

Oracle Advanced Compression の概念実証のガイドライン、 インサイト、ベスト・プラクティス

2024年7月、バージョン23ai

Copyright ©2024, Oracle and/or its affiliates 公開

本書の目的

本書では、リリース23aiの機能と強化された点の概要が説明されています。本書は、23aiへのアップグレードに関するビジネス上の利点の評価と、説明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。本文書に記載されている機能の開発、リリース、時期および価格については、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、本書に記述されているすべての機能を安全に組み込むことができず、コードの不安定化という深刻なリスクを伴う場合があります。

目次

はじめに	4
高度な行圧縮	4
高度な行圧縮の有効化	5
高度な行圧縮が起こるタイミング	6
ダイレクト・パスと従来型パスによるバルク・ロード	7
高度な索引圧縮	8
Oracle RMANバックアップ圧縮	9
高度なLOB圧縮（SecureFiles LOBセグメントの場合）	9
テストの開始前の考慮事項	10
圧縮オーバーヘッドについて	11
圧縮率の改善	11
通常の場合の実証の内容	12
無料の圧縮アドバイザ	13
付録A	13

このドキュメントについて

本書は、圧縮の概念実証を実行するためのステップ・バイ・ステップ・ガイドではありません。代わりに、本書では、ユーザーのテストから学んだ圧縮のガイドライン/ベスト・プラクティスとともに、圧縮の概念実証を計画したり、概念実証の結果を理解したりするのに役立つ、その他の洞察を提供します。

はじめに

企業は、データ量の急増という重大な問題に直面しています。企業は、収益に影響を与えることなく変化の激しいビジネス状況に迅速に適応する必要があります。ITマネージャーは、既存のインフラストラクチャを効率的に管理してコストを制御しながら、アプリケーションの問合せパフォーマンスを提供し続けていく必要があります。

Oracle Advanced CompressionとOracle Databaseを併用することによって得られる圧縮、パフォーマンス、データ・ストレージの最適化といった一連の堅牢な機能により、ITマネージャーはこの複雑な環境でも任務を遂行できます。

Oracleデータベースのデプロイメントがクラウドまたはオンプレミスのどちらであっても、Oracle Advanced Compressionにより、アプリケーションに変更を加えることなく、さまざまな環境において、堅牢な圧縮機能を実現できます。Oracle Advanced Compressionの利点には、データベース記憶域のフットプリントの縮小、バックアップにおける時間とストレージの節約、問合せパフォーマンスの向上などがあります。

一般的に評価される機能

- **高度な行圧縮**

あらゆるタイプのデータ操作の実行中に表データを圧縮できます

- **高度な索引圧縮**

サポートされるすべての一意索引と非一意索引のサイズを縮小します

- **Oracle RMANバックアップ圧縮**

Oracle Recovery Manager（Oracle RMAN）の使用時にバックアップ・データを圧縮します

- **高度なLOB圧縮**

SecureFiles LOBを圧縮します

高度な行圧縮

高度な行圧縮により、INSERTやUPDATEなどの従来のDMLを含むあらゆるタイプのデータ操作中に、データを圧縮できます。さらに、高度な行圧縮では、圧縮データに対する書き込み操作のオーバーヘッドが最小限に抑えられます。そのため、トランザクション環境やOLTP環境に加え、データウェアハウスにも適しており、圧縮の利点がすべてのアプリケーションのワークロードへと拡張されます。

高度な行圧縮では、OLTP/DWアプリケーションで動作するよう特別に設計された、独自の圧縮アルゴリズムが使用されます。このアルゴリズムは、データベース・ブロック内や複数の列間の重複値を排除することによって動作します。圧縮されたブロックには、圧縮メタデータを維持する記号表と呼ばれる構造体が含まれます。ブロックが圧縮されると、最初に重複値のコピーが記号表に1つ追加されることにより、重複値が排除されます。そして、各重複値が、記号表内の適切なエントリへの短い参照に置き換えられます。

