SYBASE TECHWAVE 2010

SQL ANYWHERE におけるセルフ管理とセルフ修復

ANIL K GOEL
DIRECTOR, SYBASE ENGINEERING
AUGUST 11, 2010

このプレゼンテーションの目的

- 自動データベース管理の必要性を説く
- SQL Anywhere の設計哲学について説明する
- SQL Anywhere の関連機能について概要を紹介する
  - セルフ管理
  - セルフチューニング
  - セルフ修復

SYBASE TECHWAVE 2010

謝辞

- 各リリースにおける優れた業績に対し、SQL Anywhere エンジニアリング・チームに謝意を表します。
- SQL Anywhere は、まぎれもなく業界で最高のチームです。

- 本プレゼンテーションについては、Glenn Paulley 氏、Mohammed Abouzour 氏、Ani Nica 氏、Ray Liu 氏にご助力いただいたことを感謝いたします。

自動データベース管理

- セルフ管理／セルフ構成
- セルフチューニング／セルフアダプト
- セルフ修復
  - 監視と修正／問題点に関するアドバイス
- セルフ保護
- 管理の容易さ
- 目標：ゼロ (手動) 管理
- 設計の自動化
- 管理ツール
  - インデックス・コンサルタント、アプリケーション・プロファイラ、その他

SYBASE TECHWAVE 2010

SQL Anywhere の設計目標

- 管理の容易さ
  - 包括的で、しかも理解しやすいツール
- デフォルト設定のままで優れたパフォーマンス
- EmbeddabilitY 機能 → セルフチューニング
  - 多くの環境で DBA が不在
- クロスプラットフォームのサポート
- 相互運用性

