

新しい世代の Oracle Real Application Clusters (Oracle RAC)

2024年8月、バージョン1.2

Copyright © 2024, Oracle and/or its affiliates 公開

概要	3
データベース設計の一部としての高可用性とスケーラビリティ	4
Oracle RACコンポーネントの概要	4
Oracle RACのスケール・アウト・アーキテクチャ	5
Oracle Clusterware	6
Oracle ASM	6
アプリケーション・コンティニューイティ	7
Fleet Patching and Provisioning	7
新しい世代のOracle RAC	8
Oracle Real Application Clusters 19c	8
エンジニアド・システム – Oracle RACを意識した設計	9
Exadataのスケール・アウト・アーキテクチャ	10
Oracle RACに役立つExadata拡張機能	10
Oracle RACの利点をOracle Cloudに拡大する	11
Oracle RACとAutonomous Database	11
Oracle RACとその他のスケール・アウト・データベース・アプローチの比較	13
まとめ	15

概要

企業が世界で成功を収められるか否かは、ミッション・クリティカルなアプリケーションの動作が高速で安定しているかどうかにかかっています。こういった重要なアプリケーションで可用性やスケーラビリティの問題が発生すると、ビジネスに支障をきたしたり、ビジネスが止まったりします。アプリケーションの可用性と応答性は、基盤となるデータベースの信頼性と応答性にかかっています。オラクルは、ミッション・クリティカルなアプリケーションに必要な透過的なデータベースの可用性とスケーラビリティを実現するため、Oracle Real Application Clusters (Oracle RAC) テクノロジーの技術開発および強化に長年、投資しています。このテクノロジーは業界に唯一であり、世界の大手企業のほとんどで、社内アプリケーションと顧客向けの重要アプリケーションの両方の実行に使われています。

このホワイト・ペーパーでは、Oracle RACについて説明し、これが、重要なアプリケーションに欠かせない以下の要件をどのように満たしているかについて解説します。

1. **高可用性** – 重要なアプリケーションを支えるデータベースは、ハードウェア障害やソフトウェア障害が起きた場合でも、アプリケーション・リクエストに応答し続けなければなりません。また、アプリケーションを中断することなく、ハードウェア、OS、データベース・ソフトウェア、データベース・スキーマといったあらゆるレベルの計画メンテナンスを実施できる必要があります。
2. **ワークロードのスケーラビリティ** – 重要なアプリケーションを支えるデータベースは、中断することなく、増加するアプリケーション・ワークロードに動的に適応できなければなりません。コンピューティング、ストレージ、メモリ、接続、ユーザー、アプリケーションの複雑さを透過的にスケーリングできる必要があります。
3. **アプリケーションの変更不要** – 重要なアプリケーションを支えるデータベースは、アプリケーションを変更することなく、上記の要件を満たさなければなりません。企業は数千年に匹敵する工数をかけてアプリケーションを開発しており、既存のアプリケーションを書き換えるのは、単純に費用対効果に優れていません。

Oracle RACは、長い間、こういった重要な要件に対応する最高水準のテクノロジーであり続けています。アプリケーションのコードを変更することなく、最大100ノードのクラスターでほぼ直線的なスケーラビリティと可用性を実現します。Oracle Exadata Database Machineと組み合わせれば、Oracle RACデータベースは、世界最大規模でもっとも要求が厳しい、OLTPや分析のワークロードを効率的に実行できます。

Oracle Database 19cでは、Oracle RACアーキテクチャの大幅な改善が行われており、アプリケーションの可用性とスケーラビリティが、わずか数年前に比べて何倍にも向上しています。こういった改善を組み合わせることで、新しい世代のOracle RACテクノロジーが生まれ、次世代のワークロードやアプリケーションのニーズに対応できます。

データベース設計の一部としての高可用性とスケーラビリティ

オラクルは当初から、高可用性とスケーラビリティの概念を構築してOracle Databaseの設計に組み込んできました。これには、ACID準拠データベース、健全なバックアップとリカバリ、レプリケーション・ソリューション（Oracle Data Guard）の提供に加え、Oracle RACをローカルの高可用性およびスケーラビリティ・ソリューションとして提供することが含まれます。

とりわけOracle RACでは、高可用性とスケーラビリティをデータベース設計に組み込むことで、コンバージドなOracle Database¹および任意のデータベース・アプリケーションのすべての機能において、何も変更することなく、これまでにないスケーリング能力を実現できました。以下に例を示します。

- Oracle RACであらゆるOracle Databaseの機能とアーキテクチャをスケーリング：
 - Oracle Multitenant Databaseと非CDBデータベース
 - Oracle Parallel SQLによる分析やバッチ処理の高速化
 - クラスタのインスタンス全体のOracle Database In-Memory
 - Multi-Instance-Redo-ApPLY（MIRA）を使用したOracle Data Guard
- Oracle RACで複雑なOLTP、DWH、分析ワークロードをスケーリング：
 - 通常、Oracle RACで動作するSAP、Oracle EBusiness Suite、Peoplesoft、Siebelなどの多くのビジネス・アプリケーション
 - Oracle RACで動作する15,000超の顧客ワークロード
- Oracle RACでクラス最高の高可用性を持つ24時間/365日稼働するシステムを実現：
 - Fortune 1000の銀行、通信企業、航空会社、E-Commerce企業の大半が重要なアプリケーションにOracle RACを使用
 - Oracle RACにより、ほとんどのハードウェア障害やソフトウェア障害、メンテナンス・イベントの影響を受けることなく、データベースの継続可用性を実現

このペーパーの以降の部分では、Oracle RACがファンクション・ SHIPPINGと分散キャッシュを組み合わせ使用して業界屈指のスケーラビリティと可用性を実現する方法について説明します。また、来るべき新しい世代のアプリケーションとDBAに向けた高可用性とスケーラビリティを実現するために、最新バージョンのOracle RACでどのような改善が行われているかにも触れます。

