の高可用性と整合性。アプリケーションにはACIDトランザクションが必要であり、データは正確でなければならず、システムの可用性を維持する必要がありました。
2. *Bitbucket*よりもデータベース管理システム (DBMS) を優先する。リレーショナル・データベースは、複雑な問合せとアプリケーション・ロジック、加えてレポーティングをサポートしました。エンタープライズ・データが分析やレポートに利用できることは不可欠でした。しかし同時に、エンタープライズSaaSベンダーは、新しいデータモデルやワークロードにも対応する必要がありました。
3. 必要に応じてモノリスやマイクロサービスを利用する。企業向けSaaSベンダーは、純粋にマイクロサービスベースのアプローチを使用する代わりに、アプリケーション・コードの大部分をモノリスのままにして、モバイル・インターフェイスなどの機能を追加したり、分離されているが重要な部分のパフォーマンスを向上させるためにマイクロサービスを利用する設計を採用していました。

両立の鍵：オラクルの“集約型” コンバージド・データベース

デジタル・トランスフォーメーションを推進する企業は、インターネットの消費者向けサービスや企業向けクラウドアプリと同じ課題に直面していますが、新しいシステムと既存のシステムを確実に連携動作させるという負担がそこに加わります。

企業は、大手インターネット企業と同じように、消費者向けのモバイル・アプリやウェブ・アプリを迅速な反復と柔軟性を持って作成する必要があります。また、商用SaaSプロバイダーと同様の敏捷性を持って複数の部門にITサービスを提供しなければなりません。さらに、既存の企業システムに新しい環境からのデータを提供したり、データを受け取ることも必要です。

ただし各環境が、運用、セキュリティ、パフォーマンスのプロファイルが異なる別々の単一目的データベース上に構築されている場合、極めて難しくなります。その代わりとして、これらのアプリ、分析、AIアルゴリズムのすべてをサポートする統一データ層が必要です。このため、データ管理における“集約”イノベーション、つまりコンバージド・データベースが必要となります。

コンバージド・データベースは、マルチモデル、マルチテナント、マルチワークロードのデータベースです (図1)。各開発チームが必要とするデータモデルとアクセス方法をサポートし、不要な機能が邪魔になることはありません。異なるチームが、望んではいなくても考えたくない、統合と分離の両方を提供します。また、それらのチームが必要とするすべてのワークロード (OLTP、分析、IoTなど) に対応します。Oracle Database 19cは世界初のコンバージド・データベースです。



マルチモデル

複数のタイプのデータモデルと、クロスモデル方式とモデル固有方式などのアクセス方式をネイティブでサポート

マルチテナント

データベースの統合、分離、俊敏性を実現するデータベース・コンテナ・アーキテクチャ

マルチワークロード

単独または連携動作し、あらゆるタイプのデータベース・ジョブにわたって並外れたコストパフォーマンスを生み出すさまざまなソフトウェアの最適化手法

図1: Oracle Database 19cは世界初のコンバインド・データベース

強力な相互効果

コンバインド・データベースの良い例として、スマートフォンがあります。スマートフォンは、電話、メッセージング、カメラ、カレンダー、音楽その他の機能を1つの製品に統合したもので、それぞれが当初は別々の製品を必要としていました。今では、これらの単独製品はスマートフォンの一機能にすぎません。スマートフォンは、小型化、低消費電力、拡張現実感、バーチャルリアリティ、3Dなどのユニークな機能を実現するために、大量生産されるカスタム集積回路を改良し続けています。スマートフォンは、機能同士の相乗効果により、全体として、単なる部品の集合体としてそれ以上の価値を生み出しています。これは、緊密な統合と、それによって可能となる新しいワークフローのおかげです。