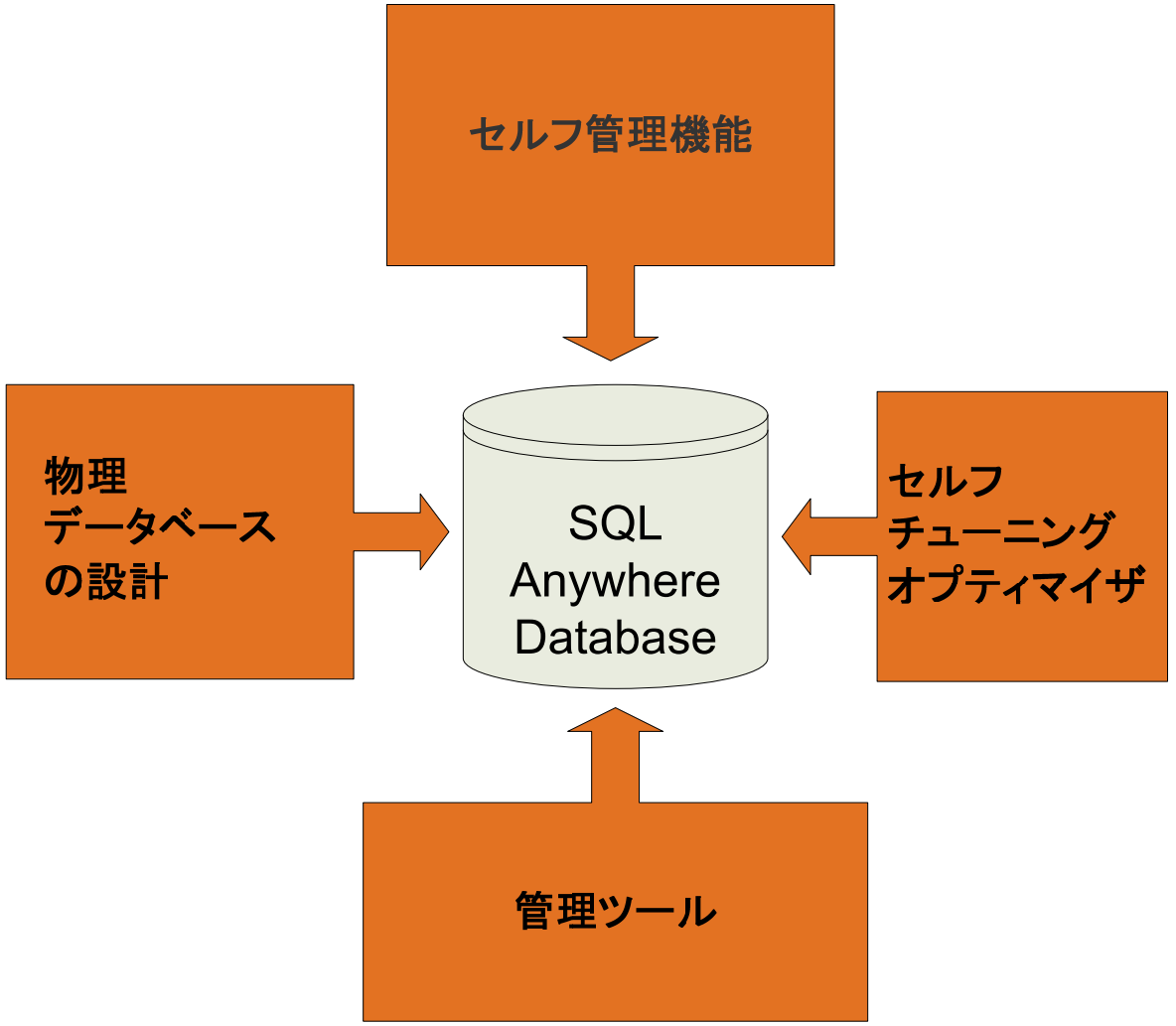
➔自動データベース管理への全体論的なアプローチ

SYBASE TECHWAVE 2010

セルフ管理が重要な理由

- ポイント：複雑さ
  - アプリケーション開発はますます複雑化している： ORM ツールキットや、データベース・レプリカ間を同期する分散コンピューティングなどの新しい開発パラダイム
  - さまざまな難題を解決するため、IT においてデータベースのユビキタス化が進む
    - 企業では依然として多種多様な DBMS 製品の所有と管理が続いているため、管理コストがかさんでいる
  - ユビキタス性にはさまざまなスケールが必要
    - TCO を安定させるには、各開発者の生産性向上を図る必要がある

EMBEDDABILITY



物理データベースの設計

配備の容易さ

SYBASE TECHWAVE 2010

物理データベースの設計

- 論理データベースと物理データベースの設計の兼ね合いは、パフォーマンスに大きく影響する
- 設計変更の潜在的な効果が、作業負荷の実行特性に応じて行き渡るようにする
- 代表的な作業負荷を設定することは困難
  - データの不均衡、相関関係、実システムのシミュレーションから生じる人為的な競合の軽減などの問題がある
- 大部分の DBMS ベンダは現在、インデックスやマテリアライズド・ビューの作成を支援するツールを提供している
  - ただし、すべての目的がそうしたツールで網羅されているわけではない
- ***SQL Anywhereは使用も配備も簡単***

SYBASE TECHWAVE 2010

SQL Anywhere による物理データベースの設計

- SQL Anywhere のデータベースは最大 15 の dbspace で構成される
  - 各 dbspace が OS のファイルに相当
  - それぞれがテンポラリ・ファイル、トランザクション・ログに使用される
  - その他はユーザ・データに使用
  - UNIX の RAW パーティションはサポート対象外
- イメージのコピーは単純な OS ファイルのコピーで実行する
  - イメージ・コピーのユーティリティは不要
  - データベースを別のマシンまたは CE デバイスに容易に配備できる
  - ファイルは、ユーザの操作を必要とせずにサポート対象のあらゆるプラットフォームで相互運用できる
    - 必要に応じてオンザフライ方式のデータ変換を行う

10

物理データベースの設計

- データベース・ファイルは必要に応じて自動的に拡張され、データ量が大きくなっても対応
  - ディスクがフルになったとき、サーバはユーザのコールアウト・プロシージャを実行できる
  - サーバはテンポラリ・ファイルのガバナーを提供し、中間結果に領域が必要なときに接続が切れないようにする
- プライマリ・キー、外部キーに対するインデックスは自動的に作成される
  - サーバは冗長な (重複または包含される) インデックスを自動的に検出する
    - 複数の論理インデックスが同じ物理構造を共有する