Oracle RACコンポーネントの概要

Oracle RACは、アプリケーションに透過的なスケール・アウト・データベース・アーキテクチャを実現します。アプリケーションは、異なるサーバーにホストされた最大100データベース・インスタンスからなるクラスタを構成する任意のデータベース・インスタンス（Oracleデータベースを実行する際に使用するサーバーのプロセスとメモリのコレクションとして定義されます）に接続します。すべてのサーバーが、実際のデータ（一連のデータベース・ファイル）を保持する共有ストレージ・システムにアクセスできる必要があります。サーバー間は、専用の高速ネットワークで接続されている必要があります。通常、このネットワークはOracle RACインターコネクトと呼ばれます。

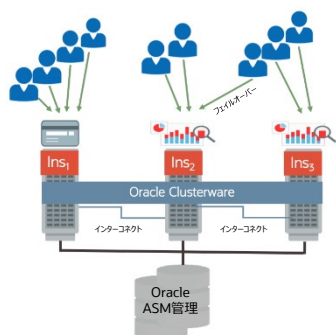


図1：Oracle RACアーキテクチャの概要

¹ オラクルのコンバージド・データベース・アーキテクチャについて詳しくは、こちらを参照してください。 <https://youtu.be/9d76-LhgMQs>

インターコネクトはこのアーキテクチャの重要な要素で、スケール・アウトしたOracle RACデータベースが、アプリケーションの観点から“1つのシステム”に見えるようにします。このシステムでは、すべてのデータベース・インスタンスが協調的な一貫性のある形ですべてのデータにアクセスし、更新できます。つまり、アプリケーションは、どのデータベース・インスタンスがワークロード・リクエストを処理しているかを意識する必要はありません。また、Oracle RACは次の5つの重要なソフトウェア・コンポーネントを活用します。

1. **スケール・アウト・データベース・インスタンス**。複数のデータベース・インスタンスがデータベースのデータに同時アクセスできるだけでは十分ではありません。協調的にアクセスして更新できるインテリジェンスが必要です。データベース・レイヤーでこのインテリジェンスを提供するのが、オラクルならではの**キャッシュ・フュージョン・テクノロジー**です。これは、クラスタのノード全体で一貫性のあるデータの分散キャッシュを実現するアルゴリズムで、効率的な水平スケーリングに役立ちます。
2. **Oracle Clusterware**は、サーバーのコレクションを高可用性クラスタに変えます。管理上の視点では、クラスタは独立したサーバーのプールとしてではなく、1つのエンティティとして管理されます。また、Oracle Clusterwareは、クラスタの障害検出、フェイルオーバー、ノード・メンバーシップ管理を提供します。
3. **Oracle Automatic Storage Management (Oracle ASM)** は、個々のOracle RACデータベース・インスタンスすべてからデータベース・ファイルに同時アクセスできるようにすることで、ストレージをスケーリングします。また、Oracle ASMは、データベース・ファイル・アクセスの最適化や、ストレージ障害から保護するデータ・ミラー化といったボリューム管理機能を提供します。
4. **アプリケーション・コンティニューイティ (AC)** は、部分的に実行された処理中のリクエストやトランザクションを中断することなく高速に再実行することで、アクティブなアプリケーション接続を、存続しているインスタンスに透過的に自動フェイルオーバーします。
5. **Oracle Fleet Patching and Provisioning (Oracle FPP)** は、以前のOracle RACにおける課題の1つだったパッチ適用とプロビジョニングを大幅に自動化および簡素化します。Oracle Fleet Patching & Provisioning (旧称Oracle Rapid Home Provisioning) は、プロビジョニング、パッチ適用、アップグレードをオーケストレーションし、アプリケーションにとってのデータベース可用性を最大化します。

Oracle RACのスケール・アウト・アーキテクチャ

データベース・システムを複数のサーバーに水平スケーリングする方法は、基本的に2つあります。1つ目は、データにアクセスするSQLを実行するサーバーにデータを移動させること（データ・シッピングまたはシェアード・ディスクと呼ばれることがあります）です。2つ目は、SQLをリモート・サーバーに送信してローカルでデータにアクセスすることです（ファンクション・シッピングまたはシェアード・ナッシングと呼ばれることがあります）。

Oracle RACは、データ・シッピング・シェアード・ディスク・アーキテクチャと言われることが多いですが、これは誤りです。実際のOracle RACは、前述の2つのアプローチを組み合わせることで、両方の長所を活かしたスケール・アウト・アーキテクチャを実現しています。

Oracle RACは、長時間実行される分析やバッチのSQLに対して、ファンクション・シッピング（シェアード・ナッシング）を使用します。Oracle RACは、大量のデータを処理するSQL文を分割してSQLフラグメントにすることで、長時間実行されるSQLを自動的に並列化します。SQLフラグメントは、クラスタ内のデータベース・インスタンス全体で並列実行されます。従来のシェアード・ナッシング・データベースとは異なり、Oracle RACインスタンスは、データベース・データの特定のサブセットに拘束されることはありません。ストレージから任意のデータを読み取れるからです。そのため、Oracle RACはクラスタのノード全体にワークロードを動的に分散させることができ、1つの遅いインスタンスが処理全体の速度を低下させるような大きな偏りを回避できます。

ファンクション・シッピングは分析には有利ですが、OLTPは得意ではありません。通常、OLTPリクエストは少量の行にアクセスし、非常に短い応答時間が求められます。リモート・ノードにRPC呼び出しを送信して1つの行にアクセスしたり、1つの索引を走査したりするのは、あまりに遅くコストがかかります。また、最後のリクエスト以降にデータが変更された可能性があるため、アプリケーションが同じデータにアクセスするたびに、リモート・ノードにリクエストを再送信する必要もあります。さらに、データが存在するインスタンスにリクエストを送信するには、複雑でコストの高い分散トランザクションと2フェーズ・コミット・プロトコルを実行し、インスタンス間で協調処理を行う必要があります。そのうえ、シェアード・ナッシングは数千個の表や索引にスケーリングできないので、ERP、CRM、HCMといった複雑なアプリケーションでは利用できません。こういった欠点から、古くからの支持者でさえ、透過的なシェアード・ナッシング・データベース・アプローチをOLTPソリューションに使うことはなくなっています。現在、ERP、CRM、HCMといった複雑なOLTPアプリケーションを実行するシェアード・ナッシング・アーキテクチャは存在しません。