この革新的な設計では、圧縮されたデータを元の状態へ変換するために使用されるメタデータがブロック・ヘッダー内に保存されるため、圧縮されたデータはデータベース・ブロック内で自己完結します。グローバルなデータベースの記号表を維持する競合他社の圧縮アルゴリズムと比較すると、圧縮されたデータにアクセスする際、（グローバルな記号表では必要な）追加のI/Oが発生しないオラクル独自のアプローチには、大幅な利点があります。

圧縮率は、圧縮されるデータ、特にデータのカーディナリティによって環境ごとに異なります。通常は、高度な行圧縮または高度な索引圧縮（あるいはその両方）を使用することで、ストレージ領域の消費が1/2～1/4に削減されると期待できます。つまり、非圧縮データ/索引が消費する領域量は、圧縮されたデータが消費する領域量の2～4倍になるということです。

しかし、高度な行圧縮の利点は、ディスク上のストレージを節約できるだけではなく、問合せパフォーマンス上の主な利点の1つは、ブロックを解凍することなく、圧縮されたブロック（データと索引）を直接メモリ内で読み取れることです。これにより、I/Oの数が減り、I/O操作に関連したシステム・コール数が削減されるため、問合せパフォーマンスを向上させることができます。さらに、メモリの追加を必要とせずにより多くのデータを保存することにより、バッファ・キャッシュの効率が向上します。

高度な行圧縮の有効化

新しい表やパーティションに対して、高度な行圧縮を有効化する方法は簡単です。単純に表またはパーティションをCREATE構文で作成し、“ROW STORE COMPRESS ADVANCED”と指定します。たとえば、次のような文を使用します。

```
CREATE TABLE emp (emp_id NUMBER, first_name VARCHAR2(128), last_name  
VARCHAR2(128)) ROW STORE COMPRESS ADVANCED;
```

既存の表に対して高度な行圧縮を有効化する方法は、多数あります。各方法に関する完全な議論は本書の範囲外ですが、本書では、一般的に使用される方法の概要を示します。

ALTER TABLE ... ROW STORE COMPRESS ADVANCED

今後のすべてのDMLについて高度な行圧縮を有効化しますが、表の既存のデータについては非圧縮のままにします。

オンライン再定義（DBMS_REDEFINITION）

この方法では、その後のDMLでの高度な行圧縮が有効化され、既存のデータが圧縮されます。DBMS_REDEFINITIONを使用すると、移行中に、読取り/書込みの両方のアクティビティに対して、表がオンライン状態で維持されます。最適なパフォーマンスを得るには、DBMS_REDEFINITIONをパラレルで実行します。

オンライン再定義では、操作の実行中に索引が仮表にクローンされます。クローンされたすべての索引の増分が同期（リフレッシュ）操作中に維持されるため、オンライン再定義の実行中も実行後も、索引の利用が中断されることはありません。

ただし、オンライン再定義をパーティションの再定義に使用する場合に限り、索引の利用は中断されます。この場合、グローバル索引がすべて無効化され、オンライン再定義の実行後にそのグローバル索引を再作成する必要があります。

ALTER TABLE ... MOVE ROW STORE COMPRESS ADVANCED

この方法では、その後のDMLでの高度な行圧縮が有効化され、既存のデータが圧縮されます。表の移行中、読取りアクティビティに対しては表がオンライン状態で維持されますが、排他（X）ロックがかかるため、移行コマンドが完了するまですべてのDMLがブロックされます。最適なパフォーマンスを得るには、ALTER TABLE ... MOVEをパラレルで実行します。

ALTER TABLE ... MOVEにより、パーティションまたは表にある索引がすべて無効化されます。ALTER TABLE ... MOVEの実行後、これらの索引を再作成する必要があります。パーティションの移行の場合、ALTER TABLE ... MOVE PARTITIONをUPDATE INDEXES句とともに使用すると、索引が維持されます（排他（X）ロックがかかるため、移行コマンドが完了するまで、すべてのDMLがブロックされます）。パーティション化されていない表に対して、この句は使用できません。

ALTER TABLE... MOVE文を使用すると、パーティション化されていない表のデータやパーティション化された表のパーティションのデータを新しいセグメントに再配置したり、オプションとして異なる表領域に再配置したりできます。ALTER TABLE...MOVE ROW STORE COMPRESS ADVANCEDにより、圧縮データ用の新しいエクステントが移行先の表領域内に作成され、データが圧縮されます。ここで、新しいセグメントは、データファイルの末尾や先頭に配置されるとは限らず、あらゆる場所に配置される可能性がある点に注意が必要です。そのため、元のセグメントが解放されるとき、エクステントの位置によっては、データファイルが縮小されない場合もあります。