TECHWAVE 2010

サーバ・キャッシュのセルフ管理

動的なメモリ管理

SYBASE TECHWAVE 2010

SQL Anywhere のメモリ管理

- 単一の異種バッファ・プールで、事前の制限は少ない
- バッファ・プールの構成
  - テーブルとインデックス・データ・ページ
  - チェックポイント・ログ・ページ
  - ビットマップ・ページ
  - ヒープ・ページ (クエリ実行プラン、最適化グラフ、接続構造、ストアド・プロシージャ、トリガのためのデータ構造)
  - 空き (使用可能) ページ
- すべてのページ・フレームは同じサイズ
- *完全内包型のメモリ・マネージャ*
  - セルフ管理のメモリ・フットプリント

SQL Anywhere のメモリ管理

- バッファ・プールに、ソート領域、ヒープ領域、テーブル・ページのプリセット境界はない
- バッファ・プールのページ置換方式では、さまざまなカテゴリのページと要求をオンデマンドで管理できる必要がある
  - 作業負荷は時間経過とともに変化する
  - 静的な構成では、管理負荷とサービス・レベルのバランスをとる必要がある

SYBASE TECHWAVE 2010

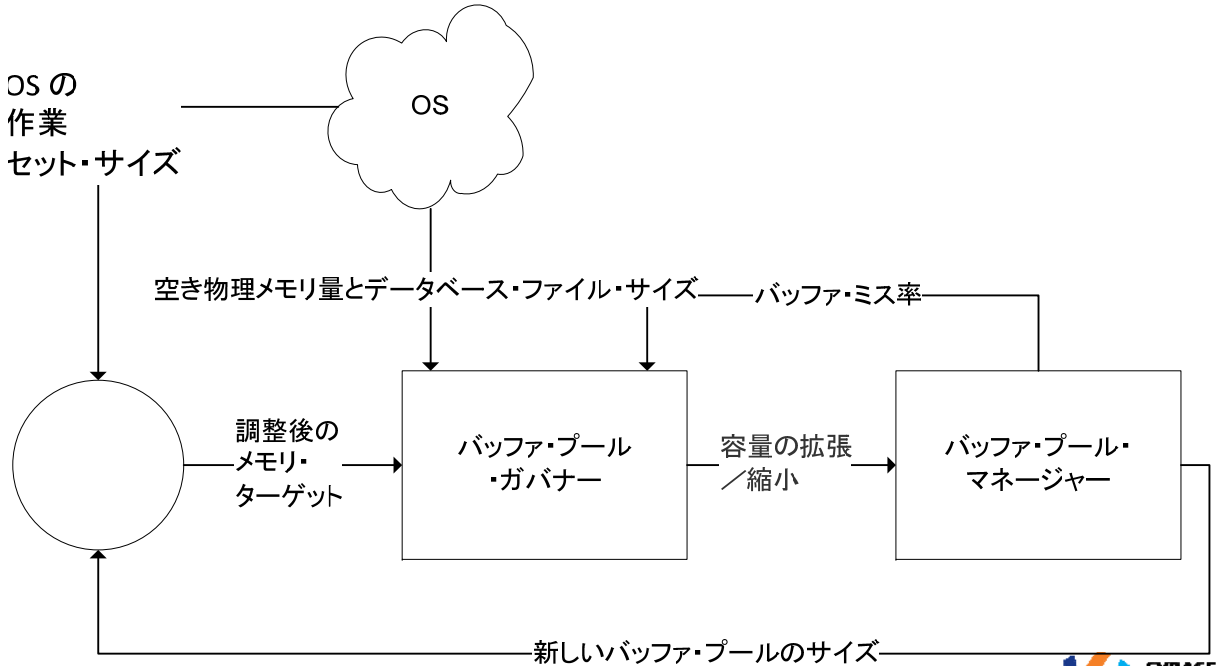
動的なメモリ管理

- SQL Anywhere のサーバは以下の***どちらにも***対応できるよう必要に応じてバッファ・プールを拡張または縮小する
  - データベース・サーバのロード
  - 他のアプリケーションの物理メモリ要件
- デフォルトではサポート対象の全プラットフォームで有効
  - 下限サイズ、上限サイズ、初期サイズをユーザが設定できる