Oracle RACは、実行時間が短いOLTP SQLを、アプリケーション・リクエストを受信したインスタンスで直接実行します。OLTP SQLがリクエストしたデータで、すでにインスタンスのローカルにキャッシュされているものは、即座にメモリから直接アクセスされます。インスタンスのローカルにないデータは、高パフォーマンス・データ転送プロトコルを使ってストレージまたはリモート・インスタンスからインスタンスに取得した後、ローカルにキャッシュします。データをローカルにキャッシュすることで、今後同じデータにアクセスする際の速度が劇的に向上し、メッセージングの必要がなくなります。キャッシュは、多くの実績があるよく知られたアプローチで、多くの場合においてリモート・アクセスの90 %から99 %を削減し、応答時間を大幅に短縮します。トランザクションをインスタンスのローカルで実行するので、コストの高い分散トランザクションや2フェーズ・コミット・プロトコルも不要になります。

Oracle RACでは、アプリケーションがデータの場所（いずれかのデータベース・インスタンスのメモリにあるか、ディスクにあるか）を意識する必要はありません。Oracle RACプロトコルが、アプリケーションが接続しているインスタンスまで自動的にデータを運んでくれるからです（データの場所の透過性）。高パフォーマンス分散キャッシュ・プロトコルを実現しつつ、完全なデータ整合性を維持するアルゴリズムを、Oracle Cache Fusionと言います。

キャッシュ・フュージョンは、プライベート・クラスタのインターコネクトを利用して、Oracle RACデータベース・インスタンスにキャッシュされたデータに共有アクセスできるようにします。これは、クラスタ全体を対象とした1つの“融合した”キャッシュを仮想的に作成するようなものです。キャッシュ・フュージョンは、インスタンス間でデータ・ブロックをメモリ対メモリで直接転送することで、高パフォーマンスを実現します。Oracle RACデータベース・インスタンスで行う必要があるのは、Oracle RAC構成全体の結合キャッシュにブロックがまだ存在しない場合に、ストレージからデータを読み取るだけです。さらに、キャッシュ・フュージョンでは、複数のインスタンスが同じデータ・ブロックを同時に更新できないようにするロック・プロトコルが実現されています。そのため、Oracle RACキャッシュはクラスタ内の複数の独立したインスタンスに分散していますが、キャッシュ全体で確実に整合性を保ちながら、最適な同時データ・アクセスを実現できます。

Oracle RACは、ファンクション・.shipping、データ・shipping、分散データ・キャッシュを組み合わせることで、分析、OLTP、バッチ処理を他にはない形で最適化できます。また、こういったワークロードを自由に組み合わせることもできます。オラクルは最近、Oracle RACのアルゴリズムを大幅に強化し、さらにパフォーマンスを向上させています。

Oracle Clusterware

Oracle Clusterwareは、Oracle RACアーキテクチャで使われているテクノロジーで、サーバーのコレクションを可用性の高い統合システムに変えます。Oracle Clusterwareでは、障害検出、ノード・メンバーシップ、ノード・フェンシング、最適なリソース配置が提供されます。また、Oracle RACやクラスタ内の他のアプリケーションのコンポーネントの相互依存性をクラスタ全体で管理します。Clusterwareは、リソース・モデルやポリシーを使用することで、コンポーネントの計画停止時間や計画外停止時間に対する応答の可用性を高めます。

Oracle Clusterwareについて詳しくは、<http://www.oracle.com/goto/clusterware>を参照してください。

Oracle ASM

Oracle Automatic Storage Management（Oracle ASM）は、データベースに組み込まれた、ファイル・システムおよびボリューム・マネージャです。数千台規模のストレージ・デバイスを対象に、24時間/365日の可用性で、ストレージ容量の共有とスケーリングを実現します。Oracle ASMは、Oracle RACデータベースのクラスタ内のすべてのサーバーのストレージを管理します。データをストライプ化することで、ホットスポットをなくしてI/Oパフォーマンスを最大限に高めます。Oracle ASMを使用すると、ストレージ容量をオンラインで追加したり、削除したりすることができます。データの冗長なミラー・コピーを保持してフォルト・トレランスを実現することも、ベンダーが提供する信頼できるストレージ・アレイで使用することもできます。データの管理は、人間がデータベース・ファイル単位で操作するのではなく、データのクラスに対して望まれる信頼性とパフォーマンス特性を選ぶことで行います。

Oracle ASMは、大規模クラスタ化データベースを管理する際の実務的な問題の多くを解決します。データベースや関連データのサイズが増加し、ストレージ・デバイスが数千台、あるいはサーバーが数十台に近づく、従来のストレージ管理手法を効率的にスケーリングできなくなり、人為的なエラーが起きやすくなります。手動でのロード・バランシングといったその他のタスクも、非常に複雑になります。Oracle ASMは、Oracle RACやOracle単一インスタンス・データベースでのこういった問題を解決します。Oracle ASMについて詳しくは、<http://www.oracle.com/goto/asm>を参照してください。

アプリケーション・コンティニューティ

Oracle RACは、アクティブ/アクティブ型のデータベース管理システムで、Oracle RACデータベース・インスタンスやサーバーに障害が発生しても、データベース・サービスは引き続き利用可能です。これは、アクティブ/パッシブ型の構成を利用するテクノロジーとは対照的です。アクティブ/パッシブ型では、アクティブなデータベース・インスタンスに障害が発生すると、再起動して処理を再開する必要がありますので、通常は、フェイルオーバーが発生した後に、一時停止が数分間継続する可能性があります。どちらのシナリオでも、障害発生時点で処理中のSQLや未コミット・トランザクションは中断され、データベース管理システムによってロールバックされます。この障害に対処するには、あらゆるアプリケーションで複雑なアプリケーション・ロジックを書かねばなりません。ただし、障害はまれにしか起こらないので、このロジックは無視されることが多く、テストもめったに行われません。そのため、アプリケーション障害に至ります。さらに、データの論理的破損につながる可能性もあります。