たとえば、スマートフォンのカメラは他の多くのアプリケーションと統合されています。写真の保存とバックアップ・プロセスが自動化され、電子メールやメッセージで写真を送信したり、人気のあるソーシャルメディアのサイトに簡単に投稿したりすることができ、リアルタイムで自動編集や色補正を行うこともできます。カレンダーはスマホのインターネット接続を利用してクラウドと同期し、継続的に更新されます。音楽アプリでは、クラウドの膨大な音楽ライブラリから、音楽を連続してストリーム再生できます。これらの個別の各機能は、同等の機能を持つ単独の単一目的の製品よりも、ある意味で高機能かつ高性能です。高価な単一目的のハイエンド・カメラやステレオ・システムは、スマートフォンのカメラやミュージック・アプリよりも高品質な写真やサウンドを提供することができますが、スマートフォンはこうしたハイエンド・システムと比較して規模の経済性を実現しているため（数百万単位のボリュームがあるのに対して、ハイエンド・カメラの場合はわずか数千単位）、長期的にはその差を縮めることができます。

スマートフォンから得られる使いやすさ、利便性、相乗効果は、コンバインド・データベースにも当てはまります。コンバインド・データベースを使用すると、アプリケーションの開発ははるかに簡単になります。複雑なメッセージングやイベントコードを書いてデータを結合する代わりに、JOINのような標準SQLの機能を使用することができます。開発者がSQLを使用せずにネイティブ・データ型に直接プログラムしたい場合、オラクルはこれをサポートするJSONやグラフなど最も一般的なデータモデル用の豊富なAPIを提供しています。

相乗効果とダイナミックな新しいワークフローもさることながら、オラクルのコンバインド・データベースに組み込まれている機能は、多くの場合、単一目的のデータベースの機能より優れています。オラクルのコンバインド・データベースは、空間、グラフ、JSON、機械学習機能など、最も完全で最高品質の機能セットを提供します。業界のアナリストもこのことに気づいており、ニッチな単一目的のデータベースの焦点とされるすべての分野でオラクルのコンバインド・データベースが第1位にランク付けされています。オラクルのコンバインド・データベースを採用すれば、各データモデル、ワークロード、または開発パラダイムの特定の要件を満たすために劣った機能を受け入れるよう強要されることはありません。

マルチモデル：単独エンジン、多くの特徴

まずマルチモデルについて考えてみましょう。異なるデータモデルとそのアクセス方法を単一のエンジンでサポートすることで、開発者の生産性とデータの生産性の両方をどのように向上させることができるのでしょうか。

1Gartner 『Critical Capabilities for Operational Database Management Systems』 <https://www.gartner.com/document/3975496>

クロスモデル・データ・アクセス

モデル固有データ・アクセス

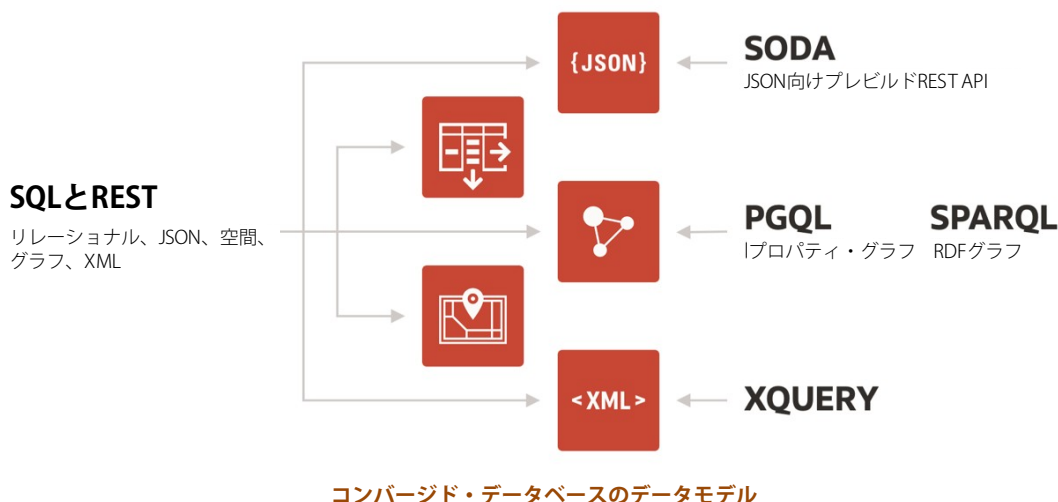


図2：複数のデータモデルとアクセス方式に対応するオラクルのコンバージド・データベース

単一目的データベースとは異なり、オラクルのコンバージド・データベースは、JSON、XML、リレーショナル、空間、グラフ、IoT、テキスト、ブロックチェーン・データをサポートし、完全な結合、トランザクション、そして企業が信頼するその他のクリティカルなSQL機能を備えています（図2）。加えてオラクルのコンバージド・データベースは、グラフおよび空間問合せのためのモデル固有のアクセス方式や、何百もの汎用的な機械学習アルゴリズムもサポートしています。これらの機能は、ステートフル接続だけでなく、RESTful APIからもアクセスでき、開発者の選択に委ねられています。