ALTER TABLE ... MOVE TABLE/PARTITION/SUBPARTITION ... ONLINE

この方法では、その後のDMLでの高度な行圧縮が有効化され、既存のデータが圧縮されます。ALTER TABLE ...MOVE TABLE/PARTITION/SUBPARTITION ... **ONLINE**を使用すると、移行中の表/パーティション/サブパーティションでDML操作を中断せずに実行し続けることができます。索引は移行操作中に維持されるため、手動による索引の再作成は不要です。

ドキュメントに関する重要な注意事項

前述の操作に関する詳細、使用例、および制限事項については、現行のOracle Databaseドキュメントを参照してください

高度な行圧縮が起こるタイミング

高度な行圧縮では、OLTPアプリケーションおよびデータウェアハウス・アプリケーションで動作するために特別に設計された独自の圧縮アルゴリズムを使用します。このアルゴリズムは、データベース・ブロック内や複数の列間の重複値を排除することによって動作します。圧縮されたブロックには、圧縮メタデータを維持する記号表と呼ばれる構造体が含まれます。

ブロックが圧縮されると、最初に重複値のコピーが記号表に1つ追加されることにより、重複値が排除されます。そして、各重複値が、記号表内の適切なエントリへの短い参照に置き換えられます。

圧縮の利点と使用される圧縮技術は似ていますが、データベース内では、高度な行圧縮の使用時にさまざまな圧縮タイプを起動できます。以下に、さまざまな圧縮タイプの例を示します。これらさまざまな圧縮タイプがいつ使用されるかを理解すると、概念実証の結果を分析するときに役立ちます。

挿入ダイレクト・ロード圧縮

APPENDヒントを使用した挿入やSQL*Loaderを使用した挿入など、ダイレクト・パスによるロードのメカニズムを使用してデータが挿入されるときに実行されます。この場合、データは、セグメントの最高水位標（セグメントの仮想最終使用ブロック・マーカー）の上に挿入されるため、データ・ブロックに極めて効率的に書き出すことができます。圧縮エンジンでは大量の行が処理され、行をバッファして圧縮し、圧縮した行をデータ・ブロックに書き出すことができます。この結果、領域をすぐに節約できます。

挿入ダイレクト・ロード圧縮の場合、行が非圧縮形式で書き込まれることは一切ありません。

再帰的圧縮

単一の行や配列の挿入と更新などの従来のDML操作で起動されます。この圧縮タイプでは、行が非圧縮形式で書き出されます。データ・ブロックが内部ブロックの使用率のしきい値に達すると、圧縮が起動されます。このようなシナリオでは、データ・ブロックを再帰的のランザクションで圧縮できます。これは、圧縮後に即時コミットされます。

圧縮によって節約された領域は、即時解放され、任意の追加トランザクションで使用できます。圧縮はユーザーのDML操作（ユーザーのトランザクション）によってトリガーされますが、データの実際の圧縮は再帰的トランザクションで行われるため、圧縮の最終結果はユーザーのトランザクションの最終結果とは紐づいていません。

ダイレクト・パスと従来型パスによるバルク・ロード

バルク・ロード操作の実行時に、ダイレクト・パスによる方法または従来型パスによる方法を選択することで、ロード・パフォーマンスに重大な影響が及ぶ可能性があります。

バルク・ロード挿入操作の実行時、特に従来型パスによるロードを使用して大量の行を挿入する場合に、挿入パフォーマンスが低下する可能性があります。

大量の行を扱う場合に従来型パスによるロードを使用すると、速度が低下する可能性があります。その理由は、既存の圧縮ブロックに新しい行が挿入されると、挿入は圧縮されずに実行され、その後、同じブロックで追加の挿入が行われるとブロックが埋まり始め、内部のしきい値に達すると、ブロックが圧縮されるからです（詳細については、前述の再帰的圧縮の説明を参照）。

