



Oracle ホワイト・ペーパー

2014 年 10 月

Oracle E-Business Suite リリース 12.2.n の アップグレード・ダウンタイム最小化の ベスト・プラクティス

ORACLE[®] 12

E-BUSINESS SUITE

概要	4
はじめに	4
このドキュメントの使用	4
準備と計画	5
パフォーマンスの問題の解決	5
R12.2.n のアップグレード・プロセスの概要	6
オンライン・パッチ	7
ダウンタイム・モード	7
オンライン・パッチとカスタム・コード	8
パフォーマンスの問題の予測	9
サマリー	10
準備と計画 - アップグレード前の環境/アクティビティ	10
準備と計画 - リリース 12.2.n のアップグレード・アクティビティ	11
パフォーマンスの問題の解決 - 診断の使用	21
パフォーマンスの問題の解決 - 一般的な解決策	26
主なリソースおよびドキュメント	29
準備と計画 - アップグレード前の環境/アクティビティ	30
OATM (Oracle Applications 表領域モデル)	32
複数組織アーキテクチャ(11i からのアップグレードの場合のみ)	33
古いデータのパージ	33
インタフェース・データのパージ	33
CBO 統計の収集	34
固定オブジェクト統計およびディクショナリ統計	35
MRC スキーマの削除(11i からのアップグレードの場合のみ)	35
準備と計画 - リリース 12.2.n のアップグレード・アクティビティ	35
計画	35
アップグレード・パスとデータベースの動作保証	36
パフォーマンスとテクノロジーの修正	37
不要なワークロード/オーバーヘッドの削除	37
アップグレードと初期化パラメータ	41
動的/ハイパー・スレッディング	49
12.2 の中間層のサイズ設定に関するガイドライン	50
オンライン・パッチでのダウンタイム・モードの使用	50
CBO 統計の収集	51

ORACLE[®] **E-BUSINESS SUITE** **12**

パフォーマンスの問題の解決	61
アップグレード中に収集する診断	61
便利なスクリプト	84
一般的な解決策	94



概要

このドキュメントでは、必要な実装とリソースを最小化しながら、Oracle E-Business Suite をリリース 12.2.n にアップグレードする際のダウンタイムも最小化する方法を説明します。

特に、次の内容に役立ちます。

- 最終繰返し中のダウンタイムの最小化。
- パフォーマンスの問題解決に使用するテスト繰返し回数の最小化。
- 問題の早期識別。理想的には最初のテスト繰返し時。
- 最終繰返しの簡略化による、必要なリソースの最小化

はじめに

このドキュメントの使用

「サマリー」の項には推奨事項の概要が記載されていますので、すべて確認してください。

詳細が説明されている「準備と計画」および「パフォーマンスの問題の解決」の項には、推奨事項に関する詳しい説明と指示が記載されています。また、気を付ける必要がある重要な注意事項が含まれている場合があります。通常、この項は、参照用としてご覧ください。

推奨のアプローチは、2つの部分で構成されています。

- 準備と計画
- パフォーマンスの問題の解決

どちらも次に説明します。



準備と計画

内容は次のとおりです。

- 推奨される構成または設定。
- リリース 12.2.n のアップグレード・プロセスの一環として実行する必要があるタスク。
- すべての実行で取得する必要がある診断。
- 一般的なパフォーマンス・モニタリングに関するアドバイス。
- 実行ごとのアップグレード・パラメータ(AD パラレル・ワーカー、バッチ・サイズ、`max_parallel_servers`)の決定に関するアドバイス。

通常、すべての実装に当てはまります。

パフォーマンスの問題の解決

それぞれの実行で発生した特定のパフォーマンスの問題を解決する方法についてのアドバイスです。

パフォーマンスの問題が発生することのある特定の領域やジョブは存在しますが、未知/未報告のパフォーマンスの問題も多く存在します。お客様ごとに、どの部分でパフォーマンスの問題が発生するかを予測することはできません。



R12.2.nのアップグレード・プロセスの概要

R12.2.n へのアップグレードでは、次の順序で主なパッチが適用されます。

- 最新の CUP および 12.2.0 へのアップグレード
- オンライン・パッチの有効化
- R12.AD.C.Delta.n および R12.TXK.C.Delta.n
- 12.2.n RUP パッチ。

オンライン・パッチの有効化までのすべてのパッチ適用で、自動パッチ(adpatch)が行われます。

オンライン・パッチ(ADOP)と、基礎となるエディションベースの再定義が行われるのは、オンライン・パッチの有効化後の R12.AD.C.Delta.n、R12.TXK.C.Delta.n、12.2.n RUP および後続のパッチのみです。

オンライン・パッチ方式が実行されるのは R12.2.n へのアップグレードの後半の段階ですが、これらのパッチが実行されるのは、ダウンタイムの間(アプリケーションが実行されていないとき)であることに注意してください。

R12.AD.C.Delta.5 以降では、ダウンタイム・モード(後で説明)を使用して、12.2.n RUP パッチを適用できます。

そのため、R12.2.0 へのアップグレードに関するパフォーマンスのアドバイスと課題は、R12.1.n へのアップグレードと概ね同じです。同じ方法で、パッチの多くを適用できます(AutoPatch)。

ただし、オンライン・パッチの有効化パッチ、オンライン・パッチおよび 12.2.n 用のリリースアップデート・パックは、メインの R12.2.0 のアップグレードとは大きく異なるため、特に、内部 SQL や ADOP SQL (AD_ZD オブジェクトからの SQL 実行または AD_ZD オブジェクトでの SQL 実行など)で、新たなパフォーマンスの課題が生じる可能性があります。

オンライン・パッチ(ADOP)には、パフォーマンスを追跡する新しいログがありますが、主要な診断は、以前の Oracle E-Business アップグレードに使用されたもの(AWR レポート、AD ジョブ・タイミング・レポート、表示カーソル・レポートおよび SQT 抽出など)と同一であることを強調しておきます。



オンライン・パッチ

オンライン・パッチでは、**cutover** フェーズの期間は最低限になっています。ただし、これにより、一部で追加のオーバーヘッドが生じることは避けられません。具体例は次のとおりです。

- 複数のフェーズ(**prepare**、**apply**、**finalize**、**cutover**、**cleanup** など)
- 2つのエディションのファイル・システムの更新。
- チェックと検証
- EBR (エディションベースの再定義)では表のエディション化が不可。そのため、追加/更新は全エディションに対して行われます。以前は(**AutoPatch** では) **SQL** 設定操作が使用されていましたが、通常は、すべての行に対して同一の転送 **crossEdition** トリガーを起動する **AD_ZD_TABLE.apply** ユーティリティを使用して実行されます。これにより、パフォーマンス・オーバーヘッドが追加される場合があります。
- **prepare**、**apply**、**finalize** および **cleanup** の各フェーズ中に、実行中のアプリケーションとリソースを共有する場合があります。ただし、**R12.2.n** のアップグレード中は当てはまりません。

オンライン・パッチには、時間がかかることを予測しておく必要があります。

最も時間がかかるのは、**Apply**、**Cleanup**(**QUICK** ではなく **FULL** の場合)および **FS Clone** の各フェーズであることが多いです。

ADOP で使用可能な新しい診断はログです。これにより、アップグレードの特定のステップにタイムスタンプが付いたり、発生したエラーがリストされたりします。ただし、パフォーマンスの問題の原因となった **SQL** や根底にあるイベントが識別されるわけではないため、**AWR** レポート、**AD** ジョブ・タイミング・レポート、表示カーソル・レポートおよび **SQT** 抽出などを使用する必要があります。

ダウンタイム・モード

R12.AD.C.Delta.5 以降では、ダウンタイム・モードと呼ばれる、新しいパッチ適用機能を使用できます。

このモードで **Oracle E-Business Suite** のパッチを適用すると、**ADOP** ではまず、アプリケーション層のサービスが停止していることを確認してから、**Oracle E-Business Suite** のデータベースやファイル・システムの実行エディションへのパッチ適用が進められます。

ORACLE[®] 12 E-BUSINESS SUITE

ダウンタイム・モードでのパッチ適用では、オンライン・パッチ・サイクルは使用されません。ダウンタイム・モードでのパッチ適用プロセスは、オンライン・モードより短時間で終わりますが、システムのダウンタイムが長くなるというデメリットがあります(R12.2.n のアップグレードには当てはまりません)。

注意: Release 12.2 のパッチは、ダウンタイム・モードでは通常はテストされません。ダウンタイム・モードがサポートされるのは、明示的に文書化されている本番使用の場合か、(12.2.4 RUPなどで) Oracle Support または Oracle Development の指示がある場合のみです。

Oracle E-Business Suite リリース 12.2.4 のアップデート・パック(17919161)および Oracle E-Business Suite リリース 12.2.4 のオンライン・ヘルプ・パッチ(17919162)は、ダウンタイム・モードで適用することをお勧めします。経過時間の短縮に役立ちます。

オンライン・パッチとカスタム・コード

R12.2.n へのアップグレードに備え、カスタム・コードがオンライン・パッチ標準に準拠するよう、更新しておく必要があることに注意してください。

R12.2.n へのアップグレード前に、計画サイクルの初期段階で、次を実行しておく必要があります。

- オンライン・パッチ準備レポート。カスタム・データベース・オブジェクトに存在する、自動的に修正される問題と、手動での作業が必要な問題が識別されます。
- 手動修正準備レポート。
- グローバル標準準拠チェッカ・スクリプト。このスクリプトによってレポートされたエラーに対応します
- オンライン・パッチ・データベース準拠チェッカ。このレポートにリストされている開発標準違反を修正します。

詳細は、次を参照してください

『Oracle E-Business Suite 開発者ガイド リリース 12.2』。第3項「オンライン・パッチ適用の準備」、第4項「リリース 12.2 アップグレードにおけるカスタマイズの準備」および第5項「オンライン・パッチ適用対応環境におけるカスタマイズの処理」



次に示す、My Oracle Support のドキュメントも参照してください。

- Deploying Customizations in Oracle E-Business Suite Release 12.2 (Document 1577661.1)
- Oracle E-Business Suite Technical Planning Guide, First Edition, Release 12.2 (Document 1585857.1)
- Using the Online Patching Readiness Report in Oracle E-Business Suite Release 12. (Document 1531121.1)

不要になったか、関係がないため、削除できるカスタマイズがないかどうか確認してください。R12.2 は機能が向上しているため、特定のカスタマイズが不要になる場合があります。

パフォーマンスの問題の予測

リリース 12.2.n のアップグレードが、(ダウンタイムという観点で)どのくらいかかるを予測するのは困難です。特に、データベース・サイズに基づいて、信頼性の高い、または正確な見積を作成するのは難しいことです。ダウンタイムは、使用しているモジュールや機能、ハードウェアの仕様によって異なります。

コストベース・オプティマイザや、競合(キューイング理論など)の背後にある数学的処理の特質により、これまでパフォーマンスに関する既知の問題がなかったジョブが、特定のお客様のところで、突然パフォーマンスの問題を発生させる場合があります。次に例を示します。

- CBO 統計、データベース初期化パラメータおよびその他の構成をわずかに変更するだけでも、異なる実行計画が使用される可能性があります。
- 負荷が特定のレベルに達すると、キュー(待機)時間が急激に長くなります。

モジュールやデータ配布、機能は、それらが使用される表やエンティティが異なる場合のみでなく、歴代のバージョンによっても、特定のアップグレード・ジョブや SQL の実行時間が大きく異なるか、別の実行計画が使用されます。あるお客様にとって効率のよい実行計画だったとしても、別のお客様にとってそうであるとはかぎりません。これまでに競合がなかったとしても、環境のパラメータや構成が競合の原因になることがあります。こうした状況では、わずかな変更が大きな影響を及ぼす可能性があります。

同様に、あるジョブで発生したパフォーマンスの問題が、同じジョブで以前に発生した問題と同じではないこともあります。



一方で、特定のモジュールを重点的に使用するすべてのお客様で、一部のジョブの実行に長い時間がかかる場合もあります。

サマリー

この項には、ダウンタイムを最小化するための推奨事項がまとめられています。より詳しい内容や説明、手順は、後続の項に記載されています。

準備と計画 - アップグレード前の環境/アクティビティ

パフォーマンスの問題を防ぐ手順やダウンタイムを短くする手順が記載されています。

これらはすべて、リリース 12.2.n へのアップグレード前に、アップグレード前の環境で実行するアクティビティです。

アップグレード前に推奨されるアクティビティのリスト

- 11i からアップグレードする場合は、OATM (Oracle Applications 表領域モデル) に変換します。12.0 または 12.1 からアップグレードする場合は、(OATM ではなく) 表領域移行ユーティリティを使用して、既存のオブジェクトを移行します。
- 新しい複数組織(Multi-Org)アーキテクチャに変換します(11i からのアップグレードの場合のみ)。
- アップグレード前に、不要になった古いデータをすべてパージします。
- 「自動インボイス」、「仕訳インポート」、「受注インポート」など、すべてのインタフェースを消去します。
- まだ残っている場合は、MRC スキーマを削除します(11i からのアップグレードの場合のみ)。
- ダウンタイムの開始が近づいたら、(すべてのスキーマに対して GATHER_AUTO オプションを指定して)スキーマ統計を収集します。FND_STATS または統計の収集コンカレント・プログラムを使用してください。
- 固定オブジェクト統計およびディクショナリ統計を収集します



準備と計画 - リリース 12.2.nのアップグレード・アクティビティ

計画

関連するアップグレード・ガイドを必ず参照してください。

- 『Oracle E-Business Suite リリース 11i から 12.2 へのアップグレード・ガイド』

または

- Oracle E-Business Suite リリース 12.0 および 12.1 から 12.2 へのアップグレード・ガイド
- 適用する最新のオンライン・パッチの有効化、R12.AD.C.Delta.n、R12.TXK.C.Delta.n および 12.2.n RUP を特定します。

12.2.n へのアップグレードを計画するに当たり必要な情報の概要は、My Oracle Support のドキュメントである『Oracle E-Business Suite Technical Planning Guide, First Edition, Release 12.2 (1585857.1)』を参照してください。このガイドの内容の大部分は、その他のリリース 12.2 のドキュメントから引用されています。

最新のパッチやドキュメントなどの情報は、次に示す My Oracle Support のドキュメントを参照してください。

- Oracle E-Business Suite Release Notes, Release 12.2 (Document 1320300.1)
- Oracle E-Business Suite Release 12.2: Suite-Wide Rollup and AD/TXK Delta Information (Document 1583092.1)
- Applying the Latest AD and TXK Release Update Packs to Oracle E-Business Suite Release 12.2 (Document 1617461.1)

必ず、最新の AD および TXK RUP を適用してください。ADOP (オンライン・パッチ)の(パフォーマンスの)修正が含まれています。

次の重要事項に注意してください。

- 本番システムと同一またはそれに相当するシステム(同じアーキテクチャ、CPU、IO、メモリーおよびデータ・ボリューム/期間)でアップグレードのパフォーマンス・テストを実行します。
- パフォーマンス・テストはできるだけ早期に開始してください。遅くならないようにしてください。



- パフォーマンス・テストを複数回(少なくとも 6 回)繰り返せる時間をとります。そうすることで、パフォーマンスの問題を解決し、稼働率を最大化して、パラメータをチューニングできます。
- テスト・サイクルを高速化します。

アップグレード・パスとデータベースの動作保証

次を確認してください。

- Database Preparation Guidelines for an Oracle E-Business Suite Release 12.2 Upgrade (Document 1349240.1)
- My Oracle Support の「動作保証」タブ
- Scenarios for Getting Started With Oracle E-Business Suite Release 12.2 (Document 1587195.1)

EBS で動作保証されているデータベースのバージョンとアップグレード・パス。

パフォーマンスとテクノロジーの修正

12.2 へのアップグレード前に適用するテクノロジー・パッチについては、12.2 へのアップグレード前に、My Oracle Support のドキュメントである『Oracle E-Business Suite Release 12.2: Consolidated List of Patches and Technology Bug Fixes (Document 1594274.1)』を参照してください。最新の EBS テクノロジー・コードレベル・チェッカのパッチ(17537119)に関する情報も記載されています。

推奨されているパフォーマンス・パッチがすべて適用されていることを確認してください。My Oracle Support のドキュメントである『Oracle E-Business Suite Pre-install Patches Report [Video] (Document 1448102.1)』を参照してください。

不要なワークロード/オーバーヘッドの削除

- カスタム・トリガーおよびビジネス・イベントを無効化します
- 必要に応じて、カスタム VPD ポリシーを確認して無効化します
- 有効化されている場合は、監査を無効化します。
- デバッグまたはロギングをすべて確認して無効化します。これは、すべてのレベル(サイト、職責、ユーザー・レベルなど)で実行してください。



- 可能な場合は、noarchivelog モードで実行します。
- フラッシュバック DB を無効化します。
- 大容量の表から TDE (透過的データ暗号化) を削除します。
- 複数ノードのシステムでは、AutoConfig を平行して実行することを検討してください。
- RDBMS アップグレード、プラットフォーム・アップグレード、OATM への変換は、別々のダウンタイム期間に分割します。
- インストールに関係のないタスクが実行されないよう、TUMS (手動アップグレード・スクリプト) を使用します(11i からのアップグレードのみ)
- アップグレード対象となる履歴データを最小限にします(要求時アップグレード) (11i からのアップグレードのみ)
- アップグレード前と後の技術アクティビティを並行して処理します
- アップグレード後ジョブにコンカレント・マネージャ・キューを別々に定義します

注意: 現在のところ、最新の 12.2.x RUP へのアップグレードが完了するまでは、12.2 Rapid Install アップグレードのファイル・システムに、ユーザーがノードを追加することは推奨されていないため、分散 AD および共有 APPL_TOP の使用は関係ありません。

リリース 12.2.n へのアップグレードでは、ステージングされたアプリケーション・システム (APPL_TOP) を使用できないことに注意してください。これには、複数の理由があります。

後続のパッチが前のパッチの情報に依存している場合があり、リリース 12.2.n のアップグレードは一連のパッチで構成されているため、通常は、パッチ情報のアップロードの遅延 (Defer Upload of Patch Information) を使用できないことに注意してください。

ADOP/自動パッチおよび pthfile オプションについては、次の文書を参照してください。

- 『Oracle E-Business Suite メンテナンス・ガイド リリース 12.2』
- Oracle E-Business Suite パッチ適用プロシージャ(12.1、12.0)
- 『Oracle Applications メンテナンス・ユーティリティ リリース 11i』

アップグレードと初期化パラメータ

32 コア以下の場合、最初は次のように設定します。

- parallel_max_servers = CPU コア数 x 2。



- AD パラレル・ワーカー - CPU コア数 x 1 から開始。可能な場合は、CPU コア数 x 1.5 まで増加。
- `job_queue_processes` = CPU コア数

32 コアを超える場合は、次のようにします。

- `parallel_max_servers` = CPU コア数 x 1。
- AD パラレル・ワーカー = CPU コア数 x 0.5 から 1.0 の間。

パフォーマンス診断に基づき、これらの値を小さくすることが必要な場合があります。また、大きくすることも可能です。AWR での競合とリソース(CPU)使用量のレベルが、これを決定する際の鍵となります。

ハードウェアに動的スレッディングまたはハイパー・スレッディングが使用されている場合は注意してください。そのような場合は、論理 CPU の数がコア数の数倍存在することがあります。その場合は、前述の値が、論理 CPU の数ではなく、CPU コアの数を使用して計算されていることに注意してください。

長時間実行されているすべての AD パラレル・ジョブで競合レベルが低い場合は、AD パラレル・ワーカーの数のみを増やします。同様に、すべてのパラレル問合せ/DML ジョブの競合レベルが低い場合は、`parallel_max_servers` のみを増やします。つまり、通常は、AD パラレル・ワーカーや `parallel_max_servers` を増やす前に、SQL チューニング(不十分な実行計画)および競合の問題を解決する必要があります。

AD パラレル・バッチ・サイズを 10,000 に設定します。

注意: エクステンツが小さい(128K = 16 ブロックなど)と、バッチ・サイズが著しく制限されます。そのため、選択されているバッチ・サイズに関係なく、1000 を大幅に下回るバッチがこれに当てはまります。

可能な場合は、SGA および PGA を最大化します。AWR アドバイザ統計からのフィードバックを確認しますが、次のことに注意してください。

- アドバイザ統計は、最後のスナップショット間隔のみを対象にレポートされています。
- 同じオブジェクトに多数のワーカーが同時にアクセスしている場合(AD パラレル・ジョブなど)、SGA ターゲット・アドバイザ(およびバッファ・プール・アドバイザ)では、増加している SGA から取得される物理読取りの減少が低く見積もられる場合があります。



これまでの経験から、一部の開始値を次に示します。

ログ・バッファ = 30 から 100MB

共有プール = 1 から 4GB

PGA ターゲット = 3 から 20GB

SGA/バッファ・キャッシュ = 数 GB: 過剰なページングの原因にならないよう余裕をもたせてください

指定されている場合は、`db_file_multiblock_read_count` を削除します。これが、Oracle E-Business Suite を正常に実行するための推奨値です。

Linux で HugePages を使用すると、SGA の割当てが改善してパフォーマンスが向上します。

`optimizer_dynamic_sampling` のレベルは、通常、2 以上にする必要があります(設定されていない場合は 2 がデフォルトです)。リリース 12.2.n のアップグレード中は、値を 4 にすることをお勧めしますが、アップグレード後は 2 に戻すか削除してください。

前述の各初期化パラメータ(`db_file_multiblock_read_count` を除く)の値は、通常の実行で使用する値と異なる場合があることに注意してください。そのため、リリース 12.2.n のアップグレードが完了した後は、必ず元に戻します。

その他の初期化パラメータについては、My Oracle Support のドキュメントである『Database Initialization Parameters for Oracle E-Business Suite Release 12 (Document 396009.1)』を参照してください。

12.2 の中間層のサイズ設定に関するガイドライン

管理対象インスタンス JVM のサイズ設定では、メモリーと CPU ドメインの両方を考慮する必要があります。

64 ビット環境では、大きなヒープ・サイズの割当ては推奨されていないため、クラスタの管理対象インスタンスを増やして、ターゲットの同時実行レベルまでスケール・アップします。

管理サーバーのサイズを設定する場合、デフォルト・サイズの 512MB は、ほとんどのインストールで不十分なため、XMS を少なくとも 1GB、XMX を 2GB に設定することをお勧めします。

サイズ設定に関する基本的な説明は、次のドキュメントに記載されています



- 『Oracle E-Business Suite インストレーション・ガイド: Rapid Install の使用方法 リリース 12.2』
- My Oracle Support のドキュメントである 『Managing Configuration of Oracle HTTP Server and Web Application Services in Oracle E-Business Suite Release 12.2 (Document 1905593.1)』

オンライン・パッチでのダウンタイム・モードの使用

Oracle E-Business Suite リリース 12.2.4 のアップデート・パック(17919161)および Oracle E-Business Suite リリース 12.2.4 のオンライン・ヘルプ・パッチ(17919162)は、ダウンタイム・モードで適用することをお勧めします。経過時間の短縮に役立ちます。

長期間実行されているジョブでの SQL チューニング問題の早期解決

実行計画が不十分な SQL では、特定リソース(I/O、CPU またはメモリーなど)の使用が不均衡になったり、(バッファ・キャッシュのブロックや I/O などでの)競合レベルが高くなったりします。一般的なパフォーマンスの問題を解決しようとしている場合や、アーキテクチャまたは構成を変更するメリットを調査している場合には、これが判断を誤る原因になり得ます。

時には、アーキテクチャや構成をこのようにして変更することで、実行計画が不十分で長期間実行されているジョブや、高いリソース使用率、競合が再び発生することがあります。そして、それらの再発が、他のメリットを上回ってしまいます。

長期間実行されているジョブの SQL チューニングの問題が解決すれば、アーキテクチャ、構成および AD パラレル・ワーカー/パラレル最大サーバー数を変更するメリットの調査や査定が非常にやりやすくなります。

動的/ハイパー・スレッディング

動的またはハイパー・スレッディングに注意してください。CPU サイクル当たりのスループットを最大化するために、動的/ハイパー・スレッディングを構成することが推奨されています(コアごとに 1 スレッドなど)。Oracle SPARC T4-4 では、動的スレッディングを max-ipc に設定すると、これを実行できます。

また、電源管理も無効化してください。電源管理により、スループットを最大化するためのスレッディングに支障が出たり、CPU 容量が調整されたりします。



CBO 統計の収集

スキーマ統計の収集

通常の場合では、リリース 12.2.n のアップグレード中に、スキーマ統計を手動で収集する必要はありません。多くの場合、(必要であれば) R12.2.0 のアップグレードの終了(フェーズ: last+63 など)に向けて、スクリプト `adsstats.sql` により自動的に収集されます。

ただし、CBO 統計は、リリース 12.2.n のすべてのアップグレードが終了し、システムをオンラインにしてユーザーに使用可能にする前に、`GATHER_AUTO` オプションを指定し、再度 `FND_STATS` (統計の収集コンカレント・プログラム)を使用して、すべての Oracle E-Business Suite スキーマを対象に収集する必要があります。

