SYBASE®

マイグレーション技術ガイド

Adaptive Server® Enterprise

15.5

ドキュメント ID：DC01065-01-1550-01
改訂：2009 年 10 月

マニュアルの注文

マニュアルの注文を承ります。ご希望の方は、サイベース株式会社営業部または代理店までご連絡ください。マニュアルの変更は、弊社の定期的なソフトウェア・リリース時にのみ提供されます。

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

目次

はじめに

対象読者

このマニュアルは、Adaptive Server®Enterprise と異なるバージョンにマイグレードするシステム管理者を対象としています。

このマニュアルの内容

「第 1 章 マイグレーション方法」では、アップグレードして新機能をすぐに使用するユーザ、あらかじめアップグレードして後で新機能を使用するユーザにその方法を説明します。

「第 2 章 QPTune」では、QPTune ツールの包括的な概要を示します。

「第 3 章 互換モードでのクエリ・プロセッサの実行」では、新しいバージョンにアップデートしながら、以前のバージョンと同様のパフォーマンス特性を維持するユーザに「互換モード」について説明します。

関連マニュアル

Adaptive Server Enterprise には次のマニュアルが用意されています。必要に応じて参照してください。

- 使用しているプラットフォームの『リリース・ノート』－ マニュアルには記載できなかった最新の情報が記載されています。

  このリリース・ノートの最新バージョン (英語版) を入手できます。製品の CD がリリースされた後で、製品またはマニュアルに関する重要な情報が追加されているかを確認するには、Sybase® Product Manuals Web サイトを使用してください。

- 使用しているプラットフォームの『インストール・ガイド』－ すべての Adaptive Server および関連する Sybase 製品のインストール、アップグレード、設定の手順について説明しています。

- 『新機能ガイド』－ Adaptive Server の新しい機能について説明しています。また、新しい機能をサポートするためのシステム変更や、既存のアプリケーションに影響を与える可能性がある変更についても説明しています。

- 『Active Messaging ユーザーズ・ガイド』－ Active Messaging を使用して、Adaptive Server Enterprise データベースでトランザクション (データ変更) を取得し、外部アプリケーションにイベントとしてリアルタイムで渡す方法について説明しています。

- 『コンポーネント統合サービス・ユーザーズ・ガイド』－ コンポーネント統合サービスを使用して、リモートの Sybase データベースおよび Sybase 以外のデータベースに接続する方法について説明しています。

- 使用しているプラットフォームの『設定ガイド』－ 特定の設定作業の手順について説明しています。

- 『用語解説』－ Adaptive Server マニュアルで使用されている技術用語について説明しています。
- 『Historical Server ユーザーズ・ガイド』－ Historical Server を使用して、Adaptive Server のパフォーマンス情報を入手する方法について説明しています。
- 『Adaptive Server Enterprise における Java』－ Adaptive Server データベースで Java クラスをデータ型、関数、ストアド・プロシージャとしてインストールして使用する方法について説明しています。
- 『Job Scheduler ユーザーズ・ガイド』－ コマンド・ラインまたはグラフィカル・ユーザ・インタフェース (GUI) を使用して、ローカルまたはリモートの Adaptive Server でジョブのインストール、設定、作成、スケジュールを行う方法について説明しています。
- 『マイグレーション技術ガイド』－ 別のバージョンの Adaptive Server にマイグレートするための方法とツールについて説明しています。
- 『Monitor Client Library プログラマーズ・ガイド』－ Adaptive Server のパフォーマンス・データにアクセスする Monitor Client Library アプリケーションの記述方法について説明しています。
- 『Monitor Server ユーザーズ･ガイド』－ Monitor Server を使用して、Adaptive Server のパフォーマンス統計を取得する方法について説明しています。
- 『モニタリング・テーブル・ダイヤグラム』－ モニタリング・テーブルと、そのエンティティの関係をポスター形式で図解しています。フル・サイズのダイアグラムは印刷版だけで参照できます。コンパクト版は PDF 形式で参照できます。
- 『パフォーマンス&チューニング・シリーズ』－ Adaptive Server で最高のパフォーマンスを実現するためのチューニング方法について説明しています。
  - 『基本』－ Adaptive Server のパフォーマンスに関する問題の理解と調査の基本について説明しています。
  - 『統計的分析によるパフォーマンスの向上』－ Adaptive Server で統計情報がどのように保存され、表示されるかについて説明しています。また、set statistics コマンドを使用して、サーバの統計情報を分析する方法について説明しています。
  - 『ロックと同時実行制御』－ ロック・スキームを使用してパフォーマンスを向上させる方法と、同時実行性を最小限に抑えるようにインデックスを選択する方法について説明しています。
  - 『sp_sysmon による Adaptive Server の監視』－ sp_sysmon を使用してパフォーマンスをモニタリングする方法について説明しています。

    - 『モニタリング・テーブル』 – Adaptive Server のモニタリング・テーブルに統計情報や診断情報を問い合わせる方法について説明しています。
    - 『物理データベースのチューニング』 – データの物理的配置、データに割り付けられた領域、テンポラリ・データベースの管理方法について説明しています。
    - 『クエリ処理と抽象プラン』 – オプティマイザがクエリを処理する方法と、抽象プランを使用してオプティマイザのプランの一部を変更する方法について説明しています。
- 『クイック・リファレンス・ガイド』 – コマンド、関数、システム・プロシージャ、拡張システム・プロシージャ、データ型、ユーティリティの名前と構文の包括的な一覧表を記載したポケット版 (PDF 版は通常サイズ) のマニュアルです。
- 『リファレンス・マニュアル』 – 詳細な Transact-SQL® 情報を記載しています。
    - 『ビルディング・ブロック』 – データ型、関数、グローバル変数、式、識別子とワイルドカード、予約語について説明しています。
    - 『コマンド』 – コマンドについて説明しています。
    - 『プロシージャ』 – システム・プロシージャ、カタログ・ストアド・プロシージャ、システム拡張ストアド・プロシージャ、dbcc ストアド・プロシージャについて説明しています。
    - 『テーブル』 – システム・テーブル、モニタリング・テーブル、dbcc テーブルについて説明しています。
- 『システム管理ガイド』でさらに詳しく説明しています。
    - 『第 1 巻』 – 設定パラメータ、リソースの問題、文字セット、ソート順、システムの問題の診断方法に関する説明を含め、システム管理の基本の概要について説明しています。『第 1 巻』の後半は、セキュリティ管理に関する詳細な説明です。
    - 『第 2 巻』 – 物理的なリソースの管理、デバイスのミラーリング、メモリとデータ・キャッシュの設定、マルチプロセッサ・サーバとユーザ・データベースの管理、データベースのマウントとマウント解除、セグメントの作成と使用、reorg コマンドの使用、データベース一貫性の検査方法についての手順とガイドラインを説明しています。『第 2 巻』の後半では、システムとユーザ・データベースをバックアップおよびリストアする方法について説明しています。
- 『システム・テーブル・ダイアグラム』 – システム・テーブルと、そのエンティティとの関係をポスター形式で図解しています。フル・サイズのダイアグラムは印刷版だけで参照できます。コンパクト版は PDF 形式で参照できます。

- 『Transact-SQL ユーザーズ・ガイド』－ リレーショナル・データベース言語の拡張版である Sybase の Transact-SQL について説明しています。まだ経験の浅いデータベース管理システムのユーザは、このマニュアルをガイドブックとして使用してください。pubs2 および pubs3 サンプル・データベースの詳細も説明しています。
- 『トラブルシューティング・シリーズ』－
  - 『トラブルシューティング：エラー・メッセージと詳細な解決方法』－ 発生する可能性のある問題について、トラブルシューティング手順を説明しています。このマニュアルで取り上げられている問題は、Sybase 製品の保守契約を結んでいるサポート・センタに最も頻繁に寄せられるものです。
  - 『トラブルシューティング＆エラー・メッセージ・ガイド』－ 発生頻度が高い Adaptive Server のエラー・メッセージの解決方法について詳しい手順を説明しています。
- 『暗号化カラム・ユーザーズ・ガイド』－ Adaptive Server を使用して暗号化カラムを設定し、使用する方法について説明しています。
- 『インメモリ・データベース・ユーザーズ・ガイド』－ メモリ内データベースの設定および使用方法について説明しています。
- 『Adaptive Server 分散トランザクション管理機能の使用』－ 分散トランザクション処理環境での Adaptive Server DTM 機能の設定、使用、トラブルシューティングについて説明しています。
- 『IBM® Tivoli® Storage Manager と Backup Server の使用』－ IBM Tivoli Storage Manager を設定および使用して Adaptive Server のバックアップを作成する方法について説明しています。
- 『高可用性システムにおける Sybase フェールオーバの使用』－ Sybase のフェールオーバ機能を使用して、Adaptive Server を高可用性システムのコンパニオン・サーバとして設定する方法について説明しています。
- 『Unified Agent および Agent Management Console』－ Unified Agent について説明しています。Unified Agent は、分散 Sybase リソースを管理、モニタ、制御するためのランタイム・サービスを提供します。
- 『ユーティリティ・ガイド』－ オペレーティング・システム・レベルで実行される isql および bcp などの、Adaptive Server のユーティリティ・プログラムについて説明しています。
- 『Web Services ユーザーズ・ガイド』－ Adaptive Server 用の Web サービスの設定、使用、トラブルシューティング方法について説明しています。
- 『XA インタフェース統合ガイド for CICS、Encina、TUXEDO』－ X/Open XA トランザクション・マネージャを備えた Sybase DTM XA インタフェースを使用する方法について説明しています。

- 『Adaptive Server Enterprise における XML サービス』では、データベースに XML 機能を導入する、Sybase ネイティブの XML プロセッサと Sybase Java ベースの XML のサポートについて、また XML サービスに準拠したクエリとマッピング用の関数について説明しています。

その他の情報

Sybase Getting Started CD、SyBooks™ CD、Sybase Product Manuals Web サイトを利用すると、製品について詳しく知ることができます。

- Getting Started CD には、PDF 形式のリリース・ノートとインストール・ガイド、SyBooks CD に含まれていないその他のマニュアルや更新情報が収録されています。この CD は製品のソフトウェアに同梱されています。Getting Started CD に収録されているマニュアルを参照または印刷するには、Adobe Acrobat Reader が必要です (CD 内のリンクを使用して Adobe の Web サイトから無料でダウンロードできます)。
- SyBooks CD には製品マニュアルが収録されています。この CD は製品のソフトウェアに同梱されています。Eclipse ベースの SyBooks ブラウザを使用すれば、使いやすい HTML 形式のマニュアルにアクセスできます。

  一部のマニュアルは PDF 形式で提供されています。これらのマニュアルは SyBooks CD の PDF ディレクトリに収録されています。PDF ファイルを開いたり印刷したりするには、Adobe Acrobat Reader が必要です。

  SyBooks をインストールして起動するまでの手順については、Getting Started CD の『SyBooks インストール・ガイド』、または SyBooks CD の *README.txt* ファイルを参照してください。
- Sybase Product Manuals Web サイトは、SyBooks CD のオンライン版であり、標準の Web ブラウザを使用してアクセスできます。また、製品マニュアルのほか、EBFs/Updates、Technical Documents、Case Management、Solved Cases、ニュース・グループ、Sybase Developer Network へのリンクもあります。

  Technical Library Product Manuals Web サイトにアクセスするには、Product Manuals (http://www.sybase.com/support/manuals/) にアクセスしてください。

Web 上の Sybase 製品の動作確認情報

Sybase Web サイトの技術的な資料は頻繁に更新されます。

❖ **製品認定の最新情報にアクセスする**

1. Web ブラウザで Technical Documents を指定します。(http://www.sybase.com/support/techdocs/)
2. [Certification Report] をクリックします。
3. [Certification Report] フィルタで製品、プラットフォーム、時間枠を指定して、[Go] をクリックします。
4. [Certification Report] のタイトルをクリックして、レポートを表示します。

❖ コンポーネント認定の最新情報にアクセスする

1 Web ブラウザで Availability and Certification Reports を指定します。(http://certification.sybase.com/)

2 [Search By Base Product] で製品ファミリとベース製品を選択するか、[Search by Platform] でプラットフォームとベース製品を選択します。

3 [Search] をクリックして、入手状況と認定レポートを表示します。

❖ Sybase Web サイト (サポート・ページを含む) の自分専用のビューを作成する

MySybase プロファイルを設定します。MySybase は無料サービスです。このサービスを使用すると、Sybase Web ページの表示方法を自分専用にカスタマイズできます。

1 Web ブラウザで Technical Documents を指定します。(http://www.sybase.com/support/techdocs/)

2 [MySybase] をクリックし、MySybase プロファイルを作成します。

Sybase EBF とソフトウェア・メンテナンス

❖ EBF とソフトウェア・メンテナンスの最新情報にアクセスする

1 Web ブラウザで Sybase Support Page を指定します。(http://www.sybase.com/support)

2 [EBFs/Maintenance] を選択します。MySybase のユーザ名とパスワードを入力します。

3 製品を選択します。

4 時間枠を指定して [Go] をクリックします。EBF/Maintenance リリースの一覧が表示されます。

鍵のアイコンは、「Technical Support Contact」として登録されていないため、一部の EBF/Maintenance リリースをダウンロードする権限がないことを示しています。未登録でも、Sybase 担当者またはサポート・コンタクトから有効な情報を得ている場合は、[Edit Roles] をクリックして、「Technical Support Contact」の役割を MySybase プロファイルに追加します。

5 EBF/Maintenance レポートを表示するには [Info] アイコンをクリックします。ソフトウェアをダウンロードするには製品の説明をクリックします。

表記規則

次の項では、このマニュアルで使用されている表記について説明します。

SQL は自由な形式の言語で、1 行内のワード数や、改行の仕方に規則はありません。このマニュアルでは、読みやすくするため、例や構文を文の句ごとに改行しています。複数の部分からなり、2 行以上にわたる場合は、字下げしています。複雑なコマンドの書式には、修正された BNF (Backus Naur Form) 記法が使用されています。

表 1 に構文の規則を示します。

表 *1:* このマニュアルでのフォントと構文規則

要素	例
コマンド名、プロシージャ名、ユーティリティ名、その他のキーワードは sans serif フォントで表記する。	select sp_configure
データベース名とデータ型は sans serif フォントで表記する。	master データベース
ファイル名、変数、パス名は斜体で表記する。	システム管理ガイド *sql.ini* ファイル *column_name* *$SYBASE/ASE* ディレクトリ

変数 (ユーザが入力する値を表す語) がクエリまたは文の一部である場合は Courier フォントの斜体で表記する。	`select column_name`
	`from table_name`
	`where search_conditions`

カッコはコマンドの一部として入力する。	`compute row_aggregate (column_name)`
2 つのコロンと等号は、構文が BNF 表記で記述されていることを示す。この記号は入力しない。「～と定義されている」ことを意味する。	`::=`
中カッコで囲まれたオプションの中から必ず 1 つ以上を選択する。コマンドには中カッコは入力しない。	`{cash, check, credit}`
角カッコは、オプションを選択しても省略してもよいことを意味する。コマンドには角カッコは入力しない。	`[cash | check | credit]`
中カッコまたは角カッコの中のカンマで区切られたオプションをいくつでも選択できることを意味する。複数のオプションを選択する場合には、オプションをカンマで区切る。	`cash, check, credit`
パイプまたは縦線は複数のオプションのうち 1 つだけを選択できることを意味する。	`cash | check | credit`
省略記号 (...) は、直前の要素を必要な回数だけ繰り返し指定できることを意味します。	`buy thing = price [cash | check | credit] [, thing = price [cash | check | credit]]...` この例では、製品 (thing) を少なくとも 1 つ購入 (buy) し、価格 (price) を指定する必要があります。支払方法を選択できる。角カッコで囲まれた項目の 1 つを選択する。追加品目を、必要な数だけ購入することもできる。各 buy に対して、購入した製品 (thing)、価格 (price)、オプションで支払方法 (cash、check、credit のいずれか) を指定します。