圧縮後に別の領域が解放されると、このブロックに対して挿入が再度実行され、同じ従来型パスによるロードでの操作中に圧縮が再度実行されるようになります（場合によっては同じブロックに対してさらに複数回の圧縮）。

つまり、従来型パスによる挿入を使用すると、同じ操作中に同じブロックが複数回圧縮され、CPUリソースと時間が消費される可能性があります。ワークロードの大半が従来型パスによる挿入である場合、高度な行圧縮アルゴリズムの一環としてブロックが繰り返し再圧縮されると、（ダイレクト・パスによるロードと比較して）I/Oが増える可能性が高くなります。

大量の行を操作する場合は、ダイレクト・パスによるロード操作が推奨されます。なぜなら、従来型パスによるロードとは異なり、ダイレクト・パスによるロードは最高水位標より上で行われるため、ブロックは埋まってから1回だけ圧縮され、その後、ディスクに書き込まれるからです。これにより、バルク挿入が効率的になり、従来型パスによるロードを使用してバルク挿入を実行するとき起こり得る、同じブロックの複数回の圧縮が回避されます。

AWRおよびダイレクト・パス/従来型パスによるバルク・ロード

概念実証テストの実行中に、バルク・ロードでダイレクト・パスによるロード方法と従来型パスによるロード方法のどちらが使用されるか分からない場合は、（AWRで）以下の提案される手順を使用して、SQL操作中に生じる圧縮の量を確認できます。

従来型パスによるロードでの圧縮の確認

AWRには、“Instance Activity Stats”セクションがあり、そこに正圧縮（HSC OLTP正圧縮）の総数と負圧縮（HSC OLTP負圧縮）の総数に関連する統計が表示されます。これら2つの統計を合計すると、試行された圧縮（再圧縮またはそれ以外）の総数を把握できます。

- HSC OLTP正圧縮 + HSC OLTP負圧縮 = 試行された圧縮と再圧縮の総数（従来型パスによるロード）

ダイレクト・パスによるロードでの圧縮の確認

“insert append”などのダイレクト・パスによる方法を使用してバルク・ロードを実行すると、データはデータ・ブロックに編成され、メモリに圧縮されます。つまり、バルク・ロード・データは一度だけ圧縮されます。

データ・ブロックは、表のPCTFREE設定で指定されているポイントまで満たされます。Oracle DatabaseのPCTFREEのデフォルト設定は10 %です（PCTFREEを使用すると、SQL UPDATE操作中に起こり得る増加に備えてデータ・ブロックの領域を予約できます）。

Create Table as Select (CTAS) やinsert appendの場合のような、最高水位標 (HWM) より上のブロック圧縮の場合、HSC IDL Compressed Blocksという統計があります。

- HSC IDL Compressed Blocks = HWMより上のブロック圧縮 (CTASやinsert appendなどのダイレクト・パスによるロード)

HSC OLTP正圧縮とHSC OLTP負圧縮の統計値のみが表示され、HSC IDL Compressed Blocksの統計値がまったく表示されないかほとんど表示されない場合、実行されているすべての圧縮は従来型パス操作によるものです（特に、1秒あたりに実行されている圧縮の数を確認してください）。

実行可能な場合は、同じ操作で従来型パスによるロードの代わりにダイレクト・パスによるロードが実行されるように、バルク挿入を変更することを検討する必要があります。その場合、HSC IDL Compressed Blocksの統計値が大きくなります。HSC IDL Compressed Blocksとラベル付けされた統計がない場合、HWMより上のブロック圧縮がなかったことを意味します。

高度な索引圧縮

Advanced Compressionの機能である高度な索引圧縮を使用すると、索引圧縮を自動化できるため、DBAは圧縮で考慮するプリフィックス列の数を指定する必要がなくなります（索引キー圧縮では必要）。

高度な索引圧縮は、複数の圧縮レベル（低および高）を利用できるテクノロジーです。圧縮率は、実行する圧縮のレベルによって異なりますが、平均で2～5倍です。高度な索引圧縮による相当なストレージ節約によって、ITマネージャーは多くの場合、ストレージの新規購入の必要性を減らすことができます。