`adsstats.sql` スクリプトの実行に膨大な時間がかかっている場合は、次のようにして、アップグレード時間を短縮できます。

- テストの実行中に(`adsstats.sql` を使用して同じポイント: フェーズ: last+63 で)収集したスキーマ統計をエクスポートします。
- `adsstats.sql` を実行するかわりに、これらの統計をインポートします。

これは、大幅な時間短縮が見込まれる場合や、(`parallel_max_servers` がコア数の 2 倍などの適切な値に設定された状態の)パラレル実行が効果的に使用されていることが確認できている場合にのみ行います。詳細が説明されている項の推奨スクリプトを使用してください。

R12.2.0 のアップグレード中、`adsstats.sql` ジョブの実行に長い時間がかかっている場合は、行数が 1 億を超える表が存在する大規模な環境に役立つ、さらなる対応方法が 2 つあります。

- `adsstats.sql` に `dbms_stats.auto_sample_size` を使用します
- 長期間実行されている特定の表を低い評価率でサンプリングします

詳しくは、詳細が説明されている項を参照してください。

稼働後は、`adsstats.sql` によってロックが解除されたもので、再度ロックする必要がある統計に注意してください。

`adsstats.sql` は `adstats.sql` と混同されることが多いですが、`adstats.sql` は、統計の自動収集を無効化し、固定オブジェクト統計やディクショナリ統計を収集するものです。

固定オブジェクト統計およびディクショナリ統計

次の場合に、固定オブジェクト統計を収集する必要があります。



- Oracle E-Business Suite 全体のアップグレードの一部である、関連プラットフォームやデータベースのアップグレード後。
- SGA/PGA パラメータの変更後。
- リリース 12.2.n のアップグレード後、システムに代表的なアクティビティが発生した場合。

次の場合に、ディクショナリ統計を収集する必要があります。

- Oracle E-Business Suite 全体のアップグレードの一部である、関連プラットフォームやデータベースのアップグレード後。
- リリース 12.2.n のアップグレード後。
- OATM への移行後。

オンライン・パッチの有効化(および基礎となるエディションベースの再定義)による、固定オブジェクトとディクショナリ・オブジェクトに対する変更もあります。そのため、オンライン・パッチの有効化の内部 SQL、R12.2.n RUP およびその他のオンライン・パッチの実行に、時間がかかる場合があります。このような場合に、固定オブジェクト統計やディクショナリ統計の収集が役立ちます。特に、エディショニング・オブジェクトに効果的です。

リリース 12.2 のアップグレードでは、固定オブジェクト統計およびディクショナリ統計は、`adstats.sql` やその他のメソッドによって自動収集されません。

アップグレード中に、固定オブジェクト統計またはディクショナリ統計の収集が必要となる、特定の状況がその他にもあります(スキーマ統計のインポート前や、AWR が著しく大きくなり、SQLT または AWR レポートを実行する前など)。

実行計画が不十分な内部 SQL の数が非常に少ない場合や、オブジェクト数が少ない場合は、すべてのディクショナリ統計または固定オブジェクト統計を収集せずに、特定のオブジェクトを対象に統計を収集することが可能です。

詳しくは、詳細が説明されている項を参照してください。固定オブジェクト統計またはディクショナリ統計の収集に、`adstats.sql` を使用しないでください(データベース・アップグレードの一環である場合は除きます)。後続の詳細が説明されている項に記載されているとおり、API を直接使用します。



アップグレード中に収集される一般的な診断

アップグレードのすべてのパッチ/フェーズには、現在でも次の診断が必要です。これには、オンライン・パッチの有効化、12.2 のパッチ後のオンライン・パッチ(ADOP)フェーズ(準備、適用、確定、カットオーバー、クリーンアップなど)、特に 12.2.n RUP が含まれます。

自動ワークロード・リポジトリ(AWR)は、スナップショットを 30 分にして有効化する必要があります(デフォルトは 60 分です)。短期間で終わるアップグレードの場合は、さらに短いスナップショットがより適しています。

AWR の保持期間は、アップグレードが実行される期間と、(診断および分析を収集するための)その後の長い期間を網羅できるよう、十分に長くする必要があります。推奨値は N+7 日で、N は推定アップグレード時間を表しますが、期間が長ければ、その後の診断および統計の収集期間が長くなります。

リリース 12.2.n のアップグレードのテスト実行期間は、`statistics_level` を ALL (または `_rowsource_execution_statistics` を TRUE)に設定することを強くお勧めします。

次に、リリース 12.2.n のアップグレードにおけるパフォーマンスの問題を分析する目的を示します。

- 無駄なテストの繰返しの防止。発生した問題を初回で解決するソリューションの提供を目的としています。
- 各繰返しで調査されるパフォーマンスの問題数の最大化。
- 解決されたパフォーマンスの問題数の最大化。

この実行には、『Express Diagnosis of Oracle E-Business Suite Release 12 Upgrade Performance Issues (Document 1583752.1)』ホワイトペーパーに概要が説明されている手順を使用することをお勧めします。主な手順は次のとおりです。

Oracle E-Business Suite のアップグレード前の手順:

- `statistics_level` を ALL に設定するか、`_rowsource_execution_statistics` を TRUE に設定します(この他に実際の統計はありません)。

Oracle E-Business Suite のアップグレード中の手順:

- AWR またはカーソル・キャッシュ(メモリー)の上位 SQL をモニターします(「便利なスクリプト」を参照してください)。これは、内部 SQL かアプリケーション SQL です。負荷のかかる SQL が発生した場合は、Enterprise Manager を使用して識別することも可能です。



- 長期間実行されている SQL の表示カーソル・レポート(ALL+ALLSTATS)を取得します
- パラレル問合せまたは DML を使用する SQL の SQL モニター・レポートを取得します
- SQL の一部がいつ実行されたかを識別します(「便利なスクリプト」を参照してください)
- 長期間実行されている SQL とジョブを照合します(「便利なスクリプト」を参照してください)
- Oracle E-Business Suite のすべての表に関する CBO 統計をレポートします(「便利なスクリプト」を参照してください)

Oracle E-Business Suite のアップグレード後の手順:

- AD ジョブ・タイミング・レポートを取得します
- 長期間実行されているアップグレード・ジョブを特定します(「便利なスクリプト」を参照してください)
- 長期間実行されているジョブのファイル・バージョンを取得します
- AWR レポートを取得します
- afxplain.sql を実行して、詳細な CBO 統計、データベース・パラメータ、CBO パラメータおよびメタデータを取得します。これは、個々の SQL に対して実行します

さらに詳細な分析を実行する場合に必要なものを次に示します。

- AD ユーティリティおよびワーカー・ログ
- 長期間実行されている SQL の SQLT (TXTRACT)

また、Oracle Support および Oracle Development では、次が必要です。

- AWR エクスポート
- AD パラレル表のエクスポート

詳細は、後続の「パフォーマンスの問題の解決」の診断に関する項を参照してください。

オンライン・パッチの有効化 - アップグレード中に収集される特定の診断

オンライン・パッチの有効化パッチでは、次の診断も使用できます。

- ADZDSHOWDDL.sql



オンライン・パッチ - アップグレード中に収集される特定の診断

オンライン・パッチを使用して適用されるパッチには、次の診断も使用できます。

- ADOP ログ・ディレクトリ
- オンライン・パッチ・ログ・アナライザ・ユーティリティ adopscanlog (AD および TXK C Delta 4 パッチにより配信)は、adop ログ・ディレクトリを分析します
- ADZDSHOWLOG.sql/adzdshowlog.out
- ADOP サイクル・ステータス
- ADOP フェーズのステータスを識別する SQL
- AD および TXK C のパッチ・レベルを識別する SQL

詳細は、次を参照してください。

- 『Oracle E-Business Suite メンテナンス・ガイド リリース 12.2』
- 『Oracle E-Business Suite リリース 11i から 12.2 へのアップグレード・ガイド』
- Oracle E-Business Suite リリース 12.0 および 12.1 から 12.2 へのアップグレード・ガイド

My Oracle Support のドキュメントも参照してください。

- Useful 12.2 Documents For Customers And ATG Support Engineers (Document 1585889.1)
- 12.2 E-Business Suite - Collecting Online Patching and fs_clone Log Files (Document 1542162.1)

詳細は、後続の「パフォーマンスの問題の解決」の診断に関する項を参照してください。

パフォーマンスの問題の解決 - 診断の使用

ファイル・バージョン

アップグレードで実際に使用されたファイルのバージョンを検出するには、統合ドライバ(またはパッチ・ドライバ)を確認します。ファイル・システムまたはデータベースにあるファイルやオブジェクトのバージョンは、後続のパッチで適用された、より新しいバージョンの可能性があるため、信頼しないでください。



Statistics_Level = ALL (または_rowsource_execution_statistics = TRUE)

これは、(SQLT および表示カーソル・レポートの)各実行計画行に実際に使用された実行統計(経過時間、物理読取り、バッファ取得を含む)を確認する最も簡単な方法です。SQL トレースおよび TKPROF の代替機能では、編集標準コードが必要です。

これにより、最初の繰返しで問題の根本原因を突きとめられるため、解決が早くなります。

また、初期化パラメータ_rowsource_execution_statistics を TRUE (statistics_level は'TYPICAL')に設定しても、実際に使用された同じ実行統計を収集できます。この場合、statistics_level=ALL とするよりも、オーバーヘッドを低くできます。

AWR が有効化されている状態で statistics_level を ALL に設定すると、SYSAUX 表領域(表 WRH\$_LATCH_CHILDREN)の使用率が大幅に増加する可能性があることに注意してください。そのため、この表領域の使用率をモニターしてください。

AWR エクスポート

AWR エクスポートを使用すると、AWR 表を別の環境にロードできます。その後、分析して、ジョブや SQL、オブジェクトなどの待機が発生している場所とそのパターンを検出できます。

AWR レポート

アップグレードが実行されている期間の AWR レポートを取得します。また、長期間実行されているジョブの期間と、個々のスナップショットの AWR レポートも取得します。

これらから、次のことを識別できます。

- 長期間実行されている SQL (SQL 統計/SQL Ordered By)
- 競合およびボトルネック(時間設定された上位 5 のフォアグラウンド・イベント/フォアグラウンド待機イベント)
- CPU 使用率

特定の待機のレベルが高い場合は、システム全体の構成やリソースの問題であると仮定する前に、まず、その待機が長期間実行されている特定の SQL やジョブでのみ発生しているかどうかを確認してください。

特定の SQL のアクティブ・セッション履歴レポートは、各行ソースまたはオブジェクトの待機/イベントの識別にも便利です。



詳細は、My Oracle Support のドキュメントである『Performance Diagnosis with Automatic Workload Repository (Document 1674086.1)』を参照してください。

固定オブジェクト統計およびディクショナリ統計は、`statistics_level` が ALL に設定されている場合や、保持期間が長い場合、スナップショット間隔が短い場合は特に、AWR レポートの実行前に収集する必要があることに注意してください。

AD パラレル表

AD_PARALLEL_UPDATES、AD_PARALLEL_UPDATE_UNITS 表には、実際に処理された行、バッチ数、進捗状況、(ロック/スリープまたはデータ配分の問題を示す可能性がある)長期間実行中のバッチに関する情報があります。

AD_TASK_TIMING は、ジョブやワーカーの開始時間と終了時間を示し、長期間実行されているすべてのジョブの識別、および長期間実行されている SQL や(AWR での)パフォーマンスの問題と特定のジョブとの照合に役立ちます。また、分割待機が原因で使用率が低いワーカー(および潜在的なリソース使用率)の識別にも役立ちます。

AD ジョブ・タイミング・レポート

ジョブ・タイミング・レポート(`adtimrpt.sql`)は、時間のかかっている上位 100 のジョブをレポートします。また、アップグレードの各フェーズや、失敗、遅延、再開およびスキップされたジョブのタイミングもレポートします。詳細レポートの `adtimdet.sql` (`adt.lst`)は、フェーズ別および経過時間別にすべてのジョブをレポートします。

標準の `adtimrpt.sql` レポートの「時間がかかっている上位 100 のジョブ」のセクションには、AD パラレル・ジョブのすべてのワーカーが個別にリストされることに注意してください。そのため、上位 100 がわずかなジョブで占められている場合もあります。

かわりに、「便利なスクリプト」の「長期間実行されているアップグレード・ジョブ」の項にある SQL も使用できます。

AD ユーティリティおよびワーカー・ログ

AD ユーティリティおよびワーカー・ログも、いつ何が起こったかがより詳細にわかるため、診断に役立ちます。AD ワーカー・ログ(`adworknnn.log`)では、各ワーカーによって実行されたアクティビティとそのタイミングを確認できます。

ORACLE[®] 12

E-BUSINESS SUITE

SQL トレース - ジョブ固有

表示カーソル、SQL モニターおよび AWR から、パフォーマンスの問題解決に必要な診断を取得できない場合は、次のテスト実行でそのジョブ用に、10046 SQL トレース(レベル 16 - ALL_EXECUTIONS (waits を指定))を取得する必要があります。

DBMS_MONITOR (10g 以上)または Event++構文(11g 以上)を使用して、その他のセッションの SQL トレースを実行することは可能ですが、SQL トレースは、(一時的に SQL スクリプトを編集して)特定のジョブに対して有効化することをお勧めします。手順は、詳細が説明されている項を参照してください。

SQL 固有

(長期間実行中のジョブで)長期間実行されている SQL が識別されたら、次のものによって、個々の SQL が診断されます。

- 表示カーソル・レポート。これは、ジョブの実行中または直後に実行できます。ただし、カーソルが共有プール(メモリー)内に存在している必要があり、ない場合は実際の統計が失われます。
- (Oracle パラレル問合せ/DML を使用する SQL の) SQL モニター・レポート。これは、ジョブの実行中または直後に実行できます。
- afxplain.sql (詳細な CBO 統計、DB/CBO パラメータおよび SQL のメタデータを提供)
- XTRACT メソッドを使用した SQLT 出力。パフォーマンスに影響するため、パッチ/フェーズの完了後に取得してください。

最初の 3 つはアップグレードの実行中に提供されますが、最後の 1 つは完了後になります。

オンライン・パッチの有効化固有

オンライン・パッチの有効化パッチは、AutoPatch (adpatch)を使用して適用されます。前述の一般的な診断に加え、次のスクリプトの出力も役に立ちます。

```
$ sqlplus apps @$AD_TOP/sql/ADZDSHOWDDLSQL
```

ADOP ログおよび診断

すべての ADOP ログは、<INSTALL BASE>/fs_ne/EBApps/log/adop ディレクトリなど、エディション化されていないファイル・システム(fs_ne)に配置されています。

```
/u01/PROD/fs_ne/EBApps/log/adop
```




ディレクトリ全体の zip を作成するのが、最も簡単かつ迅速です。

関連するメイン・ファイルは、各種の ADOP ログ(adop_YYYYMMDD_HHMISS.log など)です。

ただし、adzdshowlog.out、adworker*.log、u*.log、u*.lgi、admrgpch*.log ファイルもすべて有用で、同じパスにあります。

adrelink.log、adlibin.log、adlibout.log、adworknnn.log などの AD ユーティリティおよびワーカー・ログ(ADOP 以外)も、同じ場所に格納されています。

オンライン・パッチ・ログ・アナライザ・ユーティリティ(adopscanlog)

これは、AD および TXK C Delta 4 パッチによって配信されたものです。

このユーティリティは、adop ログ・ディレクトリにエラーや警告があるかどうかを分析し、adop の実行中に発生した問題をユーザーが迅速に識別できるよう、メッセージを表示します。そのため、ログ・ファイルを手動で確認する選択肢となります。

ADZDSHOWLOG.sql/adzdshowlog.out

これは、AD_ZD_LOGS 表のコンテンツをレポートします。オンライン・パッチの進捗状況に関するメッセージが、タイムスタンプ付きで記載されています。

ADOP サイクル・ステータス

adop -status を使用して、ADOP サイクルの現在のステータスを確認できます。

SQL を使用した ADOP、AD および TXK C のパッチ・レベルのステータスの確認

SQL を直接使用して、ADOP フェーズや AD/TXK のパッチ・レベルのステータスを確認することもできます。

長期間実行されている SQL、競合およびチューニング

長期間実行されているジョブまたは SQL の場合は、適切な実行計画が使用されているかどうかの調査から開始するのが最適です。実行計画が不十分であると(またはわずかに最善には及ばない場合でも)、特に、ある競合が特定のジョブでのみ発生する場合は、その競合の根本原因になる可能性があります。

最善ではない実行計画が原因の不要な競合が取り除かれると、AD パラレルまたはパラレル実行セッション間の少数の競合(特定の待機の 5 から 15%など)が有用なインディケータとなり、その大部分は使用可能なリソースから取得されます。



一部の待機イベントの待機時間全体は競合として分類できますが、これはすべての待機に該当するわけではなく、特に、db file 順次読取り、db file 散布読取り、ダイレクト・パス読取り/書き込みなどの I/O 待機には当てはまらないことに注意してください。

パフォーマンスの問題の解決 - 一般的な解決策

既知の問題

長期間実行されているジョブおよび SQL が識別されたら、既知の問題および可能性のある解決策や回避策について、My Oracle Support で確認してください。

ただし、修正または回避策で、発生した特定の問題を必ず解決できるわけではないことを念頭に置いておいてください。

問題が完全に同一であることを(診断から)確認できない場合、修正は適用できますが、問題が完全に解決されるまで、診断の収集と解決策の検索を続けてください。

カスタム索引

長期間実行されているジョブにカスタム索引を作成すると、新しい索引により、実行計画とパフォーマンスが大幅に向上する場合があります。

不十分な実行計画を対象とした SQL プロファイル

長期間実行されているジョブの実行計画が不十分であることがわかった場合は、SQL プロファイルを使用すると、CBO でより適切な実行計画が選択されるよう支援するヒントを適用できます。これを実行するには、SQL チューニングの専門知識が必要です。

実用的で安定した実行計画

実用的で安定した実行計画は、最適な実行計画ではありませんが、効率が悪いわけではなく、ジョブ時間の予測が可能で安定しています。

ほとんどの場合、最適な実行計画を提供し、無駄を最小限にする結合順序が 1 つは存在します。

AD パラレル・ジョブの場合は、実用的な実行計画が駆動表を使用してリードし、ある範囲の行 ID でアクセスして、後続の表でネスト・ループ結合と索引アクセスを使用します。(選択フィルタまたは表の合計行数が原因で)結合されている行数が最も少ないと考えられる表を最初に結合する必要があります。



オンライン・パッチの有効化/オンライン・パッチで長期間実行されている SQL

オンライン・パッチの有効化、R12.2.n RUP およびその他のオンライン・パッチには、実行計画が不十分で、長期間実行されている内部 SQL が存在する場合があります。

このような場合に、固定オブジェクト統計やディクショナリ統計の収集が役立ちます。特に、エディショニング・オブジェクトに効果的です。

長期間実行されている索引作成

通常、xdf および odf ジョブに当てはまります。

索引が作成されているスナップショットの AWR レポートを取得し、さらに調査します。

事前作成が役に立つ場合もありますが、別の解決策がある可能性が高いです。事前作成が必要な場合は、正しく実行するよう注意してください。詳しくは、詳細が説明されている項を参照してください。

長期間実行されている DML ジョブの索引における高レベルの競合

長期間実行中のジョブで負荷のかかる DML が実行されていて、競合レベルが高く、その競合が特定の索引に存在する場合は、ジョブの前に索引を削除し、後から再作成することを検討してください(その索引がそのときに使用されていない場合)。

索引は、まったく同一の定義で、並列に再作成するようにしてください。また、ALTER INDEX を実行して、並列度も元に戻してください(NOPARALLEL)。

高レベルの"enq: HW - contention"または"enq: HV - contention"

表および索引への挿入を行っている長期間実行中のジョブに、高レベルの待機"enq: HW - contention"または"enq: HV - contention"が存在する場合は、次のことを実行できます。

- 待機が主に特定の索引で発生している場合は、ジョブの前にその索引を削除し、後から再作成します(その索引がそのときに使用されていない場合)。
- 表および索引のエクステンツ・サイズを増やします。
- 表および索引にエクステンツを事前に割り当てます。
- 表および索引をパーティション化し、負荷を複数のパーティションに分散します。表および索引は、稼働後もパーティション化されることに注意してください。そのためこれは、本番環境で稼働した後にも明白なメリットがある場合にのみ実行します。



高レベルの REDO ログ待機であるログ・バッファ領域、ログ・ファイル同期など。

REDO ログ(特に、ログ・バッファ領域およびログ・ファイル同期)に関連した高レベルの待機が存在する場合は、次のことを検討してください。

- REDO ログの構成変更
- より高速なファイラへの移行
- サイズの増加。数値を増やすか、ログ並列度(隠しパラメータ `_log_parallelism_max`)を高くしてください。NOLOGGING の実行も可能ですが、お薦めではありません。

長期間実行されている統計の収集(adsstats.sql)

アップグレードで CBO 統計(adsstats.sql)の収集に時間がかかっている場合は、次の対応を検討してください。

- `parallel_max_servers` および `pga_aggregate_target` の増加。(adsstats.sql は、コンカレント・タスクが少ないとき、または存在しないときに実行してください。)
- すべてのノードを Oracle RAC システムで使用。
- テストの実行中に収集した統計のインポート(「準備と計画」の項で説明)。

特定のパフォーマンス問題を解決するための統計の収集または削除

パフォーマンスの問題は、特定のオブジェクトに関する統計の収集または削除が必要なことを示している場合があります。

それを実行する際の詳細は、それぞれの状況によって異なり、解決方法も簡単ではないことがあります。詳しくは、詳細が説明されている項を参照してください。

長期間実行されているマテリアライズド・ビューxdf/odfs

マテリアライズド・ビュー(MV)を作成している長期間実行中の xdf または odf ジョブがある場合は、大きな MV ログのクリーン・アップまたは切捨てを検討します(これには、MV の完全なリフレッシュが必要なことに注意してください)。

長期間実行されている不要なジョブ

長期間実行されているジョブが、実際に必要かどうかを確認します(使用されていない、またはライセンスされていないモジュールやローカライゼーションなど)。Oracle Support では、ジョブのスキップを推奨したり、修正や回避策を提供したりする場合があります。



AD パラレル・ワーカーの最大数

自動パッチ・ユーティリティにより、ワーカーの最大数が指定されます。その数はいくつかの要因によって異なり、特定の状況では、必要なワーカー数を下回る場合があります。

この最大数の計算方法や、問題の解決方法(プロセス・パラメータの増加またはその他のセッション/プロセスの削減)については、詳細が記載されている項を参照してください。

主なリソースおよびドキュメント

ドキュメント:

- 『Oracle E-Business Suite リリース 11i から 12.2 へのアップグレード・ガイド』。特に、「アップグレードの計画」(第 1 章)、「停止時間の短縮」(付録 F)および「要求時アップグレード」(付録 H)
- Oracle E-Business Suite リリース 12.0 および 12.1 から 12.2 へのアップグレード・ガイド。特に、「アップグレードの計画」(第 1 章)および「アップグレードの準備」(第 2 章)
- 『Oracle E-Business Suite セットアップ・ガイド リリース 12.2』
- 『Oracle E-Business Suite メンテナンス・ガイド リリース 12.2』
- 『Oracle E-Business Suite 開発者ガイド リリース 12.2』。特に、第 3 項「オンライン・パッチ適用の準備」、第 4 項「リリース 12.2 アップグレードにおけるカスタマイズの準備」および第 5 項「オンライン・パッチ適用対応環境におけるカスタマイズの処理」
- 『Oracle E-Business Suite インストレーション・ガイド: Rapid Install の使用方法 リリース 12.2』
- Oracle E-Business Suite 概要 リリース 12.2
- Oracle E-Business Suite パッチ適用プロシージャ リリース 12.1 およびリリース 12.0
- 『Oracle Applications メンテナンス・ユーティリティ リリース 11i』
- Oracle E-Business Suite 複数組織実装ガイド リリース 12.1
- Oracle E-Business Suite システム管理者ガイド - 構成 リリース 12.1
- 『Oracle Applications システム管理者ガイド - 構成 リリース 11i』
- 『Oracle Database VLDB およびパーティショニング』 - 「パラレル実行の仕組み」および「パラレル実行のための一般的なパラメータのチューニング」の項



- 『Oracle Database パフォーマンス・チューニング・ガイド』 - 「自動ワークロード・リポジトリ・データの転送」の項

My Oracle Support のドキュメント:

- Oracle E-Business Suite Release Notes, Release 12.2 (Document 1320300.1)
- Oracle E-Business Suite Release 12.2: Suite-Wide Rollup and AD/TXK Delta Information (Document 1583092.1)
- Applying the Latest AD and TXK Release Update Packs to Oracle E-Business Suite Release 12.2 (Document 1617461.1)
- Scenarios for Getting Started With Oracle E-Business Suite Release 12.2 (Document 1587195.1)
- Oracle E-Business Suite Technical Planning Guide, First Edition, Release 12.2 (Document 1585857.1)
- Useful 12.2 Documents For Customers And ATG Support Engineers (Document 1585889.1)
- Database Preparation Guidelines for an E-Business Suite Release 12.2 Upgrade (Document 1349240.1)
- Oracle E-Business Suite Release 12.2: Consolidated List of Patches and Technology Bug Fixes (Document 1594274.1)
- Best Practices for Gathering Statistics with Oracle E-Business Suite (Document 1586374.1)
- Fixed Objects Statistics Considerations (Document 798257.1)
- AD Command Line Options for Release R12 (Document 1078973.1)
- Oracle E-Business Suite Release 12.2: Upgrade Sizing and Best Practices (Document 1597531.1)
- Oracle E-Business Suite Pre-install Patches Report [Video] (Document 1448102.1)
- Oracle E-Business Suite Upgrades and Platform Migration (Document 1377213.1)
- Use of Multiple Organizations In Oracle Applications Release 11i (Document 210193.1)
- Reducing Your Oracle E-Business Suite Data Footprint using Archiving, Purging, and Information Lifecycle Management (Document 752322.1)
- How to Send a File to Oracle Support Using FTP (ftp.oracle.com) (Document 464666.1)



- Express Diagnosis of Oracle E-Business Suite Release 12 Upgrade Performance Issues (Document 1583752.1)
- Oracle E-Business Suite Performance Guide (Document 1672174.1)
- Oracle E-Business Suite SQL Trace and TKPROF Guide (Document 1674024.1)
- Performance Diagnosis with Automatic Workload Repository (Document 1674086.1)
- SQLT (SQLTXPLAIN) - Tool that helps to diagnose a SQL statement performing poorly or one that produces wrong results (Document 215187.1)
- How Does Adpatch Determine The Number Of Workers To Recommend? (Document 800024.1)
- Using AutoConfig to Manage System Configurations in Oracle E-Business Suite Release 12 (Document 387859.1) - section "5.1. Running AutoConfig in Parallel Across Multiple Nodes".
- How to verify or create a Database Object using an odf (adodfcmp) or xdf (FndXdfCmp) file? (Document 551325.1)
- Database Initialization Parameters for Oracle E-Business Suite Release 12 (Document 396009.1)
- R12.0 and R12.1: FAQ for the SLA Upgrade: SLA Pre-Upgrade, Post-Upgrade, and Hot Patch (Document 604893.1)
- Managing Configuration of Oracle HTTP Server and Web Application Services in Oracle E-Business Suite Release 12.2 (Document 1905593.1)
- Deploying Customizations in Oracle E-Business Suite Release 12.2 (Document 1577661.1)
- Using the Online Patching Readiness Report in Oracle E-Business Suite Release 12. (Document 1531121.1)
- HugePages on Oracle Linux 64-bit (Document 361468.1)
- Shell Script to Calculate Values Recommended Linux HugePages / HugeTLB Configuration (Document 401749.1)
- 12.2 E-Business Suite - Collecting Online Patching and fs_clone Log Files (Document 1542162.1)



準備と計画 - アップグレード前の環境/アクティビティ

パフォーマンスの問題を防ぐ手順やダウンタイムを短くする手順が記載されています。

これらはすべて、リリース 12.2.n へのアップグレード前に、アップグレード前の環境で実行するアクティビティです。

OATM (Oracle Applications表領域モデル)

既存のオブジェクトは、アップグレード前の環境で OATM に移行することをお勧めします。

11i からアップグレードする場合は、11i のうちに、OATM (Oracle Applications 表領域モデル)に変換します。

以前に環境がリリース 11i からリリース 12.0 または 12.1 にアップグレードされている場合は、アップグレード・プロセスにより、すべての新製品用の表領域の作成、新しい表領域モデル向けのデータベース構成、および新規オブジェクトの作成が行われています。ただし、既存のオブジェクトは自動的に移行されていません。そのため、(OATM ではなく)12.0 または 12.1 のうちに、表領域移行ユーティリティを使用して、既存のオブジェクトを移行してください。

まだ実行していない場合は、表領域移行ユーティリティを使用して、すぐにこの移行を実行することを強くお勧めします。オンライン・パッチが有効化された後はこのユーティリティの使用がサポートされていないため、環境をリリース 12.2.n にアップグレードした後は、移行を実行できないことに注意してください。

環境をすぐに OATM に移行しない場合は、表領域を継続して個別に管理する必要があります。

詳細は、次を参照してください。

- 『Oracle E-Business Suite リリース 11i から 12.2 へのアップグレード・ガイド』の第 1 章 (アップグレードの計画)。
- Oracle E-Business Suite リリース 12.0 および 12.1 から 12.2 へのアップグレード・ガイドの第 2 章(アップグレードの準備)。
- 『Oracle E-Business Suite セットアップ・ガイド リリース 12.2』
- Oracle E-Business Suite システム管理者ガイド - 構成 リリース 12.1
- 『Oracle Applications システム管理者ガイド - 構成 リリース 11i』



複数組織アーキテクチャ(11iからのアップグレードの場合のみ)

11i から移行している場合は、Oracle E-Business Suite 11i のうちに、新しい複数組織(Multi-Org)アーキテクチャに変換することが可能です。詳細は、『Oracle E-Business Suite リリース 11i から 12.2 へのアップグレード・ガイド』の第 1 章(アップグレードの計画)を参照してください。

関連項目:

- Oracle E-Business Suite 複数組織実装ガイド リリース 12.1
- 『Use of Multiple Organizations In Oracle Applications Release 11i (Document 210193.1)』

古いデータのパージ

アップグレード前に、不要になった古いデータをすべてパージします。

これは非常に長いプロセスで、パフォーマンス関連の問題、実行頻度および各実行でパージする期間を解決する必要があります。そのため、できるだけ早く開始してください。

11i には 130、R12 には 240 を超える標準のパージ・プログラムがあります。

パージ・プログラムの管理、構成、開始およびモニターには、(11i および R12 では)OAM パージ・プログラムを使用します。

パージ・ポータルにアクセスするには、次のナビゲーションを使用します。

「システム管理者」 > 「Oracle Applications Manager」 > 「パージ/重大アクティビティ」

パージ・プログラムのリストは、My Oracle Support のドキュメントである『Reducing Your Oracle E-Business Suite Data Footprint using Archiving, Purging, and Information Lifecycle Management (Document 752322.1)』に記載されています。これには、Archive_Purge_ILM_paper[1].pdf が含まれます。

インタフェース・データのパージ

「自動インボイス」、「仕訳インポート」、「受注インポート」など、すべてのインタフェースをパージします。



CBO統計の収集

リリース 11i、12.0 または 12.1 のうちに、すべてのスキーマに対して GATHER_AUTO オプションを指定してスキーマ統計を定期的に収集します。

その後、リリース 12.2.n へのアップグレードの開始が近づいたら、(すべてのスキーマに対して GATHER_AUTO オプションを指定して)スキーマ統計を収集します。

必ず、FND_STATS パッケージまたは統計の収集コンカレント・プログラムを使用してください。DBMS_STATS パッケージや分析は使用しないでください。

短期表/一時表のゼロまたは不正な統計

アップグレード前の環境の統計には、自動オプションで収集されたにもかかわらず、問題が存在する場合があります。

短期表、一時表またはグローバル一時(GT)表の統計を確認してください。処理(インポート/インタフェース/ステージングなど)用の行が含まれる表や、行がない(またはわずか数行の)表があります。

GT 表の場合、行が表示されるのは、それを挿入したセッションのみで、それらの行が保持されるのは、そのセッションの期間中のみです。これらの表の統計が収集された場合は、そのセッション(の表内)に存在する行の統計のみが収集され、0 行の統計になる可能性が高いです。

また、GT 表の統計はセッション固有ではないため、統計は 1 セットしかなく、その他すべてのセッションに表示されます。表の追加時に、セッション別に統計が収集されても、その他のセッションのデータのボリュームや配布が異なる場合は、不十分な実行計画になる場合があります。

同様に、その他の短期表、一時表またはステージング表では、表が空の場合でも統計が収集されていることがあります。

統計がゼロ(または行数が少ない統計)の場合、多数の行がある一時表/短期表の実行計画が不十分になる可能性があります。

これらの一時表、短期表またはグローバル一時表すべてで統計をなくし、動的サンプリングを使用するのが理想です。

ただし、特定の場合には、予想表を追加する対応(表にデータが追加されている場合)や、プロセスが実行されるたびに統計を収集する対応がとられています。Oracle E-Business Suite のシード済コードにより、統計が収集される場合もあります。



固定オブジェクト統計およびディクショナリ統計

これらは、アップグレード前の環境で収集し、修正するか最新の状態にしておく必要があります。

MRCスキーマの削除(11iからのアップグレードの場合のみ)

すべてのプログラムおよびレポートで、APPS スキーマが使用されています。MRC_APPS スキーマは不要になったため、削除すると領域が開放され、アップグレード中の処理オーバーヘッドが低減されます。

まだ存在している場合は、リリース 12.2.n へのアップグレード前にこのスキーマを削除してください。11.5.10 へのアップグレード時に、すでに削除されている必要があります。

『Oracle E-Business Suite リリース 11i から 12.2 へのアップグレード・ガイド』を参照してください。

準備と計画 - リリース 12.2.nのアップグレード・アクティビティ

注意: 1 つの Oracle RAC ノードでは、リリース 12.2.n のアップグレードのみを実行してください。リリース 12.2.n のアップグレードにおける経過時間の大部分は、DML (INSERT、UPDATE、DELETE)を実行するジョブが占めています。これらのジョブは、複数のワーカー/パラレル・サーバーを使用し、これらがすべて、同じオブジェクトとブロックに同時にアクセスします。そのため、そのクラスタのノード間における追加の通信(およびクラスタ待機)が、スループットを増やすために追加の CPU を使用することのメリットを大幅に上回ってしまいます。

計画

関連するアップグレード・ガイドを必ず参照してください。

- 『Oracle E-Business Suite リリース 11i から 12.2 へのアップグレード・ガイド』

または

- Oracle E-Business Suite リリース 12.0 および 12.1 から 12.2 へのアップグレード・ガイド

適用する最新のオンライン・パッチの有効化、R12.AD.C.Delta.n、R12.TXK.C.Delta.n および 12.2.n RUP を特定します。



最後の更新時(2014 年 10 月)には、次のものがありました。

- オンライン・パッチの有効化(12.2 CA オンライン有効化パッチ 13543062)
- R12.AD.C.Delta.5 パッチ(18283295)
- R12.TXK.C.Delta.5 パッチ(18288881)
- Oracle E-Business Suite リリース 12.2.4 アップデート・パック(17919161)
- Oracle E-Business Suite リリース 12.2.4 オンライン・ヘルプ・パッチ(17919162)

12.2.n へのアップグレードを計画するに当たり必要な情報の概要は、My Oracle Support のドキュメントである『Oracle E-Business Suite Technical Planning Guide, First Edition, Release 12.2 (1585857.1)』を参照してください。このガイドの内容の大部分は、その他のリリース 12.2 のドキュメントから引用されています。

最新のパッチやドキュメントなどの情報は、次に示す My Oracle Support のドキュメントを参照してください。

- Oracle E-Business Suite Release Notes, Release 12.2 (Document 1320300.1)
- Oracle E-Business Suite Release 12.2: Suite-Wide Rollup and AD/TXK Delta Information (Document 1583092.1)
- Applying the Latest AD and TXK Release Update Packs to Oracle E-Business Suite Release 12.2 (Document 1617461.1)

必ず、最新の AD および TXK RUP を適用してください。ADOP (オンライン・パッチ)の(パフォーマンスの)修正が含まれています。

アップグレード・パスとデータベースの動作保証

次を確認してください。

- Database Preparation Guidelines for an Oracle E-Business Suite Release 12.2 Upgrade (Document 1349240.1)
- My Oracle Support の「動作保証」タブ
- Scenarios for Getting Started With Oracle E-Business Suite Release 12.2 (Document 1587195.1)

EBS で動作保証されているデータベースのバージョンとアップグレード・パス。



パフォーマンスとテクノロジーの修正

12.2 へのアップグレード前に適用するテクノロジー・パッチについては、12.2 へのアップグレード前に、My Oracle Support のドキュメントである『Oracle E-Business Suite Release 12.2: Consolidated List of Patches and Technology Bug Fixes (Document 1594274.1)』を参照してください。最新の EBS テクノロジー・コードレベル・チェッカのパッチ(17537119)に関する情報も記載されています。

推奨されているパフォーマンス・パッチがすべて適用されていることを確認してください。My Oracle Support のドキュメントである『Oracle E-Business Suite Pre-install Patches Report [Video] (Document 1448102.1)』を参照してください。

不要なワークロード/オーバーヘッドの削除

ワークロードを削減するには、次を実行します。

- カスタム・トリガーおよびビジネス・イベントを無効化します。
- 必要に応じて、カスタム VPD ポリシーを確認して無効化します。
- 有効化されている場合は、監査を無効化します。『Oracle E-Business Suite リリース 11i から 12.2 へのアップグレード・ガイド』および Oracle E-Business Suite リリース 12.0 および 12.1 から 12.2 へのアップグレード・ガイドには、アップグレード前に、いずれにせよ Oracle AOL 監査証跡を無効化する必要があると記載されています。
- プロファイルを使用して有効化されていた、すべてのデバッグおよびロギングを確認して無効化します。これは、すべてのレベル(サイト、職責、ユーザー・レベルなど)で実行してください。
- 可能な場合は、noarchivelog モードで実行します。
- フラッシュバック DB を無効化します。
- 大容量の表から TDE (透過的データ暗号化)を削除します。
- 複数ノードのシステムでは、AutoConfig を平行して実行することを検討してください。次を参照してください。
 - 『Using AutoConfig to Manage System Configurations in Oracle E-Business Suite Release 12 (Document 387859.1) 』 - 5.1 項、複数ノード間での AutoConfig の同時実行

ORACLE[®] 12

E-BUSINESS SUITE

- 『Oracle E-Business Suite セットアップ・ガイド リリース 12.2』 - 第 3 章「技術的構成」の「高度な AutoConfig 機能および追加ユーティリティ」にある「複数ノード間での AutoConfig の同時実行」
- Oracle E-Business Suite 概要 リリース 12.2 - 第 5 章「技術的構成ツール」の管理タスクの項にある、AutoConfig のパラレル実行に関する項

注意: 現在のところ、最新の 12.2.x RUP へのアップグレードが完了するまでは、12.2 Rapid Install アップグレードのファイル・システムに、ユーザーがノードを追加することは推奨されていないため、分散 AD および共有 APPL_TOP の使用は関係ありません。

リリース 12.2.n へのアップグレードでは、ステージングされたアプリケーション・システム (APPL_TOP) を使用できないことに注意してください。これには、複数の理由があります。

後続のパッチが前のパッチの情報に依存している場合があり、リリース 12.2.n のアップグレードは一連のパッチ(R12.2.0 を継承する、オンライン・パッチの有効化、R12.AD.C.Delta.n、R12.TXK.C.Delta.n および 12.2.n RUP のすべてのパッチ)

で構成されているため、通常は、パッチ情報のアップロードの遅延 (Defer Upload of Patch Information) (My Oracle Support のドキュメントである『Patching Best Practices And Reducing Downtime (Document 225165.1)』を参照) を使用できないことに注意してください。

別々の手順への分割

本番アップグレードの前に、別々のダウンタイム期間に完了できるタスクを特定します。

次に示すいずれも候補になります。

- (現在の Oracle E-Business Suite レベルで動作保証されている最新版への) データベース・バージョンのアップグレード
- Oracle Applications 表領域モデル(OATM)への変換。
- その他、プラットフォーム(ハードウェアまたは OS)の計画済アップグレード。
- 11i からのアップグレードの場合は、『Oracle E-Business Suite リリース 11i から 12.2 へのアップグレード・ガイド』の「アップグレードの計画」(第 1 章)、「停止時間の短縮」(付録 F) および「要求時アップグレード」(付録 H) に示されているタスク。後者には、次の項で説明されているトピックが含まれており、アップグレード前およびアップグレード後のタスクが記載されています。



インストールに関係のないタスクが実行されないための TUMS (手動アップグレード・スクリプト)の使用(11i からのアップグレードのみ)

TUMS は、インストールには使用しないため、アップグレードから除外可能なタスクをリストします(使用していない製品に必要なタスクや、これまでに適用済のパッチなど)。TUMS はパッチ(18342870)で配信されるもので、システムの調査に必要なスクリプトを提供し、レポートを作成します。TUMS レポートを作成し、アップグレードの開始前に確認することを強くお勧めします。

パッチ 18342870 をダウンロードして適用し、スクリプト `adtums.sql` を実行して、レポート `tumsr12.html` を生成します。

`tumsr12.htm` レポートには、インストールでは使用しない手順(このドキュメントの TUMS 手順キーで識別)がリストされます。このレポートにリストされている手順は、無視しても問題ありません。

詳細は、『Oracle E-Business Suite リリース 11i から 12.2 へのアップグレード・ガイド』の「アップグレードの準備」(第 2 章)を参照してください。

アップグレードする履歴データの最小化(11i からのアップグレードの場合のみ)

『Oracle E-Business Suite リリース 11i から 12.2 へのアップグレード・ガイド』の「要求時アップグレード」(付録 H)を参照してください。

また、次に示す My Oracle Support のドキュメントも参照してください。

- R12.0 and R12.1: FAQ for the SLA Upgrade: SLA Pre-Upgrade, Post-Upgrade, and Hot Patch (Document 604893.1)
- Oracle E-Business Suite Release 12.2: Upgrade Sizing and Best Practices (Document 1597531.1)

11i から 12.2.0 へのアップグレード(特に Oracle Payables)では、アップグレードされるのが最新のデータのみの場合でも、一部の SLA アップグレード・ジョブで、すべての履歴データが処理されます。

リリース 12.2.0 のアップグレードでは、次のトランザクションがすべて移行されます - Payables (取引エンティティ、イベント、仕分見出し、仕分行)、Receivables (取引エンティティ)。



アップグレード前と後の技術アクティビティの並行化

効果的な場合には、技術アップグレード・タスクを平行して実行します。たとえば、データベースでアップグレード・ドライバが実行されている間に、リリース 12.2 の設定やシングル・サインオンの再登録をアプリケーション層で実行できます。

アップグレード後ジョブへの個別コンカレント・マネージャ・キューの定義

ダウンタイム中に行われるリリース 12.2.n へのアップグレードでは、多くのコンカレント・プログラムが自動的に発行されます。これらのプログラムの多くは、複数のスレッドで実行されるため、アップグレード前手順の一部を構成するコンカレント要求の合計数は非常に多くなります。

これらのプログラムは、システムが稼働すると、コンカレント・マネージャにより選択されて実行されます。その結果、それらの実行が、システムで進行中のコンカレント・ジョブの実行と交錯します。

これらの要求にコンカレント・マネージャ・キューを個別に定義し、アップグレード関連プロセスの干渉を排除して、標準のランタイム要求のアップグレード後テストを実行できるようにすることをお勧めします。

このような定義は、その他のマネージャ・キュー(標準など)がこれらの要求を選択しないようにし、この新しいマネージャ・キューがこれらの要求のみを処理できるよう、包括ルールと除外ルールを使用することで作成できます。

これを実行することにより、これらのアップグレード後のコンカレント・プログラムに割り当てられているターゲット・プロセスの数を制御できるようになりました(稼働シフトを使用して、これを動的に実行する場合も含みます)。新しいマネージャ・キュー、ターゲット・プロセス、包括/除外ルールおよび稼働シフトの構成に関するその他の詳細は、

Oracle E-Business ドキュメント・ライブラリの『Oracle Applications システム管理者ガイド』を参照してください。

詳細は、次を参照してください。

- My Oracle Support のドキュメントである『Oracle E-Business Suite Release 12.2: Upgrade Sizing and Best Practices (Document 1597531.1)』
- Oracle E-Business Suite リリース 12.0 および 12.1 から 12.2 へのアップグレード・ガイド
 - 付録 E のコンカレント・プロセスの管理に関する項



- 『Oracle E-Business Suite リリース 11i から 12.2 へのアップグレード・ガイド』 - 付録 K のコンカレント・プロセスの管理に関する項

アップグレードと初期化パラメータ

次に示す各初期化パラメータ(`db_file_multiblock_read_count`を除く)の値は、通常の実行で使用する値と異なる場合があることに注意してください。そのため、リリース 12.2.n のアップグレードが完了した後に、必ず元に戻してください。

その他の初期化パラメータについては、My Oracle Support のドキュメントである『Database Initialization Parameters for Oracle E-Business Suite Release 12 (Document 396009.1)』を参照してください。

AD パラレル・ワーカーと `parallel_max_servers`

パラレル実行スレッドの数は、初期化パラメータ `parallel_max_servers` によって決定されます。

これは、パラレル SQL および DML を使用するジョブの経過時間に影響します。そのようなジョブには、`apintbal.sql`、`apxlaupg.sql` および `adsstats.sql` があります。

(自動パッチおよび ADOP の)AD パラレル・ユーティリティで使用するワーカー数も構成できます。AD パラレル・ワーカーと `parallel_max_servers` の適切な数を取得することが重要です。

一般的には、次の設定が推奨されます。

- `parallel_max_servers` = CPU コア数 x 2。
- AD パラレル・ワーカー - CPU コア数 x 1 から開始。可能な場合は、CPU コア数 x 1.5 まで増加。

ただし、次の理由から、値が低い方がより最適なる可能性があります。

- 前述の値を推奨するのは、CPU 使用率を最大化するためです。ただし、I/O やメモリーなどに制約が生じます。そのため、メモリー使用率(スワッピング/過剰なページングなし)と、I/O 応答時間を確認してください(ヒストグラムでは、待機時間が長くなるとグラフの裾が厚くなります)。また、オブジェクト全体のバッファ・ビジー待機も確認してください(特定のオブジェクトで高レベルのときは、実行計画が最適でないことを示す場合が多いです)。
- システムの CPU/コアの数が増えると、オペレーティング・システムに存在する CPU の数も結果として増加します。ハードウェアに動的/ハイパー・スレッディング機能が使



用されていて、リリース 12.2.n のアップグレード期間にそれが無効化されていない場合は特に、これに該当します。

CPU コア数が 32 を下回る場合、通常は、前述の推奨事項が有効です。ただし、32 を超える場合は、競合レベルが著しく高くなり、ワーカー/スレッドの数を増やしたことによる CPU やリソース使用率のメリットを上回ってしまいます。

そのため、CPU コア数が 32 以上の場合は、前述のレベルより低いところから、`parallel_max_servers` と AD パラレル・ワーカーの設定を開始してください。次に例を示します。

- `parallel_max_servers` = CPU コア数 x 1。
- AD パラレル・ワーカー = CPU コア数 x 0.5 から 1.0 の間。

ハードウェアに動的スレッディングまたはハイパー・スレッディングが使用されている場合は注意してください。そのような場合は、論理 CPU の数がコア数の数倍存在することがあります。そのときには、論理 CPU の数ではなく、CPU コアの数を使用して前述の値を計算してください。

ハイパー・スレッディングまたは動的スレッディングが有効化されていて、CPU 使用率が低い場合は、ハイパー・スレッディングの構成を変更する必要があります。後続の「動的/ハイパー・スレッディング」の項を参照してください。

長時間実行されているすべての AD パラレル・ジョブで競合レベルが低い場合は、AD パラレル・ワーカーの数のみを増やします。同様に、長期間実行中のすべてのパラレル問合せ/DML ジョブで競合レベルが低い場合は、`parallel_max_servers` のみを増やします。つまり、通常は、AD パラレル・ワーカーや `parallel_max_servers` を増やす前に、SQL チューニング(不十分な実行計画)および競合の問題を解決する必要があります。

`parallel_servers_target` には、値を指定しないことに注意してください(デフォルト)。アップグレード・プロセスでは、通常、実行されているパラレル DML/SQL ジョブはいつでも一度に 1 つのみであるため、`parallel_max_servers` を下回る値は指定しないでください。

Job_queue_processes

同じように、`job_queue_processes` は、通常、CPU コア数に設定することをお勧めします。

CPU コア数が多い場合(または動的/ハイパー・スレッディングを使用している場合)は、`job_queue_processes` の値を小さくする方がより適切な場合があります。



ただし、`job_queue_processes` が多数使用されるのは、スクリプト `adobjcmp.sql/adutlrcmp.sql` のみで、オブジェクト(`UTL_RECOMP.RECOMP_PARALLEL` など)のコンパイルにジョブ・キュー(DB スケジューラ)が使用されます。

SGA、共有プール、PGA など。

可能な場合は、SGA および PGA を最大化します。

AWR アドバイザ統計および SQLT/表示カーソルのフィードバックを確認すると、(`statistics_level` が ALL に設定されているか `_rowsource_execution_statistics` が TRUE の場合は)ソート、ハッシュ結合などで使用されている一時領域が表示されます。