オラクルのコンバージド・データベースは、Webおよびモバイル・アプリケーションで一般に使用されるJSONデータをサポートしている点で、さらに優れています。Simple Oracle Document Access (SODA) は、あらかじめ構築されたRESTful APIのセットで、開発者はSQLを使用せずにJSONコレクションを作成して問合せを行うことができます。ただし、単一目的ドキュメント・データベースとは異なり、Oracle Databaseでは、JSONオブジェクトからスキーマと索引を生成して、JSONデータと空間データ、グラフデータ、リレーショナルデータとの並列SQL分析、トランザクション、結合を可能にします。実際、世界最大級の家電メーカーの1社は、Oracle DatabaseのJSONの機能を使用して世界中のPOSトランザクションに対応しています。

これらのマルチモデル機能により、企業は今すぐに開発者の生産性を高め、後からデータの生産性を向上させることができます。これにより、開発者はシンプルなAPI主導のアクセスとモデル固有の言語を利用できるようになりますが、必要なときにいつでも強力なSQL機能を利用することができます。一方、IT部門は、オラクルのコンバージド・データベースを導入する際にはいつでも、セキュリティ、アップグレード、パッチ適用、メンテナンスについて共通の方法で対応できます。

マルチテナント：統合、分離、俊敏性

コンテナは、今日のアプリにおけるもっとも強力なアイデアの1つで、共有UNIXマシン上の個々のプロセスを分離する方法として1979年に登場したのが最初です。コンテナ化することにより、より分離性を高められるだけでなく、コンテナのオーケストレーションと組み合わせることにより、基盤となるリソースを仮想化し、効率性、俊敏性、可搬性を高めることができます。

ただし、OSのコンテナ化がアプリのコンテナ化と若干意味が異なるのと同じように、データベースのコンテナ化も意味が異なります。

不要になったら削除可能なコンテナ化アプリとは異なり、データベースではデータを持続させる必要があります。データを永続的に保存し、後で使用できるようにすることは、データベースの仕事の中心的な部分です。では、コンテナ機能をデータ層にどのように取り込むのでしょうか。独自のマルチテナント機能を備えたオラクルのコンバージド・データベースでは、データ層自体にコンテナ化とオーケストレーションが組み込まれています。

オラクルのコンバージド・データベースのマルチテナント・アーキテクチャにより、単一のコンテナ・データベース上で、複数のプラグブル・データベースをサポートできます（図3）。Oracleコンテナ・データベースは、アプリ・コンテナ・エンジンが複数のアプリケーション・コンテナをサポートするのと同様に、複数のプラグブル・データベース（またはデータ・コンテナ）をサポートするという点で、アプリ・コンテナ・エンジンに似ています。