高度な索引圧縮の有効化：

高度な索引圧縮は、CREATE/ALTER INDEX句のCOMPRESS ADVANCED副次句を指定することで有効化できます。圧縮時に新しい索引を自動的に作成できます。または、既存の索引を圧縮状態に再構築することもできます。

CREATE INDEX idxname ON tablename(col1, col2, col3) COMPRESS ADVANCED LOW/HIGH;

高度な索引圧縮では、プリフィックス・エントリの列数を指定する必要がないことに注意してください。リーフ・ブロックごとに自動的に計算されるためです。

高度な索引圧縮は、サポートされるすべての索引（プリフィックス・キー圧縮に適さない索引を含む）で問題なく使用できます。高度な索引圧縮を使用して索引を作成すると、すべての一意索引および非一意索引のサイズを減らすことができます（または、少なくとも負圧縮によるサイズ増大を防ぐことができます）。また同時に、索引に効率的にアクセスしながら圧縮率を大幅に改善することができます。

高度な索引圧縮には、次の制限があります。

- ビットマップ索引では、高度な索引圧縮がサポートされない
- ファンクション索引では、高度な索引圧縮がサポートされない

高度な索引圧縮を使用すると、Bツリー索引を簡単に圧縮できます。このとき、圧縮でメリットを得られるすべての索引リーフ・ブロックが自動的に圧縮され、ブロックごとに最適なプリフィックス列長が自動的に計算されます。そのため、索引圧縮がブロック・レベルの真にローカルなものになり、索引セグメント全体にとって最適な圧縮率の実現を目指して、圧縮プリフィックス表の作成とリーフ・ブロックの圧縮方法の決定が、ブロックごとにローカルで行われます。

索引構成表（IOT）は本質的に索引であるため、高度な行圧縮または基本的な圧縮では圧縮できないことに注意してください。ただし、IOTは高度な索引圧縮（低）で圧縮できます。

Oracle RMANバックアップ圧縮

Oracle RMANがOracle Databaseと密接に統合されていることにより、すでに圧縮されているデータ/索引ブロックはOracle RMANバックアップ中に圧縮されたままであり、リカバリ前に解凍する必要がありません。そのため、ストレージ・コストが削減されます。また、バックアップとリストアにかかる時間も大幅に短縮される可能性があります。

バックアップの圧縮自体に関して言えば、Oracle RMAN基本圧縮では非常に高い圧縮率が実現しますが、CPUに負荷がかかる場合があります。CPUの可用性がバックアップとリストアのパフォーマンスを制限する要因となる可能性があります。

Advanced Compressionを使用したOracle RMANバックアップ圧縮には、3つのレベル（LOW、MEDIUM、HIGH）が用意されています。ストレージが節約される量は、LOWからHIGHに向かって増加しますが、CPUリソースの消費量が増加する可能性があります。LOW / MEDIUM / HIGHの圧縮は、通常はOracle RMAN基本圧縮よりもCPUの使用量を減らしながら、さまざまなレベルの圧縮を提供するように設計されています。

これら3つのレベルは、次のように分類されます。

- **HIGH** - ネットワーク速度に制約がある、速度が遅めなネットワークを介したバックアップに最適
- **MEDIUM** - ほとんどの環境に推奨され、圧縮比と速度のバランスに優れている
- **LOW** - バックアップ・スループットに一番影響が少なく、CPUリソースに制約がある環境に最適

I/Oに制約があるがCPUがアイドル状態の場合は、HIGHが最適な場合があります。HIGHの場合、CPUをより多く使用しますが、領域をもっとも多く節約するので、バックアップ・ファイルの書込みに必要なI/O数をもっとも少なくなります。その一方で、CPUに制約がある場合は、LOWまたはMEDIUMがより適切かもしれません。CPUの使用量が少なめで、通常、約80 %の領域が節約されます（Oracle RMANに含まれる基本圧縮と比較した場合）。

高度なLOB圧縮（SecureFiles LOBセグメントの場合）

通常、非構造化データをインラインで保存するのではなく、SecureFilesに移動（かつ高度なLOB圧縮を使用）することで、表の圧縮率を改善できます。

高度な行圧縮では、ブロックのサイズを削減するために重複排除が使用されます。インラインで保存された非構造化データの場合、その非構造化データの複製が同じブロックに存在する可能性は高くありません。つまり、ブロック内の非構造化データは、かなり大容量になる場合が多いですが、圧縮されません。このため、表の圧縮率が全般的に予想よりも低くなる可能性があります。