AWR アドバイザ統計では、必要な SGA が低く見積もられる場合があることに注意してください。

- アドバイザ統計がレポートするのは、すべて最後のスナップショット間隔のものです。AWR レポートが複数のスナップショット間隔を対象とし、最後のスナップショット間隔のワークロードが低い場合(検出されているプロセスが最後のスナップショットよりだいぶ前に完了しているなど)、アドバイザ統計の推奨値は低く見積もられている可能性があります。
- 同じオブジェクトに多数のワーカーが同時にアクセスしている場合(AD パラレル・ジョブなど)、SGA ターゲット・アドバイザ(およびバッファ・プール・アドバイザ)では、増加している SGA から取得される物理読取りの減少が低く見積もられる場合があります。

リリース 12 のアップグレードでは、AWR の SGA ターゲット・アドバイザが SGA のサイズが適切であることを示している場合でも、SGA サイズの増加時に、(`db file` 順次読取りなどの物理 I/O の減少による)パフォーマンスの向上が検知されました。推奨値の 2 倍の値を使用した場合でも、大きなメリットがありました。

SGA ターゲット・アドバイザが大きな値を推奨しない理由は不明ですが、リリース 12 のアップグレード・プロセスの特質であることが推測されます(各表のほとんどすべての行が繰り返し読み取られます。ただし、各 AD パラレルまたはパラレル実行スレッドがアクセスする行の割合は、ほんのわずかです)。

SGA/バッファ・キャッシュを非常に大きくすると、過剰なページングやスワッピングの原因となることに注意してください。また、バッファ内のブロックでは同じラッチを共有するため、過剰なラッチ待機の原因にもなります。



これまでの経験から、一部の開始値を次に示します。

ログ・バッファ = 30 から 100MB

共有プール = 1 から 4GB

PGA ターゲット = 3 から 20GB

SGA/バッファ・キャッシュ = 数 GB。過剰なページングの原因にならないよう余裕をもたせてください。

SGA および HugePages (Linux)

Linux で HugePages を使用すると、SGA の割当てが改善してパフォーマンスが向上します。

HugePages の構成方法の詳細は、My Oracle Support のドキュメントである [HugePages on Oracle Linux 64-bit \(Document 361468.1\)](#)』を参照してください。

推奨される HugePages の構成を計算するには、My Oracle Support のドキュメントである『[Shell Script to Calculate Values Recommended Linux HugePages / HugeTLB Configuration \(Document 401749.1\)](#)』を使用します。

PGA の割当てには、HugePages のメリットはないことに注意してください。

db_file_multiblock_read_count

指定されている場合は、db_file_multiblock_read_count を削除します。これが、Oracle E-Business Suite を正常に実行するための推奨値です。

動的サンプリング

リリース 12 のアップグレード中は、重要な大容量の表(特に補助元帳会計(XLA)スキーマ)が多数作成され、追加されます。CBO 統計が(adsstats.sql の実行前に)収集されていないことが多いため、動的サンプリングが役立つ場合があります。

リリース 12.2.n のアップグレードはバッチ・プロセスで、大部分の SQL は実行頻度が低いですが、大量の行に対して実行されます。そのため、動的サンプリングのレベルを高くしても、サンプル・サイズ(ブロック)の増加が著しくなければ、それほどのオーバーヘッドはありません。

Oracle E-Business Suite では、データベース・パラメータ optimizer_dynamic_sampling の設定は、通常推奨されていませんが、この場合にはデフォルト値(2)が使用されます。ただし、アップグレードの推奨値は 4 で、後からデフォルト(2)に戻す必要があります。



レベル4 の場合、サンプル・サイズは64 ブロックであることに注意してください。レベル5、6、7、8、9 および10 では、サンプル・サイズを128、256、512、1024、4086 およびすべてのブロックに増やしてください。

AD パラレル・バッチ・サイズ

ほとんどのインストールには、10K が適していますが、時間がある場合は、他の値もテストしてください。バッチ・サイズの変更は、通常は推奨されていません。駆動表のエクステント・サイズで制限されているため、いずれにしても、使用されている実際のバッチ・サイズの変更は制限されます。

バッチ・サイズを変更するのは、バッチ・サイズを増やしたり減らしたりすることで解決できる、重大なパフォーマンスの問題がある場合のみです(次を参照してください)。また、問題のSQLの実行計画が最適であることが確認された場合のみです。

リリース 12.2.0 のアップグレードで追加または変更された大規模な取引表のエクステント・サイズは、大きくすることをお勧めします。どのような場合にも当てはまるベスト・プラクティスは、急激にサイズが大きくなる表のエクステント・サイズを大きくしておくことです。

バッチ・サイズとその計算方法には、いくつかの制限事項があることに注意してください。次にそれを説明します。

エクステントが小さい(128K=16 ブロックなど)と、バッチ・サイズが著しく制限されます。選択されているバッチ・サイズに関係なく、1000 を大幅に下回るバッチがこれに当てはまります。

エクステントが大きい場合でも、ストレージ・オーバーヘッド(空領域、ブロック/行ヘッダー)と圧縮により、実際のバッチ・サイズは選択内容で大幅に変わります。

バッチ・サイズが大きい場合でも小さい場合でも、メリットとデメリットがあります。バッチ・サイズが大きければオーバーヘッドは小さくなり、不十分な実行計画のオーバーヘッドを低減できます。ただし、ワーカーが同じ操作を同時に実行しようとする危険性が高まるため、競合の原因となります(特に、複数のSQLがジョブに存在する場合)。また、リマインダ・バッチにより、ジョブの終了時間が遅れる危険性も高まります。

`AD_PARALLEL_UPDATE_UNITS` の `rows_processed` 列には、使用されている実際のバッチ・サイズに対応しない値が含まれる場合があることに注意してください。処理される行の値は通常、DML 文の後に `SQL%ROWCOUNT` が続くものであるため、SQL の実行に使用された行数ではなく、そのSQLの結果として挿入または更新された行数になります。ジョブに複数のSQL/DMLが存在する場合がありますが、それは最後のSQLから数えた数です。



AD パラレル・バッチ・サイズの変更

バッチ・サイズを変更するのは、バッチ・サイズを増やしたり減らしたりすることで解決できる、重大なパフォーマンスの問題が発生している場合のみです。実行する前には、実行計画が最適であることを確認してください。

- 同じ操作を何度も実行している繰返しの SQL のオーバーヘッドが高い場合 => バッチ・サイズを増やしてください。
- チューニングでは解決できない繰返しのフル・スキャン/範囲スキャンのオーバーヘッドが高い場合 => バッチ・サイズを増やしてください
- ジョブの開始時にワーカー間の競合が激しく、高く定義されたピークに発生している (ワーカーが同じアクティビティを同時に/同期して実行している) 場合 => ワーカーがすぐに同期しなくなるよう、バッチ・サイズを小さくしてください。
- リマインダ・バッチが大きいために、ジョブの最後にワーカー使用率が低い場合 => バッチ・サイズを小さくしてください。

バッチ・サイズ全体の変更は、非常に不明確なアクションであることに注意してください。あるジョブのパフォーマンスは向上するかもしれませんが、別のジョブのパフォーマンスが低下する可能性もあります。

個々の AD パラレル・ジョブのバッチ・サイズは、SQL スクリプトのバッチ・サイズをハード・コードすることで変更できます。ただし、(極端な値を選択する可能性や編集を誤る可能性もあるため)厳密にはサポートされていません。

例

```
--l_batch_size          VARCHAR2(30) := &&2;
l_batch_size            VARCHAR2(30) := 100000;
```

または、統合ドライバ自体。

例

```
#sql bom patch/115/sql cstmtaupg.sql none none none sqlplus_repeat &phase=upg+72
checkfile:bom:patch/115/sql:cstmtaupg.sql &un_inv &batchsize
sql bom patch/115/sql cstmtaupg.sql none none none sqlplus_repeat &phase=upg+72
checkfile:bom:patch/115/sql:cstmtaupg.sql &un_inv 100000
```

バッチ・サイズの計算とその制限

バッチ・サイズとその計算方法には、(2006 年 5 月 8 日版 adprupdb.pls 120.1 の現在のバージョンの AD_PARALLEL_UPDATES_PKG では)いくつかの制限事項があることに注意してください。



これは、実際のバッチ・サイズと入力されたサイズがかなり異なる場合も多いことを意味します。

また、エクステンツ・サイズが原因で、バッチ・サイズが制限されることもあります。

AD_PARALLEL_UPDATE_PKG のコードにより、各バッチに使用されるブロック数は次のように計算されます。

1. $B = (\text{batch_size} * \text{Average_row_length}) / 8192$
2. これは、10 の位に四捨五入されます。
3. 行の平均の長さが 0 か、結果が 0 の場合は、デフォルトの 200 ブロックが使用されます。

駆動表の各エクステンツは、それぞれに B ブロックが含まれる更新ユニットに分割され、各更新ユニットにはリマインダが配置されます。各更新ユニットはバッチとして使用されます。

次に制限事項を示します。

- B がエクステンツ・サイズより大きい場合は、エクステンツ全体が更新ユニットとして使用されます。その結果、実際のバッチ・サイズを、エクステンツに格納されている行数より大きくすることができなくなります。
- バッチに必要なブロック(B)の数を計算する際、AD_PARALLEL_UPDATES_PKG では、ブロック・ヘッダー、空領域または行ヘッダーのストレージ・オーバーヘッドが考慮されないため、必要なブロック数が低く見積もられる場合があります。行の平均の長さが短い場合は、この影響が大きくなります。
- 拡張圧縮または圧縮を使用している場合、行の平均の長さには影響しません。行の平均の長さは結果として多く見積もられ、各バッチに使用されるブロック数が多くなります。
- 必要なブロック数を 10 の位に四捨五入することで、精度がさらに下がります。
- 各エクステンツを B ブロックの更新ユニットに分割することも、精度を下げています。リマインダ・ブロックの数が B よりも大幅に少ない場合、このリマインダ・バッチはかなり小さくなります。エクステンツごとの更新ユニット数が少ない場合は、バッチ・サイズへの影響がさらに大きくなります。
- CBO 統計が追加されていない場合、200 ブロックかエクステンツ・サイズのうち、より小さい方がバッチ・サイズとして使用されます。



これらすべてのことから、実際のバッチは入力されたものより小さいことが多く、サイズが大幅に変わることがわかります。そのため、管理が難しい場合があります。

次に示す 3 つの例に、このことがすべて説明されています。

例 1

行の平均の長さが 200、バッチ・サイズは 10000、エクステント・サイズは 256 ブロックで、空領域とストレージ・オーバーヘッドは、行データが占めるブロックの 70%にとどまります。

$B = 244.1 \text{ ブロック} = 240 \text{ ブロック}$ (10 の位に四捨五入されています)。

このため、各エクステントの更新ユニットは、240 ブロックが 1 つと、16 ブロックが 1 つで 2 になります。

この結果、バッチ・サイズが 2 つになります。

$1 \times 240 \text{ ブロック} * (8192 \text{ バイト/行の平均の長さ } 200) * 70\% = \text{平均 } 6881 \text{ 行}$

$1 \times 16 \text{ ブロック} * (8192 \text{ バイト/行の平均の長さ } 200) * 70\% = \text{平均 } 459 \text{ 行。}$

例 2

行の平均の長さが 200、バッチ・サイズは 10000、エクステント・サイズは 16 ブロックで、空領域とストレージ・オーバーヘッドは、行データが占めるブロックの 70%にとどまります。

$B = 244.1 \text{ ブロック} = 240 \text{ ブロック}$ (10 の位に四捨五入されています)。

ただし、エクステント・サイズは 16 です。そのため、バッチは 16 ブロックに制限されます。

これにより、バッチ・サイズは、 $16 \text{ ブロック} * (8192/\text{実際の行の平均の長さ } 200) * 70\% = \text{平均 } 459$ になります。

例 3

行の平均の長さが 200、バッチ・サイズは 10000、エクステント・サイズは 256 ブロックで、空領域とストレージ・オーバーヘッドは、行データが占めるブロックの 70%にとどまり、圧縮率は 2 です。

$B = 244.1 \text{ ブロック} = 240 \text{ ブロック}$ (10 の位に四捨五入されています)。

このため、各エクステントの更新ユニットは 2 になります。1 つは 240 ブロックで、もう 1 つは 16 ブロックです。

この結果、バッチが 2 つになります。

$1 \times 240 \text{ ブロック} * (8192 \text{ バイト/行の平均の長さ } 200) * 70\% * \text{圧縮率 } 2 = \text{平均 } 13,762 \text{ 行}$



$1 \times 16 \text{ ブロック} \times (8192 \text{ バイト/行の平均の長さ } 200) \times 70\% \times \text{圧縮率 } 2 = \text{平均 } 918 \text{ 行。}$

バッチ・サイズが大きすぎる場合のリマインダ・バッチの問題

(実際の)バッチ・サイズが非常に大きい場合は、その後に続くその他すべてのバッチを処理する少数のバッチにより、ジョブが大幅に遅延するリスクがあります。

これをリマインダ・バッチと呼びます。

T がスレッド(AD ワーカー)数で、R が処理する合計行数、B はバッチ・サイズとします

また、B が R/T よりわずかに少ないとします($R/(T+1)$ など)

そのため、T のスレッド全体に割り当てられるバッチは、T+1 になります。

つまり、一方のスレッドには、もう一方のスレッドの 2 倍の負荷がかかります。

32 のワーカーで(駆動表の)192,000 行が処理された場合、 R/T は約 6,000 です。

このため、(エクステント・サイズが大きいと仮定した場合)バッチ・サイズを 10,000 にすると、リマインダ・バッチが発生します。そのため、この場合にふさわしいのは、より小さなバッチ・サイズ(1,000 など)です。

ただし、バッチ・サイズが小さいと、各バッチのオーバーヘッドが大きくなるため、バッチ・サイズを過剰に小さくしないでください。

動的/ハイパー・スレッディング

現在、多くのハードウェア・ベンダーが、動的スレッディングまたはハイパー・スレッディングのバージョンを提供しています。こうしたケースでは、オペレーティング・システム(そして最終的にはデータベース)に仮想 CPU(論理 CPU とも呼ばれます)を提供するよう、ハードウェアを構成できます。この仮想 CPU の数は、コア数よりも大幅に多くなります。

たとえば、Oracle SPARC T5-8 には、最大 8 つのソケット(物理 CPU)があり、それぞれのコア数が 16 であるため、合計で 128 コアになります。ただし、動的スレッディングの場合、実行できる仮想 CPU(またはスレッド)は最大 1024 です。

これは、オンライン・ユーザー数が多く、アクティブ・セッションの数が CPU 数の何倍にもなるアプリケーションに理想的です。従来、これらのアクティブ・セッションでは CPU を共有していたため、コンテキストの切替えが頻繁に行われていました。動的スレッディングにより、コンテキストの切替えが少なくなるため、パフォーマンスが向上します。



ただし、アクティブ・セッションの数が少なく、それぞれのワークロードが多い、リリース 12.2.n のアップグレードのような場合には必要ありません。この場合、各 AD パラレル(またはパラレル実行)ワーカー/スレッドで使用可能な CPU 容量が制限されます。

すべてが同じ操作を実行するワーカー/スレッドが大量(256 など)にある場合は、競合が激しくなり、CPU 使用率が高まることによるメリットをはるかに上回ることは明らかなです。

そのため、動的スレッディング/ハイパー・スレッディングは、たとえば各コアに 1 スレッドなど、CPU サイクルごとのスループットを最大化するために構成することをお勧めします。Oracle SPARC T4-4 および T5-8 では、動的スレッディングを max-ipc に設定することで対応してきました。

また、電源管理により、スループットを最大化するためのスレッディング構成に支障が出たり、CPU 容量が調整されたりするため、電源管理は無効化してください。

12.2 の中間層のサイズ設定に関するガイドライン

管理対象インスタンス JVM のサイズ設定では、メモリーと CPU ドメインの両方を考慮する必要があります。

64 ビット環境では、大きなヒープ・サイズの割当ては推奨されていないため、クラスタの管理対象インスタンスを増やして、ターゲットの同時実行レベルまでスケール・アップします。

管理サーバーのサイズを設定する場合、デフォルト・サイズの 512MB は、ほとんどのインストールで不十分なため、XMS を少なくとも 1GB、XXM を 2GB に設定することをお勧めします。

サイズ設定に関する基本的な説明は、次のドキュメントに記載されています

- 『Oracle E-Business Suite インストレーション・ガイド: Rapid Install の使用方法 リリース 12.2』
- My Oracle Support のドキュメントである [naging Configuration of Oracle HTTP Server and Web Application Services in Oracle E-Business Suite Release 12.2 \(Document 1905593.1\)](#)』

オンライン・パッチでのダウンタイム・モードの使用

Oracle E-Business Suite リリース 12.2.4 のアップデート・パック(17919161)および Oracle E-Business Suite リリース 12.2.4 のオンライン・ヘルプ・パッチ(17919162)は、ダウンタイム・モードで適用することをお勧めします。経過時間の短縮に役立ちます。



CBO統計の収集

詳細は、My Oracle Support のドキュメントである『Best Practices for Gathering Statistics with Oracle E-Business Suite (Document 1586374.1)』を参照してください。

Oracle E-Business Suite のスキーマ統計の収集

アップグレード前の環境では、Oracle E-Business Suite スキーマのスキーマ統計は、FND_STATS(または統計の収集コンカレント・プログラム)に GATHER_AUTO オプションを指定して収集する必要があります。

リリース 12.2.n のアップグレードでは、R12.2.0 のアップグレードの終了(last+63)に向けて、(adsstats.sql を使用して)すべての Oracle E-Business Suite のスキーマ統計が再度収集されます。adsstats.sql は、R12.2.n RUP では実行されないことに注意してください。

adsstats.sql と adstats.sql の違いは重要ですので気を付けてください。これらは、2つの異なるスクリプトです。

- adsstats.sql は、Oracle E-Business Suite のスキーマ統計を収集します。
- adstats.sql は、10g/11g の自動統計収集ジョブを無効化し、ディクショナリ統計および固定オブジェクト統計を収集します。

リリース 12.2.n のアップグレード後で、システムをユーザーに使用可能にする前に、GATHER_AUTO オプションを指定した FND_STATS (または統計の収集コンカレント・プログラム)を使用して、すべての Oracle E-Business Suite スキーマを対象に CBO 統計を再度収集します。DBMS_STATS や Analyze は使用しないでください。

原則として、Oracle E-Business Suite のスキーマの CBO 統計は、リリース 12.2.n のアップグレード中以外に収集する必要はありません。ただし、実際には、パフォーマンスの問題の結果として、リリース 12.2.n のアップグレードで作成された表や一時表/短期表では特に、特定オブジェクトの統計の収集または削除が必要になる場合があります(「パフォーマンスの問題の解決」の項を参照してください)。

Oracle E-Business Suite のスキーマ統計のインポート

adsstats.sql スクリプトの実行に長い時間がかかっている場合は、次のようにして、リリース 12.2.n のアップグレード時間を短縮できます。

- テストの実行中に(adsstats.sql を使用して同じポイント: フェーズ: last+63 で)収集した統計をエクスポートします。



- adsstats.sql を実行するかわりに、これらの統計をインポートします。

ただし、これにより、アップグレードが煩雑になります。そのため、大幅な時間短縮が見込まれ、この方法で実行することに確実性がある場合にのみ、これを実行してください。

parallel_max_servers がコア数の 2 倍未満の値に設定されている場合は、エクスポート/インポートに再度ソートする前に、parallel_max_servers の値の増加を検討することに注意してください。

統計のインポートにも時間がかかる場合は、インポート前の固定オブジェクト統計やディクショナリ統計の収集によって解決できる、パフォーマンスの問題が存在することがあります。通常、固定オブジェクト統計およびディクショナリ統計は、リリース 12.2 のアップグレードの完了後(R12.2.n RUP の後など)にのみ収集します。そのため、アクティビティが増えて、その経過時間が、統計をインポートしたことで得られるメリットを損なってしまいます。

統計は、次のようにしてエクスポートできます。

```
FND_STATS.backup_schema_stats(schema_name => 'ALL', statid => '<your statid>')
```

また、インポートは次のように行います。

```
FND_STATS.restore_schema_stats(schema_name => 'ALL', statid => '<your statid>')
```

FND_STATS.backup_schema_stats および restore_schema_stats を使用する必要がある理由は、2 つあります。

- Oracle E-Business Suite でサポートされているのが FND_STATS のみであるため。
- FND_STATS.backup_schema_stats および restore_schema_stats で実行されるのは、Oracle E-Business Suite のスキーマ(リリース 12.2 のデータベースに存在するオブジェクトの 85%程度)のバックアップと復元のみであるため。一方、DBMS_STATS.export_database_stats と import_database_stats では、すべてのスキーマ統計がエクスポート/インポートされ、固定オブジェクト統計、ディクショナリ統計およびシステム統計も収集されます。そのため、FND_STATS を使用した方が、負荷は大幅に低くなります。

次に、統計のエクスポートとインポートに推奨される SQL スクリプトを示します。

統計のエクスポート

```
set verify off
whenever sqlerror exit failure rollback;
whenever oserror exit failure rollback;
```



```

declare
begin
FND_STATS.BACKUP_SCHEMA_STATS(schemaname => 'ALL', statid => '<your identifier>');
exception
when others then
raise_application_error(-20000, sqlerrm || ' Error while executing
                                FND_STATS.BACKUP_SCHEMA_STATS package.');
```

統計のインポート

```

set verify off
whenever sqlerror exit failure rollback;
whenever oserror exit failure rollback;

declare
begin
FND_STATS.RESTORE_SCHEMA_STATS(schemaname => 'ALL', statid => '<your identifier>');
exception
when others then
raise_application_error(-20000, sqlerrm || ' Error while executing
                                FND_STATS.RESTORE_SCHEMA_STATS package.');
```

ただし、記述する際には、DBMS_STATS.IMPORT_SCHEMA_STATS の問題があります。スキーマに表がない場合は、完了前に FND_STATS.RESTORE_SCHEMA_STATS の失敗原因となる例外が発生します。Oracle E-Business Suite には、表のないスキーマがいくつかあります。

次のスクリプトを使用して回避します。

```

set verify off
whenever sqlerror exit failure rollback;
whenever oserror exit failure rollback;

DECLARE
CURSOR schema_cur IS
WITH
schema_tabcount AS
(SELECT c5, COUNT(*) num_obj
FROM fnd_stattab
WHERE statid = <your identifier>'
```



```

GROUP BY c5
)
SELECT upper(oracle_username) sname
FROM   fnd_oracle_userid
WHERE  oracle_id BETWEEN 900 AND 999
AND    read_only_flag = 'U'
AND    EXISTS
      (SELECT 'exists' from schema_tabcount WHERE c5 = upper(oracle_username) AND num_obj >0)
UNION ALL
SELECT DISTINCT upper(oracle_username) sname
FROM           fnd_oracle_userid a,
               fnd_product_installations b
WHERE          a.oracle_id = b.oracle_id
AND    EXISTS
      (SELECT 'exists' from schema_tabcount WHERE c5 = upper(oracle_username) AND num_obj >0)
ORDER BY      sname;

BEGIN
  FOR c_schema IN schema_cur
  LOOP
    BEGIN
      DBMS_STATS.IMPORT_SCHEMA_STATS(c_schema.sname, 'FND_STATTAB', <your
identifier>', 'APPLSYS');
    EXCEPTION
      WHEN OTHERS THEN
        dbms_output.put_line('Error on '||c_schema.sname);
        dbms_output.put_line(SUBSTR(SQLERRM,1,80));
    END;
  END LOOP;
EXCEPTION
WHEN OTHERS THEN
  raise_application_error(-20000, sqlerrm ||' Error while executing adsstats_restore');
end;
/
exit;

```

adsstats.sql の経過時間を短縮するその他の方法

R12.2.0 のアップグレード中、adsstats.sql ジョブの実行に長い時間がかかっている場合は、行数が 1 億を超える表が存在する大規模な環境に役立つ、さらなる対応方法が 2 つあります。

adsstats.sql に dbms_stats.auto_sample_size を使用します



11g の場合は、dbms_stats.auto_sample_size の estimate_percent が 10%ではなく(=0)で、FND_STATS.GATHER_SCHEMA_STATISTICS が呼び出されるように、adsstats.sql ジョブを修正します。現在これには、精度が 100%に近く、はるかに迅速に実行されるハッシュベースのサンプリングが使用されます。

より規模の大きな表(行数が 1 千万から 1 億など)の場合は、通常、評価率の 10%より少し速くなります。量がさらに多ければ、もっと速くなります。ただし、小規模な表では遅くなる可能性があります。

そのため、効果があるのは、R12 の新しい表または更新された表で、行が 1 億を超える大規模な環境のみです(XLA スキーマ、ZX スキーマ、AP_DBI_LOG、AP_LIABILITY_BALANCE、AP_INVOICE_DISTRIBUTIONS_ALL など)。