SYBASE TECHWAVE 2010

動的なメモリ管理

- 基本概念：バッファ・プールのサイズをオペレーティング・システムの作業セット・サイズに一致させる
  - フィードバック制御ループ



SYBASE TECHWAVE 2010

マルチプログラミング・レベルのセルフチューニング

SQL ANYWHERE 12 の新機能

SYBASE TECHWAVE 2010

タスク・スケジューリング・モデル

- データベース・サーバには主に 2 つのアーキテクチャがある:
  - 接続単位のワーカー:
    - ワーカーは接続の期間中を通じて接続ごとに占有となる
- 要求単位のワーカー:
  - 1 つのワーカー・プールで 1 つの要求キュー: 各ワーカーが一度に 1 つの要求を選択して完了する
  - 同じワーカーが同じ接続を処理する保証はない
- SQL Anywhere は要求単位のワーカー・モデルを利用する

SYBASE TECHWAVE 2010

SQL Anywhere のタスク・スケジューリング・アーキテクチャ

- 要求を実行するワーカーのプールは小さいのが普通
  - スケジューラはワーカー間に作業を*動的に割り当てる*
    - 協調的なマルチタスク処理
  - 未割り当ての要求はスレッドが使用可能になるまで待機
  - サーバはクエリ内の動的な並行処理をサポート
    - 並行処理の程度は利用可能なリソースによって異なる
- プールのサイズがマルチプログラミングのレベルを決定する
  - SA のデフォルト: 20
- リソースの効率的な利用
- 多数の接続の効率的な処理

SYBASE TECHWAVE 2010

SQL Anywhere のタスク・スケジューリング・アーキテクチャ

- 待機中の要求は*スワップの候補*
  - 仮想メモリの技術を利用してメモリ・フットプリントを一定に維持しようとする
  - サーバ・ロードから要求された場合には、ユーザ接続のためのヒープがディスクにスワップされる
    - 1989 年、SQL Anywhere の最初のリリースで初めて実装
  - ヒープ内のポインタを*スイズルし、オンデマンドでヒープを再ロードできる*
- 診断のための接続では、予約された特別なスレッドを使用して要求を実行できる
  - 大規模なエンジンの診断も可能

SYBASE TECHWAVE 2010

要求単位のワーカー・アーキテクチャ

- ワーカー・プールのサイズの選択方法
  - ワーカー・プールが大きい場合：
    - サーバの同時性レベルを高くする
    - サーバ・リソースに対する競合を増やす
    - サーバの作業セット・サイズを大きくする
  - ワーカー・プールが小さい場合：
    - ハードウェア・リソースの未使用率
    - 作業負荷の同時性レベルを制限する
    - 未処理の要求を処理するワーカーがないため、サーバがハングする可能性がある