ただし、高度なLOB圧縮では異なる圧縮アルゴリズムが使用されるため、インラインで保存されているときには圧縮できない非構造化データを、多くの場合、SecureFiles LOBセグメントに保存されているときには圧縮できます。

高度なLOB圧縮には、3つのレベル（LOW、MEDIUM、HIGH）が用意されています。高度なLOB圧縮では、デフォルトでMEDIUMレベルが使用されます。MEDIUMレベルでは通常、少ないCPUオーバーヘッドで適度な圧縮が実行されます。高度なLOB圧縮のLOWは、高パフォーマンス向けに最適化されています。高度なLOB圧縮のLOWでは、より少ないCPU使用量で、MEDIUMによって達成される圧縮の約80 %が維持されます。高度なLOB圧縮のHIGHでは、ストレージの節約が最大になりますが、CPUオーバーヘッドも最大になります。

Oracle Databaseでは、SecureFilesデータが圧縮可能かどうかを検出され、業界標準の圧縮アルゴリズムを使用して圧縮されます。圧縮によって領域の削減が見込めない場合や、データがすでに圧縮されている場合、SecureFilesではそのようなLOBの圧縮は無効化されます。

テストの開始前の考慮事項

概念実証のテスト前計画の一環として、Oracleの圧縮で推奨される以下のベスト・プラクティスに留意し、（必要に応じて）実行してください。

- 最新リリースにアップグレードします（または重要なパッチを現在のリリースに適用します）。次のMOS Noteを参照してください。『[List of Critical Patches Required for Oracle 11g Table Compression](#)』（Doc ID 1061366.1）
- 概念実証の成功基準を定義します（データ、索引、バックアップ・ストレージ削減、問合せ/挿入/更新のパフォーマンス、バルク・ロード操作のパフォーマンス、アプリケーションのパフォーマンスなど）。
- Oracle E-Business Suiteを使用して概念実証を実行する場合は、Oracle MOS Note 2458161.1で詳細を確認してください。
- 圧縮列に“long”データタイプが含まれていないことを確認します。このデータタイプは、高度な行圧縮ではサポートされていません。
- 圧縮表/パーティションの列数が255未満であることを確認します（この制限はOracle Database 12c以降では解消されています）。詳細については、Oracle MOS Note 1612095.1を参照してください。
- CPUオーバーヘッドは通常、最小限で済みますが、高度な行圧縮または索引圧縮は、CPUサイクルが空いているシステムに対して導入することが適しています。圧縮により、一部のDML操作では、非常に小さなものではありませんが、追加のオーバーヘッドが発生するためです。
- 圧縮の各機能の最適なテスト環境は、本番環境に合わせて再現した環境です。この環境で、もっとも現実的な（圧縮前および圧縮後の）パフォーマンス比較と機能比較を行うことができます。

- 一般には、データベースに含まれる、アプリケーション関連のすべての表を圧縮することが推奨されますが、例外が1つあります。表がキューとして使用される場合です。つまり、行が表に挿入された後、大部分またはすべての行が削除され、その後さらに多くの行が挿入された後で再び削除される場合です。この種のアクティビティは、性質上一時的な行を常に圧縮するためのオーバーヘッドが発生するため、圧縮の適切なユースケースではありません。
- 高度な行圧縮は、Oracle Transparent Data Encryption (Oracle TDE) の表領域レベルの暗号化で適切に機能します。表領域レベルの暗号化では、圧縮は暗号化の前に行われるので、圧縮率が暗号化によって影響を受けることはありません。Oracle TDEの列レベルの暗号化では、暗号化は圧縮前に実行されるため、圧縮率に悪影響を与えます。

圧縮オーバーヘッドについて

概念実証を実行する前に、ユーザーは、圧縮解除のオーバーヘッドが問合せパフォーマンスに影響する可能性があるかと推測することがあります。しかし、実際のところ、そのような可能性は低いのが通常です。