小規模な環境では、実際にかかる時間が長くなります。

長期間実行されている特定の表の低い評価率でのサンプリング

表 FND_STATS_HIST をチェックして、最も時間のかかっている表を確認します。

次の SQL を使用して、分析された表の数とともに、最後に実行された FND_STATS (request_id) を表示します。R12.2.0 のアップグレードで実行された adsstats.sql の<request_id>がわかります。

```
SELECT
request_id,
count(*) tables_analyzed,
TO_CHAR(MAX(last_gather_end_time),'DD-MON-YYYY HH24:MI:SS') max_end_time,
MAX(last_gather_end_time) max_end_time_int
FROM fnd_stats_hist
WHERE object_type = 'CASCADE'
AND last_gather_end_time IS NOT NULL
GROUP BY request_id
ORDER BY MAX(last_gather_end_time) DESC
```

request_id が判明したら、次の SQL で、統計の収集に最も時間のかかった表と、行数、ブロック数およびサンプル・サイズの情報がわかります。

```
SELECT fsh.schema_name, fsh.object_name, fsh.object_type,
ROUND((fsh.last_gather_end_time - fsh.last_gather_start_time)*24*3600,0) time_secs,
to_char(fsh.last_gather_start_time,'DD-MON-YYYY HH24:MI:SS') start_time,
to_char(fsh.last_gather_end_time,'DD-MON-YYYY HH24:MI:SS') end_time,
dt.num_rows,
dt.blocks,
dt.avg_row_len,
ROUND(DECODE(NVL(num_rows,0),0,NULL,(sample_size*100)/num_rows),0) est_pct,
```

ORACLE[®] 12

E-BUSINESS SUITE

```
sample_size
FROM fnd_stats_hist fsh
, dba_tables dt
WHERE dt.owner = fsh.schema_name
AND dt.table_name = fsh.object_name
AND fsh.request_id = <request_id>
AND fsh.object_type = 'CASCADE'
AND fsh.last_gather_end_time IS NOT NULL
ORDER BY (fsh.last_gather_end_time - fsh.last_gather_start_time)*24*3600 desc
```

最も長い時間実行されているものが、次の実行で、低い評価率(サンプル・サイズ)で実行される候補になります。低い評価率でサンプリングすることにより、CBO 統計の精度が低下するため、R12.2.0 のアクティビティ(R12.2.n RUP やアップグレード後のパッチ/ジョブなど)の後のパフォーマンスを注意して監視してください。

これらの表は、adsstats.sql の実行前に、低い評価率で直接収集できるようになります。または、adsstats.sql を直接編集し、スキーマ統計の収集(FND_STATS.GATHER_SCHEMA_STATISTICS)前に、低い評価率でこれらの表を収集できます。

コマンドは次のとおりです。

```
fnd_stats.gather_table_stats('<owner>', <table_name>', <estimate_percent>,
<parallel_degree>);
```

ここで、<parallel_degree>は、parallel_max_servers の設定です(adsstats.sql 自体の v_parallel など)。

次の値を使用することをお勧めします。

- 1 千万から 10 億行の表には 3%
- 10 億行を超える表には 1%

パーティション化された表の増分統計

増分統計の収集は、Oracle 11gR2 DBMS_STATS の新機能であり、FND_STATS で完全にサポートされています。グローバル統計は、表全体ではなく、新しいパーティションや変更されたパーティションのみをスキャンすることで導出され、大容量の Oracle E-Business Suite の表に非常に役立ちます。

アップグレード中に、パーティション化された表に影響することはありませんが、稼働後には、統計の収集が高速になります。

次に示す、パーティション化された新しい表に設定する必要があります: XLA_AE_HEADERS、XLA_AE_LINES、XLA_DISTRIBUTION_LINKS、XLA_EVENTS、



XLA_TRANSACTION_ENTITIES、XLA_TRIAL_BALANCES、AP_DBI_LOG、
AP_LIABILITY_BALANCE。

次のコマンドを実行して起動します。

```
dbms_stats.set_table_prefs('<schema_name>', '<table_name>', 'INCREMENTAL', 'TRUE');
```

ロックされた統計

一部の表では、ある理由で統計がロックされます。たとえば、表が一時表または短期表(あるいはグローバル一時表)で、代表的な統計のセットが収集されているため、常に使用する必要があり、上書きできない場合です。

そのような場合には、adsstats.sql スクリプトで統計のロックが解除され、再度収集されることに注意してください。また、リリース 12.2.n のアップグレード中、これらの表は空である(または代表的なデータ・ボリュームがない)ことが多いです。

ロックして除外できるよう、アップグレード前にそれらのリストを取得しておくくと便利です。

Oracle E-Business スイートでは、DBMS_STATS.LOCK_TABLE_STATS プロシージャを使用して統計がロックされ、FND_STATS.LOAD_XCLUD_TAB プロシージャで、後続の FND_STATS 収集ジョブから除外されます。

DBMS_STATS.LOCK_TABLE_STATS プロシージャで統計がロックされたら、*FND_STATS* では、後続の統計収集ジョブで統計がロックされている表がスキップされます。スキーマの収集/表統計コンカレント・プログラムにより、要求ログ・ファイルに、表では統計がロックされているというメッセージが表示されます。

FND_STATS.LOAD_XCLUD_TAB から *FND_STATS* に、スキーマ統計の収集コンカレント・プログラムからその表をスキップするよう伝達されます。これにより、*FND_EXCLUDE_TABLE_STATS* 表にエントリがシードされます。

固定オブジェクトおよびディクショナリ統計

これらは、アップグレード前の環境で収集しておく必要があります。

Oracle Database 10g 以降では、大部分のディクショナリ SQL に CBO が使用されるため、ディクショナリ・オブジェクトおよび固定オブジェクトに関する統計を収集する必要があります。

固定オブジェクト(X\$表)はメモリーにのみ存在し、一般的には、インスタンスまたはメモリー構造についての情報を記録します。v\$動的パフォーマンス・ビューは、V\$SQL や V\$SQL_PLAN などの X\$表に定義されます。



データ・ディクショナリ表(SYS.USER\$, SYS.TS\$, SYS.SEG\$, SYS.OBJ\$, SYS.TAB\$, SYS.FILE など)は、通常のアプリケーション表などのデータ・ファイルに格納されます。

(V\$ビューまたは SYS/SYSTEM オブジェクトに)AWR および TKPROF レポートで値が高い内部 SQL がある場合は、ディクショナリ統計と固定オブジェクト統計の収集が必要な可能性が高いです。

FND_STATS API では、ディクショナリ・オブジェクトまたは固定オブジェクトの統計が収集されないことに注意してください。DBMS_STATS API を使用する必要があります。

固定オブジェクト統計

通常、固定オブジェクト統計を収集する必要があるのは、固定オブジェクトに大幅な変更があった場合です。たとえば、次の後です。

- データベースのメジャー・アップグレード。
- プラットフォームのアップグレード(特に Exadata)。
- アプリケーションのアップグレード、または新しいアプリケーション・モジュールの追加。
- データベースの SGA/PGA 構成への変更。
- 負荷またはセッション数の大幅な変更。

パフォーマンスを最適にするには、システムに代表的なアクティビティ(通常の負荷)があったときに固定オブジェクトの統計を収集する必要があります。

実行するコマンドは次のとおりです。

```
execute DBMS_STATS.GATHER_FIXED_OBJECTS_STATS(no_invalidate=>FALSE);
```

通常は、no_invalidate=>FALSE 引数は必要ありません。ただし、プロシージャ

DBMS_STATS.set_database_prefs、set_global_prefs、set_schema_prefs または set_table_prefs を使用して、NO_INVALIDATE のデフォルト値を TRUE に設定しておく必要があります

リリース 12.2.n のアップグレードでは、次の場合に、固定オブジェクト統計を収集する必要があります。

- Oracle E-Business Suite 全体のアップグレードの一部である、関連プラットフォームやデータベースのアップグレード後。
- SGA/PGA パラメータの変更後。



- リリース 12.2.n のアップグレード後、システムに代表的なアクティビティが発生した場合。

オンライン・パッチの有効化(および基礎となるエディションベースの再定義)による、固定オブジェクトに対する変更もあります。そのため、オンライン・パッチの有効化の内部 SQL、R12.2.n RUP およびその他のオンライン・パッチの実行に、時間がかかる場合があります。このような場合に、固定オブジェクト統計の収集が役立ちます。

詳細は、次に示す My Oracle Support のドキュメントを参照してください。

- Best Practices for Gathering Statistics with Oracle E-Business Suite (Document 1586374.1)
- Fixed Objects Statistics Considerations (Document 798257.1)

ディクショナリ統計

通常、ディクショナリ統計を分析する必要があるのは、データベースに大量の DDL 操作が発生した後です(たとえば、多数の新規オブジェクトを作成または再作成して、データベースが大幅に変更された場合です)。

SYS および SYSTEM を含め、すべてのシステム・スキーマの統計を収集するには、次のコマンドを使用します(Oracle Database 10g 以降の場合)。

```
execute DBMS_STATS.GATHER_DICTIONARY_STATS( estimate_percent =>
DBMS_STATS.AUTO_SAMPLE_SIZE, options => 'GATHER AUTO',
no_invalidate=>FALSE);
```

通常は、`no_invalidate=>FALSE` 引数は必要ありません。ただし、プロシージャ `DBMS_STATS.set_database_prefs`、`set_global_prefs`、`set_schema_prefs` または `set_table_prefs` を使用して、`NO_INVALIDATE` のデフォルト値を `TRUE` に設定しておく必要があります。

一般的には、ディクショナリ統計の収集を並行して処理することは適切ではありません。

ディクショナリ統計は、表の並列度で収集されます(普通は 1)。そのため、通常は、連続して収集されます。並列度を `AUTO_DEGREE` に設定するオプションがあります。ただし、このパラメータはオプション `GATHER_AUTO` では無視されるので、かわりに `GATHER` を使用する必要があります。そのため、並行処理は、(収集する必要の有無にかかわらず)すべての表を収集することと引換えに実行することになります。これは、それぞれ々のお客様の選択です。

Oracle Database 9i の場合、コマンドは次のとおりです。



```
execute DBMS_STATS.GATHER_SCHEMA_STATS('SYS', method_opt=>'for all columns
size 1', degree=>30, estimate_percent=>100, cascade=>true);
```

リリース 12.2.n のアップグレードでは、次の場合に、ディクショナリ統計を収集する必要があります。

- Oracle E-Business Suite 全体のアップグレードの一部である、関連プラットフォームやデータベースのアップグレード後。
- リリース 12.2.n のアップグレード後。
- OATM への移行後。

オンライン・パッチの有効化(および基礎となるエディションベースの再定義)による、ディクショナリ・オブジェクトに対する変更もあります。そのため、オンライン・パッチの有効化の内部 SQL、R12.2.n RUP およびその他のオンライン・パッチの実行に、時間がかかる場合があります。このような場合に、ディクショナリ統計の収集が役立ちます。特に、エディションング・オブジェクトに効果的です。

詳細は、My Oracle Support のドキュメントである『Best Practices for Gathering Statistics with Oracle E-Business Suite (Document 1586374.1)』を参照してください。

特定のディクショナリ・オブジェクトまたは固定オブジェクトの統計の収集

実行計画が不十分な内部 SQL の数が非常に少ない場合や、オブジェクト数が少ない場合は、すべてのディクショナリ統計または固定オブジェクト統計を収集せずに、特定のオブジェクトを対象にすることが可能です。

例

```
exec dbms_stats.gather_table_stats(ownname=>'SYS', tabname=>'OBJ$', no_invalidate=>false);
exec dbms_stats.gather_table_stats(ownname=>'SYS', tabname=>'X$KQFP', no_invalidate=>false);
```

adstats.sql

このスクリプトは、次を実行します。

- Oracle 10g/11g の自動統計収集ジョブを無効化します。
- ディクショナリ統計および固定オブジェクト統計を収集します。

ただし、実行できるのは Database 10g 以上のみで、データベースを制限モードする必要があります。



(関連するデータベース・アップグレードの一環として実行される場合があります)Oracle E-Business Suite リリース 12.2.n のアップグレード自体の一環である、固定オブジェクト統計またはディクショナリ統計の収集には、このスクリプトを使用しないでください。前述のとおり、API を直接使用します。

Oracle E-Business Suite のインスタンスを Oracle Database 11g にアップグレードする一環として、このスクリプトを実行することをお勧めします。詳細は、アップグレードの説明に記載されています。

注意:

- `adstats.sql` スクリプトは、リリース 12.2.n のアップグレードの一環として、また、R12.2.0 やその他のドライバでは、自動的に実行されません。
- パフォーマンスを最適にするには、システムに通常の負荷がかかったときに固定オブジェクトの統計を収集する必要があります。このスクリプトは、データベースが制限モードのときにのみ実行されるよう設計されています。
- スクリプトでは、システム統計は収集されません。
- 前に記載したとおり、`adstats.sql` と `adsstats.sql` が混同されることがありますが、`adsstats.sql` は、Oracle E-Business Suite のスキーマ統計を収集します。

パフォーマンスの問題の解決

この項では、次の内容について説明します。

- アップグレード中に収集する診断(特に、長期間実行されているジョブや SQL がある場合に収集する診断)。
- 一般的(汎用的)な問題とその解決策。個々の問題には、My Oracle Support のバグで対応します。

アップグレード中に収集する診断

ほとんどのアップグレードで、自動ワークロード・リポジトリは、スナップショットを 30 分にして有効化することをお勧めします(デフォルトは 60 分です)。短期間で終わるアップグレードの場合は、より短い期間での分析が可能になるため、さらに短いスナップショットがより適しています。



AWR の保持期間は、アップグレードが実行される期間と、(診断および分析を収集するための) その後の長い期間を網羅できるよう、十分に長くする必要があります。推奨値は $N+7$ 日で、 N は推定アップグレード時間を表しますが、期間が長ければ、その後の診断および統計の収集期間が長くなります。

リリース 12.2.n のアップグレードのテスト実行期間は、`statistics_level` を ALL (または `_rowsource_execution_statistics` を TRUE) に設定することを強くお勧めします。

サービス・リクエストへのファイルのアップロード

My Oracle Support を使用して、ファイルを SR にアップロードすることが適切でない場合や、不可能な場合があります(ファイルが大きすぎる場合など)。その場合には、SFTP を使用してファイルをアップロードできます。My Oracle Support のドキュメントである『How to Send a File to Oracle Support Using FTP (ftp.oracle.com) (Document 464666.1)』を参照してください。

Oracle E-Business Suite リリース 12.2.n のアップグレードにおけるパフォーマンスの問題の一般的な診断

アップグレードのすべてのパッチ/フェーズには、現在でも次の診断が必要です。これには、オンライン・パッチの有効化、12.2 のパッチ後のオンライン・パッチ(ADOP)フェーズ(準備、適用、確定、カットオーバー、クリーンアップなど)、特に 12.2.n RUP が含まれます。

次に、リリース 12.2.n のアップグレードにおけるパフォーマンスの問題を分析する目的を示します。

- 無駄なテストの繰返しの防止。発生した問題を初回で解決するソリューションの提供を目的としています。
- 解決されたパフォーマンスの問題数の最大化。

アップグレード・ジョブを個別にテストすることはできません。テストできるのは、次の繰返し的时候了。修正に効果がない場合は、テストが無駄に繰返されている可能性があります。

これを実行するには、次のものがが必要です。

- 実際の統計: これにより、不十分な可能性があるものではなく、実行計画のどの手順が不十分であるかを正確に把握できます。パフォーマンスの修正により予想される影響も推定できます。パフォーマンスの問題を初回で解決する修正を提供できる可能性が高まります。また、細かな修正(例: より明確な索引アクセスを可能にすることなどで、経過時間を 10 から 50%まで削減できる修正)の識別も可能になります。多くの場合、これらの修正により、ワーカ間の競合が低減されます。



- 高速で簡単に実行できる診断。
- リリース 12.2.n のアップグレード・パフォーマンスへの影響が非常にわずかな診断。
アップグレード中に安全に実行できれば、結果がすぐに取得され、問題もより迅速に解決されます。

主要な準備の手順は、次のとおりです。

- リリース 12.2.n のアップグレード前に、`statistics_level` を `ALL` (または `_rowsource_execution_statistics = TRUE`) に設定します。
- 自動ワークロード・リポジトリ (AWR) は、スナップショットを 30 分にして有効化する必要があります (デフォルトは 60 分です)。短期間で終わるアップグレードの場合は、さらに短いスナップショットがより適しています。
- AWR の保持期間は、アップグレードが実行される期間と、(診断および分析を収集するための) その後の長い期間を網羅できるよう、十分に長くする必要があります。推奨値は `N+7` 日で、`N` は推定アップグレード時間を表しますが、期間が長ければ、その後の診断および統計の収集期間が長くなります。

アップグレード中:

- AWR またはカーソル・キャッシュ (メモリー) の上位 SQL をモニターします (「便利なスクリプト」を参照してください)。これは、内部 SQL かアプリケーション SQL です。負荷のかかる SQL が発生した場合は、Enterprise Manager を使用して識別することも可能です。
- 長期間実行されている SQL の表示カーソル・レポート (ALL+ALLSTATS) を取得します
- パラレル問合せまたは DML を使用する SQL の SQL モニター・レポートを取得します
- SQL の一部がいつ実行されたかを識別します (「便利なスクリプト」を参照してください)
- 長期間実行されている SQL とジョブを照合します (「便利なスクリプト」を参照してください)
- Oracle E-Business Suite のすべての表に関する CBO 統計をレポートします (「便利なスクリプト」を参照してください)

リリース 12.2.n のアップグレード後:

- AD ジョブ・タイミング・レポートを取得します



- 長期間実行されているアップグレード・ジョブを特定します(「便利なスクリプト」を参照してください)
- 長期間実行されているジョブのファイル・バージョンを取得します
- AWR レポートを取得します
- afxplain.sql スクリプトを実行して、詳細な CBO 統計、データベース・パラメータ、CBO パラメータおよびメタデータを取得します。これは、個々の SQL に対して実行します。

その他の診断

さらに詳細な分析を実行する場合に必要なものを次に示します。

- AD ユーティリティおよびワーカー・ログ。
- 長期間実行されている SQL の SQLT (TXTRACT)

また、Oracle Support および Oracle Development では、次が必要です。

- AWR エクスポート
- AD パラレル表のエクスポート

オンライン・パッチの有効化 - アップグレード中に収集される特定の診断

オンライン・パッチの有効化パッチでは、次の診断も使用できます。

- ADZDSHOWDDL.sql

オンライン・パッチ - アップグレード中に収集される特定の診断

オンライン・パッチを使用して適用されるパッチには、次の診断も使用できます。

- ADOP ログ・ディレクトリ
- オンライン・パッチ・ログ・アナライザ・ユーティリティ adopscanlog (AD および TXK C Delta 4 パッチにより配信)は、adop ログ・ディレクトリを分析します
- ADZDSHOWLOG.sql/adzdshowlog.out
- ADOP サイクル・ステータス
- ADOP フェーズのステータスを識別する SQL
- AD および TXK C のパッチ・レベルを識別する SQL



詳細は、次を参照してください。

- 『Oracle E-Business Suite メンテナンス・ガイド リリース 12.2』
- 『Oracle E-Business Suite リリース 11i から 12.2 へのアップグレード・ガイド』
- Oracle E-Business Suite リリース 12.0 および 12.1 から 12.2 へのアップグレード・ガイド

My Oracle Support のドキュメントも参照してください。

- Useful 12.2 Documents For Customers And ATG Support Engineers (Document 1585889.1)
- 12.2 E-Business Suite - Collecting Online Patching and fs_clone Log Files (Document 1542162.1)

Statistics_Level = ALL (または_rowsource_execution_statistics = TRUE)

これは、次のコマンドを使用して設定できます。

```
SQL>alter session set statistics_level='ALL'
```

これは、(SQLT および表示カーソル・レポートの)各実行計画行に実際に使用された実行統計(経過時間、物理読取り、バッファ取得などを含む)を確認する最も簡単な方法です。SQL トレース および TKPROF の代替機能では、編集標準コードが必要です。

この方法を使用すると、通常は、問題の解決が大幅に早くなり、正しい解決策が初回でわかる場合もあります。

また、初期化パラメータ_rowsource_execution_statistics を TRUE (statistics_level は'TYPICAL')に設定しても、実際に使用された同じ実行統計を収集できます。この場合、statistics_level=ALL とするよりも、オーバーヘッドを低くできます。

お客様の技術アーキテクトや DBA (または実装パートナー)の多くが、statistics_level を ALL (あるいは_rowsource_execution_statistics = TRUE)にすると、パフォーマンスが大幅に低下すると考え、この設定に抵抗を感じる場合があります。

ここでは、次の 2 点が関係します。

- statistics_level = ALL/_rowsource_execution_statistics = TRUE に設定すると、パフォーマンスに多少の影響がありますが、その規模は小さく、大きくはない可能性が高いです。リリース 12 のアップグレードはバッチ・プロセスで構成されているため、統計の負荷が全体に占める割合は非常に低いです。



- ・ パフォーマンスへの影響が大きい場合でも、目標は、
(`statistics_level/_rowsource_execution_statistics` を元の値に戻すことが適切になったときに)後続のテスト実行の経過時間を低減して稼働させることです。そのため、テストの初期段階で、経過時間が増加しても問題はありません。

経過時間への多少の影響や、最初に実行が必要な作業がありますが、その後は、経過時間と実行が必要な再作業量の低減に役立ちます。

AWR が有効化されている状態で `statistics_level` を ALL に設定すると、AWR の `WRH$_LATCH_CHILDREN` 表に挿入される行数が大幅に増加する可能性があることに注意してください。そのため、領域が不足しないよう、`SYSAUX` 表領域をモニターしてください。

SQL トレース - SQL スクリプト固有

表示カーソル、SQL モニターおよび AWR 診断から、パフォーマンスの問題を解決できない場合は、次のテスト実行でそのスクリプト/ジョブ用に、10046 SQL トレース(レベル 16 - `ALL_EXECUTIONS` (`waits` を指定))を取得します。

(特に AD パラレル・ジョブでは)最後の方の実行までパフォーマンスの問題が発生しなかったり、最初の実行(デフォルトでレベル 8)で十分な情報が得られなかったりする場合があるため、レベル 16 (`ALL_EXECUTIONS`)が必要になります。

SQL トレースは、システム・レベルで有効化できます。ただし、大量のトレース・ファイルが作成されるため、正しいトレースを特定するのが難しくなる場合があります。(ヒント: UNIX では、`grep` を使用して、主要な SQL 文字列またはスクリプト/モジュール名を検索します。)

ただし、`DBMS_MONITOR` (10g 以上)または `Event++` 構文(11g 以上)を使用して、その他のセッションの SQL トレースを実行することは可能です。

My Oracle Support のドキュメントである『Oracle E-Business Suite SQL Trace and TKPROF Guide (Document 1674024.1)』の、E-Business Suite でのトレース(TKPROF)の取得にある、`DBMS_MONITOR` を使用したその他のデータベース・セッション向け(10g 以上)/`Event++` 構文を使用したその他のデータベース・セッション向け(11g 以上)の項を参照してください。

この場合、次の制限事項があります。

- ・ 必要な診断(フェーズ、実行とフェッチの統計、行ソース統計)を取得するには、プロセスまたは SQL の開始前に、トレースを有効化しておく必要があります。
- ・ トレースするセッションを識別できる方法が必要です。たとえば、クライアント識別子(R12.2.n のアップグレード・スクリプトでは設定されません)、サービス、モジュール(設



定されている場合もあります)、アクション、セッションID とシリアル番号、OS プロセスID、Oracle プロセスID またはプロセス名です。これらは、ジョブ/スクリプトが実行されるまでわかりません。

- パラレル・スレーブやAD ワーカーが複数存在する場合もあります。

このため、SQL トレースは、次のようにして一時的に SQL スクリプトを編集し、特定のスクリプトに対して有効化することをお勧めします。

スクリプトの本体の前に次を追加します。

- `alter session set tracefile_identifier='<identifier string>';`
- `alter session set events '10046 trace name context forever, level 16';`
- `alter session set statistics_level='ALL';`
- `alter session set max_dump_file_size='UNLIMITED';`

スクリプトの本体の後に次を追加します。

- `alter session set events '10046 trace name context off';`

スクリプト・レベルでトレースを有効化するその他のメリットは、各スクリプトに異なるトレース識別子を指定できることで、トレースの検索が簡単になります。

詳細は、My Oracle Support のドキュメントである『Oracle E-Business Suite SQL Trace and TKPROF Guide (Document 1674024.1)』で、特に、E-Business Suite でのトレース(TKPROF)の取得にある、SQL*Plus からの項を参照してください。

SQL 固有

AWR または TKPROF を使用して、(長期間実行中のジョブで)長期間実行されている SQL が識別されたら、次に示すような出力が表示されます。

これらを便利にするには、`statistics_level` を `ALL` (または `rowsource_execution_statistics = TRUE`) に設定する必要があることに注意してください。実際の統計は、負荷のかかる実行計画の手順、またはわずかに最善には及ばない実行計画の識別に必要です。(この情報がなくても、実行計画自体に問題ありません。)

表示カーソル・レポート

カーソル・キャッシュにロードされたカーソルの実際の実行計画を表示します。基本レベルでは、ランタイム実行計画を表示します。ただし、フォーマット `ALL` にも、プルーニング、パラ