Oracle RACにアプリケーション・コンティニューティを含めて利用すると、アクティブなアプリケーション接続を、存続しているインスタンスに透過的に自動フェイルオーバーできます。アプリケーション・コンティニューティは、部分的に実行された処理中のリクエストやトランザクションを中断することなく高速に再実行します。再実行が成功すると、障害が発生しなかったかのように、アプリケーションが透過的に継続されます。アプリケーション・コンティニューティを利用することで、アプリケーション開発者は、意図せずにアプリケーションでトランザクションが2回実行されないかと心配したり、複雑なエラー処理やリトライを記述したり、さらに悪い場合、失敗したリクエストを手動でリカバリする方法を探したりする必要がなくなります。アプリケーション・コンティニューティがあれば、アプリケーションの障害耐性が格段に高まり、速く楽にアプリケーションのコーディングを行えるようになります。

Oracle Application Continuityについて詳しくは、<http://www.oracle.com/goto/ac>を参照してください。

Fleet Patching and Provisioning

かつて、Clusterware、Oracle ASM、Oracle RACデータベースのプロビジョニングとパッチ適用は複雑なものでした。Oracle Fleet Patching and Provisioning（Oracle FPP）は、このプロセスを大幅に自動化および簡素化します。Oracle Fleet Patching & Provisioning（旧称Oracle Rapid Home Provisioning）は、プロビジョニング、パッチ適用、アップグレードをオーケストレーションし、アプリケーションにとってのデータベース可用性を最大化します。

Oracle FPPは、Oracle RACアーキテクチャと完全に統合されています。Oracle FPPでは、標準化されたソフトウェア・ホーム（ゴールド・イメージ）の、スペース効率に優れたリポジトリが保持され、これを任意の数のターゲット・マシンにプロビジョニングできます。指定したゴールド・イメージから、任意の数のホームをプロビジョニングできます。また、Oracle FPPには系統情報が保持されているため、デプロイしたソフトウェアの来歴を常に確認できます。

この機能や追加機能のおかげで、Oracle FPPは、オンプレミスでもOracle Cloudでも使用される理想的なフリート・ソフトウェア・パッチ適用システムとなっています。Oracle FPPは、パッチ適用プロセスを全自動化するために、Oracle Autonomous Databaseで使われています。Oracle FPPはOracle RACライセンスに含まれているので、オンプレミスのOracle Databaseは、追加料金なしに同じ利点を得られます。

Oracle FPPについて詳しくは、<https://www.oracle.com/database/technologies/rac/fpp.html>を参照してください。

新しい世代のOracle RAC

オラクルはOracle RACテクノロジーを継続的に強化し、簡素化および高速化して、スケーラビリティと可用性を向上させています。その結果、画期的なOracle RAC 19cリリースが誕生し、次世代のあらゆるワークロードやアプリケーションのニーズを満たすOracle RACが実現しています。

Oracle Real Application Clusters 19c

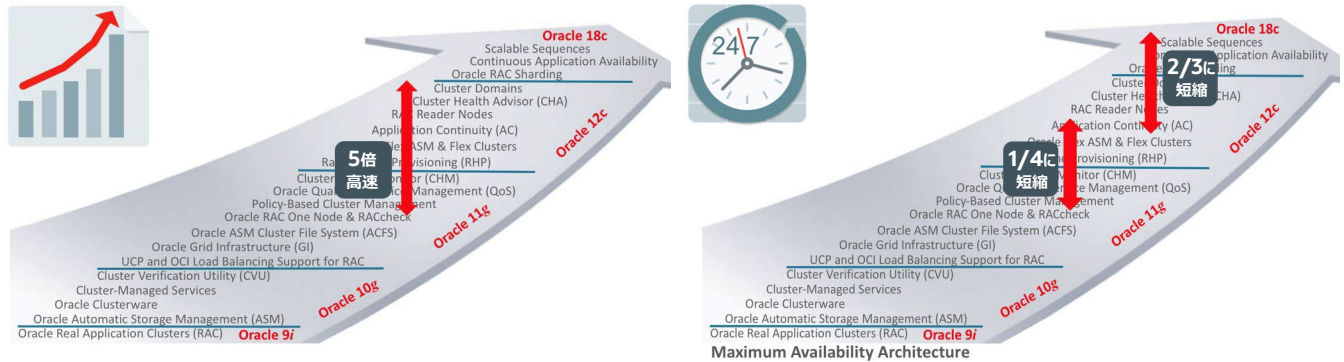


図2：Oracle RAC 19cでのスケーラビリティ（左）と可用性（右）の向上

図2は、Oracle RAC 19cがこれまでのOracle RACバージョンの中で、もっともスケーラブルで信頼性が高い理由を示しています。Oracle RACでの高競合ワークロードのパフォーマンスは、Oracle RAC 11g Release 2の5倍になっています。一方で、クラスタ再構成時やインスタンス障害後の一時停止時間は、Oracle RAC 11g Release 2とOracle RAC 12c Release 2の間で4分の1に、さらにOracle RAC 12c Release 2とOracle RAC 19cの間で3分の2に減少しています。つまり、Oracle RAC 11g Release 2で1分間の一時停止に見舞われていたアプリケーションは、Oracle RAC 19cでは15秒未満の停止で、完全に動作する状態に戻るでしょう。

Oracle RACデータベース・インスタンスやサーバーの障害からのリカバリには、クラスタの再構成が必要です。クラスタの再構成中は、障害発生時点で変更可能だったデータへのアクセスが遅延し、インスタンスがリカバリを終えるまで待たされる可能性があります。Oracle RACデータベースでは、障害が発生したインスタンスのリカバリは、存続しているいずれかのインスタンスで行われます。Oracle RAC 12c Rel. 2より前のバージョンのOracle RACの場合、このリカバリ・プロセスは、障害後にリカバリを実行するデータベース・インスタンスの選択や、標準REDOログの読み取りと適用を含む5段階で行われていました。

Oracle RAC 12c Release 2以降は、さまざまな改善の一環として、Recovery Buddy機能によってこの5段階のうちの2つが完全になくなっています。この機能により、新しいインスタンスが起動するたびに、リカバリを行うインスタンスが事前に割り当てられるので、インスタンスのリカバリを行うインスタンスを選択する必要がなくなります。さらに、新しいプロセスが導入され、割り当てられたRecovery Buddyインスタンスにインターコネクト経由でREDOログ・リカバリ・データを継続的に直接送信するようになっています。Buddyインスタンスは、この情報をメモリに保存し、リカバリを実行するときにREDOログを読み取らなくてもよいようにします。これにより、新しいトランザクションが再びデータに完全にアクセスできるようになるまでの時間が短縮されます。