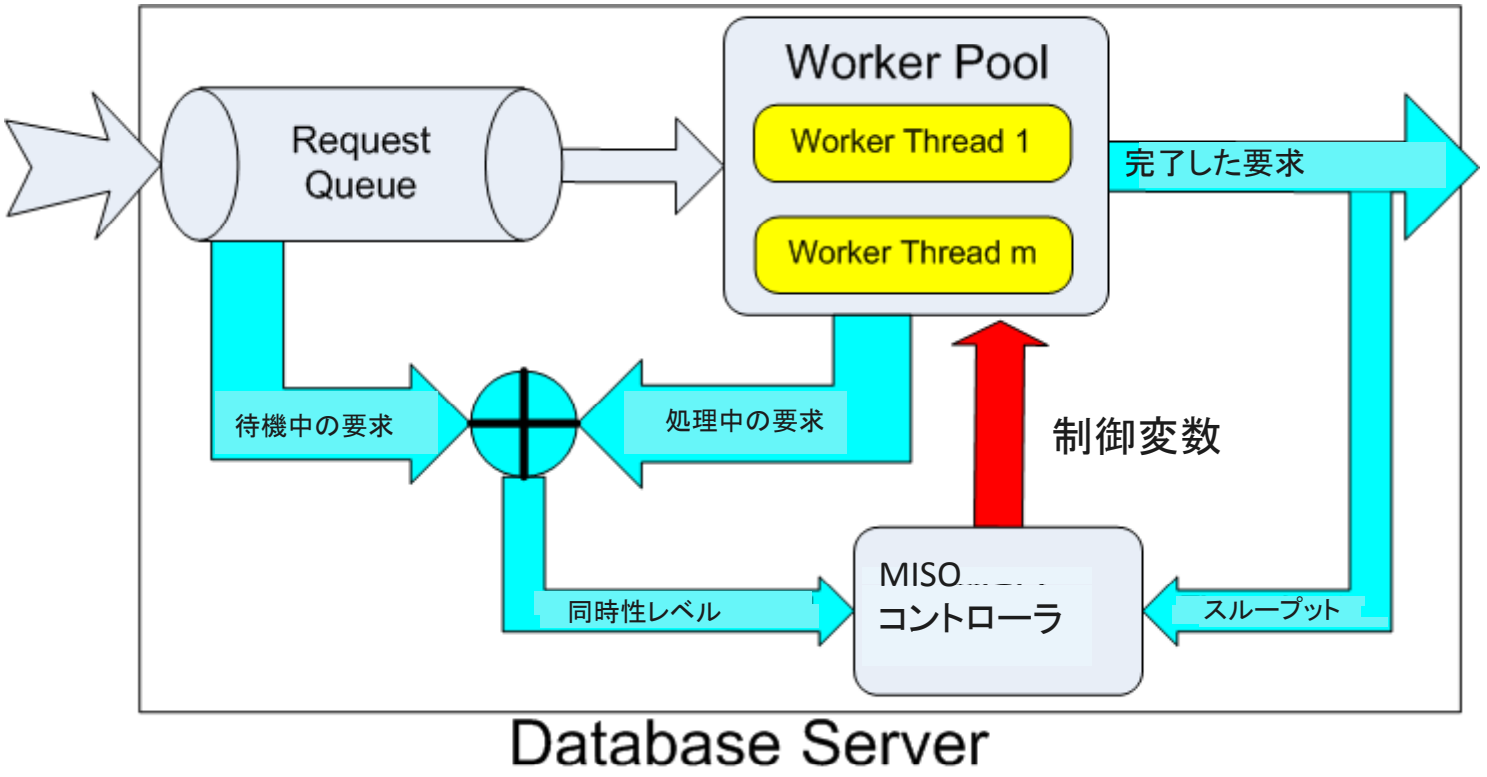
SYBASE TECHWAVE 2010

SQL Anywhere での解決策

- ワーカー・プールのサイズを、つまりサーバのマルチプログラミング・レベル (MPL) を動的に調整する
- 動的な MPL の効果 :
  - DBA が配慮しなければならないパラメータが 1 つ減る
  - 異なる作業負荷でサーバのスループットが向上する
  - 作業負荷のトランザクションが複合のとき、変化を処理しやすい

SYBASE TECHWAVE 2010

SQL Anywhere スケジューラー



要求単位のワーカー

セルフチューニングとクエリ最適化

SYBASE TECHWAVE 2010

自動統計管理

- 1992 年以来の SQL Anywhere の機能
  - 初期の実装では、ハッシュベースの構造を使用してカラム密度と、頻度-値統計を管理していた
- 現在は
  - セルフチューニング・カラムのヒストグラム
    - ベース・テーブルとテンポラリ・テーブルの両方で
    - 統計はオンザフライ方式で自動的に更新される
  - ジョインのヒストグラムは、最適化プロセスの中間結果分析で構築される
  - サーバは永続的なインデックス統計をリアルタイムで維持
  - 最適化に際してインデックスをサンプリング