- 次は、オプション句のあるコマンドの構文の例です。

  sp_dropdevice [*device_name*]

  複数のオプションを持つコマンドの例を示します。

  select *column_name*
  from *table_name*
  where *search_conditions*

  構文では、キーワード (コマンド) は通常のフォントで表記し、識別子は小文字で表記します。ユーザが提供するワードは斜体で表記します。

- Transact-SQL コマンドの使用例は次のように表記します。

```
select * from publishers
```

- 次は、コンピュータからの出力例です。

```
pub_id    pub_name                city          state
------    --------------------    -----------   -----
0736      New Age Books           Boston        MA
0877      Binnet & Hardley        Washington    DC
1389      Algodata Infosystems    Berkeley      CA

(3 rows affected)
```

このマニュアルでは、例に使用する文字はほとんどが小文字ですが、Transact-SQL のキーワードを入力するときは、大文字と小文字は区別されません。たとえば、SELECT、Select、select はすべて同じです。

テーブル名などのデータベース・オブジェクトの大文字と小文字を Adaptive Server が区別するかどうかは、Adaptive Server にインストールされたソート順によって決まります。シングルバイト文字セットを使用している場合は、Adaptive Server のソート順を再設定することによって、大文字と小文字の区別の取り扱い方を変更できます。詳細については、『システム管理ガイド』を参照してください。

アクセシビリティ機能

このマニュアルには、アクセシビリティを重視した HTML 版もあります。この HTML 版マニュアルは、スクリーン・リーダーで読み上げる、または画面を拡大表示するなどの方法により、その内容を理解できるよう配慮されています。

Adaptive Server HTML マニュアルは、連邦リハビリテーション法第 508 条のアクセシビリティ規定に準拠していることがテストにより確認されています。第 508 条に準拠しているマニュアルは通常、World Wide Web Consortium (W3C) の Web サイト用ガイドラインなど、米国以外のアクセシビリティ・ガイドラインにも準拠しています。

注意 アクセシビリティ・ツールを効率的に使用するには、設定が必要な場合もあります。一部のスクリーン・リーダーは、テキストの大文字と小文字を区別して発音します。たとえば、すべて大文字のテキスト (ALL UPPERCASE TEXT など) はイニシャルで発音し、大文字と小文字の混在したテキスト (Mixed Case Text など) は単語として発音します。構文規則を発音するようにツールを設定すると便利かもしれません。詳細については、ツールのマニュアルを参照してください。

Sybase のアクセシビリティに対する取り組みについては、Sybase Accessibility (http://www.sybase.com/accessibility) を参照してください。Sybase Accessibility サイトには、第 508 条と W3C 標準に関する情報へのリンクもあります。

不明な点があるときは

Sybase ソフトウェアがインストールされているサイトには、Sybase 製品の保守契約を結んでいるサポート・センタとの連絡担当の方 (コンタクト・パーソン) を決めてあります。マニュアルだけでは解決できない問題があった場合には、担当の方を通して Sybase のサポート・センタまでご連絡ください。

第 1 章

マイグレーション方法

Sybase Adaptive Server Enterprise には、高度なアルゴリズムを使用して問い合わせの対象となるテーブルの統計を分析し、パフォーマンスを向上させる洗練されたクエリ・オプティマイザが用意されています。Adaptive Server 15.0.3 ESD #1 以降には、Adaptive Server 15.0 の高度なクエリ・オプティマイザの有効活用をサポートするツールが組み込まれています。

このマニュアルでは、Adaptive Server 12.5 から Adaptive Server 15 にアップグレードし、2 つのバージョン間のパフォーマンスの違いを分析して、Adaptive Server 15.0 インストールをチューニングする際の最適化目標と推奨事項について説明します。

また、最適なクエリ・プラン、最適化目標、その他の構成設定を特定し、適用できる QPTune というツールについても説明します。

アップグレード前の考慮事項

この項では最適化目標と基準、および Adaptive Server 15.0 にアップグレードする前に完了する必要のある手順について説明します。運用サーバをアップグレードした後のパフォーマンスの評価に役立つ、アップグレード前のテストの概要も記載されています。

最適化目標について

Adaptive Server 15.0 のクエリ処理エンジンの中心的概念は、最適化するクエリの性質を示す「最適化目標」です。Adaptive Server クエリ・オプティマイザは、最適化目標に基づいて、クエリを最適化する最善の方法を判断します。

たとえば、通常のオンライン・トランザクション処理 (OLTP: online transaction processing) クエリと通常の意思決定支援システム (DSS: decision-support system) クエリでは、クエリで使用されるデータのアクセス・パターンが異なるため、まったく異なるクエリ・プランが作成されます。OLTP クエリでは通常、影響するのは 1 つまたは少数のローだけで、インデックスが適切に指定されているテーブルだけがジョインされます。一方、DSS クエリでは通常、多くのローに影響し、少数のローが返され、多くのテーブルがジョインされます。

アクセス・パターンが異なるため、OLTP クエリでは従来の「ネストループ・ジョイン」を使用すると実行効率が最も高く、DSS クエリでは「ハッシュ・ジョイン」を使用すると実行速度が速くなります。クエリの用途が OLTP か DSS かを指定すれば、オプティマイザはこの情報を使用して、時間、メモリ、および CPU 使用率の節約につながるクエリ・プランを生成します。

Adaptive Server 15.0 には 3 つの最適化目標があります。オプティマイザで検証可能なオプションと方法の数が少ない順に次に示します。

- allrows_oltp – OLTP クエリに適しています。allrows_oltp を使用すると選択できるジョイン方法が最も少なく、クエリ・オプティマイザはネストループ・ジョインのみを検証します。
- allrows_mix – Adaptive Server 15.0 にアップグレードした後のデフォルトの設定です。allrows_mix を使用すると、マージ・ジョインとともに並列プランもオプティマイザで検証できます (Adaptive Server が並列処理の可能な構成になっている場合)。
- allrows_dss – DSS クエリに適しています。allrows_dss を使用すると選択できるジョイン方法が最も多く、オプティマイザはハッシュ・ジョイン、ネストループ・ジョイン、マージ・ジョイン、および並列プランを検証します。

allrows_mix と allrows_dss を使用する場合、SQL オペレーションでその他の低レベル処理アルゴリズムも有効になります。これらのアルゴリズムは allrows_oltp を使用すると無効になります。

選択肢の多い最適化目標を指定すると、クエリ・オプティマイザがクエリ・プランの生成時に使用するリソース (時間およびプロシージャ・キャッシュ) が大幅に増加します。ネストループ・ジョインだけを使用して allrows_dss と allrows_oltp で同じクエリ・プランを生成すると、allrows_oltp より allrows_dss で最適化を行った方が多くの時間とプロシージャ・キャッシュを必要とします。

どの最適化目標を選択するかによって、クエリのパフォーマンスに大きな影響を与える可能性があります。特定のアプリケーションの負荷特性がシステムの他の部分と異なることがわかっている場合、そのアプリケーションに適したセッションレベルの最適化目標を設定できます。QPTune ユーティリティを使用するか、さまざまな最適化目標を手動で試し、一連のアプリケーションとクエリの全体的なパフォーマンスがよくなる最適化目標を選択してください。詳細については、「第 2 章 QPTune」を参照してください。

サーバレベル、セッションレベル、または個々のクエリ・レベルで最適化目標を設定できます。

- サーバワイドなデフォルト :

```
sp_configure 'optimization goal', 0, 'allrows_dss'
```

- セッションレベルの設定 (サーバワイド設定を上書き) :

```
set plan optgoal allrows_dss
```

- クエリレベルの設定 (サーバワイド設定とセッションレベル設定を上書き):

```
select * from T1, T2 where T1.a = T2.b
plan '(use optgoal allrows_dss)'
```

注意 login trigger を使用してセッションレベルの最適化目標を設定することもできます。

最適化基準

最適化目標は、「最適化基準」と呼ばれる一連のプロパティの “on/off” 設定を集めたものです。最適化基準によって、アクセス・メソッド、ジョイン、グループ化、ソートなどについてオプティマイザが特定のアルゴリズムを検証可能かどうかが決定されます。

たとえば、ハッシュ・ジョインを有効にするには、次の最適化基準を使用します。

```
set hash_join on
```

また、“store_index” アルゴリズム (再フォーマット) を無効にするには、次の最適化基準を使用します。

```
set store_index off
```

セマンティック上の理由から、オプティマイザが特定の基準または目標を無視することがあります。たとえば、すべての join 演算子を無効にすると、新しいオプティマイザは「ネストループ」を自動的に有効にします。

注意 Sybase では Sybase 製品の保守契約を結んでいるサポート・センタからの指示がないかぎり、最適化基準を明示的に設定するのではなく、最適化目標を使用することをおすすめします。

Adaptive Server 15.0 での並列クエリ処理

Adaptive Server バージョン 11.5 以降、複数のワーカー・プロセスで 1 つのクエリを処理する、クエリでの並列処理がサポートされています。並列処理を使用すると、多数のローにアクセスして小さい結果セットを返す DSS タイプのクエリの応答時間を改善できます。

Adaptive Server 15.0 のクエリ処理機能では DSS タイプのクエリでパフォーマンスが向上する可能性がありますが、Adaptive Server 15.0 にアップグレードした場合、最初は並列処理を使用しないことをおすすめします。

逐次処理は並列処理に比べてリソース効率が高いですが、並列処理を使用すると同じハードウェアで全体的なパフォーマンスを改善できます。また、逐次モードで Adaptive Server 15.0 を使用すると、以前のバージョンの Adaptive Server で並列処理を使用した場合よりもクエリの実行速度が速くなります。

ただし、Adaptive Server 15.0 でセマンティック・テーブル分割を使用するクエリや、create index などの DDL コマンドでは、並列処理を使用すると逐次処理より応答時間が短縮する場合があります。

Adaptive Server 15.0 のリソース推奨事項

Adaptive Server 15.0 にはバージョン 12.5 よりも多くのプロシージャ・キャッシュが必要です。このメモリ要件の増加は、クエリ実行だけでなく、最適化にも適用されます。プロシージャ・キャッシュのサイズを Adaptive Server 12.5 の場合の 2 ～ 6 倍にすることをおすすめします。

また、場合によっては、Adaptive Server 15.0 でクエリ処理を行うために tempdb の領域を大きくする必要があります。

Adaptive Server 15.0 での統計値の組み込み

Adaptive Server ではコストベースのクエリ・オプティマイザを使用して、特定のクエリに最適なプランを選択できます。オプティマイザは、クエリ内で参照されるテーブル、インデックス、パーティション、およびカラムに関する統計値に基づいて、さまざまなプランのコストを見積もります。I/O と CPU 時間についてコストが計算されます。オプティマイザは、最小コストのクエリ・プラン方法を選択します。統計値が不正確な場合、コストの見積もりが正しく算出されないため、最適でないプランを選ぶ結果になったりパフォーマンス低下につながることがあります。

統計値の中には、テーブル中のページ数やロー数 (systabstats に格納) のように、クエリ処理中に自動更新されるものがあります。その他の統計値は、update statistics の実行時とインデックスの作成時にだけ更新されます。その例として、sysstatistics に格納されているカラムのヒストグラムと密度情報があります。

Adaptive Server 15.0 では、ソート、グループ化、共用体、ジョイン、その他のオペレーションなどに複数のアルゴリズムを使用しているため、以前のバージョンの Adaptive Server に比べて不正確な統計データの影響を受けやすくなっています。また、Adaptive Server 15.0 は Adaptive Server 12.x と比べてさまざまな用途に統計値を使用します。たとえば、Adaptive Server 15.0 は統計値を使用して、複数テーブルのクエリのジョイン順を特定します。

where 句をジョイン述部に使用する場合と探索引数に使用する場合、where 句で参照するすべてのカラムのヒストグラムを最新の状態に保つことをおすすめします。重要な統計値や欠落した統計値を特定するには、QPTune の統計アドバイザを使用します。

アップグレード前の推奨テスト

運用システムを Adaptive Server 15.0 にアップグレードする前に、バージョン 15 より前の Adaptive Server を使用した現在の運用環境でのアプリケーションのパフォーマンス特性について詳細情報を集めます。集めた詳細データは、パフォーマンス分析の統計の基礎となります。

Adaptive Server 12.x と 15.0 のパフォーマンスを比較するため、次のテストを実行します。

- できるだけ多くのアプリケーション関数をテストし、重要な関数は入念にテストします。関数ごとに応答時間またはスループットを測定します。可能であれば、アプリケーションで実行したクエリごとにこれらの測定を行ってください。
- 現在の Adaptive Server 12.x 運用システムのパフォーマンスを測定します。
- 詳細に設定した Adaptive Server 15.0 の「テスト」システムで、Adaptive Server 12.x 運用データベースの完全コピーと現実的な負荷を使用して、同じ関数およびパフォーマンス測定をテストします。Adaptive Server 12.x と同じクエリを同じレベルの同時ユーザのアクティビティとして実行します。現在の Adaptive Server 12.x 運用環境の「パフォーマンス到達範囲」を取得することで、Adaptive Server 15.0 と比較するときの適切なベースラインが得られます。取得する測定値には、論理 I/O オペレーションの数、経過時間、コンパイル時間、CPU 使用率、showplan 出力などを含める必要があります。Adaptive Server 12.x と Adaptive Server 15.0 の間のパフォーマンスを的確に比較するには、次の 2 つのレベルのパフォーマンス・データを収集します。
  - 個々のクエリを個別に実行した状態と、複数のユーザが実行した全体的な負荷
  - サーバワイドのリソース使用量の観点から見た、全体としての Adaptive Server

Adaptive Server 12.x と 15 のパフォーマンス数値に影響を与える重要な要因がいくつかあります。紛らわしいパフォーマンス数値が算出されるのを防ぐため、次の点に注意してください。