高競合ワークロードでパフォーマンスが5倍に向上した理由を、数個の機能に言及するだけで説明することはできません。この重要な改善には、次の図3に示すさまざまな機能が貢献しています。図3に記載されているそれぞれのパフォーマンス強化機能が高競合ワークロードのパフォーマンス改善にどのように役立ち、貢献しているかについては、このペーパーでは説明しません。

SAP、オラクルのEBusiness Suite、Siebelなどの複雑なアプリケーションを含むほとんどのアプリケーションや、一般的に、Oracle RACシステムのほとんどのOLTPや分析のワークロードでは、シングル・インスタンス・システムで垂直スケーリングを行う場合、ほぼ直線的なスケーラビリティが実現します。世界中のあらゆる業界で、もっとも複雑なワークロードを20年超にわたって実行してきたことが、この事実を証明しています。

アプリケーションのスケーラビリティ率が80 %未満の場合、そのアプリケーションは競合によってパフォーマンスが低下している可能性が高いと考えられます。このような競合の原因として、次の3つのいずれか、またはその組合せが挙げられます。

1. アプリケーションのトランザクションによって、同じデータ・ブロックに対する変更が頻繁に行われている（“書き込みホットスポット”）。
2. REDOログの書き込みが必要なため、新しいブロック変更を他のインスタンスに分散させる処理が遅くなっている（“REDOログ遅延”）。
3. アプリケーションでメタデータの競合が発生している（“Right Growing Indexと索引の競合”）。

- ほとんどのOLTP書き込みホットスポットは索引で発生

Oracle RAC 19cでは、さまざまな機能を使用することで、こういった競合原因によるパフォーマンスへの影響を大幅に軽減しています。

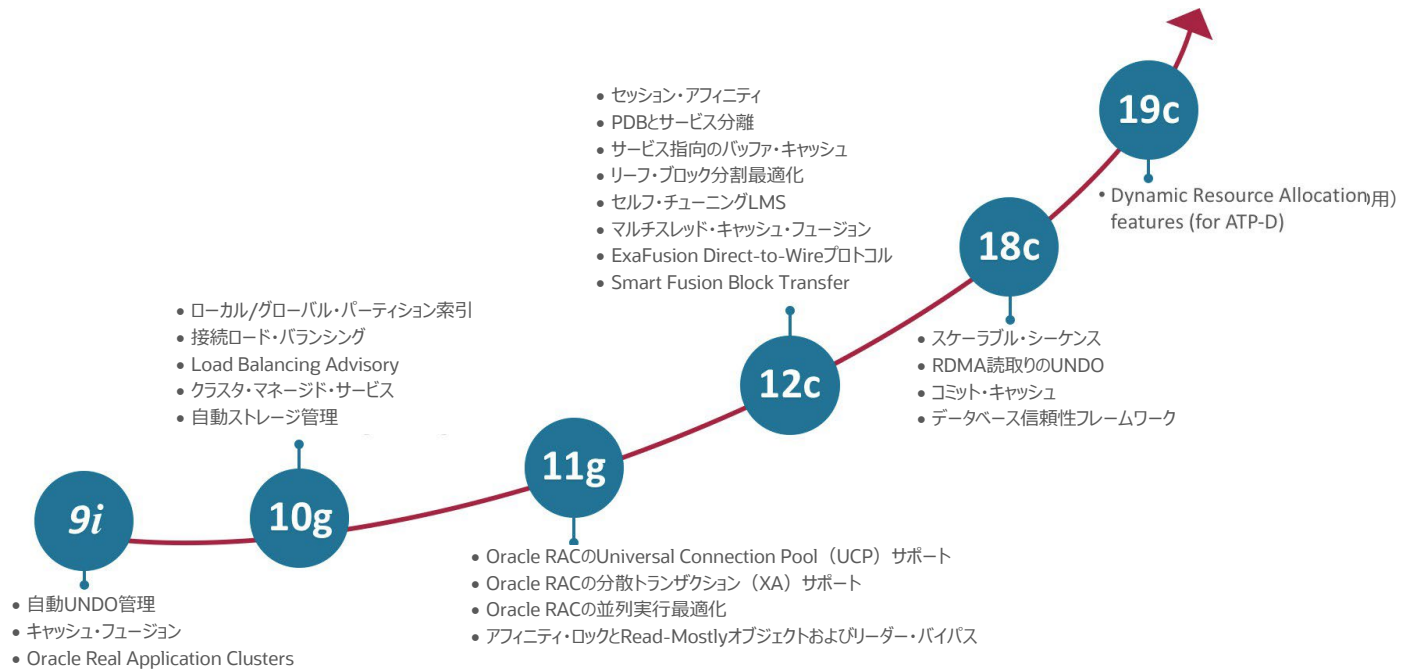


図3：Oracle RACの各リリースにおけるパフォーマンス強化機能

エンジニアド・システム – Oracle RACを意識した設計

オラクルは10年以上前、一般的にOracleエンジニアド・システムと呼ばれる高パフォーマンスで堅牢なデータベース・インフラストラクチャの提供に向けて、新たな戦略に乗り出しました。Oracleエンジニアド・システムは、Oracle Databaseに最適化されたサーバー、ネットワーク、ストレージによる事前構成済み、事前チューニング済み、事前テスト済みの統合システムです。もっとも重要な2つのエンジニアド・システム製品として、Oracle Exadata Database Machine (Exadata) とOracle Database Appliance (ODA) が挙げられます。2つのシステムのうち、シンプルの方がODAです。これは主に、Oracle RACデプロイメントを簡単にできるように設計されています。ODAのハードウェアはOracle RAC対応ですぐに利用でき、ODAのツールはオペレーティング・システム、ファームウェア、Oracle Clusterware、Oracle RACデータベース・ソフトウェアのデプロイメントとパッチ適用をオーケストレーションします。Exadataはさらにスケーラブルで高パフォーマンスな製品で、汎用ハードウェア・システムにデータベースをデプロイするお客様には実現できない多くの機能を提供します。