カラム・ヒストグラム

- セルフチューニングの実装
  - 標準 (レンジ) バケットと、頻度-値統計のどちらも組み込み
  - 述部の評価と更新 DML 文の結果を利用してリアルタイムで更新
  - デフォルトでは、統計は各 DML 要求の実行中に計算される
- LIKE 述部の最適化に使用する文字列についての統計をキャプチャする新しい技法
- ヒストグラムは LOAD TABLE 文または CREATE INDEX 文で自動的に計算される
  - 必要な場合には明示的にも作成／削除が可能
  - ただし、デフォルトではアンロード／再ロードをまたがって維持される

SQL Anywhere のクエリ・オプティマイザ

- SA は実行するたびに要求を最適化する
- サーバ・コンテキストを考慮する
- クエリが複雑になっても優れた最適化時間
  - 主として左側の深いツリーを構築する、プロプライエタリなコストベースのジョイン列挙アルゴリズム
  - Bushy ツリーを列挙し、複雑にネストした LEFT、RIGHT、FULL OUTER JOINS、および派生テーブルを得る
  - 最適化プロセスには、ヒューリスティックおよびコストベースの複雑なリライトが含まれる
  - ハードウェア上に制約はない - 単一のブロックにおける 500 の修飾子でテスト済み
- 利点 : サーバ環境、バッファ・プールの内容／サイズ、データの不均衡に応じたプラン。パッケージとして管理する必要がない

クエリ・オプティマイザの自動バイパス

- シングルテーブルの単純なクエリは、クエリ・オプティマイザ外で最適化され、きわめて高速にアクセス・プランが生成される
- ストアド・プロシージャ、トリガ、イベントにおけるクエリのアクセス・プランはキャッシュされ、以降の実行に再利用される
  - プランは「訓練期間」を経て、プランのバリエーションが決定される
  - バリエーションがない (可変値を除いても) 場合には、プランがキャッシュされて再利用される
  - クエリは、プランが最適を下回らないように対数的に再度最適化される

SYBASE TECHWAVE 2010

適応クエリ処理

- 場合によってはオプティマイザが、実際の中間結果のサイズ見積りが不十分な場合に実行できる代替のアクセス・プランを生成する
  - サーバは実行時に自動的に代替プランに切り替わる
- ハッシュ・ジョインのようにメモリ負荷の高い演算子では、バッファ・プールの利用率が高いときに使用されるメモリが少なくなる

SYBASE TECHWAVE 2010

適応クエリ処理

- データベースのバックアップなど一部の処理には、ストレージに利用される I/O デバイスの特性を判定するためのサンプリング・プロセスが伴う
  - 主な目的は、利用可能なディスク・ヘッド数を決定すること
  - プロセスは適切な数の CPU を使用してスループットを最大化できる
  - アルゴリズムは、システムの他の CPU 要件による影響を受けやすく、必要に応じて CPU 使用率は自動的にスケール・ダウンされる

SYBASE TECHWAVE 2010

適応クエリ内並行処理

- SQL Anywhere は、有利な場合にアクセス・プランを並行処理するアプローチをとる
  - 並行処理の程度は、列挙プロセス中にコストに基づいて決定される
- 作業はワーカー・プール・サイズとは独立してパーティション化される
  - 並行処理の程度という点では、プランは概ねセルフチューニングされる
  - 一定期間に利用可能なワーカーの数が最適より少ない場合でも、クエリ・フラグメントが不足しない

SYBASE TECHWAVE 2010

セルフ修復統計

SQL ANYWHRE 12 の新機能

SYBASE TECHWAVE 2010

セルフ修復統計が必要な理由

- 統計のセルフチューニングは非トランザクション型
  - オーバヘッドを低く保つ
  - ロールバックの際に同期がとれなくなる場合がある
- セルフチューニングのアルゴリズムは、設計上、概算による
  - 単一ローの DML には負荷が低く損失も多い独立した方法が用いられる
  - 統計の生成はデータを 1 回しか参照せず、不正確
  - セルフチューニングを完全にすることが目的ではない
    - データの不均衡が激しい場合でも同期が維持される
- セルフチューニングでは、サーバがビジーな場合に最新状態を維持できない可能性がある
- ➔セルフチューニングされた統計の精度は恣意的に低くなることがある
  - システムは自身の監視が必要