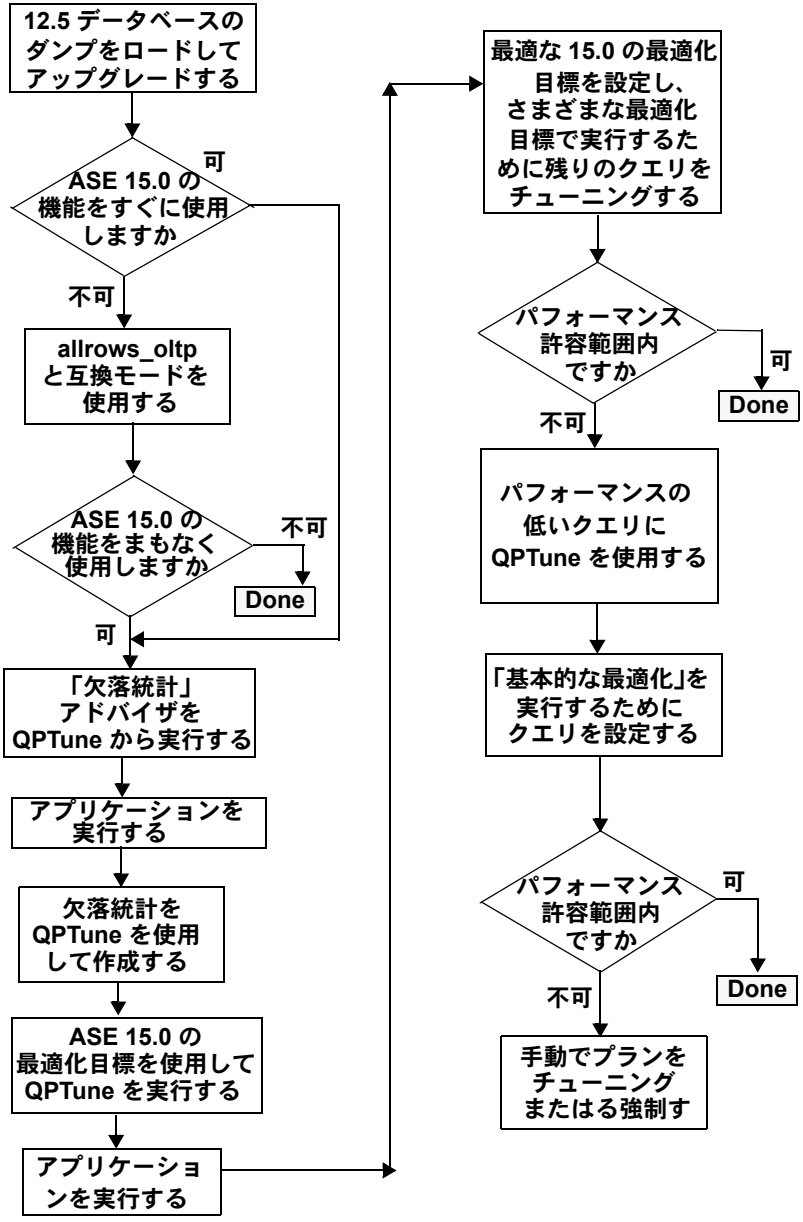
- Adaptive Server 12.x と Adaptive Server 15.0 の両方のテストで同じ方法でキャッシュを「ウォーム・アップ」“Warm up” してください。
- 同一のキャッシュおよびバッファ・プールの設定を使用してください。
- Adaptive Server 15.0 でのプロシージャ・キャッシュを Adaptive Server 12.x で使用するサイズの 2 ～ 6 倍にしてください。
- 特にログ・デバイスと tempdb で、同様のデータ・デバイスのレイアウトおよび配置を使用してください。
- 各テストの実行後、特にテスト時にデータを変更した場合に、元のデータベースを簡単にリストアできるようにテスト・システムを設定してください。

注意 場合によっては、Adaptive Server 15.0 ではデータ・キャッシュのサイズを大きくする必要があります。

Adaptive Server 15.0 機能へのマイグレーション

Adaptive Server 15.0 の機能は、アップグレード直後からすぐに使用することも、新機能を後で使用することもできます。次のフローチャートに、Adaptive Server のマイグレーションに使用可能ないくつかの方法を示します。

図 *1-1: Adaptive Server* マイグレーション方法のフローチャート



アップグレードして新機能をすぐに使用する場合

Adaptive Server 15.0 にアップグレードして新機能をすぐに使用するには、互換モードの設定を省略し、QPTune を使用してアプリケーションをチューニングすることをおすすめします。

1 Adaptive Server 12.5 およびデータベースを、QPTune を使用したマイグレーションがサポートされている Adaptive Server 15.0.3 ESD #1 以降にアップグレードします。

2 QPTune を使用して、統計アドバイザを有効にします。

3 アプリケーションのクエリを実行します。QPTune によって重要な統計値が示され、それらの統計値が作成されます。通常、ほとんどのクエリのチューニングがこの時点で行われます。

4 Adaptive Server 15.0 の最適化目標を使用して QPTune を実行します。

5 さまざまな最適化目標でクエリを実行し、最適なパフォーマンスが得られる最適化目標を選択します。

6 アプリケーションのクエリを再度実行します。

7 チューニングが必要なクエリがないかどうかを確認します。

8 さらにチューニングが必要なクエリに対して QPTune を実行します。

9 残りのクエリを手動でチューニングします。従来の方法でクエリ・プランを分析し、必要なパフォーマンスが得られるようにクエリ・プランを書き直すか、または抽象クエリ・プランなどの方法を使用します。

アップグレードして新機能を後で使用する場合

Adaptive Server 15.0 にアップグレードして、Adaptive Server 15.0 の機能の使用を段階的に開始する場合は、QPTune を使用したマイグレーションがサポートされている Adaptive Server 15.0.3 ESD #1 以降に Adaptive Server 12.5 とデータベースをアップグレードします。最適化目標に allrows_oltp を使用し、アップグレード用に互換モードを有効にします。

Adaptive Server 15.0 の機能を使用する準備ができたら、次の手順に従います。

1 QPTune を使用して、統計アドバイザを有効にします。

2 アプリケーションのクエリを実行します。QPTune によって重要な統計値が示され、それらの統計値が作成されます。通常、ほとんどのクエリのチューニングがこの時点で行われます。

3 Adaptive Server 15.0 の最適化目標を使用して QPTune を実行します。

4 さまざまな最適化目標でクエリを実行し、最適なパフォーマンスが得られる最適化目標を選択します。

5 アプリケーションのクエリを再度実行します。

6　チューニングが必要なクエリがないかどうかを確認します。

7　さらにチューニングが必要なクエリに対して QPTune を実行します。

8　抽象クエリ・プランを使用して手動で残りのクエリをチューニングします。

注意　同じ方法でストアド・プロシージャの段階的マイグレーションを行うことができます。

アップグレードして新機能は使用しない場合

Adaptive Server 12.5 からアップグレードしても Adaptive Server 15.0 の機能を使用する予定がない場合は、allrows_oltp を最適化目標に使用し、互換モードを有効にします。

互換モードの詳細については、「第 3 章 互換モードでのクエリ・プロセッサの実行」を参照してください。

トラブルシューティング

この項では、最適化の問題に対処するためのクエリ処理のパフォーマンスと方法について説明します。

クエリ処理のヒント

Adaptive Server 15.0 では、クエリ処理環境が大幅に改善されましたが、クエリ・プランまたはクエリのパフォーマンスが予想と異なる場合は、問題を特定できるいくつかの方法があります。

- さまざまな最適化目標を使用している場合は、キャッシュされたプランが使用されていないことを確認してください。セッションレベルまたはサーバワイドで最適化目標を変更しても、キャッシュされたプランの再コンパイルは行われません。ストアド・プロシージャの場合、再コンパイルして実行するか、またはアクセス中のテーブルの 1 つで sp_recompile を実行します。バッチの場合は、最初に set statement_cache off を実行して、ステートメント・キャッシュが無効になっていることを確認します。
- サーバワイド設定であるか、またはセッションレベル設定であるかにかかわらず、常に特定の最適化目標でストアド・プロシージャを最適化するには、set plan optgoal allrows_xxx をストアド・プロシージャの最初の文で使用します。これは、Adaptive Server 15.0.2 ESD #2 以降でのみ機能します。

- Adaptive Server 12.x の SQL コードに、明示的に強制されたジョイン順 (set forceplan を使用) が含まれている場合は、Adaptive Server 15.0 にアップグレードする前にジョイン順を再調査します。そのような構造では、Adaptive Server 15.0 の機能の利点を十分に活用できないことがあります。

  Adaptive Server 15.0.1 ESD #2 以降では、2 つのトレース・フラグを有効にできます。

  - トレース・フラグ 15307 は、クエリ・プランのコンパイル時に、set forceplan 文の影響を無効にします。
  - トレース・フラグ 15308 は、インデックス、プリフェッチ、並列処理、バッファ置換などの方法の明示的な強制を無効にします。

  サーバ起動時にこれらのトレース・フラグ (15307 と 15308) を設定することも、dbcc traceon を使用して動的に有効にすることもできます。これらの影響はサーバワイドで、抽象クエリ・プランで定義されたクエリ・プランのプロパティに影響することはありません。

- システムで消費される tempdb の領域が多すぎる場合は、モニタリングと診断のアクセス・テーブルを使用して、特定のセッションがワークテーブルの領域を大量に消費していないか確認します。モニタリング・テーブルを有効にし、次のクエリを実行します。

  ```
  select SPID, DBName, ObjectName, PartitionSize
  from master..monProcessObject
  where DBID = tempdb_id(SPID)
  order by SPID
  ```

  PartitionSize の値が大きいセッションを探します。ワークテーブルには “temp worktable” の ObjectName があります。master データベースで monProcessSQLText または monProcessStatement に対してクエリを発行し、対応する SQL 文を特定します。

  tempdb を満杯にして tempdb 領域を必要とする他のセッションに影響するセッションを停止するには、“tempdb_space” タイプのリソース制限を作成します。また、複数のテンポラリ・データベースを作成し、特定のユーザに割り当てることもできます。1 つのセッションで使用されている tempdb 領域を確認するには、次のコマンドを使用します。

  ```
  select pssinfo(spid|0,'tempdb_pages')
  ```

- Adaptive Serer 15.0.1 以降で、同一、または類似したクライアント生成の SQL クエリを大量に実行する場合は、ステートメント・キャッシュ設定とリテラルの自動パラメータ化設定を有効にします。これにはストアド・プロシージャや execute-immediate クエリ形式は含まれず、クエリの違いはその探索パラメータのみです。クエリの最適化に要する時間とリソースを大幅に削減できるため、全体的なパフォーマンスが向上します。

ステートメント・キャッシュを有効にするとクエリのプランがキャッシュされるため、同一のクエリをコンパイルする必要がなくなり、時間とリソースを節約できます。ステートメント・キャッシュは、statement cache size 設定パラメータを使用するとサーバワイドで有効になります。セッション・レベルでは、set statement_cache off を使用してステートメント・キャッシュを無効にします。

リテラルの自動パラメータ化は、enable literal autoparam 設定パラメータを使用するとサーバワイドで有効になり、セッション・レベルでは set literal_autoparam on で有効になります。enable literal autoparam は、ステートメント・キャッシュが有効な場合にのみ適用されます。リテラルの自動パラメータ化を有効にすると、キャッシュは定数値のみ異なるほぼ同一のクエリにも拡張されます。たとえば、次の 2 つのクエリは同一と見なされません。

```
select CustName from Customers where CustID = 123
select CustName from Customers where CustID = 456
```

ただし、同じクエリ・プランを生成する可能性があります。リテラルの自動パラメータ化を有効にすると、ステートメント・キャッシュで where 句の定数値が除外され、次のようなすべてのクエリのプランがキャッシュされます。

```
select CustName from Customers where CustID= <integer-
constant>
```

Adaptive Server 15.0 で廃止された最適化コマンド

12.x のさまざまな最適化関連設定が、Adaptive Server 15.0 では使用されなくなりました。次のコマンドは Adaptive Server 15.0 にも存在しますが、互換モードにのみ関連し、Adaptive Server 15.0 の最適化プロセスには影響しません。

- set sort_merge – set merge_join、最適化目標、および設定パラメータ enable merge join で置き換えられました。
- set jtc – ジョイン推移閉包は Adaptive Server 15.0 では常に有効です。
- set table count – この設定は Adaptive Server 15.0 では使用されません。
- enable sort-merge join および JTC – この設定パラメータは最適化目標および設定パラメータ enable merge join で置き換えられました。
- トレース・フラグ 334 と 384 の起動 – これらのフラグで有効にするマージ・ジョインおよび JTC は使用されなくなりました。

これらの機能の参照をアプリケーションから削除することをおすすめします。

サポート・センタに問い合わせる前に用意する情報

サポート・センタに問い合わせる前に、特に問題を再現できる場合は、できるだけ多くの診断統計情報を収集してください。

701 エラー

通常のクエリ (update index statistics を除く) で 701 エラーが発生した場合、Adaptive Server でプロシージャ・キャッシュ領域が不足していることを示します。デフォルト・サイズのプロシージャ・キャッシュで実行している場合は、プロシージャ・キャッシュを増やして再度実行してください。バージョン 15.0 以降の一般的なガイドラインとして、プロシージャ・キャッシュのサイズを 12.5.x の場合の 2 ～ 6 倍にします。一部の状況、特に最適化目標として allrows_dss を使用している場合は、プロシージャ・キャッシュをさらに大きくする必要があります。

プロシージャ・キャッシュを大きくしても 701 エラーが解決せず、問題を特定できない場合は、プロシージャ・キャッシュ・ページを含む設定可能な共有メモリ・ダンプを設定します。

```
sp_configure 'dump on conditions', 1
go

sp_shmdumpconfig 'add', 'error', 701, 1,
'my_dump_directory',null,include_proc
go
```

sp_shmdumpconfig は、エラー 701 条件を追加して、メモリ・ダンプを開始します。4 番目のパラメータ (上の例では 1) は、取得するメモリ・ダンプの数を示します。Adaptive Server を再起動した場合、または手動でカウンタを再設定した場合を除き、Adaptive Server がこの条件で追加のメモリ・ダンプを取得することはありません。

my_dump_directory パラメータは、メモリ・ダンプを保持するディレクトリの名前です。ディレクトリが置かれているファイル・システムには、メモリ・ダンプ・ファイルが大きい場合でも保持できる十分な空き領域があります。パラメータなしで sp_shmdumpconfig を実行し、現在定義されているダンプ条件を確認します。これにより、取得するメモリ・ダンプの推定サイズも表示されます。

include_proc パラメータを使用すると、設定可能な共有メモリ・ダンプにプロシージャ・キャッシュの情報を含めることができます。

メモリ・ダンプの日時を含むファイル名が自動生成されます。メモリ・ダンプが取得されたら、次のコマンドでシステムを再設定します。

```
sp_shmdumpconfig 'drop', 'error', 701
go
```

デフォルトでは Adaptive Server はクライアントに 701 エラー・メッセージを送信します。次のコマンドを実行して、このメッセージをエラー・ログに報告することもできます。

```
sp_altermessage 701,'with_log',true
```

設定可能なすべての共有メモリ・ダンプを停止するには、dump on conditions を 0 に設定します。メモリ・ダンプを取得したら、サポート・センタに事例を報告し、メモリ・ダンプを FTP サイトにアップロードします。以下の SQL 文の出力を含めてください。この文は Adaptive Server 内のモニタリング・テーブルを使用します。

```
select * from master..monProcedureCacheMemoryUsage
select * from master..monProcedureCacheModuleUsage
go
```

モニタリング・テーブルは、Adaptive Server 15.0.2 以降で installmaster スクリプトを実行すると、自動的に設定されます。以前のバージョンの Adaptive Server のインストール・プロセスでは installmontables スクリプトが実行されます。モニタリング・テーブルの設定の詳細については、Adaptive Server 15.0 のマニュアルを参照してください。

限られたクエリで発生しているパフォーマンスの問題

最適でないクエリ・プランやリソース消費が原因で、限られたクエリが適切に実行されない場合は、開発用サーバに最新の Adaptive Server 15.0.x バージョンをインストールします。問題が解決しない場合は、問題の再現方法や診断情報をサポート・センタにお送りください。診断情報を収集するには、次の手順に従います。

1　次のコマンドを含むスクリプト・ファイル *sql.txt* を作成します。

```
select @@version
go
select @@optgoal
go
sp_cacheconfig
go
sp_configure 'nondefault' (only if you're running Adaptive
Server 15.0.2 or later)
go
dbcc traceon(3604)
set showplan on
set statistics time, io, plancost on
set option show long
go
<your query text>
go
```

注意　set option show long を複雑なクエリに使用すると、多くの出力が生成される場合があります。

2 isql を使用して *sql.txt* を実行し、ファイルに出力を取得します。

```
isql -Usa -P yourpassword -S YOUR_SERVER_NAME
-i sql.txt -o sql.out
```

isql の -w オプションを使用して出力をフォーマットします。

3 次の情報をサポート・センタに送信します。

- *sql.txt* ファイルと *sql.out* ファイル。可能であれば、“fast” (*sql.fast.txt*) クエリ・プランと “slow” (*sql.slow.txt*) クエリ・プラン、および対応する出力ファイル *sql.fast.out* と *sql.slow.out.* を含めます。
- ベース・テーブルとインデックスの DDL。ddlgen ユーティリティを使用して生成できます。
- optdiag を使用して、ベース・テーブルの統計出力をシミュレートします :

```
optdiag statistics simulate <table-name>
-Usa -P yourpassword -S YOUR_SERVER_NAME
-o <output-file>
```