Exadataは、オラクルの推奨プラットフォームであり、OLTP、DWH、レポート作成、バッチ処理、統合、混合データベース・ワークロードなどのデータベース・アプリケーションに極めて優れたパフォーマンスを発揮します。Oracle Exadataは、優れた価格性能比、可用性、サポート性を提供します。Oracle Exadataを使うと、ユーザーはシステムの構築、テスト、保守の必要性から解放され、より価値の高い、ビジネスの問題に集中できるようになります。さらに、オラクルはExadata Database Machineのすべての要素を提供して統合しているので、他にはない多くのソフトウェア機能を開発し、汎用プラットフォームでは得られないメリットを実現することができます。

Exadataのスケール・アウト・アーキテクチャ

Oracle Exadata Database Machineでは、データベース・サーバーとストレージ・サーバーの両方でスケール・アウト・アーキテクチャが採用されています。Oracle Exadata Database Machineの成長に伴い、データベースのCPU、ストレージ、およびネットワークを平衡のとれた方法で追加し、ボトルネックのないスケーラビリティを確保できます。Exadataは、Oracle RACが提供するスケール・アウト・データベース・コンピューティングをストレージ層のスケール・アウト・コンピューティングで補完することで、データ処理集中型処理を高速化します。ストレージのコンピューティング・スケール・アウトは、アプリケーションにとってシームレスで透過的な形で動作します。

Oracle Exadataのアーキテクチャは、Oracle Database管理システムに特化して構築されています。多くのOracle Databaseリリースを通して、Exadataのアーキテクチャの豊富な機能を活用する、ますます多くの機能がコア・データベースに追加されています。Oracle Exadata Database Machineで使われ、Oracle Databaseと併用して相乗効果を生む機能の例として、ストレージ索引、ストレージ・サーバーによるSQLオフロード処理、Hybrid Columnar Compression、Smart Cache Flashが挙げられます。こういった機能は、シングル・インスタンス・データベースにもOracle RACデータベースにもメリットをもたらしますが、データベースやExadataの機能強化の中には、Oracle RACデータベースに他にはないメリットをもたらすものもあります。

Oracle RACに役立つExadata拡張機能

- Exadata Instant Failure Detection

調整されたクラスタとして動作する、独立した複数のサーバーで構成されたOracle RACシステムでは、クラスタ内のサーバーで障害が発生した場合、すばやくそれを検出してリカバリできる必要があります。そのため、Exadataにはいくつかの機能強化が組み込まれており、ノード障害を検出して、わずか2秒ほどで再構成できるようになっています。この機能を実現できるのは、オラクルがExadataのデータベースやネットワーク・インフラストラクチャをエンド・ツー・エンドで制御しているからです。

- キャッシュ・フュージョンの機能強化

Oracle RACのパフォーマンスは、基盤となる通信の速度に依存します。ExadataのOracle Cache Fusionは、Exadataが提供する、待機時間が短くて広帯域幅のクラスタ・インターコネクトを活用します。さらに、ExadataはExafusionネットワーク・プロトコルを実装しているので、Oracleプロセスは、オペレーティング・システムのカーネルをバイパスし、データベース・サーバー間でRDMAの読取りや書き込みを直接行うことができます。これにより、プロセスがネットワーク・スタックの横断やコンテキスト・スイッチの発生によるオーバーヘッドの影響を受けることがなくなるため、パフォーマンスが劇的に向上します。

Exadataのみで利用できるキャッシュ・フュージョンの最適化としては他にも、Exadata Commit Cacheがあります。この機能は、RDMAプロトコルを用いてトランザクションのコミット時間を記録するインメモリ・キャッシュです。Oracle RACデータベース・インスタンスでは、たとえ別のデータベース・インスタンスが同じデータをアクティブに処理していても、継続中のトランザクションのデータの整合性を保たなければなりません。Exadataでは、最初のデータベース・インスタンスにUNDOデータの読取りが必要かどうかを、コミット・キャッシュからすばやく判断するという最適化が行われています。この機能の結果、データ・ブロックのUNDOデータのトラフィックが60 %減少するワークロードもあります。

- インメモリのフォルト・トレランス

既存のデータベースにインメモリ列指向およびベクトル処理データベース機能を提供するOracle Database In-Memoryによって、分析速度が透過的かつ桁違いに向上すると同時に、混合ワークロードのOLTP速度も向上します。Oracle Database In-MemoryをOracle RACデータベースなどの既存のデータベースに導入するのは簡単です。アプリケーションを変更する必要がないからです。

一般的なOracle RAC構成でデータベース・インスタンスに障害が発生すると、そのインスタンスのインメモリ・データは利用できなくなります。問合せは、存続しているデータベース・インスタンスで引き続き実行できますが、ストレージからインメモリにデータを再移入するには時間がかかり、その間、分析問合せの実行速度は大幅に低下します。これに対処するため、Exadataではフォルト・トレラントなOracle Database In-Memoryが提供されます。これにより、必要に応じてOracle RACクラスタのインスタンスにデータを複製できるため、速度低下がなくなります。ストレージ・サブシステムがディスク全体にデータをストライプ化およびミラー化して高パフォーマンスと高可用性を実現すると同様に、Oracle Database In-Memoryも、Exadata Database MachineのOracle RACインスタンス全体にインメモリ・データを分散させ、複製します。その結果、1つのサーバーやデータベース・インスタンスに障害が発生しても、インメモリ問合せで、存続しているサーバー上の複製されたデータ・コピーを透過的に使用できます。