統合

- アプリまたはサービスごとの自己完結型プラグブル・データベース
- コンテナ・データベース・レベルでの共通操作（バックアップ、アップグレードなど）



分離

- ロックダウン・プロファイル
- 透過的暗号化
- リソース分離
- 必要に応じて、コンテナ・データベースあたり1つのプラグブル・データベース



俊敏性

- 高速プロビジョニング
- パブリック・クラウド、ローカル・クラウド、オンプレミスにわたるオンライン再配置

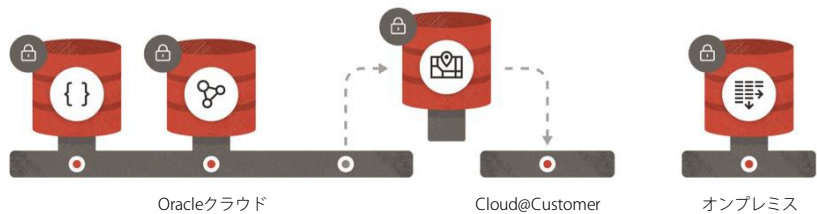


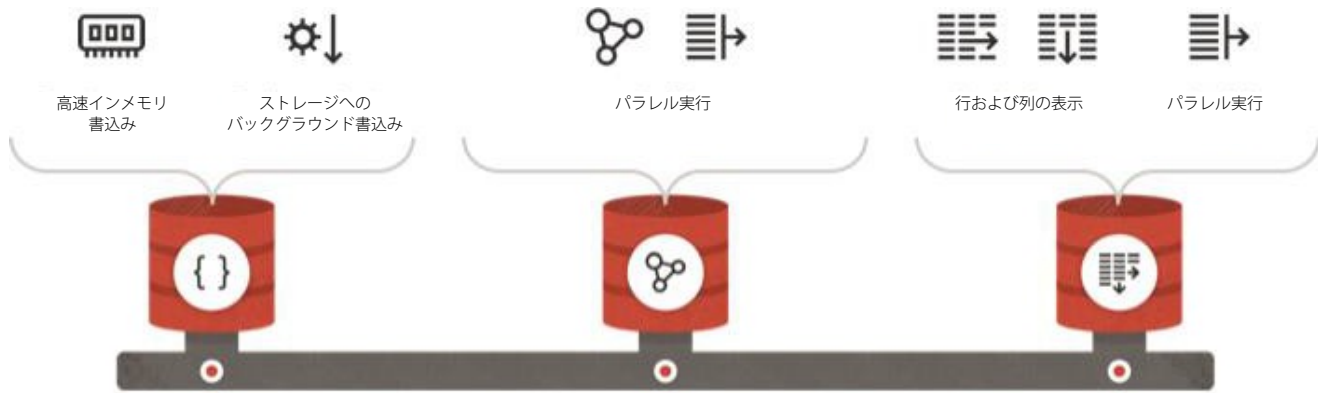
図3：オラクルのコンバージド・データベースによるデータ層でのコンテナ化

Kubernetesなどのアプリケーション層コンテナ・オーケストレーション・フレームワークのように、オラクルのマルチテナント・クラウド・アーキテクチャでは、コンテナ・データベースを利用して、統一データ層でプラグブル・データベースをオーケストレートします。このマルチテナント・アーキテクチャにより、オラクルのコンバージド・データベースは次のことを実現します。

- **技術面の効率**：複数のデータベースをそれぞれ分離されたワークスペースでサポートしつつ、共通のインフラストラクチャ部を共有します。これにより、冗長で重複したオーバーヘッドが排除され、1台のサーバーあたりのデータベース数が増えます。
- **運用面の効率**：共通環境の管理コストは、複数のデータベース間で分割して負担し、多くのデータベースを1つのデータベースのようにして効率よく管理できます。
- **俊敏性**：新しいデータベースのプロビジョニングや既存データベースのクローニングが簡単に行うことができ、統合や分離、あるいはパフォーマンス向上に向け他のプラットフォームに再デプロイする事も簡単です。
- **使いやすさ**：データベース（とそのスキーマ）は変更なしで動作します。スケーラビリティは透過的です。集中管理によって複雑さが緩和されます。シンプルなSQL拡張機能により、迅速なプロビジョニングとクローニングなどの新しいマルチテナント機能を制御します。

オラクルのコンバージド・データベースのマルチテナント・アーキテクチャは、マイクロサービスごとにプラグブル・データベースを提供することで、マイクロサービスの開発と導入を容易にします。各プラグブル・データベースは、分離・保護され、まさに特定のマイクロサービスが必要とするデータ型またはワークロードを提供します。オペレーション・チームがコンテナ・データベースを介して多くのデータベースを1つのデータベースとして管理する場合でも変わりません。

マルチテナント・アーキテクチャを活用した統一データ層の効率性の向上と管理の簡素化により、開発者の生産性が向上します。開発者は、データ管理タスクよりもアプリケーション開発に集中することができます。加えて、統一層のプラグブル・データベースには、データ・サイエンティストやビジネス・アナリストが好みのツールとアクセス方式を通じて容易にアクセスすることができ、レポートを生成したり、分析を実行したり、AIモデルを構築したりすることができ、データの生産性を高めることができます。