レル実行、述語、見積計算、別名およびリモート SQL 情報など、その他の情報が含まれています。

(IOSTATS および MEMSTATS が含まれる)+ALLSTATS オプションには、各実行計画手順の実際の統計が含まれます。次に例を示します。

- 経過時間
- 物理読取り
- バッファ取得
- メモリー集中操作(ハッシュ結合、ソート、ビットマップ演算子など)に使用されたメモリー(PGA)

ただし、これが提供されるのは、`statistics_level=ALL/_rowsource_execution_statistics = TRUE` の場合のみです

XTRACT を使用した SQLT でも、同じ状況で、実際の行ソース統計がレポートされることに注意してください。ただし、`display_cursor` では、提供される情報の量が少ないです。また、長期間実行されている SQL が進行している間、少ないオーバーヘッドで、アップグレード中に実行できます。

表示カーソル・レポートは、できるだけ早く実行する必要があることに注意してください。遅くなると、カーソルがメモリーからフラッシュされたり、無効になったりします。そのような場合は、データを使用できません。

レポートは、次の SQL スクリプトを実行して生成できます。

```
SET pages 0
SET lines 300
SET LONG 10000
SET LONGCHUNKSIZE 10000
SPOOL<report_name>.txt
SELECT * FROM TABLE(dbms_xplan.display_cursor('<sql_id>', NULL, 'ALL +ALLSTATS'));
SPOOL OFF;
```

詳細は、My Oracle Support のドキュメントである『Oracle E-Business Suite Performance Guide (Document 1672174.1)』の表示カーソルに関する項を参照してください。

SQL がメモリーにはなく、AWR にある場合は、かわりに表示 AWR レポート (DBMS_XPLAN.DISPLAY_AWR)を使用します。ただし、実際の内容に関するレポートではありません



ません。+ALLSTATS オプションがないため、AWR に格納されている実行計画手順に関する実際の統計はありません。この場合、XTRACT メソッドを使用した SQLT でも、実際の統計はレポートされないことに注意してください。

表示カーソル・レポートおよび AWR レポートに表示されるのは、sql_text (最初の 1000 文字) のみで、full_text ではありません。

そのため、必要な場合には、次の SQL スクリプトを実行して、完全な SQL テキストを取得してください。

```
SET pages 0
SET lines 300
SET LONG 10000
SET LONGCHUNKSIZE 10000
SPOOL<report_name>.txt
SELECT sql_id, sql_text, sql_fulltext FROM v$sql
WHERE sql_id = '<sql_id>';
SPOOL OFF;
```

SQL モニター・レポート

主なメリットは、計画やパラレル・スレーブの各ステージに渡り、パラレル SQL/DML がどのように実行されているかをよく把握できることです。ただし、使用できるのは、Oracle Database 11g 以上のみです。

また、SQL が実行された時点で、statistics_level 初期化パラメータが ALL (または _rowsource_execution_statistics が TRUE) に設定されていない場合でも、各実行計画行の実際の実行や行数を確認できます。

長期間実行されている SQL が進行している間、少ないオーバーヘッドで、アップグレード中に実行できます。

次の SQL スクリプトを実行して生成できます。

```
set trimspool on
set trim on
set pages 0
set long 10000000
set longchunksize 10000000
set linesize 200
set termout off
spool sql_monitor_for_<sql_id>.htm
variable my_rept CLOB;
BEGIN
```



```

:my_rept := dbms_sqltune.report_sql_monitor(sql_id => '<sql_id>', report_level => 'ALL',
type => 'HTML');
END;
/
print :my_rept

spool off;

set termout on

```

statistics_level は ALL (または _rowsource_execution_statistics = TRUE) に設定する必要があることを、再度記載しておきます。

詳細は、My Oracle Support のドキュメントである『Oracle E-Business Suite Performance Guide (Document 1672174.1)』の SQL モニター・レポートに関する項を参照してください。

afxplain.sql

ときには、環境で SQLT を使用できない場合があります。たとえば、顧客が本番環境への SQLT のインストールに反対することや、本番システムへのパフォーマンスの影響を心配することがあります。

\$FND_TOP/sql/afxplain.sql を使用すると、一部の出力を SQLT と同一にすることができます。

スクリプト・ファイル自体を除き、環境へのインストールが必要なものはなく、パフォーマンスへの影響も無視できる程度です。アップグレードの進行中に実行できます。

次の情報が提供されます。

- データベース/アプリケーションのバージョン、O/S のバージョンおよびインスタンス/データベース名
- コストと述語の情報を含む Explain Plan
- 共有メモリーの量
- パッケージ、ビューおよびトリガーなどのオブジェクト情報
- SQL で参照されたすべてのオブジェクトに関する統計(列と索引を含む)
- デフォルト以外のデータベース・パラメータ (init.ora)

出力には、次のものが含まれないことに注意してください。

- 各表の実際の行数。



- 実際の実行計画(ただし、`display_cursor` (または `display_awr`)で取得できます)。

診断が必要な SQL を含むファイルを作成します。この例では、`query.sql` と呼びます。

次を実行します。

```
afxplain.sql query.sql Y N
```

Y N パラメータはそれぞれ、CBO トレース(10053)は必要だが、統計はエクスポートしないことを指定しています。

出力はファイル `query.sql.out` に記載されます

10053 イベントのトレースは、ユーザー・ダンプ保存先に格納されます。

詳細は、My Oracle Support のドキュメントである『Oracle E-Business Suite Performance Guide (Document 1672174.1)』の SQLT にある、SQLT (`afxplain.sql`)の代替に関する項を参照してください。

XTRACT メソッドを使用した SQLT 出力

大量のシステム・リソースを使用するため、アップグレード中には実行しないでください。かわりに、表示カーソル・レポートと `afxplain.sql` を使用して、SQLT 出力に含まれる大部分の情報を取得できます。

詳細は、次に示す My Oracle Support のドキュメントを参照してください。

- Oracle E-Business Suite Performance Guide (Document 1672174.1), section "SQLT"
- SQLT Diagnostic Tool (Document 215187.1)

SQLT の環境は、パフォーマンスの問題が発生した環境と同一にし、できるだけ、関連するプログラム/プロセスの直後に実行する必要があります。

SQLT が取得しているデータを変更する可能性があるアクションに注意してください(つまり、発生したパフォーマンスの問題の後に実行され、SQLT より前に実行されるアクションです)。たとえば、統計の収集、削除、ロック、インポートや、一時表またはインタフェース表からのデータの削除です。

XTRACT を使用して SQL を実行するには、`sql_id` または `hash_value` が必要になります。

Oracle Database 11g では、TKPROF により、SQL ID とハッシュ値が提供されます。

Oracle 11g の場合は、未処理のトレースにより、`sql_id` (`sqlid=`)が提供されます。10g 以前では、ハッシュ値が提供されます。



sql_id を取得するその他の方法には、AWR や Oracle Enterprise Manager を使用する方法、および V\$SQL を直接問い合わせる方法があります。

SQLT には、実行計画、CBO 統計、データベース・パラメータ、CBO パラメータ、パフォーマンス統計および SQL に関連付けられたメタデータ(スキーマ・オブジェクト定義など)があります。

SQLT パラメータによっては、AWR レポート、ASH レポート、ADDM レポート、CBO (10053) トレース、SQL モニター・レポート、EBS 初期化パラメータのヘルスチェック・レポート (bde_chk_cbo)、テスト事例ビルダーおよび SQL チューニング・アドバイザなどのサポート情報が含まれます。

インスタンスにアクセスできない場合に特に便利です。アクセスが可能な場合でも、1 つの場所(1 つの zip ファイル)にすべての情報が収集されます。

次の理由から、XTRACT メソッドが必要です。

- メモリーまたは AWR からランタイム実行計画を取得します
- RAC 対応です。
- statistics_level が ALL (または _rowsource_execution_statistics が TRUE) に設定されているときに SQL を実行すると、カーソルがメモリー(カーソル・キャッシュ)にある場合は、実行計画に関する実際の行ソース統計(I/O、バッファ取得、経過時間)が含まれます。

AWR および ASH レポートは、SQLT によって実行されることに注意してください。一部のディクショナリ・オブジェクト(特に WRH\$_LATCH_CHILDREN)は、アップグレード中に大幅に増えます。そのため、固定オブジェクト統計およびディクショナリ統計は、statistics_level が ALL (通常の設定は TYPICAL または BASIC) に設定されている場合は特に、SQLT の実行前に収集する必要があります。

SQLT の実行には、かなりの時間がかかります。

ワークロードを削減するには、sqltxtract.sql の実行前に、(SQL*Plus から)次を実行することをお勧めします。

テスト事例ビルダー(TCB)または SQL チューニング・アドバイザ(あるいはその両方)を無効化する場合

```
EXEC sqltxplain.sqlt$a.set_param('test_case_builder', 'N');

EXEC sqltxplain.sqlt$a.set_param('sta_time_limit_secs', '30');
```




テスト事例リポジトリの自動エクスポートを無効化する場合

```
EXEC sqltxplain.sqlt$a.set_param('export_repository', 'N');
```

それでも SQLT の実行に時間がかかり、SQL で使用されるスキーマ・オブジェクトに大量のサブパーティションが含まれる場合は、次のようにして、収集されるデータの粒度を下げることはできます。

```
EXEC sqltxplain.sqlt$a.set_param('c_gran_segm', 'PARTITION');
```

```
EXEC sqltxplain.sqlt$a.set_param('c_gran_cols', 'PARTITION');
```

```
EXEC sqltxplain.sqlt$a.set_param('c_gran_hgrm', 'PARTITION');
```

これらのコマンドはすべて、APPS として実行できることに注意してください。ユーザー SQLTXPLAIN として実行する必要はありません。

これらの値は、SQLTXPLAIN.SQLI\$_PARAMETER という表に格納されます。一度設定したら、SQLT の実行ごとに再設定する必要はありません。現在の値は、表 SQLTXPLAIN.SQLI\$_PARAMETER を問い合わせることで確認できます。

さらに時間を短くするには、次を実行して、表の行数のカウントを無効化できます。ただし、各表の実際の行数に関する情報は失われます。

```
EXEC sqltxplain.sqlt$a.set_param('count_star_threshold', '0');
```

これらはすべて、1.4.4.4 (2012 年 4 月 2 日)以上のバージョンの SQLT が使用されていることを仮定しています。

AWR エクスポート

AWR は、\$ORACLE_HOME/rdbms/admin/awrextr.sql を使用して、ダンプ・ファイル(データ・ポンプ・エクスポート・ファイル)にエクスポートできます。

\$ORACLE_HOME/rdbms/admin/awrload.sql スクリプトは、エクスポート・ダンプをローカル・データベースにロードする際に使用でき、ローカル・データベースでは、様々な AWR 表で SQL を実行して、どこ(ジョブ、SQL、オブジェクトなど)で待機が発生し、パターンがあるかどうかについての詳細な分析を取得できます。

主な表は、DBA_HIST_ACTIVE_SESS_HISTORY、DBA_HIST_SEG_STAT_OBJ、DBA_HIST_SYSTEM_EVENT、DBA_HIST_SEG_STAT、DBA_HIST_WAITSTAT、DBA_HIST_ACTIVE_SESS_HISTORY、DBA_HIST_SQLSTAT、DBA_HIST_SQLTEXT です。



また、(エクスポート/インポートを実行することなく)ソース環境で直接 SQL を実行することも可能です。

エクスポート/インポートする方法の 1 つのメリットは、当初の環境がリフレッシュされた後も、診断を保持できることです。

『Oracle Database パフォーマンス・チューニング・ガイド』の「自動ワークロード・リポジット・データの転送」を参照してください。

AWR レポート

次の AWR レポートを取得します。

- アップグレードが実行されている期間全体。
- 長期間実行されているジョブの期間(ジョブの開始直前に取得されたスナップショットから終了直後のスナップショットの間など)。
- 個々のスナップショット。

通常、スナップショット間隔は 30 分にする必要があります(デフォルトは 1 時間です)。これは、アップグレード/期間の経過時間全体に合うよう変更できます。アップグレード/期間が短い場合は、わずかな遅延が重大であるため、スナップショットは短くする必要があります。8 から 15 時間かかるアップグレードの場合は、15 分のスナップショットがより適しています。特定ジョブに対して AWR レポートを実行している場合は、そのジョブのはるか前や後は入れずに、実行されていた期間をレポートに含める必要があります。

AWR の保持期間は、アップグレードが実行される期間と、(診断および分析を収集するための)その後の長い期間を網羅できるよう、十分に長くする必要があります。推奨値は N+7 日で、N は推定アップグレード時間を表しますが、期間は長い方がより適切です。

awrrpt.sql は通常、AWR レポートの生成に使用されます。必ず、HTML レポート・タイプを選択してください。Oracle RAC インスタンスでは、1 つの Oracle RAC ノードでのみアップグレードが実行されるため、多くの場合は、awrrpti.sql で十分です。

AWR レポートで、最初に確認する 3 つの主な点は、長期間実行中の SQL (SQL 統計/SQL Ordered By)、競合/ボトルネック(時間設定された上位 5 のフォアグラウンド・イベント/フォアグラウンド待機イベント)および CPU 使用率です。ただし、状況に応じて、その他にも多数ある便利な統計を AWR レポートから取得できます。一部は、このドキュメントの他の箇所で説明しています。



特定の待機のレベルが高い場合は、システム構成やリソースの問題であると仮定する前に、まず、その待機が長期間実行されている特定の SQL やジョブでのみ発生しているかどうかを確認してください。

次を実行することも可能です。

- 長期間実行されている特定の SQL の実行計画をレポートするための `awrsqrpt.sql` (または `awrsqrpi.sql`)。
- 特定の SQL のアクティブ・セッション履歴をレポートするための `ashrpt.sql` (または `ashrpti.sql`)。これにより、各行ソースまたはオブジェクトの待機やイベントに関する有用な情報が得られます。

一部のディクショナリ・オブジェクト(特に `WRH$_LATCH_CHILDREN`)および固定オブジェクトは、アップグレード中に大幅に増えることに注意してください。`statistics_level = ALL` の場合、保持期間が長い場合またはスナップショット間隔が短い場合は特に、これが当てはまります。

そのため、固定オブジェクト統計やディクショナリ統計は、AWR の実行前に収集する必要があります。

詳細は、My Oracle Support のドキュメントである『Performance Diagnosis with Automatic Workload Repository (Document 1674086.1)』を参照してください。

AD パラレル表のエクスポート

(データ・ポンプ・エクスポートまたは元のエクスポートを使用して)エクスポートする主な表は、次のとおりです。

- `AD_PARALLEL_UPDATES`
- `AD_PARALLEL_UPDATE_UNITS`
- `AD_TASK_TIMING`

これらは、データ・ポンプ・インポートまたは元のインポートを使用して、ローカル・データベースにインポートし、分析できます。

ここで確認する主な項目は、各バッチで実際に処理された行数、バッチ数、進捗状況(低下しているジョブを特定するため)、各バッチで処理された行数/所要時間のうち、ロック/スリープまたはデータ配分の問題を示す可能性がある大きな異常です。また、ジョブやワーカーの開始時間と終了時間(`AD_TASK_TIMING`)は、長期間実行されている SQL や(AWR での)パフォーマンスの問題と特定のジョブの照合に役立ちます。

ORACLE[®] 12 E-BUSINESS SUITE

AD ジョブ・タイミング・レポート(次を参照)は、時間のかかっている上位 100 のジョブのみをレポートします。ただし、AD パラレル・ジョブでは、各ワーカーが別々のジョブとしてレポートされるため、AD ワーカーの数が多いため(16、24 または 32 など)場合は、レポートされるジョブが少ない可能性があります。この場合は、AD_TASK_TIMING 表を直接問い合わせ、長期間実行されているジョブをすべて識別できます。

また、AD_TASK_TIMING を問い合わせ、分割待機が原因で利用率が低いワーカー(および潜在的なリソース利用率)を識別することもできます。

長期間実行されているジョブや分割待機を表示するレポートについては、このドキュメントの一部である「便利なスクリプト」を参照してください。

AD ジョブ・タイミング・レポート

レポートされる内容は次のとおりです。

- 成功、失敗、遅延、再開またはスキップされたジョブの数。
- 時間のかかっている上位 100 のジョブ。
- 失敗、遅延、再開およびスキップされたジョブ。
- アップグレードの各フェーズのタイミング、ジョブの合計数、および遅延、再開、スキップされた数。

ただし、レポートされるのは、時間のかかっている上位 100 のジョブのみで、AD パラレル・ジョブでは、各ワーカーが別のジョブとみなされます。つまり、レポートされるジョブの数が少ない可能性があるため、AD_TASK_TIMING 表を直接問い合わせる必要があります(前述の項と、「便利なスクリプト」の項を参照してください)。

ADOP、AutoPatch または AD 管理を実行している場合は、AD ジョブ・タイミング・レポート (adt<session_id>.lst)が自動的に生成されます。このレポートのコンテンツには、Oracle Application Manager からアクセス可能で、完了したアップグレード・セッションのレポートを APPL_TOP/admin/<SID>/out ディレクトリから取得することも可能です。レポートは、adt<session_id>.lst という名前です。

AD 管理ジョブ向けに、コマンド・ラインから AD ジョブ・タイミング・レポートを実行することもできます。

```
$ cd $APPL_TOP/admin/<SID>/out
```



```
$ sqlplus <APPS username> /<APPS password> @$AD_TOP/admin/sql/adtimrpt.sql ¥
<session id> <output file>
```

ここで、<session_id>は必要なタイミング統計のセッションで、<output file>は統計が書き込まれるファイルの名前です。

\$AD_TOP/admin/sql/adtimdet.sql も、同じように実行できます。これにより、フェーズ別および経過時間別にすべてのジョブの詳細が取得されます。ジョブの実行にかかった時間や、時間のかかっている上位 100 のジョブが、わずかなジョブの複数のワーカーで占められている場所の検出にも便利です。

SQL スクリプトが、(admin/sql ではなく) \$AD_TOP/sql に配置されている場合があることに注意してください。

詳細は、『Oracle E-Business Suite メンテナンス・ガイド リリース 12.2』を参照してください。

AD ユーティリティおよびワーカー・ログ

すべての AD ユーティリティは、処理アクションと発生したエラーをログ・ファイルに記録します。多くのユーティリティではログ・ファイルの名前が必要ですが、デフォルトでは、<utility_name>.log に設定されます。たとえば、AD 管理の場合、デフォルトのログ・ファイルは adadmin.log です。AutoPatch の場合は、adpatch.log です。

ジョブを並行処理する AD ユーティリティも、ワーカー・ログ・ファイルに詳細を書き込みます。adwork<number>.log ファイル(adwork001.log、adwork002.log...)は、\$APPL_TOP/admin/<SID>/log ディレクトリにあります(<SID>は ORACLE_SID または TWO_TASK 変数(UNIX)の値です)。

オンライン・パッチ(ADOP)の場合、AD ユーティリティおよびワーカー・ログは、<INSTALL BASE>/fs_ne/EBApps/log/adop ディレクトリなど、エディション化されていないファイル・システム(fs_ne)に配置されています。

```
/u01/PROD/fs_ne/EBApps/log/adop
```

後続の「ADOP ログおよび診断」を参照してください。

オンライン・パッチの有効化 - 診断固有

オンライン・パッチの有効化パッチは、AutoPatch (adpatch)を使用して適用されます。前述の一般的な診断に加え、次のスクリプトの出力も役に立ちます。

```
$ sqlplus apps @$AD_TOP/sql/ADZDSHOWDDL.Sql
```



ADOP ログおよび診断

すべての ADOP ログは、<INSTALL BASE>/fs_ne/EBSapps/log/adop ディレクトリなど、エディション化されていないファイル・システム(fs_ne)に配置されています。

/u01/PROD/fs_ne/EBSapps/log/adop

ADOP の各サイクルでは、パッチ・セッション ID などに対応するサブディレクトリが作成されます。

/u01/PROD/fs_ne/EBSapps/log/adop/n

n はセッション ID です。

ディレクトリ全体の zip を作成するのが、最も簡単かつ迅速です。

関連するメイン・ファイルは、各種の ADOP ログ(adop_YYYYMMDD_HHMISS.log など)です。

ただし、adzdshowlog.out、adworker*.log、u*.log、u*.lgi、admrgpch*.log ファイルもすべて有用で、同じパスにあります。

ADOP の実行中には、どの ADOP セッション ID が使用されているかが、端末の画面上に表示されます。

例: /u01/PROD/fs_ne/EBSapps/log/adop/9/apply_20121011_024437

セッション ID のディレクトリには、各フェーズのトレース・ファイル(adop_20130316_091340.log など)と、その他のログ(apply_20130316_091340 など)が含まれる、各フェーズに対応したログ・ディレクトリがあります。

トレース・ファイルと対応するログ・ディレクトリのタイムスタンプは同一です。

ADOP 以外のログ

各フェーズの同じログ・ディレクトリ(apply_20130316_091340 など)には、AD ユーティリティおよびワーカーのログもいくつか含まれています。

adrelink.log、adlibin.log、adlibout.log、adworknnn.log がこれに当たります。最も有用なファイルは adworknnn.log で、それぞれの AD パラレル・ワーカーで実行されたジョブがタイムスタンプ付きで表示されます。

オンライン・パッチ・ログ・アナライザ・ユーティリティ

これは、AD および TXK C Delta 4 パッチによって配信されたものです。



このユーティリティは、adop ログ・ディレクトリにエラーや警告があるかどうかを分析し、adop の実行中に発生した問題をユーザーが迅速に識別できるよう、メッセージを表示します。そのため、ログ・ファイルを手動で確認する選択肢となります。

ログ・アナライザ・ユーティリティは、オプションなしで実行できます。

エラーがあるかどうか、最新の adop セッションのログ・ディレクトリすべてをスキャンするには、次のようにします。

```
$ adopscanlog
```

ユーティリティは、様々なオプションを指定して実行することも可能です。次に例を示します。

最後のセッションで、最新の adop 実行に関連するログ・ディレクトリをスキャンするには、次のようにします。

```
$ adopscanlog -latest=yes
```

最後のセッションで、指定されたフェーズの最新の実行に関連するログ・ディレクトリをスキャンするには、次のようにします。

```
$ adopscanlog -latest=yes -phase=<phase_name>
```

エラーがあるかどうか、(session_id で表して)指定されたセッションのログ・ディレクトリすべてをスキャンするには、次のようにします。

```
$ adopscanlog -session_id=<number>
```

サポートされているパラメータの完全なリストを表示するには、次のようにします。

```
$ adopscanlog -help
```

adzdshowlog.out

これは、AD_ZD_LOGS 表のコンテンツをレポートします。オンライン・パッチの進捗状況に関するメッセージが、タイムスタンプ付きで記載されています。この表のコンテンツは、クリーンアップ/準備フェーズが実行されるたびに切り捨てられます。

また、次のスクリプトを実行すると、これまでのフェーズの内容を取得できます。

```
$ sqlplus apps @$AD_TOP/sql/ADZDSHOWLOG.sql
```

または、次の SQL を実行します

```
SELECT * FROM ad_zd_logs ORDER BY log_sequence desc;
```



ADOP フェーズのステータスを確認する SQL

次の SQL 文により、各 ADOP フェーズのステータスが、対応するセッション ID とともに表示されます。

```
SELECT adop_session_id, prepare_status, apply_status, finalize_status,  
       cutover_status, cleanup_status, abort_status, status, abandon_flag, node_name  
FROM ad_adop_sessions  
ORDER BY adop_session_id;
```

実際にはこれは、環境におけるオンライン・パッチの履歴です。

どのフェーズにも次のステータスが適用されます。

Y: フェーズは実行済です

N: フェーズは未完了です

X: フェーズを適用できません

R: フェーズは実行中(進行中)です

F: フェーズは失敗しました

P: 少なくとも 1 つのパッチが、セッション ID に適用されています(使用されるのは適用フェーズのみです)

C: この ADOP セッションのステータスは完了です

注意: 数字のステータスが関連するのはカットオーバー・フェーズのみです。これらのステータス値は、手順の完了時に次のように更新されます。

N: フェーズは未完了です

0: cutover/force_shutdown が開始されました

1: force_shutdown の手順が正常に実行されました

3: db_cutover の手順が正常に実行されました

4: fs_cutover の手順が正常に実行されました

6: force_startup の手順が正常に実行されました

Y: フェーズは実行済です



カットオーバーのステータス

```

cutover_status='Y' 'COMPLETED'
cutover_status not in ('N','Y','X') and status='F' 'FAILED'
cutover_status='0' 'CUTOVER STARTED'
cutover_status='1' 'SERVICES SHUTDOWN COMPLETED'
cutover_status='3' 'DB CUTOVER COMPLETED'
cutover_status='D' 'FLIP SNAPSHOTS COMPLETED'
cutover_status='4' 'FS CUTOVER COMPLETED'
cutover_status='5' 'ADMIN STARTUP COMPLETED'
cutover_status='6' 'SERVICES STARTUP COMPLETED'
cutover_status='N' 'NOT STARTED'
cutover_status='X' 'NOT APPLICABLE'