SYBASE TECHWAVE 2010

SQL Anywhere での解決策

- **統計ガバナー**
  - バックグラウンド・サーバ・プロセスの内部システム
  - 利用されるとき統計の精度をセルフ監視
  - 精度の低い統計をセルフ修復
  - 精度の極端に悪い統計は削除
- **方法**
  - QP 中の概算エラーを分類および記録する
  - 精度が一定以下の統計を自動的に修復する
  - 統計の保守可能性を判断する
  - 統計の使用状況を監視する
  - エンジンとクエリの実行に対しオーバヘッドは小さい

SYBASE TECHWAVE 2010

統計フラッシャ・プロセス

- 30 分ごとに実行されるデーモン・プロセス
- 使用されていない統計をメモリからアンロード
- カラム統計の健全性についてアドバイス
  - 通常、修復が必要、削除が必要、など
- カラム統計の使用状況についてアドバイス
- 統計を作成すべきか削除すべきかのアドバイス
- **エラー値を計算し、ヒストグラムの修復が必要かどうか判断**
  - 概算と実際の選択値の集計差異に基づく
  - 小さいクエリの選択値では若干エラー寄りに偏る

SYBASE TECHWAVE 2010

統計クリーナ・プロセス

- フラッシャ・プロセスによりトリガされるデーモン・プロセス
- 他の方法では修復できない統計を修復する
- 不良な統計が見つかったテーブルの ID を追跡する
- 修復対象とマークされた統計のみを修復する
- 一度に 1 テーブルずつスキャンし、そのテーブルの不良統計をすべて修復する
- バックグラウンドの優先順位 - エンジンがビジー、またはリソースを使いすぎた場合には一時停止し、後で再開する

SYBASE TECHWAVE 2010

統計の修復

- 統計精度を自動的に向上するために複数の方法が利用される
- ユーザ・クエリのピギーバック・オフ
  - テーブルの大きい部分を参照するアクセス・プランを利用
  - クエリの実行中にインライン統計収集を実行する
  - 置き換え、または元を修復
- インデックスから再作成
  - ピギーバックのフォールバック・メカニズム
  - *浅いインデックス・スキャンによってヒストグラムを再作成*

SYBASE TECHWAVE 2010

統計の修復

- *サンプリング後のテーブル・スキャンを実行*
  - テーブル・カラムにインデックスがない場合には、テーブルをスキャンして統計を取得する必要がある
  - 小数のテーブル・ページからランダムなサンプルを読み込む
- *問題のある状況を検出し、セルフ修復を防ぐか、ヒストグラムを破棄する*

SYBASE TECHWAVE 2010

結論

SYBASE TECHWAVE 2010

他のセルフ管理機能

- キャッシュ・ウォーミング
  - 起動時に、バッファ・プールにデータベース・ページを自動的にロードする。このページは前回の起動時に最初に要求されたページ
- セルフチューニングのパラレル・バックアップ
  - 並行処理の程度は自動的に検出される
- SQL Anywhere は自動化のイベントをサポート
  - スケジュールに基づいて起動、あるいは監査や監視、修正アクションを必要とする特定のサーバ・イベントが発生したときに起動

まとめ

- SQL Anywhere では、どの機能も可能なかぎりのセルフ管理を前提として設計されている
- セルフ管理機能は全体論的に考えねばならない
  - ジャストインタイムの最適化と適応実行戦略を両方用いなければ、動的なバッファ・プール・サイズもあまり意味がない
- 実装上は多くの課題：バランシング処理が必要
  - オーバヘッドを低く保つことが困難
  - 暴走による制御不能な動作を回避する
- フィードバック制御メカニズムを利用するセルフチューニングのチャンスは多い
  - メモリ割り当て、マルチプログラミング・レベル、クエリ内並行処理制御

SYBASE TECHWAVE 2010

THANK YOU

QUESTIONS?

Anil.Goel@sybase.com

SYBASE TECHWAVE 2010