モノのインターネット

スマート・センサーなどからのストリーミング・データをメモリ最適化書込み

機械学習

グラフデータに対する機械学習アルゴリズムをパラレル実行

リアルタイム分析

インメモリで行と列を管理、トランザクション・データに対するリアルタイム分析をパラレル実行

図4：オラクルのコンバージド・データベースによるさまざまなデータ・ワークロードの広範な最適化

マルチワークロード：一度に様々なジョブを実行

十分なパフォーマンスを維持するためには、データベースのワークロードの種類に応じた様々な方法でソフトウェア最適化が必要です（図4）。たとえば、頻繁に測定を行うスマート・センサーには、大量の新規レコードを極めて素早く取り込むことができるデータベースが必要です。オラクルのコンバージド・データベースは、IoTアプリがそのデータを直接メモリ・バッファに書き込むと同時に、別のバックグラウンド・プロセスでデータベースがそれらのレコードをディスクにコミットする仕掛けで対応します。この機能により、データがディスクに保存されるまでの待ち時間をなくし、2ソケット・サーバーで毎秒2500万回の挿入を処理します。

一方、機械学習モデルのトレーニングは、まったく異なる種類の仕事です。これには、比較的単純な計算を非常に大量に行う必要があり、多くのCPUサイクルを消費します。オラクルのコンバージド・データベースには、数百もの汎用的な機械学習アルゴリズムが組み込まれています。アルゴリズムは、多くのCPUコア間で並列化できるように変更されているため、この種の計算を劇的に高速化することができます。

時には、開発チームは1つのコンバージド・データベース上で複数の異なるワークロードに同時に対応する必要があります。たとえば、最新のトランザクション・アプリケーションでは、業務系データについて何らかの分析をリアルタイムで実行することがもとめられます。オラクルのコンバージド・データベースでは、行形式（高速トランザクション処理用）と列形式（高速分析用）の両方で同じデータを表し、それら2つをデータ変更と完全に同期された状態に維持することで、両方のワークロードを舞台裏で最適化します。

コアデータベースエンジンが、過度なカスタマイズやアプリのチューニングを必要とせずに、さまざまな種類のアプリケーションのワークロードを完全にサポートすることができれば、開発者の生産性は向上します。同様に、同じコアエンジンがデータ・サイエンティストやビジネス・アナリストにとって重要なさまざまな分析ワークロードも完全にサポートすることができれば、データの生産性が向上します。

オラクルのコンバージド・データベースを使用すると、DevOpsチームは、アプリケーションに変更を加えることなく、さまざまな最適化のオン/オフを切り替えることができます。これにより、データベースでの要求が増加した場合や、要求の種類が変化した場合でも、パフォーマンス・チューニングが可能でます。

こうしたソフトウェア最適化はすべて、物理的なコンピューティング・リソースが関係しています。プロセッサ、メモリ、ディスク、そしてそれらのコンポーネント間の入出力などです。多くの企業では、ブレード・サーバーとネットワーク接続されたストレージを組み合わせた高機能なシステムを構築し、オラクルのコンバージド・データベースに並外れた機能をサポートするための動作環境を提供してきました。しかし、もっと優れた方法があります。それはOracle Exadataです。



ソフトウェアだけでは 不可能なスケールアウト、 スケールアップ アーキテクチャ

最適なハードウェア構成

データベース・サーバー、ストレージ・サーバー
ストレージからコンピュータを分離し、独立してス
ケーリングできるようにする

ストレージでの階層化キャッシング
永続メモリ、フラッシュ・メモリ、ディスクを使用
し、アクセス頻度に基づいてデータを階層化する

ネットワークを介したダイレクト・メモリアクセス
データベース・サーバーは、ストレージ・サーバーの
メモリへのダイレクト・アクセス権限を持ち、一般
的なネットワーク接続ストレージのボトルネックをバ
イパスする