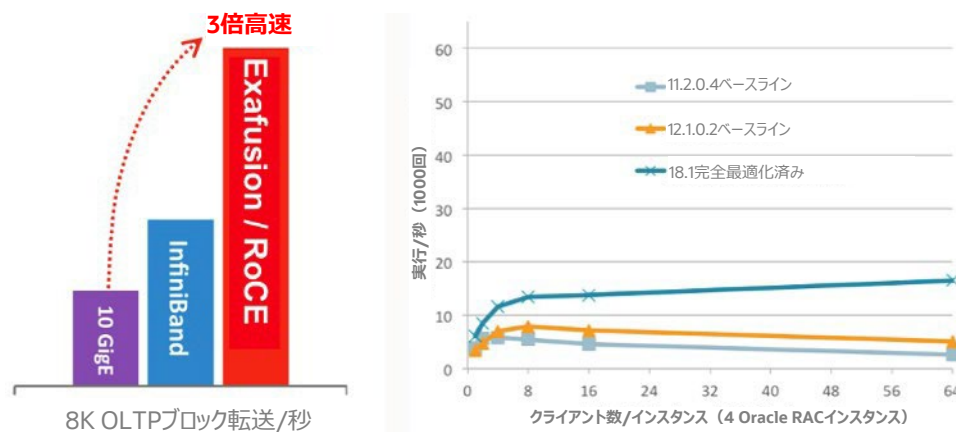


図4：ExadataベースのOracle RACの改善と、改善がパフォーマンスに及ぼす影響の概要

Oracle RACの利点をOracle Cloudに拡大する

Oracle RACは、オラクルのクラウド製品の重要な要素です。Oracle RACは、Oracle RAC Database Cloud Service (Oracle DBCS)、Oracle Exadata Service、Autonomous Databaseで利用できます。また、Oracle RACは、Oracle SaaSの基盤データベース・インフラストラクチャの一部でもあります。オンプレミスのOracle RACを導入している数千社がすでに経験している優れたパフォーマンスと可用性を同じように実現しながら、Oracle Cloud上でOracle RAC Databaseを実行できます。こういったデータベース・サービスには、オンプレミスにデプロイされているデータベースと100 %互換性があるため、Oracle Cloudへの移行を円滑に行い、効率よくハイブリッド・クラウド戦略を実施できます。

Oracle CloudにOracle RAC構成をデプロイすると、オンプレミスへのデプロイと同じように、ClusterwareやOracle ASMといったOracle RACインフラストラクチャのすべての基盤コンポーネントが活用されます。違うのは、Oracle CloudのOracle RACデプロイメントは高度に自動化され、数分で完了する点です。

オラクルは、Oracle RACを完全にサポートするインフラストラクチャを作成する唯一のクラウド・ベンダーです。このステップには、仮想マシン間で効率的にストレージを共有することや、Oracle RAC対応のネットワーク・インフラストラクチャを提供することが含まれます。このような理由により、Oracle RACはオラクルのクラウドでしかサポートされていません。Cloud InfrastructureへのOracle RACのデプロイについて詳しくは、MOS Note 2093394.1を参照してください。

Oracle RACとAutonomous Database

オラクルは、ミッション・クリティカルなアプリケーションが必要とする高可用性とスケーラビリティを提供するため、主力データベース・クラウド・サービスであるAutonomous Database Serviceの基盤にOracle RACを選択しました。次の図5をご覧ください。

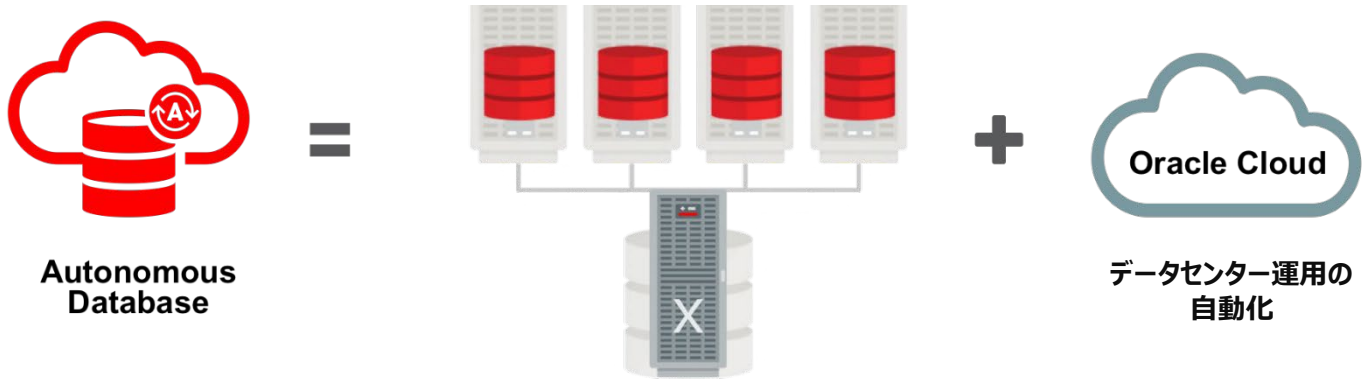


図5 : Oracle RACはAutonomous Databaseの基盤

ExadataのOracle RACは、Autonomous Databaseで実行されるワークロードを透過的にスケール・アウトできます。Oracle RACの構成と運用の全側面が完全に自動化されます。かなりの数のCPUを使用して実行されるデータベースが自動的に構成され、Oracle RACクラスタの複数のサーバーで稼働します。Oracle RACのローリング・パッチ適用とアプリケーション・コンティニューイティを使用し、サーバー、OS、Clusterware、データベースのメンテナンス時にアプリケーションの停止時間が発生しないようにします。クラスタのノードに分散させると競合を引き起こすアプリケーション・モジュールがある場合は、接続文字列に`collocationタグ`を使うと、同じインスタンスで実行するように構成できます。

Oracle RACとその他のスケール・アウト・データベース・アプローチの比較

現在市場に出回っているデータベースの数や、そういったデータベースで利用できる高可用性およびスケーラビリティ・ソリューションの多様性を考えれば、次のような疑問が生じます。Oracle DatabaseとReal Application Clustersをソリューションとして選ぶ理由は何でしょうか。この問いの答えは簡単です。オラクルは、既存のすべてのスケーラビリティおよび可用性アーキテクチャの最高の要素を他にはない形で組み合わせ、あらゆる種類や規模のアプリケーションのニーズを満たせるようにしているからです。