高度な行圧縮、索引キー圧縮、および高度な索引圧縮が実行されたブロックは、ブロック・レベルで"圧縮解除"されることは一切なく、ほとんどの問合せでは、個々の行が圧縮解除されることもありません。ほとんどの問合せは、メモリ内のデータベース・ブロックの圧縮形式で直接実行可能であり、ほとんどの問合せ条件は、圧縮されたデータ形式で直接実行されます。そして、問合せの後の段階で必要となる値のみが圧縮解除されます。

通常、圧縮されたデータ/索引への問合せでのオーバーヘッドは増加せずに減少します。これは、一定量のユーザー・データに問合せを実行するためのI/Oが減少するからです。たとえば、3倍の圧縮率でデータを圧縮する場合、圧縮の使用時にそのデータをディスクから読み取ってバッファ・キャッシュへ読み込むためのI/Oの量はわずか3分の1になります。確かに、圧縮されたデータ・ブロック内のポインタを間接参照して列の値を抽出するために"追加の"命令サイクルがいくつか生じる可能性がありますが、通常はI/Oが減少することで十二分に相殺されます。

しかし、圧縮に関連する潜在的なオーバーヘッドを確実に把握するために、組織データ、アプリケーション、および本番環境で圧縮を使用する方法をシミュレートするテスト環境を使用してテストすることを推奨します。

圧縮率の改善

特定の表/パーティションの圧縮率は主に、その表やパーティションにブロック・レベルで存在する重複の量に関連します。重複の量が多いほど圧縮率は高くなり、データの一意性が高いほど、圧縮率は低くなります。データが一意である場合、表やパーティションは十分に圧縮されないか、まったく圧縮されない可能性が非常に高くなります。

特定の表の圧縮率をできる限り高めるために試せることがいくつかあります。通常どおりに、データ、アプリケーション、システムを使用して変更をテストし、このような変更が環境に及ぼす影響を確認する必要があります。

データのソート

データのロード時にデータを事前にソートしておくことで、表の圧縮率を改善できる場合があります。各列のデータのカーディナリティに基づいて、ソートする列を決定する必要があります。個別値が少ない列をソートできる場合は、圧縮率が向上する可能性があります。

ただし、事前にソートするには、ロード前にデータの準備が追加で必要となるため、追加される時間と圧縮率の向上を比較検討する必要があります。データのソートが圧縮率に影響を及ぼすかどうかを確認する場合は、独自のデータを使用してテストしてください。

より大きいブロック・サイズ

より大きいブロック上により重複したデータがある場合、より大きいブロックの方が圧縮率は高くなる可能性があります。

ただし、より大きいブロックの方が高度な行圧縮のより高い圧縮率を常年实现できるわけではありません。より大きいブロック・サイズが圧縮率に影響を及ぼすかどうかを確認する場合は、独自のデータを使用してテストしてください。

通常概念実証の内容

前述のとおり、圧縮の各機能の最適なテスト環境は、本番環境に合わせて再現した環境であることに留意することが重要です。この環境で、もっとも現実的な（圧縮前および圧縮後の）パフォーマンス比較と機能比較を行うことができます。

通常、圧縮の概念実証の実行中にテストされるAdvanced Compressionの機能には、以下が含まれます。

- 高度な行圧縮
- 高度な索引圧縮
- Oracle RMANバックアップ圧縮
- Advanced LOB Compression
- Oracle Data Guard REDO転送の圧縮

Advanced Compressionには他にも多数の機能が含まれますが、上記の機能が概念実証にもっとも一般的に含まれます。Advanced Compressionの他の機能を含めることも、上記の機能の一部を含めないこともできます。