- Adaptive Server の設定ファイルのコピー。Adaptive Server15.0.2 の場合は sp_configure 'nondefault' の出力も含めてください。
- クエリでビューまたはストアド・プロシージャが使用されている場合は、defncopy または ddlgen を使用して取得した SQL ソース・コードも含めます。
- sp_monitorconfig ‘all' と sp_helpsort の出力。

システムワイドのパフォーマンスの問題

サーバ・レベルで Adaptive Server のパフォーマンスが低く、15.0.2 ESD #3 以降を実行している場合は、Adaptive Server を停止し、RUN_server ファイルでトレース・フラグ 757 を設定して再起動してください。この方法は、マルチエンジン Adaptive Server を実行中に、明確な原因もなく異常に高い CPU 使用率が報告された場合にも効果的です。

プロシージャ・キャッシュがキャッシュされたアイドル状態のプランで満杯になった場合、CPU 使用率が高くなければ、代わりに次の dbcc コマンドを実行します。ただし、これらのコマンドを使用した場合、サーバを再起動するほどの効果は期待できません。

```
dbcc traceon(757)
go
dbcc proc_cache(free_unused)
go
```

注意 Adaptive Server の 15.0.2 ESD #3 より前のバージョンではトレース・フラグ 757 を使用しないでください。

サポート・センタへの診断情報のアップロード

診断ファイルを作成したら、サポート・センタに事例を報告します。Sybase FTP サイトに診断情報をアップロードする際は、最新のアップロード手順についてサポート・センタに問い合わせてください。

第 2 章

QPTune

QPTune は Java および XML で書かれた Adaptive Server ユーティリティです。QPTune を使用すると、最適なクエリ・プラン、最適化目標、その他の構成設定を特定し、それらをサーバまたはクエリのレベルで適用できます。これにより、その後のクエリの実行で最適なパフォーマンスが得られます。アプリケーションのクエリに最適な設定が見つかったら、設定をエクスポートして運用サーバに適用できます。

QPTune を使用すると、次の処理を実行できます。

- アプリケーションの欠落統計を修正する
- アプリケーションをチューニングして、複数のクエリに最適なオプティマイザ設定を見つける
- ユーザ定義のルールを使用し、カスタマイズされた設定や標準設定を特定のクエリに選択的に適用する

QPTune を使用すると、複数の構成設定や Adaptive Server インストールを分析および比較してパフォーマンスの影響分析レポートを生成したり、Adaptive Server のパフォーマンスを低下させることなくプラン修正を実行したりできます。Adaptive Server は単純な select 文を使用して測定基準を収集し、ストアド・プロシージャまたはステートメント・キャッシュに格納します。また、システムの全体的なパフォーマンスにほとんど影響しない DDL 文を使用してクエリ・プランを修正します。さらに QPTune を使用するとさまざまなスレッショルド・レベルでのモニタリングが可能なため、収集する測定基準を減らすことができます。

システムの設定

QPTune を起動する前に、次の環境変数を設定します。

- SYBASE_JRE6 および JAVA_HOME － Java Runtime インストールに設定する
- SYBASE － マシン上の最新の Sybase インストールに設定する
- SYBASE_ASE － マシン上のインストールの Adaptive Server コンポーネント (ディレクトリ) に設定する

QPTune 実行プログラム名は、UNIX の場合は *QPTune*、Windows の場合は *QPTune.bat* で、次の場所にあります。

$SYBASE/$SYBASE_ASE/qptune (UNIX)
%SYBASE%¥%SYBASE_ASE%¥qptune (Windows)

QPTune での構文および参照情報については、「QPTune リファレンス情報」(48 ページ) を参照してください。

環境およびインストールを確認し、基本構文に関する情報を表示するには、QPTune を -h オプションを指定して実行します。

```
QPTune -h
```

Windows 環境での出力例

```
QPTune <Version 3.0> Windows/Unix Built: Fri Jan 21 14:00:15 PDT 2009
Syntax:
QPTune [-U <username>] [-P <password>] [-S <hostname:port/database>]
[-A <action
[start|collect(_full)|compare|fix|(start|collect|fix|undo_fix)_stats]>]
[-M <mode>] [-T <appTime>] [-i <inputFile>] [-o <outputFile>]
[-f <fileList(,)>] [-c <configFile>] [-l <limit>] [-e <evalField>]
[-d <diff%(,diff_abs)>] [-m <missingCount>] [-n <login>] [-J <charset>]
[-N (noexec)] [-g (applyOptgoal)][-v (verbose)] [-s (sort)] [-h (help)]
Example:
QPTune -U sa -P -S WUXP:5000/scenario -A collect -M allrows_mix -T 0
-o metrics.xml -c config.xml -e elap_avg -d 5,5 -l 5 -i metrics.xml
-f a1.xml,a2.xml,a3.xml -v -s
```

注意 sa_role と sso_role を持っているユーザのみが QPTune アクションを実行できます。ただし compare は任意のユーザが実行できます。

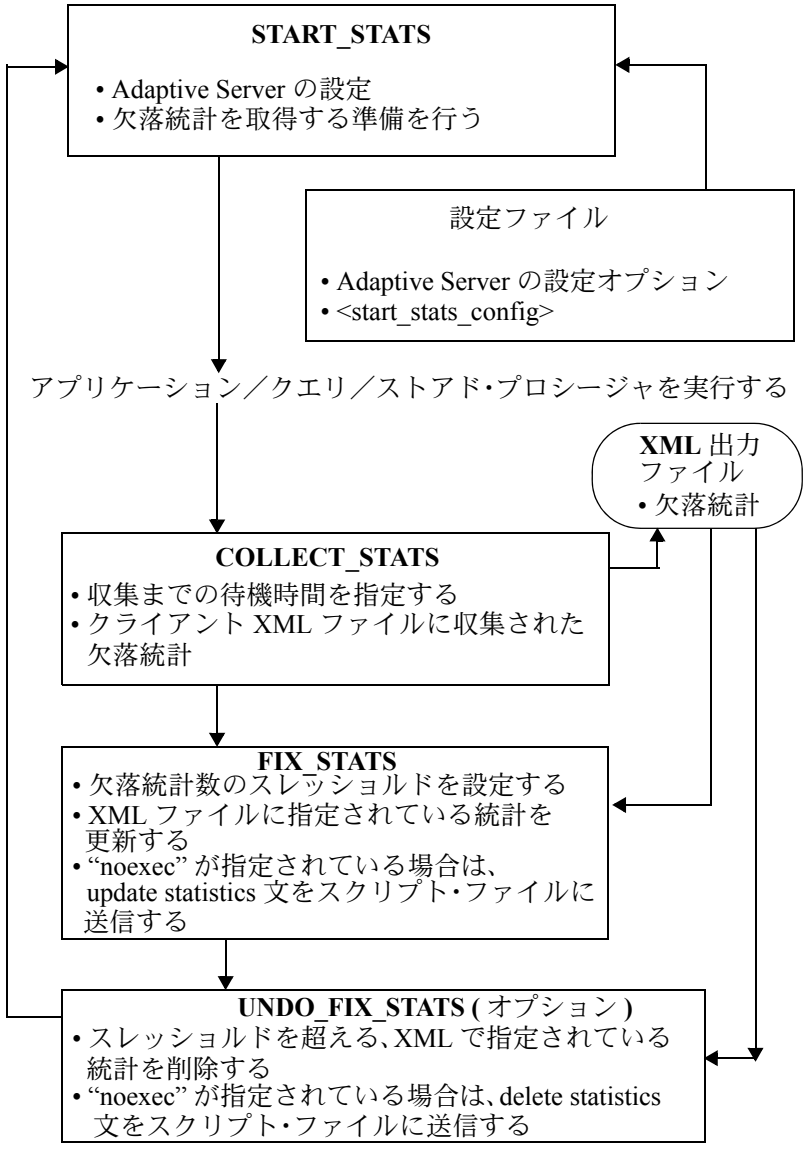
QPTune を使用した欠落統計の修正

QPTune を使用して、サーバのアップグレード後に、欠落統計を修正または更新します。QPTune を使用して欠落統計を修正する主な手順は、次のとおりです。

- start_stats アクションを使用して QPTune を起動します。
- アプリケーション、クエリ、またはストアド・プロシージャを実行します。
- 欠落統計情報があれば、指定の XML ファイルに収集します。「統計情報の収集」(21 ページ) を参照してください。
- fix_stats アクションを使用して、上記の XML ファイルで指定した統計情報を更新します。「統計値の修正」(22 ページ) を参照してください。
- (オプション) undo_fix_stats アクションを使用して、欠落統計の修正を取り消します。「undo_fix_stats の使用」(23 ページ) を参照してください。

欠落統計を修正するチューニング・サイクルを以下に示します。

図 2-1: 欠落統計を修正するチューニング・サイクル



欠落統計を修正するための QPTune の起動

start_stats アクションを使用してユーティリティを起動します。次に例を示します。

```
QPTune -A start_stats -S my_host:4816/my_database
       -v

Executing : QPTune -U sa -P [unshown]
-S jdbc:sybase:Tds:my_host:4816/my_database
-A start_stats -M allrows_dss -T 0 -i null
-o metrics.xml -f null -c config.xml -l 5
-e elap_avg -d 5,5 -m 5 -n null -v
You are now connected to database: my_database
[INFO] Config: sp_configure 'capture missing statistics', 1
[INFO] Config: sp_configure 'system table', 1
[INFO] Config: delete sysstatistics where formatid =110
```

-c オプションを使用して設定ファイルを指定することもできます。これにより、設定ファイルの <start_stats> セクションからサーバレベルの構成設定が抽出されます。「設定ファイル」(30 ページ) を参照してください。

統計情報の収集

QPTune で start_stats アクションを実行してシステムの準備が完了したら、collect_stats アクションを実行して欠落統計情報の収集を開始できます。QPTune ですぐにアクションを実行することも、後で実行することもできます。この機能を使用すると、start_stats と collect_stats の手順を自動化できます。

collect_stats は、欠落統計数に指定されたスレッショルドを超えている統計を求めるため、欠落統計情報を sysstatistics テーブルから取得します。QPTune は欠落統計を統合し、更新する必要がある最小セットの統計を特定します。

-m オプションは、欠落統計数のスレッショルドを指定します。クエリの統計の欠落回数がスレッショルド値以上になると、統計が収集され、XML ファイルにエクスポートされます。デフォルトのスレッショルド数は 5 です。

-o オプションは、欠落統計を保持する出力 XML ファイルを指定します。collect_stats の出力 XML を fix_stats アクションと undo_fix_stats アクションへの入力に使用します。

たとえば、次のように指定する。

```
QPTune -A collect_stats -m 1 -o missingstats.xml -v
        -S my_host:4816/my_database

Executing : QPTune -U sa -P [unshown] -S
jdbc:sybase:Tds:my_host:4816/my_database -A collect_stats -M
allrows_dss -T 0 -i null -o missingstats.xml -f null -c
config.xml -l 5 -e elap_avg -d 5,5 -m 1 -n null -v
You are now connected to database: my_database
Now collecting missing statistics information from
```

```
sysstatistics on "Fri Sep 26 10:08:06 PDT 2008".
<?xml version="1.0" encoding="UTF-8"?>
<server url="jdbc:sybase:Tds:my_host:4816/my_database"
file="missingstats.xml"
type="missing stats" datetime="Fri Sep 26 10:08:06 PDT 2008" >
<missingStat id="1">
<id>1068527809</id>
<stats>Y(y4,y2)</stats>
<count>2</count>
</missingStat>
<missingStat id="2">
<id>1068527809</id>
<stats>Y(y3)</stats>
<count>1</count>
</missingStat>
<missingStat id="3">
<id>1068527809</id>
<stats>Y(y2,y1)</stats>
<count>1</count>
</missingStat>
<missingStat id="4">
<id>1068527809</id>
<stats>Y(y1)</stats>
<count>1</count>
</missingStat>
</server>
The missing statistics information is written into XML file:
missingstats.xml
[INFO] End config: sp_configure 'enable metrics capture', 0
[INFO] End config: sp_configure 'abstract plan dump', 0
[INFO] End config: sp_configure 'system table', 0
[INFO] End config: sp_configure 'capture missing statistics',
0
Program has restored the data source for metrics collection.
----- QPTune finished executing. ------
```

統計値の修正

欠落統計情報を XML ファイルに収集したら、-m オプションで指定した欠落統計数のスレッショルド以上の統計値を更新できます。fix_stats アクションを使用して統計値を更新します。

-i オプションは、すべての欠落統計を含む入力 XML ファイルを指定します。

“noexec” を示す -N オプションと、出力スクリプト・ファイルを示す -o オプションを使用することで、実際の更新を実行しないで統計値を更新する SQL スクリプトを生成できます。生成されたすべての update statistics 文を使用して出力ファイルが作成されますが、文は実行されません。生成されたスクリプトは SQL ファイル・フォーマットです。-N オプションを使用すると、後で SQL スクリプトを実行してリソースを最適化することを選択できます。

たとえば、次のように指定する。

```
QPTune -A fix_stats -m 1 -i missingstats.xml
        -v -S my_host:4816/my_database

Executing : QPTune -U sa -P [unshown] -S
jdbc:sybase:Tds:my_host:4816/my_database -A fix_stats -M
allrows_dss -T 0 -i missingstats.xml -o metrics.xml -f null -
c config.xml -l 5 -e elap_avg -d 5,5 -m 1 -n null -v
You are now connected to database: my_database
Fix statistics on "Fri Sep 26 10:14:59 PDT 2008"
------------------------------------------------------------
Details of statements(s) fixed:
------------------------------
Fixed statistics:[Update] Y(y4,y2)
[INFO] Fix Statement = update statistics Y(y4,y2)
Fixed statistics:[Update] Y(y3)
[INFO] Fix Statement = update statistics Y(y3)
Fixed statistics:[Update] Y(y2,y1)
[INFO] Fix Statement = update statistics Y(y2,y1)
Fixed statistics:[Update] Y(y1)
[INFO] Fix Statement = update statistics Y(y1)
----- QPTune finished executing. ------
```

たとえば、次のように結果が表示されます。

```
QPTune -U sa -P -S my_host:5000/my_database
       -A fix_stats -m 5 -i missingstats.xml
        -N -o missingstats.sql
```

undo_fix_stats の使用

修正した欠落統計を復元するには、undo_fix_stats アクションを使用します。undo_fix_stats は、XML ファイルで指定された統計値の欠落数が -m オプションで指定した数以上になると、その統計値を削除します。

たとえば、次のように指定する。

```
QPTune -A undo_fix_stats -m 1 -i missingstats.xml
        -v -S my_host:4816/my_database

Executing : QPTune -U sa -P [unshown] -S
jdbc:sybase:Tds:my_host:4816/my_database -A undo_fix_stats -M
allrows_dss -T 0 -i missingstats.xml -o metrics.xml -f null -
c config.xml -l 5 -e elap_avg -d 5,5 -m 1 -n null -v
You are now connected to database: my_database
Fix statistics on "Fri Sep 26 10:20:23 PDT 2008"
------------------------------------------------------------
Details of statements(s) fixed:
------------------------------
Fixed statistics:[Delete] Y(y4,y2)
[INFO] Fix Statement = delete statistics Y(y4,y2)
```

```
Fixed statistics:[Delete] Y(y3)
[INFO] Fix Statement = delete statistics Y(y3)
Fixed statistics:[Delete] Y(y2,y1)
[INFO] Fix Statement = delete statistics Y(y2,y1)
Fixed statistics:[Delete] Y(y1)
[INFO] Fix Statement = delete statistics Y(y1)
----- QPTune finished executing. ------
```