大まかに言えば、データベースの高可用性およびスケーラビリティ・ソリューションは、表1に記載されたアプローチによって分類できます。

表1：データベースの高可用性およびスケーラビリティ・ソリューション – 概要

テクノロジー	利点	欠点	Oracle RACとの比較
クラスタ・フェイルオーバー – 現在実行中のサーバーに障害が発生した場合に、データベースを別のサーバーで実行します。	利用および実装が比較的簡単です。	シンプルでソリューションで、スケーリングや透過的なメンテナンスができません。	Oracle RACではスケーラビリティ、はるかに高速なフェイルオーバー、透過的なメンテナンスが追加されているので、フェイルオーバー・クラスタの上位機能が提供されます。
シェアード・ナッシング – ストレージを共有しない複数のコンピュータでデータベースを透過的に実行します。	共有ストレージを実装することなく、複数のノードにスケーリングできます。分析ワークロードに効果的です。	メッセージングとトランザクションの調整コストが高いため、OLTPワークロードのパフォーマンスとスケーラビリティが低下します。 数千個の表や索引にスケーリングできないので、ERP、CRM、HCMといった非常に複雑なアプリケーションでは利用できません。	シェアード・ナッシングの優れた点とシェアード・ストレージの優れた点を組み合わせています。 シェアード・ナッシングと同様の方法でSQLファンクション・.shippingを使って、長時間実行されるSQLをクラスタのノード全体で自動的に並列化します。これにより、分析ワークロードやバッチ処理をノード全体で透過的にスケーリングできます。 OLTPワークロードの場合は、極めて複雑な（ERP）アプリケーションであっても、データ・shippingと分散キャッシュを使って優れた応答時間と透過的なスケーリングを実現します。
データベース・シャーディング – 複数の物理データベースから論理データベースを構築し、シャーディング・キーに基づいてリクエストをルーティングします。	物理データベースが独立しているため、高可用性と障害の分離を実現できます。	アプリケーションに対して透過的ではありません。 シャーディング・キーでパーティション化できるアプリケーションにしか効果がありません。 クロス・シャード操作に十分対応していないか、まったく対応していません。	Oracle Globally Distributed Databaseは、複数の物理Oracle Databaseへのシャーディングを実装しており、シャーディング・キーを使用してパーティション化できるアプリケーションで最大級のスケーラビリティと障害の分離を実現します。

		非常に複雑な（ERP）アプリケーションに対応していません。 分析に十分対応していないか、まったく対応していません。	それぞれのシャード・データベースで Oracle RACを使用することで、シャードの高可用性や計画メンテナンスを実現でき、両方のテクノロジーの利点を組み合わせることができます。
マスター/スレーブ型レプリケーション – マスター・データベースから1台または複数台のスレーブ・データベースにデータをレプリケーションします。	障害からの保護とフェイルオーバー、ソリューションによっては読取りスケーラビリティが実現します。	書き込みスケーラビリティがありません。 通常はフェイルオーバーによって一部のデータが失われます。 通常は読取りレプリカでデータの整合性が失われます。 接続とユーザーのスケーラビリティの点で劣ります。 レプリカに分析のスケーラビリティがありません。	Oracle RACの透過的なスケーラビリティおよびフェイルオーバーと、レプリケーションによる障害からの保護を組み合わせることで、両方の利点が得られます。 Oracle Active Data Guardでは、レプリカ・データベース（Oracle Database 19cではレプリカ・データベースの透過的な更新）を使用して、読取りスケーラビリティを高めることができます。
アクティブ/アクティブ型レプリケーション – 完全にアクティブな複数のデータベースにデータをレプリケーションします。	可用性が非常に高く、障害からの保護や読取りスケーラビリティに優れています。	複数のレプリカへの書き込みが競合し、データ整合性の問題が起きる可能性があります。 アプリケーションに対して透過的ではありません。 データベース（ファイル）の複数個のコピーが必要になります。 レプリカに分析のスケーラビリティがありません。	それぞれのレプリカ内で Oracle RACを使用することで、書き込みと分析のスケーラビリティが実現し、レプリカが少なくなることで、競合の回避に役立ちます。 Oracle RACには、業界をリードするGoldenGateレプリケーションが完全に統合されています。

以上の内容からわかるように、Oracle RACは、OLTP、分析、混合ワークロードのどのような組合せに対しても、他にはない形で高可用性とスケーラビリティを発揮します。また、オラクルのネイティブ・シャーディング、Data Guard、レプリケーション・テクノロジーにシームレスに統合されているので、ディザスタ・リカバリだけでなく、さらに高いスケーラビリティや可用性が実現します。Oracle RACには、最新のデータベース・アーキテクチャの最高の特性がすべて備わっており、欠点は最低限またはゼロに抑えられています。

まとめ

オラクルは、ミッション・クリティカルなアプリケーションに必要となる透過的なデータベースの可用性とスケーラビリティを実現するため、Oracle Real Application Clusters (Oracle RAC) の開発および強化に数千年に匹敵する工数をかけています。このテクノロジーは業界に唯一であり、世界の大手企業のほとんどで、社内アプリケーションと顧客向けの重要アプリケーションの両方の実行に使われています。

最近、Oracle RACアーキテクチャの大幅な改善が行われており、アプリケーションの可用性とスケーラビリティが、わずか数年前に比べて何倍にも向上しています。こういった改善を組み合わせることで、新しい世代のOracle RACテクノロジーが生まれ、次世代のワークロードやアプリケーションのニーズに対応できます。

Exadata Database Machineなどのエンジニアド・システムは、Oracle RACを意識した設計になっており、可用性とスケーラビリティをこれまでにないレベルまで、さらに高めます。同時に、エンジニアド・システムによって、Oracle RACシステムのデプロイや管理が簡単になります。

Oracle Cloud、とりわけオラクルのAutonomous Databaseは、Oracle RACとエンジニアド・システムをフル活用し、市場に最高のデータベース・サービスを提供します。また、Oracle CloudでOracle RACのデプロイと管理が容易になるので、大規模なデプロイ環境でも簡単かつ効率的に管理できます。

さらに、Oracle RACアーキテクチャの一部であるアプリケーション・コンティニューイティやFleet Patching and Provisioningといった機能を利用すれば、オンプレミスでもOracle Cloudでも、そういった管理タスクを全自動で実行できます。そしてもっとも重要なことは、アプリケーションとユーザーに透過的であることです。

結論として、Oracle RACデータベースは、ミッション・クリティカル・アプリケーション環境として理想的です。重要なアプリケーションに絶対に欠かせない要件は、高可用性、動的なスケーラビリティ、そしてアプリケーションの変更が必要ないことです。Oracle RAC、とりわけOracle RAC 19cなら、それらを完全に満たすことができます。

Connect with us

+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 © 2024, Oracle and/or its affiliates.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の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、Java、MySQLおよびNetSuite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。