実際の概念実証について、多くの場合、お客様は以下のように指摘しています。

- 概念実証テストの前に、構造化データ、索引、および非構造化データの圧縮率（ストレージ削減）を見積もります。圧縮アドバイザ（後述の説明を参照）を使用して、高度な行圧縮、高度な索引圧縮、および高度なLOB圧縮の圧縮率を見積もることができます。
- テストを使用して、圧縮によるパフォーマンスの向上と、圧縮がパフォーマンスに及ぼす可能性がある影響を確認します。これを確認するには、アプリケーションを実行し、テスト・プラットフォーム（本番環境のハードウェアと同様のもの）上でデータを使用して、圧縮前と圧縮後のパフォーマンスをプロファイリングします。アプリケーション・テストには、アプリケーション間問合せ、従来型パスによるロードとダイレクト・パスによるロードの両方を使用するバルク・ロード操作、単一行DML（従来型の挿入、更新、および削除操作）、およびOracle RMANバックアップを含めるのが理想です。
- 一般的にはすべての表を圧縮することが推奨されますが、中にはデータ・ストレージ要件のおよそ80 %以上を占める最大の表のみを圧縮することを選択している組織もあります。
- Data Guard REDO転送の圧縮を使用する場合の圧縮による節約の見積りについて詳しくは、MOS Note Doc ID 729551.1を参照してください。

- 使用可能な場合、オラクルのReal Application Testing (RAT) 製品が圧縮の概念実証に役立つツールとなることがあります。

無料の圧縮アドバイザ

Advanced Compressionを簡単に始める方法の1つは、圧縮アドバイザを使用することです。“DBMS_COMPRESSION” PL/SQLパッケージ（一般的に圧縮アドバイザと呼ばれる）が、データベース環境内の圧縮関連の情報を収集します。

これには、非圧縮のパーティション化された表とパーティション化されていない表の双方の圧縮率の見積りや、以前に圧縮された表/パーティションの行レベルの圧縮情報の収集が含まれます。圧縮アドバイザにより、圧縮に関連する使用状況を判断するのに必要な、ストレージの節約情報を入手できます。

実行中の圧縮アドバイザからは、圧縮アドバイザのターゲットであった特定の表またはパーティションの推定圧縮率が出力されます。その出力で、“COMPRESSION_RATIO”は、2.1のような数値で表されます。この特定の表またはパーティションで推定圧縮率が2.1倍である場合、この数値は、圧縮が有効になると、表またはパーティションのフットプリントが50 %程度削減されることを表しています。

DBMS_COMPRESSIONは、Oracle Database Enterprise Editionに付属しています。

マルチステップ・プロセスで示したような、圧縮の概念実証の簡単な例がどのようなものであるかは、以下の**付録A**を参照してください。

付録A

マルチステップ・プロセスでの圧縮の概念実証の例

関連するすべてのパッチを適用（任意で最新リリースにアップグレード）

- 最新リリースにアップグレード（該当する場合）
- パッチを適用

成功基準の定義

- データベースのパフォーマンス
- データベース・サイズ
- バックアップ領域の削減
- バックアップ時間/リストア時間
- アプリケーション・パフォーマンス
- Data Guard（該当する場合）
- Oracle Data Pumpによる圧縮（該当する場合）

圧縮アドバイザ -- DBMS_COMPRESSION

- データ/索引の圧縮率の見積もりを取得
- 圧縮される表/索引のリスト全体を確認

テスト環境での圧縮前のベースライン：本番ワークロード/データ

- データベースのパフォーマンス・データを収集（バルク・ロード操作、問合せ、挿入/更新などを含む）
- バックアップ/リストア時間を収集

- Data Guardのパフォーマンス・データを収集（該当する場合）
- 表/索引のデータベース・サイズを収集
- バックアップ・サイズを収集
- Data Pumpのファイル・サイズを収集
- AWRLレポートを収集

テスト環境での圧縮の実装

- 推奨される方法を使用して特定された、候補となるすべての表/索引を圧縮（オンライン/オフライン）
- バルク・ロード操作を実行してベースラインと比較
- SQL文（問合せ/挿入/更新）を実行してベースラインと比較
- 非実行の問合せ（存在する場合）に対してSQLチューニングの調整を実行
- 本番ワークロードを実行してパフォーマンスを検証
- AWRLレポートを収集してベースラインと比較

本番カットオーバーの準備

- 得られた教訓
- 概念実証で得られた利点および課題/解決策をすべて文書化
- カットオーバー計画を定義

Connect with us

+1.800.ORACLE1までご連絡いただくか、[oracle.com](https://www.oracle.com)をご覧ください。北米以外の地域では、[oracle.com/contact](https://www.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。