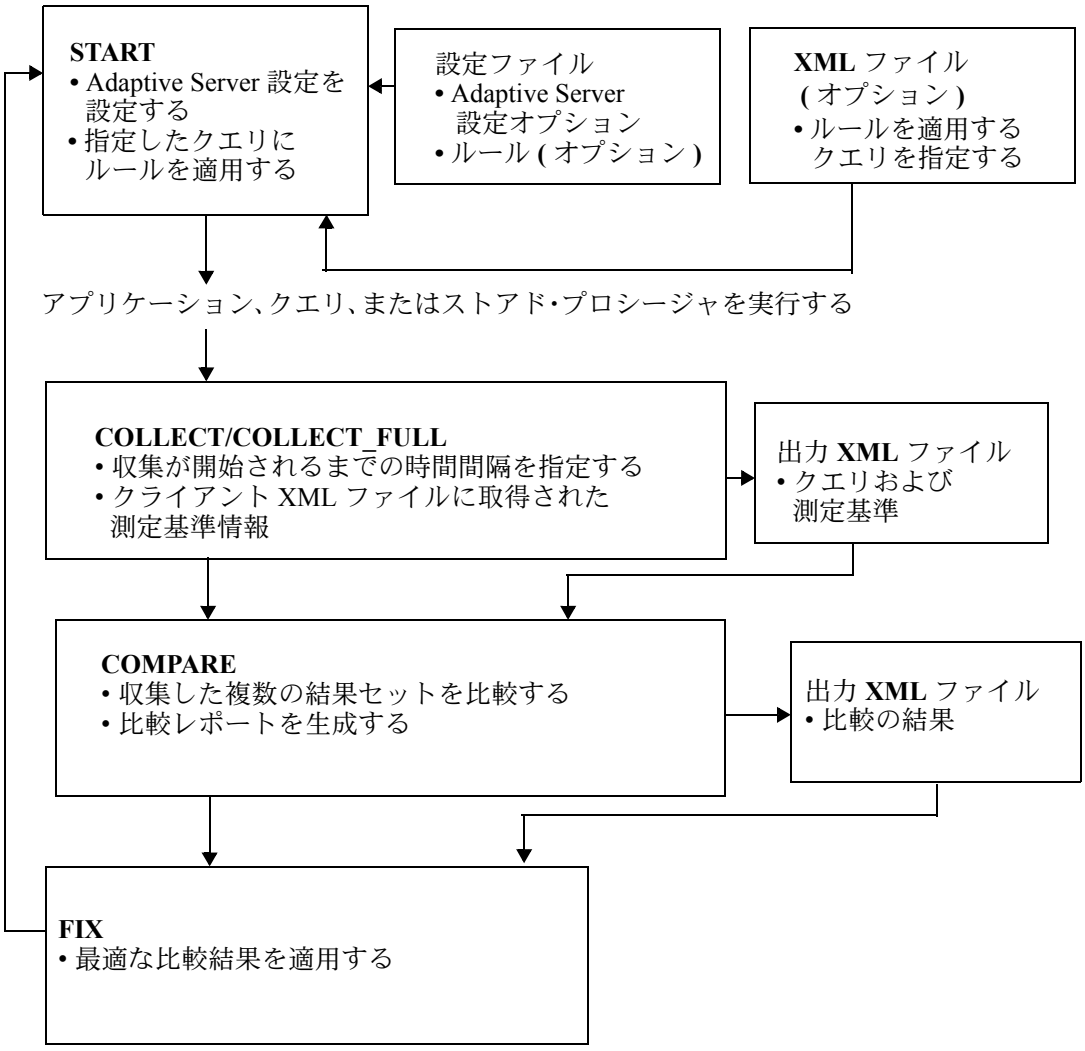
QPTune を使用したクエリまたはアプリケーションのチューニング

QPTune を使用してアプリケーションまたはクエリのチューニングを行うための主なタスクは次のとおりです。

- 次のいずれかの方法で QPTune を起動します。
  - 標準の最適化目標設定を適用する場合は、「簡単な start」(26 ページ) の手順に従います。
  - 特別なカスタム・ルールを指定のクエリに適用する場合は、「カスタムの start」(26 ページ) の手順に従います。
- チューニングするアプリケーション、クエリ、またはストアド・プロシージャを実行します。
- 測定基準を指定の XML ファイルに収集します。「測定基準の収集」(27 ページ) を参照してください。
- さまざまな最適化目標用に収集した測定基準のセットを比較します。この手順では上記の手順の XML ファイルを入力に使用し、パフォーマンス比較レポートを生成します。「測定基準の比較」(28 ページ) を参照してください。
- 比較の最適な結果をターゲット・サーバ上の指定の各クエリに適用します。「最適な結果の適用」(29 ページ) を参照してください。

QPTune のアプリケーションまたはクエリのチューニング・サイクルを以下に示します。

図 2-2: *QPTune* のアプリケーションまたはクエリのチューニング・サイクル



クエリまたはアプリケーションをチューニングするための QPTune の起動

QPTune の start アクションは、正しいサーバレベルの構成設定でサーバの準備を行います。設定ファイルを使用すると、設定は設定ファイルの <start> セクションから抽出されます。設定ファイルの <end> セクションは、QPTune の collect アクションの最後にシステムを元の状態に復元できる設定を指定します。設定ファイルの詳細については、「設定ファイル」(30 ページ) を参照してください。

簡単な start

標準の最適化目標設定を適用する場合は、次のコマンドを使用して QPTune を起動します。

```
QPTune -S host:port/database -A start
        [-M {allrows_oltp, allrows_dss, allrows_mix}]
```

-M オプションを使用して、Adaptive Server の 3 つの最適化目標に対応する、あらかじめプログラミングされたモードの 1 つを呼び出します。

- allrows_mix
- allrows_oltp
- allrows_dss (デフォルト)

カスタムの start

カスタム・ルールを指定のクエリに適用する場合は、次のコマンドを使用します。

```
QPTune -S host:port/database -A start -M custom_1
        -i input.xml -l 3 [-v]
```

-M オプションを使用して、カスタム・モードを指定します。カスタム・モードは設定ファイルの <mode> セクションで指定される特別なルールのグループです。ルールは抽象クエリ・プランを使用してクエリ・レベルで適用される Adaptive Server 15.0 の最適化基準です。

上記の例では、*custom_1* というカスタム・モードを使用し、次のようなルールが組み合わされています。

- use optgoal allrows_mix
- use merge_join off
- use opttimeoutlimit 15

-i オプションを使用して、collect フェーズで標準目標設定を適用したときに QPTune によって生成される入力 XML ファイルを指定します。収集した複数の測定基準ファイルを比較して、最適な目標設定のファイルを生成できます。入力ファイルにはクエリの SQL テキストが含まれています。

-I オプションを -i オプションと一緒に使用して、これらの特別なルールを適用するクエリの数を指定します。クエリはファイルの最初からカウントされます。-I オプションのデフォルト値は 0 で、これは入力ファイルのすべてのクエリが適用されることを示します。

測定基準の収集

システムを起動したら、アプリケーションを実行して測定基準を XML ファイルに収集します。-o オプションを使用して、出力測定基準ファイルを指定します。-v オプションは冗長出力を指定します。-M オプションは、カスタム・モードか標準モードかを指定します。

次のいずれかの方法で測定基準を収集できます。

- -T 0 オプションを使用してすぐに収集する。
- *t* 分後に -T *t* オプションを使用して収集する。

たとえば、次のコマンドは XML を *a2.xml* というファイルに書き込みます。カスタム・モードは <bestmode> タグ内に記述されています。

```
QPTune -S host:port/database -A collect -T 0
        -o a2.xml -v
```

```
Program has configured the data source for metrics collection.
Now collecting information from sysquerymetrics on "Tue Feb 19 22:16:04 PST 2008".
<?xml version="1.0" encoding="UTF-8"?>
 <server url="jdbc:sybase:Tds:SHANGHI:5000" type="ASE" mode="custom_1" datetime="Tue Feb
19 22:16:04 PST 2008">
<query id="1">
<qtext> select count(T.title_id) from authors A, titleauthor T
where A.au_id = T.au_id </qtext>
<elap_avg>300</elap_avg>
<bestmode> custom_1
</bestmode>
</query>
</server>
```

注意　collect オペレーションの出力 XML ファイルを compare、fix、start オペレーションへの入力に使用できます。

測定基準の比較

測定基準を収集したら、さまざまな XML ファイルを比較して、クエリごとに最適なクエリ最適化目標または基準を取得できます。次に例を示します。

```
QPTune -A compare -f a1.xml,a2.xml[,a3.xml..] -d 51,10
     -o best.xml -S my_host:5000/my_database
```

The -f option specifies a list of two or more collected metrics sample files separated by commas. ファイル名がスペースを含む場合は、引用符でファイル名をカプセル化します。

-d オプションは、スレッショルドのパーセンテージと絶対値を指定します。パフォーマンスの改善がスレッショルドのパーセンテージと絶対値を超える場合は、fix オペレーションで「未処理」と見なされます。これらの未処理のクエリの最適化目標／基準はプラン修正としてサーバに適用されます。

スレッショルドのパーセンテージと絶対値の組み合わせのデフォルトは “5,5” です。パーセンテージのみを指定すると、絶対値のデフォルトは 0 になります。パーセンテージ値は 0 ～ 100 で、絶対値には 0 より大きい任意の数字を指定できます。

-o オプションはファイルの比較結果を指定します。ファイルには分析されたすべてのクエリの最適な設定が保持されます。

-s オプションは、ファイル・サイズが最大から最小の順にファイルをソートします。最大のクエリ・セットを含むファイルが比較の基準に使用されます。

次の例は、compare オペレーションの結果を示しています。

```
Compare all the files: | a1.xml, a2.xml|
Report generated on "Tue Aug 19 21:13:04 PST 2008"
--------------------------------------------------------------------------------
File #1: [name= a1.xml  :  mode=allrows_mix]
File #2: [name= a2.xml  :  mode=custom_1]
Query count in File #1 : [mode=allrows_mix]         6
Query count in File #2 : [mode=custom_1]            7
================================================================================
Query count improved in File #2: [mode=allrows_mix] 3

Total performance improved [from 422 to 129]: 69 %

Following queries run better in File #2:
[mode=allrows_mix]
--------------------------------------------------------------------------------
Group 1: improved by no more than 25% [0 queries]
Group 2: improved by 25% to 50% [1 queries]
Query: select count(T.title_id) from authors A, titleauthors T where A.au_id = T.au_id
Average elapsed time (ms): File #1=100  File #2=50   Improvement=50.0%  Outstanding=No
Group 3: improved by 50% to 75% [0 queries]
Group 4: improved by 75% to 100% [2 queries]
Query: select count(*) from titlles T, titleauthors TA where T.title_id = TA.title_id
Average elapsed time (ms): File #1=34  File #2=7      Improvement=79.0%  Outstanding=Yes
```

```
Query: select au_lname, au_fname from authors where state in ("CA", "AZ")
Average elapsed time (ms): File #1=9   File #2=0       Improvement=100.0% Outstanding=No
```

上記の例は、2 つの XML 測定基準ファイル間の比較を示しています。*a1.xml* には 6 つのクエリがあり、*a2.xml* には 7 つのクエリがあります。両方のファイルに共通するクエリ間でのみ比較できます。*a2.xml* で高速に実行されたクエリは 3 つあります。改善は 4 つのグループに分類されます。

- グループ 1 – 0 ～ 25%
- グループ 2 – 25 ～ 50%
- グループ 3 – 50 ～ 75%
- グループ 4 – 75 ～ 100%

25 ～ 50% のクエリが 1 つ、75 ～ 100% のクエリが 2 つあります。グループ 2 のクエリは “Outstanding=No” とマークされていますが、これは 51% というスレッショルドを踏まえ、このクエリは修正されないことを意味します。

複数のファイルを比較する場合、QPTune は 2 つのファイルの最適値で最初のファイルを更新し、続いて新しいファイルを 3 番目のファイルと比較します。以降も同様です。

最適な結果の適用

すべてのクエリの結果を分析したら、fix アクションを使用して、最適な設定をデータベース・システムのクエリに適用します。

たとえば、次のように指定する。

```
QPTune  -S host:port/database -A fix -i best.xml
         -v -g
```

-i オプションは、比較結果として得られるクエリとその最適なプランを指定します。

-g オプションは、fix アクションとともに使用した場合に、デフォルトの目標を適用します。デフォルトの目標は、QPTune の fix アクションを使用する最適なプランとして、ほとんどのクエリが使用した最良の optgoal 設定です。このオプションは、サーバのデフォルトの最適化目標を現在使用していないクエリのプランのみを生成します。

上記の例の fix アクションの結果として得られる出力は次のとおりです。

```
Query Plan(s) fixed on "Wed Sep 17 17:44:09 PDT 2008"
------------------------------------------------------------
Fixed 2 queries using mode "custom_1" with following optimizer settings": '(use optgoal
allrows_mix) (use merge_join off) (use opttimeoutlimit 15)'

Fixed 4 query using mode "allrows_mix"

Apply 都p_configure optimization_goal, 0, allrows_mixî as the default optgoal
```

```
Details of statement(s) fixed:
---------------------------------------
Query: 'select count(T.title_id) from authors A, titleauthor T where A.au_id = T.au_id '
Fixed using: 'custom_1'
[INFO] Fix Statement = create plan 'select count(T.title_id) from authors A, titleauthor
T where A.au_id = T.au_id' '(use optgoal allrows_mix) (use merge_join off) (use
opttimeoutlimit 15)'
Query: 'select * from titleauthors where au_id  > 20 and title_id < 100'
Fixed using: 'custom_1'
[INFO] Fix Statement = create plan 'select * from titleauthors where au_id  > 20 and
title_id < 100' '(use optgoal allrows_mix) (use merge_join off) (use opttimeoutlimit 15)'
```

次に、QPTune は最適化されたクエリ・プランを作成します。このプランは現在のデータベースの sysqueryplans システム・テーブルに保存されます。一致する SQL を持つクエリが見つかると、この最適化されたプランが使用されます。受信した SQL と持続プランの SQL は、2 つの SQL 文のハッシュのチェックサムの種類が一致した場合、一致すると見なされます。リテラルのパラメータ化が明示的に有効になっている場合、2 つの文は次のような探索引数の静的な値のみ異なることがあります。

```
where CustomerID = ì12345î
```

この場合、値 “12345” はプレースホルダ変数で置き換えられるため、探索値に関係なくハッシュ値は同じになります。

新しい述部を追加するなど、アプリケーションで SQL に何らかの変更が加えられた場合、持続プランと一致しなくなり、オプティマイザは現在の設定と使用可能な統計値に基づいてクエリ・プランを作成します。

設定ファイル

設定ファイルにカスタム・モードを定義することができます。QPTune インストールには、いくつかのカスタム・モードを含む標準設定ファイルが組み込まれています。カスタム・モードの “_basic_” は「基本的な最適化」用に予約されています。

QPTune の設定ファイルには、<start>、<start_stats>、<fix>、および <end> セクションを含める必要があります。<mode> セクションはオプションです。

<start> セクションは、測定基準を収集する前の Adaptive Server の構成設定を示します。たとえば、次のように指定する。

```
 <start>
<!-- Recommended server settings -->
<start_config>sp_configure 'enable metrics capture', 1</start_config>
<start_config>sp_configure 'abstract plan dump', 1</start_config>

<!-- Clean up sysqueryplans & sysquerymetrics tables -->
```

```
<start_config>sp_configure 'system table', 1</start_config>
<start_config>sp_metrics 'flush'</start_config>
<start_config>delete sysqueryplans where gid=1 or gid=2</start_config>

<!-- Optional settings that users can change or remove -->

<!-- <start_config>sp_configure 'enable literal autoparam', 1</start_config> ->
<!-- <start_config>sp_configure 'metrics elap max', 0</start_config> -->

<!-- Hint: sp_add_resource_limit can be added to limit resource usage  -->
<!-- Specify a query plan group name to save all existing plans from ap_stdin -->
<!-- Existing plans from ap_stdout will be saved to the corresponding group name added
with '_out'. -->

<save_plans_pre_start>pre_start_qpgroup</save_plans_pre_start>
</start>
```