スマート・ストレージ・ ソフトウェア

スマート・スキャン
SQL処理をストレージ・サーバーに自動的にオフロー
ドする

自動形式最適化
分析の場合はHybrid Columnar Compressionに、トラ
ンザクションの場合は行形式でデータを保存する

自動ストレージ索引
列ごとに複数の索引を作成し、不要データの処理を
回避する

自動データ階層化
フラッシュ・メモリ、永続メモリ、ディスクの全体
にわたって使用頻度別にデータに優先順位付けする

図5：Oracle Exadataによりハードウェアとソフトウェアのイノベーションを組み合わせ、あらゆるデータ・ワークロードに対応可能な卓越したコストパフォーマンスを実現

Oracle Exadata

Oracle Exadataは、データベースのワークロードを処理することに特化して設計されたハードウェアとソフトウェアの組合せ製品です（図5）。コンピュータとストレージの機能を分離したアーキテクチャを採用することで、Oracle Exadataは、データベース・ワークロードで業界最高レベルのコストパフォーマンスを備えたモジュール式のスケールアウト/スケールアップ・ソリューションを提供します。Oracle Exadataが他のすべてのデータベース・アーキテクチャを凌駕している理由は、オラクルの共同開発プロセスにあります。最新のハードウェア・イノベーションから最後の一滴にいたるまでパフォーマンスを絞りだすよう既知のアーキテクチャに組み込み、データベース・ソフトウェアを最適化しています。

ストレージ・サーバーにIntel Optane Persistent Memoryを採用しているOracle Exadata X8Mは、このプロセスの好例です。この永続メモリは、通常のダイナミックメモリのように極めて高速に動作しますが、電源が切れてもディスクのようにデータを持続します。つまりデータベース・ソフトウェアを変更して永続メモリを活用することができれば、パフォーマンスを飛躍的に向上させる機会となります。

データベース・サーバーとストレージの間の往復に要する平均ネットワーク待機時間により、永続メモリによって得られる速度の利点が失われかねません。そこで、Oracle Exadata X8Mでは、ストレージ・サーバー上のメモリにキャッシュされたデータを利用するために、Random Direct Memory Access over Converged Ethernet（RoCE）を使用し利点を引き出しています。これは、Oracle Exadataをあらゆるデータベース・ワークロードに対応する最もコスト効率の高いプラットフォームにしている数千ものハードウェアとソフトウェアの連携のうちの1つにすぎません。

オラクルのコンバインド・データベースをOracle Exadataインフラストラクチャ上でご利用いただく場合は、Oracleクラウドのパブリック・クラウド・サービス、Oracle Exadata Cloud Service、Oracle Exadata Cloud@Customer、Oracle Dedicated Regionを介して、あるいはオンプレミスのOracle Exadataが選択可能です。

Oracle Autonomous Database：サービスとしてのコンバインド・データベース

オラクルのコンバインド・データベースの究極の姿はOracle Autonomous Databaseです。これは、Oracle Exadataインフラストラクチャ上で動作するオラクルのコンバインド・データベースで、フルマネージド・クラウド・サービスです。以下の理由で、Oracle Autonomous Databaseの管理作業は不要です。

- **自己稼働。** Autonomous Databaseに達成すべきサービス・レベルを指示すれば、あとはAutonomous Databaseが処理します。Autonomous Databaseは、データベースのプロビジョニング、セキュリティ保護、監視、バックアップ、リカバリ、トラブルシューティング、およびチューニングを自動化します。これにより、データベースのメンテナンス・タスクの手間が大幅に減り、コストが削減され、管理者のリソースをより価値の高い作業に充てることができますようになります。
- **自己保護。** Autonomous Databaseは、自己保護機能があるため、従来のデータベース導入よりも安全です。この保護は、外部攻撃と内部攻撃の両方に対する防御に対応します。

- **自己修復。** Autonomous Databaseは、従来のデータベース導入よりも信頼できます。起動時には、1つのリージョンのクラウド・データセンターで三重にミラー化されたスケールアウト構成を自動的に構築し、別のリージョンに完全なスタンバイ・コピーを構築することも選択可能です。Autonomous Databaseは、サーバーまたはデータセンター・レベルでの物理的な障害から自動的に回復します。ユーザーの誤操作からデータを守るため、過去のある時点に戻す機能も備えています。ソフトウェア更新をクラスタの全ノードでローリング方式で適用することにより、データベース、クラスタウェア、OS、VM、ハイパーバイザ、ファームウェアなどの更新中もアプリケーションはオンラインのまま維持されます。