```

adop サイクルの現在のステータスの確認

元になる実行ファイルシステムの環境ファイルと実行コマンドを明らかにします

```
adop -status
```

用途:

adop -status は、サマリー・レポートを生成します

adop -status <sessionID>は、そのセッション ID のサマリー・レポートを生成します

adop -status -detail は、詳細レポートを生成します

AD および TXK C のパッチ・レベルの確認

次を実行します。

```
SELECT codelevel FROM AD_TRACKABLE_ENTITIES WHERE abbreviation in ('txk','ad');
```



長期間実行されている SQL、競合およびチューニング

長期間実行されているジョブまたは SQL の場合は、まず、適切な実行計画が使用されているかどうかの調査から開始するのが最適です。実行計画が不十分であると(またはわずかに最善には及ばない場合でも)、特に、ある競合が特定のジョブでのみ発生する場合は、その競合の根本原因になる可能性があります。

実行計画が最適でないと、不要なデータベース・ブロックにアクセスすることになり、競合や追加の待機の原因になることが多々あります。表のフル・スキャンや、選択が特別に限定的ではない索引が原因の場合もあります。これにより、異なる AD ワーカーが、同じブロックに同時にアクセスする可能性が高まります。実行計画が不要なブロックにアクセスしなくなれば、競合/待機が低減するか、なくなります。

ただし、最適でない実行計画を特定するには、各実行計画手順の実際の行数、バッファ取得および経過時間を確認します。選択が限定的な索引を使用しているなど、一見したところ問題のない実行計画が、最善ではないことがよくあります。また、非常に適切な実行計画に、いくつかの不要なブロックにアクセスする手順が 1 つあると、ワーカーやスレッドが追加されたときに、競合のリスクが高まることにも注意してください。

statistics_level=ALL/_rowsource_execution_statistics = TRUE に設定することをお薦めするのは、このためです。

ファイル・バージョン

個別の問題を解決する際、Oracle Support や Oracle Development では、アップグレードに使用したファイルのバージョンを伺います。

特定のジョブで実行された内容や、ファイルの正確なメタデータ/オブジェクト定義を確認する場合に必要になります。これを実行するときには、アップグレードに使用したファイルのバージョンを必ず特定してください。

たとえば、R12.2.n のアップグレードでは、R12.2.n RUP やアップグレード後の手順を含め、すべてのアップグレード手順の後に示されるバージョンが、R12.2.0 のアップグレード中に使用したものと異なる場合があります。

R12.2.0 のアップグレード中に使用したファイルのバージョンを検出するには、アップグレードのその部分に使用された統合ドライバを確認します。同じことが、12.2.n RUP にも当てはまります。(たとえば、

\$ cat u_merged.drv|grep -A5 cstpostimportaad.sql です。)

ORACLE[®] 12

E-BUSINESS SUITE

問題のないわずかな競合

最善ではない実行計画が原因の不必要な競合が取り除かれると、AD パラレルまたはパラレル実行セッション間の少数の競合(特定の待機の 5 から 15%など)が有用なインディケータとなり、その大部分は使用可能なリソースから取得されます。

ただし、一部のイベントの競合レベルを測定するのは困難です。典型的な競合では、(バッファ内の)同じブロック、ラッチまたは行にアクセスするセッションが原因で待機が発生しますが、その待機全体が、

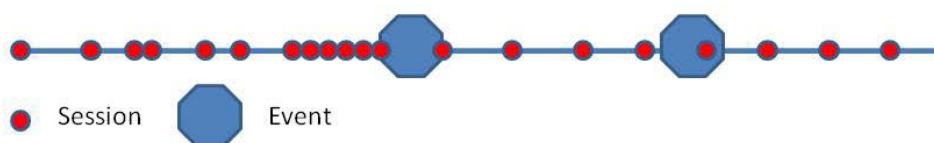
競合とみなされます。ただし、ディスク(たとえば db file 順次読取り)などのリソースの待機では、リソースがビジーではない場合でも、最低限の平均待機時間があります。そのような場合は、待機がどの程度長くなったかと、ヒストグラムの裾の長さ/厚さを確認してください。

ボトルネックの移動

ボトルネックの移動に注意してください。特定の待機を大幅に減らすアクションを実行する場合がありますが、別の待機が著しく増加します。そのアクションが新しい待機の原因になっていることは明らかです。ただし、それが問題なわけではありません。それによって、待機が発生していることが明らかになったのです。

これは、新しい待機をなくすアクションを実行してはいけないという意味ではありません。最初のアクションを元に戻すことが解決策ではない場合があるということです。

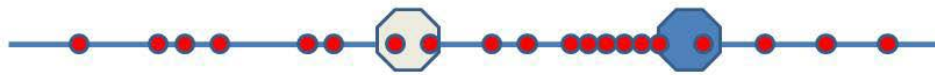
次に、その影響を図で示します。



セッションが 1 つ目のイベントにランダムなパターンで到達し、待機します。そのため、1 つ目のイベントがボトルネックです。1 つ目のイベント/ボトルネックがフローの制御に影響しているため、2 つ目のイベントにもいくつかの待機が発生しています。

ORACLE[®] 12

E-BUSINESS SUITE



1 つ目のボトルネックがなくなると、セッションが 2 つ目のイベントにランダムなパターンで到達し、待機します。今度は、2 つ目のイベントがボトルネックになります。

便利なスクリプト

カーソル・キャッシュまたは AWR の 2 つのスナップショット間の上位 SQL

SQL がメモリー(カーソル・キャッシュ)に残っている場合は、次を使用して、(最後のスナップショットで)AWR に書き込まれていない長期間実行中の SQL を特定できます。

```
SELECT * FROM
(SELECT
  ss.sql_id,
  ROUND(SUM(ss.elapsed_time/1000000),0) elapsed_time_secs,
  ROUND(SUM(ss.cpu_time/1000000),0) cpu_time_secs,
  SUM(ss.disk_reads) disk_reads,
  SUM(ss.direct_writes) direct_writes,
  SUM(ss.buffer_gets) buffer_gets,
  SUM(ss.px_servers_executions) px_server_execs,
  SUM(ss.rows_processed) rows_processed,
  SUM(ss.executions) executions,
  SUM(ss.application_wait_time) apwait_secs,
  SUM(ss.sharable_mem) sharable_mem,
  SUM(ss.total_sharable_mem) total_sharable_mem
FROM v$sqlstats ss
GROUP BY ss.sql_id
ORDER BY 2 DESC)
WHERE ROWNUM <= 100;
```

次の SQL スクリプトは、2 つの AWR スナップショット間で、最も長期間実行されている SQL をレポートします。

```
SELECT * FROM
(SELECT
```



```
dhs.sql_id,
ROUND(SUM(dhs.elapsed_time_delta/1000000),0) elapsed_time_secs,
ROUND(SUM(dhs.cpu_time_delta/1000000),0) cpu_time_secs,
SUM(dhs.disk_reads_delta) disk_reads,
SUM(dhs.buffer_gets_delta) buffer_gets,
SUM(dhs.px_servers_execs_delta) px_server_execs,
SUM(dhs.rows_processed_delta) rows_processed,
SUM(dhs.executions_delta) executions,
ROUND(SUM(dhs.iowait_delta/1000000),0) iowait_secs,
ROUND(SUM(dhs.clwait_delta/1000000),0) clwait_secs,
ROUND(SUM(dhs.ccwait_delta/1000000),0) ccwait_secs,
ROUND(SUM(dhs.apwait_delta/1000000),0) apwait_secs
FROM dba_hist_sqlstat dhs
,v$database d
WHERE dhs.dbid = d.dbid
AND snap_id > <begin snap> and snap_id <= <end snap>
GROUP BY dhs.sql_id
ORDER BY 2 DESC)
WHERE ROWNUM <= 100;
```

ここで、<begin snap>および<end snap>は、スナップショットの開始 ID と終了 ID です。

SQL が実行された時期の特定

次の SQL は、特定の SQL が実行された時期(どのスナップショット間であるかなど)を表示します。これは、SQL とジョブの照合に便利です。

```
SELECT
dhs.sql_id,
dsn.snap_id,
dsn.begin_interval_time,
dsn.end_interval_time,
ROUND(SUM(dhs.elapsed_time_delta/1000000),0) elapsed_time_secs
FROM dba_hist_sqlstat dhs
,v$database d
,dba_hist_snapshot dsn
WHERE dhs.dbid = d.dbid
AND dsn.snap_id = dhs.snap_id
AND dsn.dbid = dhs.dbid
AND dsn.instance_number = dhs.instance_number
AND dhs.sql_id = '<sql_id>'
AND dsn.snap_id > <begin_snap> and dsn.snap_id <= <end_snap>
GROUP BY dhs.sql_id, dsn.snap_id, dsn.begin_interval_time, dsn.end_interval_time
ORDER BY dsn.snap_id;
```



ここで、<begin snap>および<end snap>は、スナップショットの開始 ID と終了 ID です。

長期間実行されているアップグレード・ジョブ

標準の `adtimrpt.sql` レポートの「時間がかかっている上位 100 のジョブ」のセクションには、AD パラレル・ジョブのすべてのワーカーが個別にリストされることに注意してください。そのため、上位 100 がわずかなジョブで占められている場合もあります。

次の SQL は、最大経過時間の順(降順)ですべてのジョブをリストし、AD パラレル・ジョブのすべてのワーカーを 1 行でレポートする際に使用できます。

<session_id>は、ユーザー・セッションではなく、アップグレード・セッションの ID であることに注意してください。

同じフェーズで何度も呼び出されるジョブ(`akload.class`、`LoadMap.class`、`XDOLoader.class` など)に注意が必要です。

最後に、スクリプトがレポートするのは、完了済のジョブのみであることにも気を付けてください。

```
SELECT
phase,
phase_name,
product,
job_name,
max_elapsed_time,
min_start_time,
max_end_time,
workers
FROM
(SELECT
    phase,
    phase_name,
    product,
    job_name,
    MAX(elapsed_time) elapsed_time_unconv,
    LPAD(FLOOR(MAX(elapsed_time)*24), 4)||':'||
    LPAD(FLOOR((MAX(elapsed_time)*24-floor(MAX(elapsed_time)*24))*60), 2, '0')||':'||
    LPAD(MOD(ROUND(MAX(elapsed_time)*86400), 60), 2, '0')
    max_elapsed_time,
    INITCAP(TO_CHAR(MIN(start_time), ' MON DD HH24:MI',
        'NLS_DATE_LANGUAGE = american')) min_start_time,
    INITCAP(TO_CHAR(MAX(end_time), ' MON DD HH24:MI',
        'NLS_DATE_LANGUAGE = american')) max_end_time,
    count(worker_id) workers
```

ORACLE[®] 12

E-BUSINESS SUITE

```
FROM ad_task_timing
WHERE session_id = <session_id>
GROUP BY phase, phase_name, product, job_name)
ORDER BY elapsed_time_unconv DESC;
```

長期間実行されている SQL とジョブの照合

次のバリエーションは、2つの時間間隔の間の任意の時点で実行されているジョブで、最も長期間実行されているジョブを先に示します。これは、SQL とジョブの照合に便利です。

<start_time> = レポートを開始する期間で、書式は YYYYMMDDHH24MISS です

<end_time> = レポートを終了する期間で、書式は YYYYMMDDHH24MISS です

このスクリプトでレポートするには、ジョブが完了している必要があることに注意してください。

```
SELECT
phase,
phase_name,
product,
job_name,
max_elapsed_time,
min_start_time,
max_end_time,
workers
FROM
(
SELECT
phase,
phase_name,
product,
job_name,
MAX(elapsed_time) elapsed_time_unconv,
LPAD(FLOOR(MAX(elapsed_time)*24), 4)||':'||
LPAD(FLOOR((MAX(elapsed_time)*24-floor(MAX(elapsed_time)*24))*60), 2, '0')||':'||
LPAD(MOD(ROUND(MAX(elapsed_time)*86400), 60), 2, '0')
max_elapsed_time,
INITCAP(TO_CHAR(MIN(start_time), ' MON DD HH24:MI',
'NLS_DATE_LANGUAGE = american')) min_start_time,
INITCAP(TO_CHAR(MAX(end_time), ' MON DD HH24:MI',
'NLS_DATE_LANGUAGE = american')) max_end_time,
count(worker_id) workers
FROM ad_task_timing
WHERE session_id = <session_id>
AND
```



```
(
start_time BETWEEN TO_DATE('<start_time>','YYYYMMDDHH24MISS')
AND TO_DATE('<end_time>','YYYYMMDDHH24MISS')
OR
NVL(end_time, start_time+elapsed_time)
BETWEEN TO_DATE('<start_time>','YYYYMMDDHH24MISS')
AND TO_DATE('<end_time>','YYYYMMDDHH24MISS')
)
GROUP BY phase, phase_name, product, job_name)
ORDER BY elapsed_time_unconv DESC;
```

進行中のアップグレード AD ジョブを検出するには、adctrl オプション 1 (ワーカー・ステータスの表示)を使用します。

パッチ・ログ・ファイルを確認することで、開始時期を特定できます。

例

```
$ cat u_merged.log|grep -A2 cstpostimportaad.sql
```

```
Assigned: file cstpostimportaad.sql on worker 48 for product bom username BOM. Time is: Fri
Mar 22 2013 22:48:54
```

アップグレードの異なる時期に実行されるジョブのレポート

次の内容の把握に役立ちます。

- 特定の時期にどのジョブが実行されていたか。(AWR で)長期間実行中の SQL が属するジョブの特定に役立ちます。
- AWR のイベント(競合/待機など)の原因となっている特定のジョブ。
- どのジョブが別のジョブとして同時に実行されていたか。
- どのジョブがワーカー使用率の低かった長い期間中に実行されていたか。ただし、これは間違ったことではないので注意してください。たとえば、パラレル SQL/DML を実行するジョブや、平行してオブジェクトを作成するジョブです。

次のスクリプトは、アップグレード中に、指定の間隔で実行されていたジョブについてレポートします。

<session_id> = アップグレード・セッションの ID

<interval> = 分単位の間隔

<start_time> = レポートを開始する期間で、書式は YYYYMMDDHH24MISS です



<end_time> = レポートを終了する期間で、書式は YYYYMMDDHH24MISS です

日時にこの書式を使用するのは、すべての地域で同一であり、デリミタがないためです。

<start time>と<end time>を-1 にすると、アップグレード・セッションの期間全体がレポートされます

<end time>を-1 にすると、開始時に実行されていたジョブのみがレポートされます

このスクリプトで表示されるのは、任意の間隔に実行されていたすべてのジョブではなく、サンプル期間に実行されていたジョブのみであることに注意してください。

FND_TABLES は、間隔行を生成するためのダミー表として使用されますが、これにより、最大で約 20,000 の間隔ができるため、これを超える間隔が作成されない間隔サイズを選択してください。

```
BREAK ON time_slot
```

```
WITH
ad_start_end AS
(SELECT MIN(start_time) start_time, MAX(end_time) end_time
FROM ad_task_timing
WHERE session_id = <session_id>
)
,intervals AS
(SELECT
(NVL(TO_DATE(DECODE(<start_time>,'-1',NULL,<start_time>),'YYYYMMDDHH24MISS'),st.start_time) + (((rownum-1)* TO_NUMBER(<interval>))/(24*60))) interval_time
FROM
fnd_tables ft,
ad_start_end st
WHERE rownum <= DECODE(<start_time>
, '-1'
, CEIL(((st.end_time-
st.start_time)*24*60)/TO_NUMBER(<interval>))+1
, DECODE(<end_time>
, '-1'
, 1
, CEIL(((TO_DATE(DECODE(<end_time>,'-1',NULL,<end_time>),'YYYYMMDDHH24MISS')
-TO_DATE(DECODE(<start_time>,'-1',NULL,<start_time>),'YYYYMMDDHH24MISS'))
*24*60)
/TO_NUMBER(<interval>))+1
```



```

        )
    )
)
SELECT
TO_CHAR(di.interval_time, 'DD-MON-YYYY HH24:MI:SS') time_slot,
adt.job_name job_running,
COUNT(adt.worker_id) workers
FROM
intervals di,
ad_task_timing adt
WHERE di.interval_time BETWEEN adt.start_time
AND NVL(adt.end_time,
        DECODE(adt.elapsed_time
                , NULL
                , di.interval_time
                , adt.start_time+adt.elapsed_time)
        )
AND adt.session_id = <session_id>
GROUP BY di.interval_time, job_name
ORDER BY di.interval_time, job_name;

```

ワーカー使用率のレポート

次のスクリプトは、アップグレード中の中特定の間隔におけるワーカー使用率のレポートに使用できます。

<session_id> = アップグレード・セッションの ID

<interval> = 分単位の間隔

前のスクリプト同様、FND_TABLES は、間隔行を生成するためのダミー表として使用されますが、これにより、最大で約 20,000 の間隔ができるため、これを超える間隔が作成されない間隔サイズを選択してください。

ワーカー使用率が AD ワーカーの数より大幅に低くなっている長い期間を探します。ただし、これは間違ったことではないので注意してください。たとえば、パラレル SQL/DML を実行するジョブや、平行してオブジェクトを作成するジョブです。

```

WITH
ad_start_end AS
(SELECT MIN(start_time) start_time, MAX(end_time) end_time
FROM ad_task_timing
WHERE session_id = <session_id>
)

```



```
,intervals AS
(SELECT (st.start_time + (((rownum-1)* TO_NUMBER(<interval>))/(24*60))) interval_time
FROM
fnd_tables ft,
ad_start_end st
WHERE rownum <= CEIL(((st.end_time-st.start_time)*24*60)/TO_NUMBER(<interval>))+1
)
SELECT
TO_CHAR(di.interval_time,'DD-MON-YYYY HH24:MI:SS') time_slot,
COUNT(ad.worker_id) workers,
DECODE(MIN(ad.job_name),MAX(ad.job_name),MIN(ad.job_name),'Multiple') job_running
FROM
intervals di,
ad_task_timing adt
WHERE di.interval_time BETWEEN adt.start_time
AND NVL(adt.end_time,
        DECODE(adt.elapsed_time
               ,NULL
               ,di.interval_time
               ,adt.start_time+adt.elapsed_time)
        )
AND adt.session_id = <session_id>
GROUP BY di.interval_time
ORDER BY di.interval_time;
```

特定ジョブのバッチ/ワーカー使用率のレポート

次のスクリプトは、特定ジョブの間の間隔で進行しているバッチ数のレポートに使用できます。また、バッチ・サイズや処理速度の増減も示します。

<script_name> = 分析対象のスクリプト/ジョブ

<interval> = 分単位の間隔

FND_TABLES は、間隔行を生成するためのダミー表として使用されますが、これにより、最大で約 20,000 の間隔ができるため、これを超える間隔が作成されない間隔サイズを選択する必要があります。再度記載しておきます。

以前のアップグレードに、同じスクリプト名が存在する可能性があることに注意してください。そのため、スクリプトが最後に実行された日付のデータのみが選択されるよう、このスクリプトを編集する必要があります。



また、script_name は、タスク・タイミング・レポートの job_name と同じではない場合があることにも注意してください。たとえば、バージョン番号が追加されていることがあります。スクリプト名は通常、実際のスクリプトの変数 l_script_name に割り当てられています。

```
WITH
ad_start_end AS
(SELECT MIN(start_date) start_time, MAX(end_date) end_time
FROM ad_parallel_update_units aduu,
ad_parallel_updates adu
WHERE aduu.update_id = adu.update_id
AND adu.script_name = '<script_name>'
)
,intervals AS
(SELECT (st.start_time + (((rownum-1)* TO_NUMBER(<interval>))/(24*60))) interval_time
FROM
fnd_tables ft,
ad_start_end st
WHERE rownum <= CEIL(((st.end_time-st.start_time)*24*60)/TO_NUMBER(<interval>))+1
)
SELECT
adu.script_name,
TO_CHAR(di.interval_time, 'DD-MON-YYYY HH24:MI:SS') time_slot,
COUNT(*) batches,
ROUND(AVG(aduu.rows_processed),0) avg_row_process,
ROUND(AVG((aduu.end_date - aduu.start_date)*(24*60*60)),0) avg_bat_tim_secs,
DECODE(SUM(aduu.rows_processed),0,0,ROUND(((SUM(aduu.end_date-aduu.start_date))*(24*60*60)*10000)/(SUM(aduu.rows_processed)),0)) sec_per_10k_rows,
ROUND(AVG(aduu.end_block + 1 - aduu.start_block),1) avg_blocks_per_bat
FROM
intervals di,
ad_parallel_update_units aduu,
ad_parallel_updates adu
WHERE aduu.update_id = adu.update_id
AND adu.script_name = '<script_name>'
AND di.interval_time BETWEEN aduu.start_date AND NVL(aduu.end_date,di.interval_time)
GROUP BY adu.script_name, di.interval_time
ORDER BY adu.script_name, di.interval_time;
```

Oracle E-Business Suite のすべての表に関する CBO 統計のレポート

アップグレード中、特に adsstats.sql を実行する前に、Oracle E-Business Suite の表に関する CBO 統計のレポートが必要になる場合があります。



スクリプト `adsstats.sql` により、アップグレードが終わるまでに、統計が正しく追加されます。アップグレード中の表の統計が正しくないことはわかりません。そのため、表に `null` やゼロがあったこと、CBO 統計が不正確であったこと、また、これが実行計画の負荷が高い理由であることには気が付きません。

```
SELECT owner, table_name, num_rows, TO_CHAR(last_analyzed, 'DD-MON-YYYY HH24:MI:SS')
last_analyzed
FROM all_tables
WHERE owner IN
(SELECT upper(oracle_username) sname
FROM fnd_oracle_userid
WHERE oracle_id BETWEEN 900 AND 999
AND read_only_flag = 'U'
UNION ALL
SELECT DISTINCT upper(oracle_username) sname
FROM fnd_oracle_userid a,
fnd_product_installations b
WHERE a.oracle_id = b.oracle_id
)
ORDER BY owner, table_name;
```

競合(待機)とその発生場所の分析

次は、AWR の 2 つのスナップショット間で、特定の待機が発生している `sql_ids` を表示します。

<db_id>はデータベース ID、<inst_num>はインスタンス番号、<wait name>は待機の名前、<begin snap>および<end snap>はスナップショットの開始 ID と終了 ID です。

```
select ss.sql_id, ss.time_waited, ss.counts_waited, tt.total_time,
ROUND((ss.time_waited*100/tt.total_time),1) percent
from
(select s.sql_id
, COUNT(*) counts_waited, SUM(time_waited) time_waited
from DBA_HIST_ACTIVE_SESS_HISTORY s, DBA_HIST_SEG_STAT_OBJ o
where s.dbid = <db_id> and s.instance_number = <inst_num>
and o.dbid (+) = s.dbid
and o.obj# (+) = s.current_obj#
and s.event = '<wait name>'
and snap_id > <begin_snap> and snap_id <= <end_snap>
group by s.sql_id) ss
, (select SUM(time_waited) total_time
from DBA_HIST_ACTIVE_SESS_HISTORY t
where t.dbid = <db_id> and t.instance_number = <inst_num>
and t.event = '<wait name>')
```



```
and t.snap_id between <begin_snap> and <end_snap>) tt
order by ss.counts_waited desc;
```

次の SQL は、(AWR の 2 つのスナップショット間の)指定された SQL ID で特定の待機が発生しているオブジェクトを表示します。

ここで、<db_id>はデータベース ID、<inst_num>はインスタンス番号、<wait name>は待機の名前、<sql_id>は SQL ID、<begin snap>および<end snap>はスナップショットの開始 ID と終了 ID です。

```
select ss.sql_id, ss.event, ss.current_obj#, ss.owner, ss.object_name, ss.object_type,
ss.time_waited, ss.counts_waited, tt.total_time,
ROUND((ss.time_waited*100/tt.total_time),1) percent
from
(select s.sql_id, s.event, s.current_obj#, o.owner, o.object_name, o.object_type
, COUNT(*) counts_waited, SUM(time_waited) time_waited
from DBA_HIST_ACTIVE_SESS_HISTORY s, DBA_HIST_SEG_STAT_OBJ o
where s.dbid = <db_id> and s.instance_number = <inst_num>
and s.sql_id = '<sql_id>'
and s.event = '<wait name>'
and o.dbid (+) = s.dbid
and o.obj# (+) = s.current_obj#
and snap_id between <begin_snap> and <end_snap>
group by s.sql_id, s.event, s.current_obj#, o.owner, o.object_name, o.object_type) ss
, (select SUM(time_waited) total_time
from DBA_HIST_ACTIVE_SESS_HISTORY t
where t.dbid = <db_id> and t.instance_number = <inst_num>
and t.sql_id = '<sql_id>'
and t.event = '<wait name>'
and t.snap_id > <begin_snap> and t.snap_id <= <end_snap>) tt
order by ss.counts_waited desc
```

一般的な解決策

既知の問題

長期間実行されているジョブおよび SQL が識別されたら、既知の問題および可能性のある解決策や回避策について、My Oracle Support で確認してください。

ただし、修正または回避策で、発生する特定の問題を必ず解決できるわけではないことを念頭に置いておいてください。一部のバグや SR には、パフォーマンスの問題の症状は記述されず、解決策のみが示されています。