<end> セクションは <start> セクションに対応し、測定基準の収集後に適用する構成設定が含まれています。たとえば、次のように指定する。

```
<end>
<end_config>sp_configure 'enable metrics capture', 0</end_config>
<end_config>sp_configure 'abstract plan dump', 0</end_config>
<end_config>sp_configure 'system table', 0</end_config>
<end_config>sp_configure 'capture missing statistics', 0</end_config>

<!-- <end_config>sp_configure 'enable literal autoparam', 0</end_config> -->
<!-- <end_config>sp_configure 'metrics elap max', 0</end_config> -->
</end>
```

<start_stats> セクションには統計設定が含まれています。たとえば、次のように指定する。

```
<start_stats>
<!-- Recommended server settings -->
<start_stats_config>sp_configure 'capture missing statistics',1</start_stats_config>
<!-- Reset counter of missing statistics -->
<start_stats_config>
sp_configure 'system table',1
</start_stats_config>
<start_stats_config>
delete sysstatistics where formatid=110
</start_stats_config>
</start_stats>
```

<fix_stats> セクションは次のようになります。

```
<!-- The following set of configurations apply at "-A
fix" -->
<fix>
<!-- Recommended server settings -->
<fix_config>sp_configure 'abstract plan load',1</fix_config>
<!-- Clean up sysqueryplans & sysquerymetrics tables -->
<fix_config>sp_configure 'system table', 1</fix_config>
<fix_config>sp_metrics 'flush'</fix_config>
<fix_config>delete sysqueryplans where gid=1 orgid=2</fix_config>
<!-- Optional settings that users can change or remove -->
<fix_config>sp_configure 'enable metrics capture',1</fix_config>
<!-- <fix_config>sp_configure 'enable literal autoparam',1</fix_config> -->
<!-- <fix_config>sp_configure 'metrics elap max',0</fix_config>-->
<!-- Specify a query plan group name to save all existing plans from ap_stdin -->
<!-- Existing plans from ap_stdout will be saved to the corresponding group name added
with '_out'. -->
<save_plans_pre_fix>pre_fix_qpgroup</save_plans_pre_fix>
</fix>
```

オプションの <mode> セクションを使用すると、別の入力ファイルで指定された 1 つまたは複数のクエリに対してカスタム最適化設定を指定することができます。start アクションと collect アクションの -M オプションでモード設定を指定します。-M オプションで標準最適化目標設定以外のモードを指定すると、QPTune はこのモードをカスタマイズされたモードとして処理し、設定ファイルの <mode> セクションから指定された名前の最適化目標とルール設定を取得します。QPTune は指定されたクエリのリストにカスタム設定を適用します。

例

❖ QPTune を使用した欠落統計の修正

1 QPTune で start_stats を実行し、サーバが欠落統計を収集できるよう準備します。

```
QPTune -A start_stats -v
       -S my_host:4816/my_database
```

出力例：

```
Executing : QPTune -U sa -P [unshown]
-S jdbc:sybase:Tds:my_host:4816/my_database
-A start_stats -M allrows_dss -T 0 -i null -o metrics.xml
-f null -c config.xml -l 5 -e elap_avg -d 5,5
-m 5 -n null -v
You are now connected to database: my_database
[INFO] Config: sp_configure 'capture missing statistics', 1
[INFO] Config: sp_configure 'system table', 1
[INFO] Config: delete sysstatistics where formatid =110
```

2　クライアント・アプリケーション、ストアド・プロシージャ、またはクエリを実行します。

3　QPTune で collect_stats アクションを実行し、欠落統計数のスレッショルドを超える統計値を収集します。欠落統計情報を収集する前に、一定の期間 (-T オプションで指定)、ユーティリティを待機させることもできます。

```
QPTune -A collect_stats -m 1 -o missingstats.xml
       -v -S my_host:4816/my_database
```

出力例：

```
Executing : QPTune -U sa -P [unshown]
-S jdbc:sybase:Tds:my_host:4816/my_database
-A collect_stats -M allrows_dss -T 0 -i null -o missingstats.xml -f null -c
config.xml -l 5 -e elap_avg -d 5,5 -m 1 -n null -v
You are now connected to database: my_database
Now collecting missing statistics information from sysstatistics on "Fri Sep 26
10:08:06 PDT 2008".
QPTune Utility
<?xml version="1.0" encoding="UTF-8"?><server
url="jdbc:sybase:Tds:my_host:4816/my_database"
file="missingstats.xml"
type="missing stats" datetime="Fri Sep 26 10:08:06 PDT 2008" >
  <missingStat id="1">
     <id>1068527809</id>
     <stats>Y(y4,y2)</stats>
     <count>2</count>
  </missingStat>
  <missingStat id="2">
     <id>1068527809</id>
     <stats>Y(y3)</stats>
     <count>1</count>
  </missingStat>
  <missingStat id="3">
     <id>1068527809</id>
     <stats>Y(y2,y1)</stats>
     <count>1</count>
  </missingStat>
  <missingStat id="4">
     <id>1068527809</id>
     <stats>Y(y1)</stats>
     <count>1</count>
  </missingStat>
</server>
The missing statistics information is written into XML file:
missingstats.xml
[INFO] End config: sp_configure 'enable metrics capture', 0
[INFO] End config: sp_configure 'abstract plan dump', 0
[INFO] End config: sp_configure 'system table', 0
[INFO] End config: sp_configure 'capture missing statistics', 0
Program has restored the data source for metrics collection
----- QPTune finished executing. ------
```

4 -m オプションで指定した欠落統計数のスレッショルド以上の統計値を更新します。入力ファイル *missingstats.xml* で指定された欠落統計を修正するには、次のコマンドを使用します。

```
QPTune -U sa -P -A fix_stats -m 1 -i missingstats.xml
       -v -S my_host:4816/my_database
```

出力例：

```
Executing : QPTune -U sa -P
-S jdbc:sybase:Tds:my_host:4816/my_database -A fix_stats
-M allrows_dss -T 0 -i missingstats.1 xml -o metrics.xml -f null
-c config.xml -l 5 -e elap_avg -d 5,5 -m 1 -n null -v
You are now connected to database: my_database
Fix statistics on "Fri Sep 26 10:14:59 PDT 2008"
------------------------------------------------------------
Details of statements(s) fixed:
-------------------------------
Fixed statistics:[Update] Y(y4,y2)
[INFO] Fix Statement = update statistics Y(y4,y2)
Fixed statistics:[Update] Y(y3)
[INFO] Fix Statement = update statistics Y(y3)
Fixed statistics:[Update] Y(y2,y1)
[INFO] Fix Statement = update statistics Y(y2,y1)
Fixed statistics:[Update] Y(y1)
[INFO] Fix Statement = update statistics Y(y1)
----- QPTune finished executing. ------
```

注意 -N オプションと一緒に fix_stats アクションを使用した場合、QPTune は欠落統計を修正する文を実行しないで、代わりにそれらを -o *output_file* で指定した出力ファイルに送信します。

5 (オプション) undo_fix_stats コマンドは -i XML ファイルで指定された統計値を削除します。削除されるのは、欠落数が -m で指定された数以上の統計値です。入力ファイル *missingstats.xml* で指定された欠落統計の修正を取り消すには、次のコマンドを使用します。

```
QPTune -U sa -P -A undo_fix_stats -m 1
       -i missingstats.xml -v
       -S my_host:4816/my_database
```

出力例：

```
Executing : QPTune -U sa -P [unshown]
-S jdbc:sybase:Tds:my_host:4816/my_database
-A undo_fix_stats -M allrows_dss -T 0
-i missingstats.xml -o metrics.xml -f null
-c config.xml -l 5 -e elap_avg -d 5,5 -m 1
-n null -v
You are now connected to database: my_database
Fix statistics on "Fri Sep 26 10:20:23 PDT 2008"
```

```
------------------------------------------------------------
Details of statements(s) fixed:
------------------------------
Fixed statistics:[Delete] Y(y4,y2)
[INFO] Fix Statement = delete statistics Y(y4,y2)
Fixed statistics:[Delete] Y(y3)
[INFO] Fix Statement = delete statistics Y(y3)
QPTune Utility
Fixed statistics:[1 Delete] Y(y2,y1)
[INFO] Fix Statement = delete statistics Y(y2,y1)
Fixed statistics:[Delete] Y(y1)
[INFO] Fix Statement = delete statistics Y(y1)
----- QPTune finished executing. ------
```

❖ QPTune を使用したアプリケーションの最適化

1 QPTune で start を実行し、allrows_oltp、allrows_mix、allrows_dss のいずれかを指定します。

```
QPTune -U sa -P -S my_host:11030/my_database
       -A start -M allrows_mix -v
```

この例では、Adaptive Server は “my_host” というマシンで、ポート番号 11030 と my_database というデータベースを使用して実行されています。

出力例：

```
Executing : QPTune -Usa -P [unshown] 亡
jdbc:sybase:Tds:my_host:11030/my_database
-A start -M allrows_mix -T 0 -i null -o metrics.xml -f null -c config.xml -l 5
-e elap_avg -d 5,5 -n null -v
You are now connected to database: my_database
[INFO] Config: sp_configure 'enable metrics capture', 1
[INFO] Config: sp_configure 'abstract plan dump', 1
[INFO] Config: sp_configure 'system table', 1
[INFO] Config: sp_metrics 'flush'
[INFO] Config: delete sysqueryplans
Apply "sp_configure optimization_goal, 0, allrows_mix" to the data source.
Program has configured the data source for metrics collection.
```

2 クライアント・アプリケーション、ストアド・プロシージャ、またはクエリを実行します。クライアント・アプリケーションは、GUI ベースまたは Web ベースのプログラム、ストアド・プロシージャのセット、スクリプト内の SQL クエリのバッチなどです。たとえば、次のように指定する。

```
isql -Usa -P < sp_telco.sql > sp_telco_allrows_mix.out
```

3 QPTune で collect を実行し、アプリケーションの各クエリの測定基準を収集します。測定基準はファイル *sp_telco_allrows_mix.xml* に収集されます。

```
QPTune -U sa -P -S my_host:11030/my_database -A collect
     -M allrows_mix -o sp_telco_allrows_mix.xml -v
```

その他の各最適化目標またはカスタム・モードについて手順 1 ～ 3 を繰り返します。たとえば、allrows_dss を使用する場合、次のコマンドを実行します。

```
QPTune -U sa -P -S my_host:11030/my_database -A start
     -M allrows_dss

isql -Usa -P < sp_telco.sql > sp_telco_allrows_dss.out
QPTune -U sa -P -S my_host:11030/my_database -A collect
     -M allrows_dss -o sp_telco_allrows_dss.xml
```

allrows_mix モードの出力例：

```
Executing : QPTune -U sa -P [not shown]
     -S jdbc:sybase:Tds:my_host:11030/my_database -A collect
     -M allrows_mix -T 0 -i null -o sp_telco_allrows_mix.xml -f null
     -c config.xml-l 5 -e elap_avg -d 5,5 -n null -v
You are now connected to database: my_database
Now collecting information from sysquery tables on "Tue Aug 26 22:00:49 PDT 2008".
Metrics are flushed.
<?xml version="1.0" encoding="UTF-8"?>
<server url="jdbc:sybase:Tds:my_host:11030/my_database"
file="sp_telco_allrows_mix.xml" mode="allrows_mix"
datetime="Tue Aug 26 22:00:49 PDT 2008" >
<query id="1">
<qtext>SELECT service_key , year , fiscal_period , count(*)
FROM telco_facts T , month M , status S
WHERE T.month_key=M.month_key AND S.status_key = T.status_key
AND call_waiting_status="Dropped"
GROUP BY year , fiscal_period , service_key
ORDER BY year , fiscal_period , service_key </qtext>
<hashkey>323626785</hashkey>
<id>1568005586</id>
<elap_avg>27408</elap_avg>
<bestmode>allrows_mix</bestmode>
</query>
<query id="2">
<qtext>SELECT customer_last_name , customer_first_name FROM
residential_customer R , telco_facts T , service S , month M
WHERE M.month_text = 'February ' AND M.year = 1998
AND S.isdn_flag = 'Y' AND M.month_key = T.month_key
AND S.service_key = T.service_key
AND R.customer_key = T.customer_key -- end comment i
</qtext>
<hashkey>727793461</hashkey>
```

```
<id>1552005529</id>
<elap_avg>3355</elap_avg>
<bestmode>allrows_mix</bestmode>
</query>
. . . . . .
<query id="10">
<qtext>SELECT month_key , service_key , count(*)
FROM telco_facts WHERE month_key = 1
GROUP BY month_key , service_key
</qtext>
<hashkey>1561133104</hashkey>
<id>1680005985</id>
<elap_avg>58</elap_avg>
<bestmode>allrows_mix</bestmode>
</query>
</server>
The metrics information is written into
XML file: sp_telco_allrows_mix.xml
[INFO] End config: sp_configure 'enable metrics capture', 0
[INFO] End config: sp_configure 'abstract plan dump', 0
[INFO] End config: sp_configure 'system table', 0
Program has restored the data source for metrics collection.
----- QPTune finished executing. ------
```

4　すべての実行内容から収集された測定基準をファイル *best.xml* 内の各クエリの最適な測定基準と比較します。この測定基準に新しいモード “new_mode” を定義できます。

```
QPTune -U sa -P -S my_host:11030/my_database -v -A compare -M new_mode
     -f sp_telco_allrows_dss.xml,
     sp_telco_allrows_mix.xml,sp_telco_allrows_oltp -o best.xml
```

出力例：

```
Executing: QPTune -U sa -P [unshown]

     -S jdbc:sybase:Tds:my_host:11030/my_database
     -A compare -M new_mode -T 0 -I null -o best.xml
     -f sp_telco_allrows_mix.xml, sp_telco_allrows_dss.xml,
     sp_telco_allrows_oltp.xml
     -c config.xml -l 5 -e elap_avg -d 5,5 -n null -v
Compare all the files:
sp_telco_allrows_mix.xml,sp_telco_allrows_dss.xml,sp_telco_allrows_oltp.xml
Report generated on "Wed Aug 27 16:29:01 PDT 2008"
-------------------------------------------------------
Sorted List By File Size (Desc.) =
sp_telco_allrows_mix.xml, sp_telco_allrows_dss.xml,
sp_telco_allrows_oltp.xml
File #1 : [name=sp_telco_allrows_mix.xml : mode=allrows_mix]
File #2 : [name=sp_telco_allrows_dss.xml : mode=allrows_dss]
Query count in File #1 [mode=allrows_mix]: 14
Query count in File #2 [mode=allrows_dss]: 12
================================================
```

```
Query count improved in File #2[mode=allrows_dss]: 7
Total performance improved [from 37234 to 7781]: 79 %
Following queries run better in File #2 [mode=allrows_dss]:
-------------------------------------------------
Group 1: improved by no more than 25% [2 queries]
Query: SELECT state, count(*) FROM telco_facts T, service S,
residential_customer C, month M
WHERE T.service_key = S.service_key
AND T.customer_key = C.customer_key AND T.month_key = M.month_key
AND call_waiting_flag = 'Y' AND caller_id_flag = 'Y'
AND voice_mail_flag = 'N' AND state in ('NY', 'NJ', 'PA')
AND fiscal_period = 'Q1'
GROUP BY state
Average elapsed time(ms): File #1=837 File #2=803 Improvement=4.0%
Outstanding=No
Query: SELECT fiscal_period, T.service_key, sum(local_call_minutes),
sum(local_call_count) , count(*)
FROM telco_facts T ,residential_customer C, service S , month M
WHERE T.customer_key = C.customer_key
AND T.service_key = S.service_key AND T.month_key = M.month_key
AND fiscal_period = 'Q4' AND T.service_key in (02, 03)
AND state = 'CA'
GROUP BY fiscal_period , T.service_key
Average elapsed time(ms): File #1=832 File #2=635 Improvement=23.0%
Outstanding=Yes
-------------------------------------------------
Group 2: improved by 25% to 50% [2 queries]Group 3: improved by 50% to 75% [0
queries]
. . . . . .
Group 4: improved by 75% to 100% [3 queries]
Query: SELECT service_key , year , fiscal_period , count(*)
FROM telco_facts T , month M , status S
WHERE T.month_key=M.month_key AND S.status_key = T.status_key
AND call_waiting_status="Dropped" GROUP BY year, fiscal_period,
service_key
ORDER BY year , fiscal_period , service_key -- end comment--
Average elapsed time(ms): File #1=27408 File #2=2126 Improvement=92.0%
Outstanding=Yes
. . . . . .
File #3 : [name=sp_telco_allrows_oltp.xml : mode=allrows_oltp]
Query count in File #3[mode=allrows_oltp]: 13
=================================================
Query count improved in File #3[mode=allrows_oltp]: 4
Total performance improved [from 7781 to 6523]: 16 %
Following queries run better in File #3:

-------------------------------------------------
Group 1: improved by no more than 25% [2 queries]
Query: SELECT fiscal_period , count(*) , sum(local_call_minutes)
FROM residential_customer R , telco_facts T , status S , month M
WHERE S.call_waiting_status=@status AND state =
. . . . . .
```