Oracle Autonomous Databaseは、開発者、データ・サイエンティスト、アナリストが本当に関心のあるアプリケーションやAI、アナリティクスに取り組むための時間を節約できるように、いくつかの特徴的な設定で提供されています。Oracle Autonomous Databaseのサービス・ファミリーには、以下が含まれます。

- **Autonomous Transaction Processing (ATP)** リアルタイムまたはバッチでの分析を必要とするOLTPと混合ワークロードのデータベース操作を簡素化します。ATPにより、ランタイム・コストを最大90%削減し、組込み機械学習ベースの自動化により比類ない規模、パフォーマンス、セキュリティを提供します。
- **Autonomous Data Warehousing (ADW)** 分析処理用に最適化されています。コンピューターやストレージを自動的にスケールアップし、高速の問合せパフォーマンスを実現し、オブジェクト・ストレージやKafkaストリームなど、他のクラウド・サービス上のデータウェアハウスのデータを問い合わせることができます。
- **Autonomous JSON Database (AJD)** JSON中心のアプリ開発をサポートするように調整されています。高度な索引付けと透過的なスケールアウトを備えたネイティブなJSON管理機能に加え、JSONドキュメント上での完全なACIDトランザクションを提供します。これらのすべての機能は、REST APIを介して使用可能です。

オラクルのコンバインド・データベースによって実現される統一データ層

オラクルのコンバインド・データベースは、パブリック・クラウド、ローカル・クラウド、オンプレミスなど、さまざまな導入シナリオで利用できます。さまざまな利用環境ごとに最適なメカニズムで対応可能な同一のコアエンジンを活用することで、企業は選択の自由度を維持しながら、統一されたデータ層を構築することができます (図6)。この集約されたデータ層を利用することで、企業は以下のことが可能になります。

- **マイクロサービス・アプリとモノリシック・アプリの両方を作成する。** オラクルのコンバインド・データベースは、マイクロサービス・アプリケーションをサポートする新しい方法です。各サービスに必要な、独自のモデルとアクセス方法を備えたデータベースがあります。これらのデータベースはそれぞれ、リアル・アプリケーション・クラスタ (RAC) のような洗練された方法や、シャーディングのような比較的単純な方法を使い、必要に応じてスケールアップまたはスケールアウトすることができます。これらのデータベースはコンテナ・データベースのテナントとして、同一のセキュリティ・モデル、パッチ適用、アップグレード・パスにより一元管理できます。もちろん、同じエンジンを持つ他のインスタンスでは、従来のエンタープライズ・アプリケーションの構築や実行を、これまで通りに行うことができます。
- **パブリック・クラウド、ローカル・クラウド、またはオンプレミスでワークロードを実行する。** Oracle Exadataインフラストラクチャは、オラクルのコンバインド・データベースを3つすべてのコンピューティング環境で全く同じデータ層アーキテクチャを実現する唯一の方法です。このアーキテクチャは、Oracle Cloudでは、Oracle Autonomous DatabaseおよびOracle Exadata Cloud Serviceとして利用できます。ローカル・クラウドでは、Oracle Exadata Cloud@Customerとして、また新しいOracle Dedicated Cloud Region製品の一部として利用できます。さらにオンプレミスでは、Oracle Exadataとして利用できます。
- **業務系データや、データベースの外部データに対し、分析やデータ・サイエンスを実施する。** オラクルのコンバインド・データベースのマルチワークロード機能により、本番アプリケーションをサポートしている業務系データベースのデータに対して、分析問合せを実行することが可能になります。つまり、本番データをリアルタイムで分析できます。さらに、Autonomous Data Warehouseのインスタンスを使うと、アナリストはOracleクラウドだけでなく他のクラウドのオブジェクト・ストレージに存在するデータについて、問合せができます。
- **多数のデータベースを1つのデータベースとして管理する。** 運用チームが管理しなければならないすべてのデータベースインスタンスが異なる構成であるものの同じコアエンジンであれば、フリート管理ははるかにシンプルになります。フリート全体が同じセキュリティ・プロトコル、パッチ適用方法、およびアップグレード・パスに従います。Oracle Autonomous Databaseを使用すると、フリート全体が同じ管理方法に従うだけでなく、独自に管理を行うことができます。