可能な場合は、推奨された修正が適切であることを示すために、証拠(診断)を取得してください。

問題が完全に同一であることを(診断から)確認できない場合、修正は適用できますが、問題が完全に解決されるまで、診断の収集と解決策の検索を続けてください。

特定の既知の問題や修正に固執しても、結局は、発生したパフォーマンスの問題がその既知の問題とは異なるものとわかり、必要な修正は違うものだったということが頻繁にあります。

これは、それぞれのお客様によって、Oracle E-Business Suite のモジュールや機能の使用方法が異なり、データの容量や配布も違っているためです。データベースのバージョンや構成、ハードウェア構成、CBO およびデータベース初期化パラメータも異なります。

カスタム索引

長期間実行されているジョブの実行計画が不十分で、新しい索引に大きく影響することがあります。

一般的に、実行計画では表のフル・スキャン、選択が限定的ではない索引範囲スキャン、または索引スキップ・スキャンが行われますが、このいずれも、表がアクセスされたときに大量の行が除外される原因になります。

そのような場合は、カスタム索引を作成して行を除外すれば、表でアクセスされる行数を減らすことや、索引スキップ・スキャンの必要を低減することができます。

その場合でも、パフォーマンスの問題として、必ず(診断の証拠も添えて)SR を作成してください。パフォーマンスの問題は、標準のコードで解決する必要があります。

不十分な実行計画を対象とした SQL プロファイル

前述の診断(特に、表示カーソル・レポートまたは SQL トレース)を使用している際に、長期間実行されているジョブの実行計画が不十分であることがわかった場合は、SQL プロファイルを使用すると、CBO でより適切な実行計画が選択されるよう支援するヒントを適用できます。これを実行するには、SQL チューニングの専門知識が必要です。

その場合でも、パフォーマンスの問題として、必ず(診断の証拠とともに)SR を作成してください。パフォーマンスの問題は、標準のコードで解決する必要があります。

SQL プロファイルのヒントの構文は、SQL で直接適用するヒントより厳密です。構文が間違っていると無視されます。大/小文字は使用できますが、引用符(")は使用できず、先頭の問合せブロック名は省略されます。



次に、最もよい方法を示します。

1. SQL で(ヒントを追加した状態で)EXPLAIN PLAN を実行します。
2. 次を実行して、アウトラインから完全な形式のヒントを取得します: SELECT * FROM TABLE(dbms_xplan.display(FORMAT => 'ALL +OUTLINE'))。
これにより、正しい構文が取得されます。
3. SQL プロファイル・スクリプトを作成して実行します。
4. SQL で(ヒントが追加されていない状態で)EXPLAIN PLAN を実行します。これにより、SQL プロファイルが適用されたかどうかがわかり(出力の最後にある「Notes」セクションには、「SQL profile "<profile_name>" used for this statement」と記載されています)、EXPLAIN PLAN によってヒントが適用されたことが表示されます。

次に、ヒントを適用して、SQL プロファイルを作成するスクリプトの例を示します。

```
DECLARE

l_sql_fulltext clob := NULL;
lv_hint SYS.SQLPROF_ATTR := SYS.SQLPROF_ATTR ();

BEGIN

l_sql_fulltext := 'SELECT AAGC.APPLICATION_ID, AAGC.APPROVAL_GROUP_ID FROM
AME_APPROVAL_GROUP_CONFIG AAGC WHERE SYSDATE BETWEEN AAGC.START_DATE AND NVL(AAGC.END_DATE
- (1/86400), SYSDATE) AND NOT EXISTS (SELECT NULL FROM AME_ACTION_USAGES AAU, AME_RULE_USAGES
ARU, AME_ACTIONS AA, AME_ACTION_TYPES AAT WHERE AA.ACTION_ID = AAU.ACTION_ID AND AAU.RULE_ID
= ARU.RULE_ID AND ARU.ITEM_ID = AAGC.APPLICATION_ID AND SYSDATE BETWEEN AA.START_DATE AND
NVL(AA.END_DATE - (1/86400), SYSDATE) AND SYSDATE BETWEEN AAT.START_DATE AND
NVL(AAT.END_DATE - (1/86400), SYSDATE) AND AA.PARAMETER = TO_CHAR(AAGC.APPROVAL_GROUP_ID)
AND AAT.ACTION_TYPE_ID = AA.ACTION_TYPE_ID AND AAT.NAME IN ('pre-chain-of-authority
approvals', 'post-chain-of-authority approvals', 'approval-group chain of authority'))
AND ROWNUM < 2) ORDER BY AAGC.APPLICATION_ID';

lv_hint.EXTEND;
lv_hint(1) := 'BEGIN_OUTLINE_DATA';
lv_hint.EXTEND;
lv_hint(2) := 'USE_NL(@"SEL$2" "AAT"@"SEL$2")';
lv_hint.EXTEND;
lv_hint(3) := 'USE_NL(@"SEL$2" "ARU"@"SEL$2")';
lv_hint.EXTEND;
lv_hint(4) := 'USE_NL(@"SEL$2" "AAU"@"SEL$2")';
lv_hint.EXTEND;
lv_hint(5) := 'LEADING(@"SEL$2" "AA"@"SEL$2" "AAU"@"SEL$2" "ARU"@"SEL$2" "AAT"@"SEL$2")';
```




```

lv_hint.EXTEND;
lv_hint(6) := 'INDEX(@"SEL$2" "AAT"@"SEL$2" ("AME_ACTION_TYPES"."ACTION_TYPE_ID"
"AME_ACTION_TYPES"."START_DATE" "AME_ACTION_TYPES"."END_DATE"))';
lv_hint.EXTEND;
lv_hint(7) := 'INDEX(@"SEL$2" "ARU"@"SEL$2" ("AME_RULE_USAGES"."RULE_ID"
"AME_RULE_USAGES"."START_DATE" "AME_RULE_USAGES"."END_DATE"))';
lv_hint.EXTEND;
lv_hint(8) := 'INDEX(@"SEL$2" "AAU"@"SEL$2" ("AME_ACTION_USAGES"."ACTION_ID"))';
lv_hint.EXTEND;
lv_hint(9) := 'INDEX(@"SEL$2" "AA"@"SEL$2" ("AME_ACTIONS"."PARAMETER"))';
lv_hint.EXTEND;
lv_hint(10) := 'FULL(@"SEL$1" "AAGC"@"SEL$1")';
lv_hint.EXTEND;
lv_hint(11) := 'LEADING(@"SEL$1" "AAGC"@"SEL$1")';
lv_hint.EXTEND;
lv_hint(12) := 'IGNORE_OPTIM_EMBEDDED_HINTS';
lv_hint.EXTEND;
lv_hint(13) := 'END_OUTLINE_DATA';

dbms_sqlltune.drop_sql_profile
(name => 'R1220_AMEMIGCFG_1'
,ignore => TRUE
);

dbms_sqlltune.import_sql_profile(
sql_text => l_sql_fulltext
,category => 'DEFAULT'
,name => 'R1220_AMEMIGCFG_1'
,profile => lv_hint
,description => 'R1220 amemigcfg.sql 23y6y8d0r9v61'
,force_match => TRUE
);

END;
/

```

SQL ベースラインも、カーソル・キャッシュ、AWR または SQL チューニング・セットからの実行計画のインポートに使用できます。ただし、ヒントをインポートすることはできません。

次に、SQL プロファイルについてまとめます。

- 適用が簡単です。
- 結合順序、アクセスおよび結合メソッドを直接指定する際に使用できます。
- 安定しています。



SQL チューニング・アドバイザーの制限事項。

R12.2.n のアップグレード中に、SQL チューニング・アドバイザー(STA)を使用してパフォーマンスの問題を解決することには、次のデメリットがあります。

- 診断および後続の解決策の提供が遅れる場合があります。
- 使用可能な実行計画の取得に STA を使用する必要があるということは、解決が必要な問題が根底にあることを示しています。
- STA の実行時、または STA が実行されているセッションに(代表的なデータを含む)表が追加されている場合、特定できるのはより効率のよい実行計画のみです。そのため、グローバル一時表、一時表、内部表、暫定表には使用できません。
- CBO 統計が正しい場合、特定できるのは、より効率のよい実行計画のみです。
- STA により生成される SQL プロファイルでは、実際の結合順序、アクセスおよび結合メソッドの指定にヒントが使用されません。かわりに、OPT_ESTIMATE ヒント(および通常は SCALE_ROWS)を使用して、不十分な実行計画の原因となっている過剰/過小なカーディナリティを修正します。後から CBO 統計を変更する場合は、安定している必要はありません。ヒントを使用する SQL プロファイルをお勧めします。

実用的で安定した実行計画

ほとんどの場合、最適な実行計画を提供し、無駄(つまり、後続のステージで除外される行への不必要なアクセス)を最小限にする結合順序が 1 つは存在します。多くの場合、その結合順序は、アプリケーションのフローに従った自然なものです。

AD パラレル・ジョブ

AD パラレル・ジョブは、(AD パラレルを初期化するために
AD_PARALLEL_UPDATES_PKG.INITIALIZE_ROWID_RANGE を呼び出す際に指定された)駆動表から行 ID のバッチにアクセスします。

ほとんどの場合、これらのバッチには、合計のわずかな割合の行しか含まれていません。そのため、実行計画が(行 ID でアクセスして)駆動表を使用してリードし、ネスト・ループ結合と索引アクセスを使用する必要があります。これにより、AD パラレル・ジョブの実用的な計画が提供されます。最適な実行計画ではありませんが、特に効率が悪いわけでもありません。

小規模な表(特に検索や参照)の場合は、完全な表スキャンおよびハッシュ結合が適していますが、(ワークロード全体と比較した場合)その違いは無視できる程度です。



通常は、rowid、leading、use_nl および index ヒントが使用されます。ドキュメントに記載されていないカーディナリティ・ヒントも、駆動表の低いカーディナリティ(cardinality(ai 1)など)の指定に使用できます。

パラレル SQL

パラレル SQL を使用するジョブには、常にではありませんが多くの場合、表のフル・スキャンおよびハッシュ結合が適しています。また、無駄を最小限にする結合順序の使用にもメリットがあります。

オンライン・パッチの有効化/オンライン・パッチで長期間実行されている SQL

オンライン・パッチの有効化、R12.2.n RUP およびその他のオンライン・パッチには、実行計画が不十分で、長期間実行されている内部 SQL が存在する場合があります。これは、AWR および TKPROF レポートに、(V\$ビューまたは SYS/SYSTEM オブジェクトの)値が高い内部 SQL として現れます。

このような場合に、固定オブジェクト統計やディクショナリ統計の収集が役立ちます。特に、エディショニング・オブジェクトに効果的です。

実行計画が不十分な内部 SQL の数が非常に少ない場合や、オブジェクト数が少ない場合は、すべてのディクショナリ統計または固定オブジェクト統計を収集せずに、特定のオブジェクトを対象にすることが可能です。

長期間実行されている索引作成

通常、xdf および odf ジョブに当てはまります。

索引が作成されているスナップショットの AWR レポートを取得します。

パラレル DML を使用しない実行

まず、odf ファイルが AutoPatch によって処理されるときに、索引が平行して作成されていること(CREATE INDEX PARALLEL など)を確認します。AWR に DDL コマンドが表示されず(SQL Ordered By)。

索引が順番に作成されている場合は、AutoPatch パラメータ parallel_index_threshold を減らす必要があります。このパラメータにより、表内のブロック数が指定されます。表に含まれるブロック数がスレッドの設定より少ない場合は、パラレル AD ワーカーとシリアル DML を使用して索引が作成されます(各ワーカーが異なる索引を作成します)。表に含まれるブロック数がスレッ



ドの設定より多い場合は、1つのワーカーとパラレル DML を使用して索引が作成されます。有効な値

は、0 から 2147483647 です。0 に設定すると、パラレル・ワーカーとシリアル DML を使用して索引が作成されます。デフォルト値の 20000 は、20,000 ブロックのスレッドであることを意味します。

それ以外の場合は、表の統計を収集する必要があります。統計がない場合や正しくない(少なすぎるなど)場合は、AutoPatch (adodfcmp)で誤った数のブロックが取得され、シリアル DML を使用した作成が選択されている可能性があります。

索引を平行して作成できない場合は、次を参照して、索引の事前作成を検討してください。

parallel_max_servers/並列度

parallel_max_servers を増やす必要があるかどうかを確認します。

SQL で使用されているパラレル・スレーブの数が予想より少ない場合は、その他の同時タスクによって、パラレル・スレーブの数が制限されている可能性があります。

索引の作成と同時に、同時タスクが実行されているかどうか、AWR を確認できます。

別の制限は、CPU_COUNT * PARALLEL_THREADS_PER_CPU パラメータです。

設定されている場合、CPU_COUNT は CPU コア数かゼロにする必要があります。CPU_COUNT が設定されていない場合、またはゼロに設定されている場合は、デフォルトで、オペレーティング・システムからレポートされた CPU 数に設定されます。そのため、そのまま使用できます。

存在する場合、PARALLEL_THREADS_PER_CPU は 2 にする必要があります。設定されていないときのデフォルトはプラットフォームに依存し(通常は 2)、ほとんどの場合はそれで適切です。

自動並列度ポリシー(PARALLEL_DEGREE_POLICY)は、有効化しないことに注意してください(デフォルトは MANUAL です)。

『Oracle Database VLDB およびパーティショニング』の「パラレル実行の仕組み」および「パラレル実行のための一般的なパラメータのチューニング」の項を参照してください。

pga_aggregate_target

索引の作成では、PGA の領域を使用してソートが行われます。一時領域が大量に使用されている場合(およびそのときのみ)は、pga_aggregate_target の増加を検討する必要があります。



パラレル・スレーブ間の競合

索引が並行して作成されている場合(および十分なスレーブが使用されている場合)は、索引を作成しているパラレル・スレーブ間の競合、または別の同時プロセスとの競合が、パフォーマンスの問題の原因になっている可能性があります。

並列度が高すぎても、競合の原因になることに注意してください。

索引が作成されている期間の AWR レポートを取得し、競合(待機や CPU 使用率)の有無を確認します。

AWR に競合がない場合は、索引の事前作成で問題が解決する可能性は低いです。同じコマンドと同じ並列度で、索引が自動作成されます。

事前作成

索引が(odfによって)順番に作成されている場合で、並行して作成できないか、(odfで)並行して索引を作成する際に競合が多数発生する場合には、索引の事前作成が解決につながる場合があります。ただし、いくつかの注意点があります。

- 早まって実行しないでください。(odfを使用して)通常の作成が始まる前に、その表で負荷のかかる DML (索引で使用する列の挿入、削除または更新)が続いて実行される場合、その DML にかかる時間が大幅に長くなる可能性があります。そのため、可能な場合には、表で負荷のかかる DML を実行するジョブが終わってから、索引を事前作成してください。
- また、索引は、odf ファイルと完全に同一の定義で作成するようにしてください。
adodfcmp により、事前作成された索引と、odf ファイルの定義が比較されます。一致しない場合は、adodfcmp により、索引が変更または再作成されます。
- ALTER INDEX を実行して、後から必ず、並列度を元に戻してください(NOPARALLEL など)。

これらのことから、索引を手動で事前作成する場合は、できるだけ作成を遅らせてください。

長期間実行されている DML ジョブの索引における高レベルの競合

長期間実行中のジョブで負荷のかかる DML (一般的には挿入ですが、更新や削除の場合もあります)が実行されていて、競合レベルが高く、その競合が特定の索引に存在する場合は、ジョブの前に索引を削除し、後から再作成することを検討してください(その索引がそのときに、どの実行計画にも使用されていない場合)。



または、アップグレード前に索引を削除して、後から再作成することを検討してください。ただし、アップグレード中に、索引が使用されないようにしてください。

コマンド `ALTER INDEX <index> MONITORING USAGE` を使用し、ビュー `V$OBJECT_USAGE` を問い合せて、索引が特定の期間に使用されたかどうかを確認します。

また、AWR の表 `DBA_HIST_SQL_PLAN` により、どの索引が使用されているかが示され、`DBA_HIST_ACTIVE_SESS_HISTORY` により、DML ごとに索引の保守にかかった時間が示されます。

索引は、まったく同一の定義で、並行して再作成するようにしてください。また、`ALTER INDEX` を実行して、後から必ず、並列度を元に戻してください(`NOPARALLEL` など)。

索引の再作成はすべての行に対して行われるため、これを決定する際には、DML の実行中に挿入/削除/更新される行の割合を考慮してください。カスタム索引では、(カスタム索引により、いずれかのアップグレード・ジョブが大幅に高速になることがなければ)アップグレード前に索引を削除し、後から再作成することが推奨されているため、カスタム索引の場合に特に便利です。

高レベルの"enq: HW - contention"または"enq: HV - contention"

これは、前述のケースに関連します。

表および索引への挿入を行っている長期間実行中のジョブに、高レベルの待機("enq: HW - contention"または"enq: HV - contention")が存在する場合は、次のことを実行できます。

- 待機が主に特定の索引で発生している場合は、ジョブの前にその索引を削除し、その索引がそのときに使用されていなければ、(前述のように)後から再作成します。
- 表および索引のエクステンツ・サイズを増やします。
- 表および索引にエクステンツを事前に割り当てます。
- 表および索引をパーティション化し、負荷を複数のパーティションに分散します。表および索引は、稼働後もパーティション化されることに注意してください。そのためこれは、パーティショニングすることが、本番環境で稼働した後にもメリットがある場合にのみ実行します。

これらの"enq: HW - contention"または"enq: HV - contention"には、"enq: TX - contention"やバッファ・ビジー待機など、その他の高レベルの待機が伴うことが多いです。



HW エンキューは、セグメントの上限を超える、領域の割当て管理に使用されます。そのため、一般的に、待機"enq: HW - contention"は、セッションがエクステンツの割当てを待機していることを意味します。

待機"enq: HV - contention"も、パラレル・スレーブがエクステンツの割当てを待機しているときに発生することを除けば、同じです。

AWR 表を直接問い合わせ、待機の発生場所(どのオブジェクトかなど)をより詳細に取得することが可能です。「便利なスクリプト」の項を参照してください。

高レベルの REDO ログ待機であるログ・バッファ領域、ログ・ファイル同期など。

REDO ログ(特に、ログ・バッファ領域およびログ・ファイル同期)に関連した高レベルの待機が存在する場合は、REDO ログの構成の変更、より高速なファイラへの移行、ログのサイズまたは数の増加、あるいはログの並列度(隠し初期化パラメータ `_log_parallelism_max`)の増加を検討してください。

NOLOGGING の実行も可能ですが、お薦めではありません。すべての表と索引でロギングをオフにし、オフに切り換えたすべての表と索引で、アップグレード後にオンに戻す必要があります。

長期間実行されている統計の収集(adsstats.sql)

アップグレードで CBO 統計(adsstats.sql)の収集に時間がかかっている場合は、次の対応を検討してください。

- `parallel_max_servers` および `pga_aggregate_target` の増加(adsstats.sql は、コンカレント・タスクが少ないとき、または存在しないときに実行してください)。
- すべてのノードを Oracle RAC システムで使用。
- テストの実行中に収集した統計のインポート。

最後の提案は、リリース 12.2.n のすべてのアップグレードで行われる、非常に重要なアクションであることに注意してください。そのため、「準備と計画」の項で、より詳細に検討して説明されています。

特定のパフォーマンス問題を解決するための統計の収集または削除

パフォーマンスの問題は、特定のオブジェクトに関する統計の収集または削除が必要なことを示している場合があります(「パフォーマンスの問題の解決」の項を参照してください)。

ORACLE[®] 12

E-BUSINESS SUITE

新しい表

リリース 12 のアップグレードでは、多くの表が作成されて追加されます。大部分の表では、アップグレードの終盤まで統計は収集されません(たとえば、R12.2.0 のアップグレードの last+63 フェーズにおける adsstats.sql によるもの)。

そのため、(リリース 12 の新しい表では)統計が空で、動的サンプリングでは、適切な実行計画の結果が出ません。または、統計が正しくない場合もあります。

そのような場合には、(FND_STATS.GATHER_TABLE_STATS を使用して)アップグレードの特定ステージで個別の表の統計を収集することをお勧めします。

ただし、次のことがわかっている場合にのみ、個別の表の統計を収集してください。

- 表の統計が正しくないこと、または空であることが、明らかにパフォーマンスの問題の原因になっている場合。
- その表に、後から大量の挿入が行われない場合(または主要な結合/フィルタ列の個別値 (NDV)の数値やヒストグラムを含む列の値の配布を大幅に変更する場合)。そのような場合は、CBO 統計が正確でなくなり、後続の実行計画が不十分になります。

(動的サンプリングが行われるため)不正確な統計があるよりも統計がない方がよいこと、(表が追加されていないときに収集された)ゼロの統計よりも明らかによいことを念頭に置いてください。

統計の収集に関するガイドは、Oracle Support から探すことも可能です。

一時表

パフォーマンスの問題が短期表または一時表(グローバル一時表も含む)に起因していて、これらの表の統計に含まれるものが代表的な統計ではない場合、次に示す 2 つのいずれかを実行することをお勧めします。

- アップグレード前に、DBMS_STATS.DELETE_TABLE_STATS を使用して統計を削除し、DBMS_STATS.LOCK_TABLE_STATS でロックして、FND_STATS.LOAD_XCLUD_TAB で除外します。その後、動的サンプリングを行います。
- 適切な時点で FND_STATS.GATHER_TABLE_STATS を使用して代表的な統計を収集するか、(CASCADE = TRUE(デフォルト)を指定して) FND_STATS.RESTORE_TABLE_STATS で代表的な統計をインポートします。次に、DBMS_STATS.LOCK_TABLE_STATS でロックし、FND_STATS.LOAD_XCLUD_TAB を



使用して除外します。代表的な統計は、(CASCADE = TRUE(デフォルト)を指定して) FND_STATS.BACKUP_TABLE_STATS で、あらかじめエクスポートしておきます。

「準備と計画 - アップグレード前の環境/アクティビティ」、「CBO 統計の収集」、「短期表/一時表のゼロまたは不正な統計」の項を参照してください。

長期間実行されているマテリアライズド・ビューxdf/odfs

マテリアライズド・ビュー(MV)を作成している長期間実行中の xdf または odf ジョブがある場合は、大きな MV ログのクリーン・アップまたは切捨てを検討します(これには、MV の完全なリフレッシュが必要なことに注意してください)。

長期間実行されている不要なジョブ

長期間実行されているジョブが実際に必要かどうか、特に、データの変更(行の挿入、更新または削除)を行っていないかどうかを確認します。中には、使用されていない、またはライセンスされていないモジュールやローカライゼーション用のジョブがあります。

Oracle Support では、ジョブのスキップを推奨したり、修正や回避策を提供したりする場合があります。

ジョブのスキップ

Oracle Support により、特定のジョブがインスタンスに不要なものであると特定された場合は、そのジョブをスキップする方法が 2 つあります。

- インストール用の統合ドライバ(u ドライバ)からそのジョブをコメント・アウトします。
- adctrl ユーティリティを実行して、隠しオプション 8 を選択します。

AD パラレル・ワーカーの最大数

AutoPatch により、ワーカーの最大数が指定されることに注意してください。その数はいくつかの要因によって異なり、特定の状況では、必要なワーカー数を下回る場合があります。そのため、問題を解決する必要があります。

次のようなエラー・メッセージが表示されます。

AD ユーティリティでサポートできる最大ワーカー数は 999 です。現在のデータベース構成では、最大 Z のワーカーがサポートされています。X から Y の間のワーカーを使用することをお勧めします。



上のメッセージで、X、Y および Z は次のようにして計算されています。

$$X = 2 * \text{cpu_count}$$

$$Y = 4 * \text{cpu_count}$$

$$Z = (\text{processes} - ((\text{total rows of v\$process}) + \text{parallel_max_servers})) / 2$$

意味:

`processes` = `processes` DB 初期化パラメータ。

`parallel_max_servers` = `parallel_max_servers` DB 初期化パラメータ。

`total rows of v\$process` = `v\$process` から `count(1)` を選択

`cpu_count` = `cpu_count` DB 初期化パラメータ(またはデフォルト値)。

このことから、最大数は基本的に(データベース初期化ファイルの)使用可能な `processes` の 50% であり、現在の `processes` 数(`v$processes` の行数)と、パラレル問合せ/DML で使用可能な最大数が減算されています。

My Oracle Support のドキュメントである『How Does Adpatch Determine The Number Of Workers To Recommend? (Document 800024.1)』を参照してください。



Oracle E-Business Suite リリース 12.2.n の
アップグレード・ダウンタイム最小化の
ベスト・プラクティス
2014 年
著者: Jim Machin

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2014, 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。本文書はオラクル社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 Corporation およびその関連企業の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

Intel, Intel Xeon は、Intel Corporation の商標または登録商標です。すべての SPARC の商標はライセンスをもとに使用し、SPARC International, Inc. の商標または登録商標です。AMD、Opteron、AMD ロゴ、AMD Opteron ロゴは、Advanced Micro Devices, Inc. の商標または登録商標です。UNIX は、The Open Group の登録商標です。 0113

Hardware and Software, Engineered to Work Together