5　比較の結果得られた最適なプランを使用して、アプリケーションでクエリ・プランを修正します。

```
QPTune -U sa -P -S my_host:11030/my_database -g
       -A fix -i best.xml
```

出力例：

```
Executing : QPTune -U sa -P [unshown]
-S jdbc:sybase:Tds:my_host:11030/my_database
-A fix -M allrows_dss -T 0 -i best.xml
-o metrics.xml -f null -c config.xml -l 5
-e elap_avg -d 5,5 -n null -v
You are now connected to database: my_database
[INFO] Config: sp_configure 'abstract plan load', 1
[INFO] Config: sp_configure 'system table', 1
[INFO] Config: sp_metrics 'flush'
[INFO] Config: delete sysqueryplans
[INFO] Config: sp_configure 'enable metrics capture', 1
You are now connected to database: my_database
Query plan(s) fixed on "Wed Aug 27 17:05:46 PDT 2008"
------------------------------------------------------------
Fixed 3 queries using mode "allrows_oltp"
Fixed 3 queries using mode "allrows_dss"
Fixed 8 queries using mode "allrows_mix"
Apply "sp_configure optimization_goal, 0, allrows_mix" as the
default optgoal.
Details of statements(s) fixed:
------------------------------
Query: SELECT service_key , year , fiscal_period , count(*) -- comment 1 it''s a
comment. whatever "statments"
/* comment 3 */ FROM telco_facts T , month M , status S
WHERE T.month_key=M.month_key AND S.status_key = T.status_key
AND call_waiting_status="Dropped" GROUP BY year, fiscal_period, service_key
ORDER BY year , fiscal_period , service_key -- end comment
-- *** Query #9 ***
. . . . . .
```

❖ QPTune カスタム・モードの使用

1　設定ファイルで定義した独自のカスタム・モードを使用して、特定のクエリを実行できます。QPTune には、Adaptive Server 12.5 の基本的な最適化を表す “_basic_” などのカスタム・モードがいくつか用意されています。たとえば、デフォルトの設定ファイル *config.xml* に含まれるカスタム・モード “custom1” では、allrows_oltp の最適化目標を merge_join_off ルールと一緒に使用できます。

```
<!-- "default" custom mode -->
<mode name="default">
<optgoal>use optgoal allrows_mix</optgoal>
<rule>use merge_join off</rule>
<rule>use opttimeoutlimit 15</rule>
```

```
</mode>
<!-- "_basic_" mode is a reserved system mode. -->
<mode name="_basic_">
</mode>
<mode name="custom1">
<optgoal>use optgoal allrows_oltp</optgoal>
<rule>use merge_join off</rule>
</mode>
```

2　次の例は、“_basic_” カスタム・モードの使用法を示します。

```
QPTune -A start -M _basic_ -Usa -P -S my_host:11030/my_database
       -i best.xml -l 0 -v
isql -Usa -P < sp_telco_2.sql > sp_telco_basic.out
QPTune -A collect -M _basic_ -Usa -P -S my_host:11030/my_database
       -o sp_telco_basic.xml -v
QPTune -A compare -M best -Usa -P -Smy_host:11030/my_database -v
       -f sp_telco_basic.xml,best.xml -o best_basic.xml -d 1,0
```

出力例：

```
Report generated on "Fri Aug 29 13:29:17 EDT 2008"
-------------------------------------------------------
{INFO]Sorted List By File Size (Desc.)=sp_telco_basic.xml,best.xml
File #1 : [name=sp_telco_basic.xml : mode=_basic_]
File #2 : [name=best.xml : mode=best]
Query count in File #1: 14
Query count in File #2: 14
===============================================
Query count improved in File #2: 7
Total performance improved [from 2441 to 1529]: 37 %
Following queries run better in File #2:
-----------------------------------------------
Group 1: improved by no more than 25% [4 queries]
Query: SELECT customer_last_name , customer_first_name
FROM residential_customer R , telco_facts T , service S , month M
WHERE M.month_text = 'February ' AND M.year = 1998
AND S.isdn_flag = 'Y' AND M.month_key = T.month_key
AND S.service_key = T.service_key AND R.customer_key = T.customer_key
Average elapsed time(ms): File #1=393 File #2=306 Improvement=22.0%
Outstanding=Yes
. . . . . .
```

アップグレードに関する問題

QPTune を使用すると、Adaptive Server 15.0 にアップグレードしたときに最適なパフォーマンスを得ることができます。15.0 より前のバージョンのサーバと比べて適切に実行されないクエリがある場合、QPTune を使用すると、Adaptive Server Enterprise でバージョン 12.5.4 と同様のクエリ・プランを生成できます。これらのプランの方がバージョン 15.0 のクエリ・プランよりも実行速度が速い場合、QPTune はこれらのプランを保持し、以降のすべてのクエリ実行にこれらのプランを使用します。

❖ Adaptive Server 15.0 へのマイグレーション中の QPTune の使用

1　アプリケーションに応じて、あらかじめプログラミングされた 3 つの Adaptive Server 15.0 のモード (“mix”、“dss”、“oltp”) の測定基準情報を取得します。

```
QPTune -A start -M allrows_oltp
        -S my_host:5000/my_database
```

< クエリ、アプリケーション、またはストアド・プロシージャを実行します >

```
QPTune -A collect -M allrows_oltp -T 0 -o oltp.xml
        -S my_host:5000/my_database

QPTune -A start -M allrows_mix
        -S my_host:5000/my_database
```

< クエリ、アプリケーション、またはストアド・プロシージャを実行します >

```
QPTune -A collect -M allrows_mix -T 0 -o mix.xml
        -S my_host:5000/my_database

QPTune -A start -M allrows_dss
        -S my_host:5000/my_database
```

< クエリ、アプリケーション、またはストアド・プロシージャを実行します >

```
QPTune -A collect -M allrows_dss -T 0 -o dss.xml
        -S my_host:5000/my_database
```

2　バージョン 12.5.4 と同様の最適化を使用して、Adaptive Server 15.0 の測定基準情報を取得します。

```
QPTune -A start -M _basic_ -i oltp.xml -l 10
        -S my_host:5000/my_database
```

< クエリ、アプリケーション、またはストアド・プロシージャを実行します >

```
QPTune -A collect -M _basic_ -T 0 -o basic.xml
        -S my_host:5000/my_database
```

3　測定基準情報を比較します。

```
QPTune -A compare -d 10 -o best.xml
        -f basic.xml,oltp.xml,mix.xml,dss.xml
        -S my_host:5000/my_database
```

4 比較結果の中から最適な結果を使用してクエリ・プランを修正します。

```
QPTune -A fix -i best.xml
        -S my_host:5000/my_database
```

5 (オプション) プランの修正後にパフォーマンスの改善を確認し、アプリケーションを再実行して、測定基準情報を収集します。

```
QPTune -A collect -T 0 -o new_best.xml
        -S my_host:5000/my_database
```

他の XML 出力ファイルに対して *new_best.xml* との compare を実行すると、その *new_best.xml* に最適な結果が得られます。

ローカリゼーション

QPTune コマンド・ライン・ユーティリティはローカライズされており、英語以外に 中国語、フランス語、ドイツ語、日本語、韓国語、ポーランド語、ポルトガル語、スペイン語、タイ語の 9 か国語でメッセージを表示できます。言語プロパティ・ファイルは、*qptune.jar* ファイルにパッケージ化されています。QPTune はシステムのデフォルト・ロケールの設定言語に従って表示を設定します。

QPTune GUI

QPTune GUI は、Adaptive Server プラグインによって使用される一連の Java ライブラリです。

環境とシステムの稼働条件

QPTune 機能にアクセスするには、Adaptive Server バージョン 15.0.3 ESD #1 以降を使用する必要があります。

QPTune の実行ファイルとライブラリは、次の場所にインストールされます。

(Unix)

$SYBASE/$SYBASE_ASE

$SYBASE/$SYBASE_ASE/lib

(Windows)

%SYBASE%¥%SYBASE_ASE%

%SYBASE%¥%SYBASE_ASE%¥lib

QPTune GUI を実行するには、次のファイルがインストールに存在している必要があります。

- (UNIX)
  - *$SYBASE/$SYBASE_ASE/qptune/config.xml*
  - *$SYBASE/$SYBASE_ASE/lib/qptune.jar*
  - *$SYBASE/$SYBASE_ASE/qptune/lib/xercesImp.jar*
  - *$SYBASE/$SYBASE_ASE/qptune/lib/xml-apis.jar*
  - *$SYBASE/jConnect-6_0/classes/jconn3.jar*
  - *$SYBASE_JRE6/bin/java*
- (Windows)
  - *%SYBASE%¥%SYBASE_ASE%¥qptune¥config.xml*
  - *%SYBASE%¥%SYBASE_ASE%¥lib¥qptune.jar*
  - *%SYBASE%¥%SYBASE_ASE%¥qptune¥lib¥xercesImp.jar*
  - *%SYBASE%¥%SYBASE_ASE%¥qptune¥lib¥xml-apis.jar*
  - *%SYBASE%¥jConnect-6_0¥classes¥jconn3.jar*
  - *%SYBASE_JRE6%¥bin¥java*

次の環境変数を設定します。

- SYBASE_JRE6 － Java Runtime インストールに設定する
- SYBASE － マシン上の最新の Sybase インストールに設定する
- SYBASE_ASE － マシン上のインストールの Adaptive Server コンポーネント (ディレクトリ) に設定する

QPTune GUI の起動

QPTune GUI は、Sybase Central™ の ASE プラグインを使用します。QPTune GUI にアクセスするには、Sybase Central がインストールされている必要があります。Sybase Central の ASE プラグインの詳細については、『システム管理ガイド第 1 巻』を参照してください。

Sybase Central を起動し、ASE プラグインを使用してサーバを構成します。プラグインを使用してアクセスできるサーバでは、QPTune GUI を使用して次のことを実行できます。

- Adaptive Server での欠落統計の修正

  サーバのアップグレード後、およびサーバのチューニング前に、QPTune を使用してサーバで欠落統計を更新します。

- Adaptive Server でのタスクのチューニング

  この機能を使用して、QPTune が実行およびレポートで分析できるチューニング・タスクを定義します。さらに修正をサーバに適用できます。

欠落統計の修正

QPTune GUI を使用して、サーバのアップグレード後に、欠落統計を修正または更新します。QPTune の「欠落統計の修正」機能にアクセスするには、サーバ名を右クリックして、欠落統計の修正に使用できる 2 つのメニュー・オプションの 1 つを選択します。

- 欠落統計の修正：このオプションを選択すると、欠落統計の修正ウィザードが表示されます。QPTune は、欠落統計に関する情報を XML ファイルに収集し、そのファイルを使用して欠落統計を修正します。XML ファイルの名前とターゲット・データベースを指定し、[次へ] をクリックして [オプション] ページに進みます。

  [オプション] ページで、欠落統計数のスレッショルドを指定します。デフォルトのスレッショルド数は 5 です。欠落統計数のスレッショルドの詳細については、「統計情報の収集」(21 ページ) を参照してください。

  また、update statistics 文のスクリプト・ファイルへの送信のみを行うように選択することもできます。これを行うには、文を保存するファイル名を入力し、[完了] をクリックして、チューニング・タスクを実行せずに保存します。

  [実行] をクリックした場合、QPTune による欠落統計の修正手順に進みます。その後、QPTune によって、ウィザードが発行するコマンドの概要ページが表示されます。

- 欠落統計の修正の取り消し：このオプションを選択すると、欠落統計の修正の取り消しウィザードが表示されます。指定された XML ファイルの statistics 文を使用することで、QPTune はサーバへの以前の修正を取り消すことができます。XML ファイルの名前とターゲット・データベースを指定し、[次へ] をクリックして [オプション] ページに進みます。

  [オプション] ページで、欠落統計数のスレッショルドを指定します。デフォルトのスレッショルド数は 5 です。欠落統計数のスレッショルドの詳細については、「統計情報の収集」(21 ページ) を参照してください。

  [完了] をクリックすると、QPTune によって、以前の欠落統計の修正を取り消す手順が行われます。その後、QPTune によって、ウィザードが発行するコマンドの概要ページが表示されます。

欠落統計の修正ウィザードまたは欠落統計の修正の取り消しウィザードを使用するには、sa_role および sso_role を持っている必要があります。QPTune の欠落統計の修正サイクルの詳細については、「QPTune を使用した欠落統計の修正」(19 ページ) を参照してください。

注意　QPTune で欠落統計を収集または修正するには、少なくとも 1 回はアプリケーションを実行する必要があります。

チューニング・タスク

QPTune には、条件を満たすすべてのサーバ上の既存のチューニング・タスクを表示する [チューニング・タスク] というパネルがあります。ウィザードの手順に従って、QPTune のチューニング・サイクルを進めることができます。チューニング・タスクの定義は ASEP が実行されているクライアント・マシンに格納され、そのマシン上でのみアクセスできます。

QPTune での、アプリケーションまたはクエリのチューニング・サイクルにはいくつかの段階があります。QPTune のアプリケーションまたはクエリのチューニング・サイクルの詳細については、「QPTune を使用したクエリまたはアプリケーションのチューニング」(24 ページ) を参照してください。

サーバに新しいチューニング・タスクを作成するには、次の手順に従います。

- クエリをチューニングするサーバに接続します。
- サーバ名をクリックします。[チューニング・タスク] タブが表示されます。

  注意　[チューニング・タスク] タブが表示されない場合は、環境変数が正しく設定されていることと、必要なファイルおよびディレクトリがインストールにすべて含まれていることを確認してください。

- [チューニング・タスク] タブをクリックして、ウィンドウ内を右クリックし、[新規作成] - [チューニング・タスク] メニュー項目を表示します。
- [新規作成] - [チューニング・タスク] メニュー項目を選択します。QPTune ウィザードが表示されます。