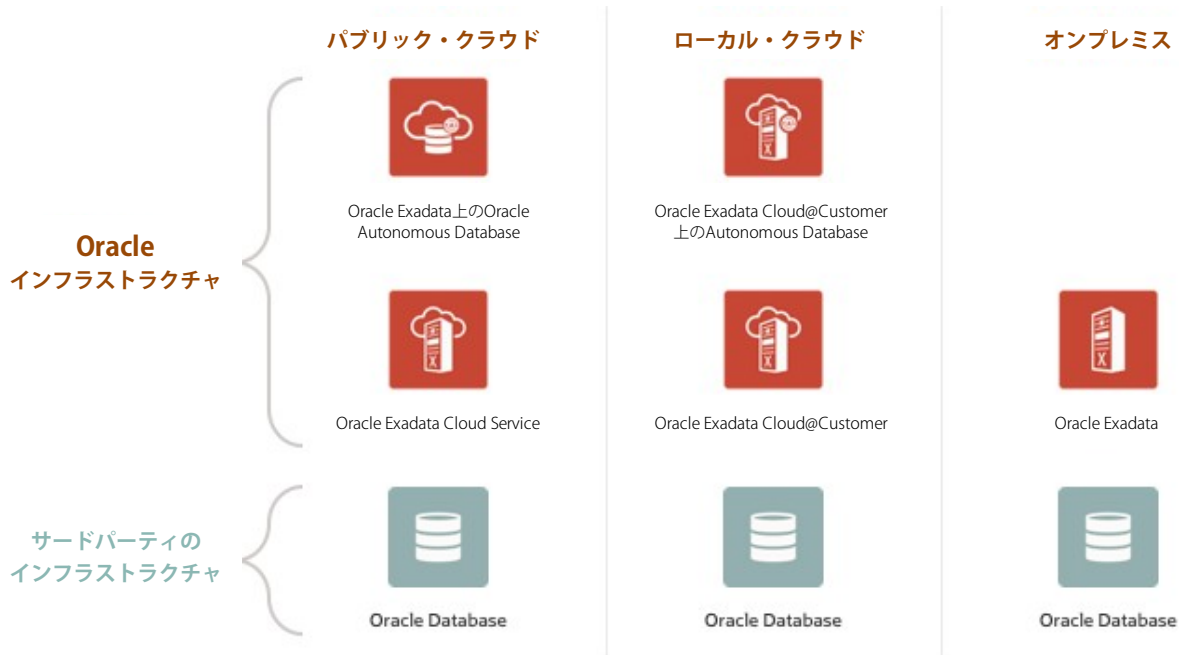


図6：企業は、さまざまなシナリオでオラクルのコンバージド・データベースを使用して、デプロイメント環境を横断する統一データ層を作成できる

結論

オラクルのコンバージド・データベースは、Oracle Databaseを徹底的に考え詰めた製品です。マルチモデル、マルチテナント、マルチワークロードのこのデータベースにより、企業は一見相反する2つの目標を満たすことができます。すなわち、開発者には各業務に適したツールを提供して生産性を高め、IT部門には統一されたデータ階層を提供してデータの再利用と全体的な管理をよりシンプルかつ容易にします。これは企業にとって、デジタル・トランスフォーメーションのシンプルな基盤となります。オラクルのコンバージド・データベースを使用することで、企業はより多くのビジネス・プロセスをより迅速にデジタル化し、より多くのデータをより自信を持ってより多くの意思決定ポイントに投入することができます。

オラクルの情報を発信しています

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

 blogs.oracle.com

 facebook.com/oracle

 twitter.com/oracle

Copyright © 2020, 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。

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

オラクルのコンパージド・データベース：開発者とデータの生産性を向上させる方

法 Matthew O'Keefe, Maria Colgan, Paul Sonderegger

2020年10月