または、ツールバーに表示される [チューニング・タスク] 作成ボタンを使用してウィザードを表示することもできます。

注意　メニュー項目および作成ツールバー・ボタンを使用するには、sa_role と sso_role を持っている必要があります。

QPTune ウィザードには、Adaptive Server のチューニングにおけるさまざまな段階に対応する次の画面があります。

- 設定：
  - 名前および設定

    タスクに関連付けられるタスク名と設定ファイルを指定します。両方を指定すると、[次へ] ボタンと [完了] ボタンが有効になります。設定ファイルが既に存在する場合は、ウィザードにより、ファイル名の下に注意が表示されて、そのファイルが示されます。

    [冗長モード] オプションを選択して、より詳細な出力を生成することもできます。

  - サーバ設定

    QPTune 実行のさまざまな段階で発行されたサーバ設定コマンドを表示または編集できます。コマンドに対する変更内容は、[比較] ページでの実行の直前に設定ファイルに書き込まれます。

  - モードの選択

    さまざまなモードを選択して QPTune を実行できます。デフォルトでは、次のあらかじめプログラミングされた 3 つのモードすべてが選択されます。

    - 意思決定支援システム (DSS)
    - オンライン・トランザクション処理 (OLTP)
    - 混合負荷 (MIX)

    カスタマイズされたモードを定義するには、[追加] ボタンをクリックします。モードの順序を変更するには、上下の移動ボタンを使用します。チューニング・タスクでは、少なくとも 2 つの収集結果を後で比較できるように、2 つ以上のモードを選択する必要があります。

    カスタマイズされたモードは、一連のクエリの最適化目標で分類されたチューニング・パラメータの集まりです。名前、1 つ以上のルール、および結果ファイルを指定した場合にのみ [OK] ボタンが有効になります。

    ルールを追加または編集するには、ポップアップ・テキスト入力ボックスを使用します。ルールを削除するには、[削除] ボタンを使用します。

- 収集：
  - アプリケーション

    収集フェーズが開始される前に、含める実行プログラムまたはスクリプト・ファイルを指定できます。
  - 収集

    このページでは、収集設定を指定できます。デフォルトでは、QPTune は、収集遅延のない最適化目標設定のみを収集し、収集の平均経過時間を評価します。
- 比較：
  - 比較

    このページでは、比較のスレッショルド設定 (パーセンテージと絶対値) と出力ファイル名を指定します。

    [完了] をクリックして、タスク定義と設定ファイルを保存します。

    [実行] をクリックすると、QPTune は、指定されたすべてのモードを実行して、測定基準を収集し、結果を比較して出力ファイルに保存します。[プレビュー] ボタンを使用すると、発行されるコマンドがリストされます。
  - 結果

    このページには、チューニング・プロセスの出力が表示されます。各クエリに最適なプランが選択された場合、比較結果でパフォーマンスの改善が示されます。出力 XML ファイルには、各クエリの最適なプランまたは optgoal 設定が含まれています。

    修正をサーバに適用することもできます。[修正] をクリックして、クエリに最適なプランまたは optgoal 設定を適用します。これにより、修正されるクエリについて、sysqueryplans テーブルにエントリが生成されます。

    fix オペレーションでデフォルトの最適化目標をサーバに適用する場合は、[デフォルトの最適化目標の適用] を選択します。デフォルトの最適化目標は、compare オペレーションで大部分のクエリが最良の optgoal として選択した optgoal 設定です。

    [デフォルトの最適化目標の適用] を選択した場合、以後の fix オペレーションで、このデフォルトの optgoal 設定を最良の optgoal に選択しなかった残りのクエリに最適な結果が適用されます。

    [デフォルトの最適化目標の適用] を選択しない場合は、fix オペレーションが結果ファイルのすべてのクエリに適用されます。

QPTune リファレンス情報

説明

QPTune は Java および XML で書かれた Adaptive Server ユーティリティです。QPTune を使用すると、最適なクエリ・プラン、最適化目標、その他の構成設定を特定し、それらをクエリまたはサーバのレベルで適用できます。これにより、その後のクエリの実行で最適なパフォーマンスが得られます。

構文

```
QPTune [-U <username>] [-P <password>] [-S <hostname:port/database>]
[-A <action [start|collect(_full)|compare|fix|(start|collect|fix|undo_fix)_stats]>]
[-M <mode>] [-T <appTime>] [-i <inputFile>] [-o <outputFile>]
[-f <fileList(,)>] [-c <configFile>] [-l <limit>] [-e <evalField>]
[-d <diff%(,diff_abs)>] [-m <missingCount>] [-n <login>] [-J <charset>]
[-N (noexec)] [-g (applyOptgoal)][-v (verbose)] [-s (sort)] [-h (help)]
```

例：

```
QPTune -U sa -P -S my_host:5000/my_database -A collect -M
allrows_mix -T 0
-o metrics.xml -c config.xml -e elap_avg -d 5,5 -l 5 -i
metrics.xml
-fa1.xml,a2.xml,a3.xml -v -s
```

パラメータ

-U *username*

データベースのユーザ名を指定します。

-P *password*

データベースのパスワードを指定します。

-S *server*

データベース・サーバを指定します。データベース・サーバは *host*:*port*/*database* で示されます。

注意 QPTune アクションを使用するときは、-S オプションを指定してください。

-A *action*

実行するアクションを指定します。start | collect | collect_full | compare | fix | start_stats | collect_stats | fix_stats | undo_fix_stats のいずれかを指定します。

-J *charset*

Adaptive Server への接続に使用する文字セットを指定します。このオプションを指定しない場合、Adaptive Server ではサーバのデフォルト文字セットが使用されます。

注意 インストールされた JRE がサーバのデフォルト文字セット・コードをサポートしていない場合は、ログイン・プロセス時にエラー・メッセージが表示されます。-J オプションを使用して、-J utf8 などの汎用文字セットを指定してください。

-M *mode*

アプリケーションの最適化目標またはカスタム・モードを指定します。allrows_oltp、allrows_dss、allrows_mix のいずれかを指定します。カスタム・モードも指定でき、_basic_ is がシステム予約カスタム・モードです。

-T *appTime*

アプリケーションの実行時間を分単位で指定します。

-o *outputFile*

出力ファイルを指定します。

-i *inputFile*

fix、fix_stats、undo_fix_stats アクションの入力ファイルを指定します。また、-i を使用して、カスタム・モードの start で特別なルールを指定のクエリに適用できます。

-f *fileList*

ファイルのリストを比較して最適なプランを取得します。複数のファイル名を区切るには、カンマを使用してください。

-c *configFile*

設定ファイルを指定します。

-l *limit*

特別なルールを使用して解析および適用する必要があるクエリの数の制限を指定します。

-e *evalField*

パフォーマンスの比較に使用する評価フィールドを指定します。

-d *difference*

パフォーマンスの改善が未処理であると判断するためのパーセンテージと絶対値の差分を指定します。

-N

fix_stats および undo_fix_stats とともに使用し、-N は、update statistics 文または delete statistics 文を使用する SQL スクリプトを生成します。update 文または delete 文は QPTune からは実行されません。文は -o オプションで指定した SQL スクリプトに記述されます。

-n *login*

クエリ実行を収集および解析するユーザのログインを指定します。

-m *missingCount*

欠落統計のスレッショルド値を指定します。デフォルト値は 5 です。

-v

冗長モードを指定します。

-g

fix アクションとともに使用した場合に、デフォルトの目標を適用します。デフォルトの目標は、QPTune の fix アクションを使用する最適なプランとして、ほとんどのクエリが使用した最良の optgoal 設定です。このオプションは、サーバのデフォルトの最適化目標を現在使用していないクエリのプランのみを生成します。

パラメータに特定の値を指定しない場合は、以下のデフォルトが使用されます。

- -A : collect
- -M : allrows_dss
- -T : 0
- -o：metrics.xml
- -c : config.xml
- -e：elap_avg
- -d : 5,5. パーセンテージのみを指定した場合、絶対値はデフォルトで 0 に設定されます。
- -l *limit*
- -m 5

パーミッション

QPTune で compare 以外のアクションを実行できるのは、sa_role および sso_role を持つユーザのみです。

第 3 章

互換モードでのクエリ・プロセッサの実行

Adaptive Server バージョン 15.0 では、クエリ・プロセッサに大幅な変更が加えられています。ほとんどのユーザにとって、新しいクエリ・プロセッサは環境の高速化と効率化をもたらすものです。ただし、Adaptive Server バージョン 12.5.4 以前の制限の多いクエリ・プロセッサに合わせてサーバとアプリケーションをチューニングしている場合、一部の状況ではバージョン 15.0 のクエリ・プロセッサの利点が適さないことがあります。その場合は、互換モードを使用して、Adaptive Server をバージョン 12.5.x から 15.0 にアップグレードしながら、同時にバージョン 12.5.x と同様のパフォーマンス特性を維持できます。互換モードを有効にする場合、Adaptive Server 15.0 は、バージョン 12.5.4 で使用されていたものと同様のクエリ・エンジンを使用し、代わりの最適化および実行方法を提供します。

互換モードの有効化

Adaptive Server 15.0.3 ESD #1 以降では、セッションまたはサーバワイド・レベルで互換モードを有効にできます。

- セッション・レベル – set compatibility_mode on | off を使用して、現在のセッションの互換モードの有効と無効を切り替えます。
- サーバワイド・レベル – sp_configure 'enable compatibility mode', 1 | 0 を使用して、サーバの互換モードの有効と無効を切り替えます。

セッション・レベルで設定した互換モードは、サーバ・レベルの設定よりも優先されます。

enable compatibility mode は動的設定パラメータです。有効にするために Adaptive Server を再起動する必要はありません。ただし、ストアド・プロシージャ、またはアドホック・クエリのコンパイル済みプランはすべてステートメント・キャッシュから削除する必要があります。

注意 sp_configure は、互換モードを有効にしても、すでにプロシージャまたはステートメント・キャッシュ内にあるキャッシュされたクエリ・プランには影響しないことを示す警告を表示します。

sp_configure は、次のように互換モードと競合する設定オプションが検出された場合も警告を表示します。

- abstract plan dump、abstract plan load または abstract plan replace のいずれかが設定されている場合。
- statement cache と literal autoparam が設定されている場合。
- ヒストグラムのチューニング係数が 1 以外の値に設定されている場合。

互換モードでの機能のサポート

互換モードを有効にすると、Adaptive Server はすべての insert、delete、update、および select クエリにクエリ・プロセッサを使用します。

クエリ・プロセッサは、完全互換モードまたは部分互換モードを使用します。

- 完全互換モード – Adaptive Server 15.0 はバージョン 12.5.x で使用されている最適化および実行方法と同様の方法を使用します。
- 制限付き互換モード – Adaptive Server はバージョン 12.5.x で使用されている最適化方法と同様の方法を使用します。

通常 Adaptive Server は可能なかぎり完全互換モードを使用します。完全互換モードを使用できない場合、制限付き互換モードに切り替えます。

表 3-1 に Adaptive Server バージョン 15.0 より前のバージョンの機能に対するサポートを示します。

表 *3-1:* 互換モードにおけるバージョン *12.5* の機能のサポート

互換モードでサポートが制限されている 12.5.4 の機能	完全互換モードでサポートされているか	制限付き互換モードでサポートされているか
text カラムと image カラムのクエリ	いいえ	はい
参照整合性	いいえ	はい
参照整合性を必要とする挿入	いいえ	いいえ
プロキシ・テーブル	いいえ	はい
ラウンドロビン分割	いいえ	いいえ

互換モードでサポートが制限されている 12.5.4 の機能	完全互換モードでサポートされているか	制限付き互換モードでサポートされているか
暗号化および暗号テキスト	いいえ	はい
rand2 関数を含むクエリ	いいえ	はい
明示的 (plan 句を使用) または暗黙的 (plan dump または plan load) な抽象プラン	いいえ	はい
拡張データ型 (Java ADT や Java UDF など)	いいえ	はい
XML 関数	いいえ	はい
ブラウズ・モード	いいえ	はい
並列ヒント	いいえ	いいえ
並列ソート	いいえ	いいえ

互換モードでは、Adaptive Server 15.0 の次の機能がサポートされていません。

- 分割されたテーブル
- 32 以上のカラムに対する group by
- スクロール可能な非反映型カーソル
- 計算カラムに対するコマンド
- “instead-of-triggers” を呼び出すクエリ
- “instead-of-triggers” 内で実行されたクエリ
- ステートメント・キャッシュでパラメータ化されたリテラルを発行するクエリ。ただし、クエリに insert...values コマンドが含まれる場合を除きます。
- showplan_in_xml で使用されるクエリ処理診断
- hash または hashbyte 関数を含むクエリ
- ユーザ定義関数 (SQL UDF)
- 明示的なタイムスタンプの挿入 (Adaptive Server バージョン 15.0.2 以降および Replication Server® で使用可能)
- SQL ベースの複写 (Adaptive Server バージョン 15.0.3 以降で使用可能)
- データ・ページに収まる最大サイズよりも広い group by 結果ロー
- xmltable 関数

注意　互換モードのクエリ・プランは並列プランとして実行されません。

診断用の追加のトレース・フラグ

トレース・フラグ 477 は互換モードを変更します。評価される各クエリに対して、Adaptive Server はこのメッセージをエラー・ログに出力します。このメッセージは Adaptive Server がクエリの処理に完全互換モードを使用したかどうかを示します。

```
Compatibility = true | false
```

このメッセージには、互換モードが選択されなかった理由と、使用可能であればクエリ・テキストが含まれます。

新しいストアド・プロシージャ *sp_compatmode*

Adaptive Server 15.0.3 ESD #1 以降で sp_compatmode を使用すると、完全互換モードを効果的に使用できるかどうかを確認できます。互換モードと競合する次のような設定オプションが検出されると、sp_compatmode は警告を生成します。

- abstract plan dump、abstract plan load または abstract plan replace のいずれかが設定されている場合
- statement cache と literal autoparam が設定されている場合
- ヒストグラムのチューニング係数が 1 以外の値に設定されている場合

例 1: 次のようにサーバ・オプションを設定して sp_compatmode を実行します。

- 互換モードを設定する
- dump/load/replace を “on” にする
- ステートメント・キャッシュを “on” にする
- literal autoparam を “on” にする
- ヒストグラムのチューニング係数を 20 に設定する

```
1> sp_compatmode
Compatibility mode is enabled.
WARNING: Compatibility mode will not be used when 'abstract
plan dump/load/replace' is on.
WARNING: Compatibility mode may not be used when statement
cache and literal autoparam are enabled.
WARNING: The configuration option 'histogram tuning factor' is
configured with value '20', which is not the default value in
ASE 12.5. This may lead to different accuracy of statistics and
different query plans.
(return status = 0)
```

例 2：互換モードを設定しないで sp_compatmode を実行します。

```
1> sp_compatmode
Compatibility mode is not enabled.
(return status = 0)
```

注意 ヒストグラムのチューニング係数の設定を Adaptive Server 15.0 のデフォルト (20) から Adaptive Server 12.5 のデフォルト (1) に変更すると、Adaptive Server 12.5 との整合性が高いプランが作成されます。

@@qpmode グローバル変数の変更

互換モードで *@@qpmode* を使用すると、直前に実行したクエリが処理されたときのクエリ処理モードが表示されます。クエリ処理モードは 4 つあります。

- 0 – 最適化できないクエリ。create table、set など。
- 1 – 完全互換モードで実行されたクエリ。
- 2 – 制限付き互換モードで実行されたクエリ。
- 3 – 15.0 クエリ・プロセッサで実行されたクエリ。

診断ツール

set showplan 出力を使用すると、完全互換モードでクエリを処理した場合、Adaptive Server 12.5.4 と同様のフォーマットでクエリ・プランが表示されます。

索引

記号

数字

A

B

D

F

Q

S

U

あ

え

お

か

き

く

け

こ

さ

せ

た

ち

て

と

は

ま

も

り

れ

ろ