

## 最新データベース パフォーマンス・チューニング

日本アイ・ビー・エム株式会社  
SW事業部SWテクニカルサポート  
データ・マネージメント技術部  
岡口 純子

2002年10月11日

IBM Software Group

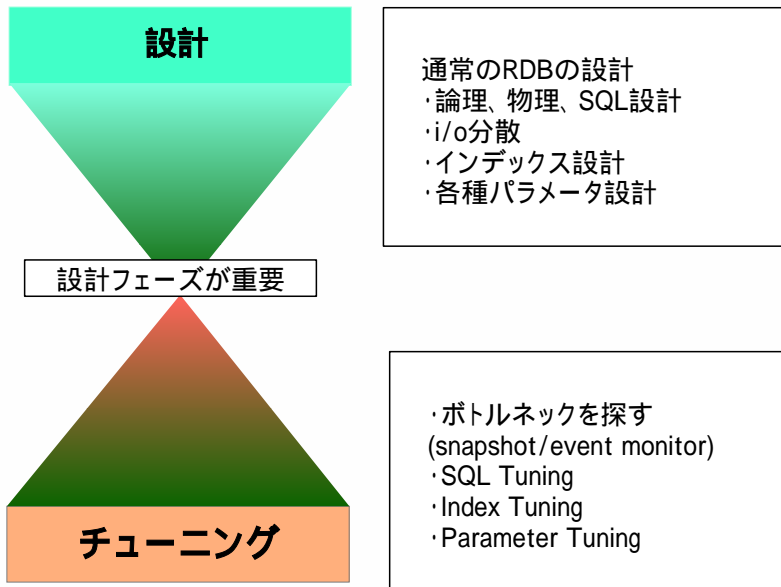
## 目次

- パフォーマンス・チューニングの一般的なポイント
  - ▶ 設計とチューニング
    - 物理設計
    - メモリー
    - CPU
  - ▶ INDEX、SQL、アクセスパス
- チューニング手順
- V8でのチューニング方法とツールの紹介
  - ▶ 構成アドバイザー
  - ▶ 設計アドバイザー
  - ▶ Storage管理ツール
  - ▶ パフォーマンスエキスパート

### お断り

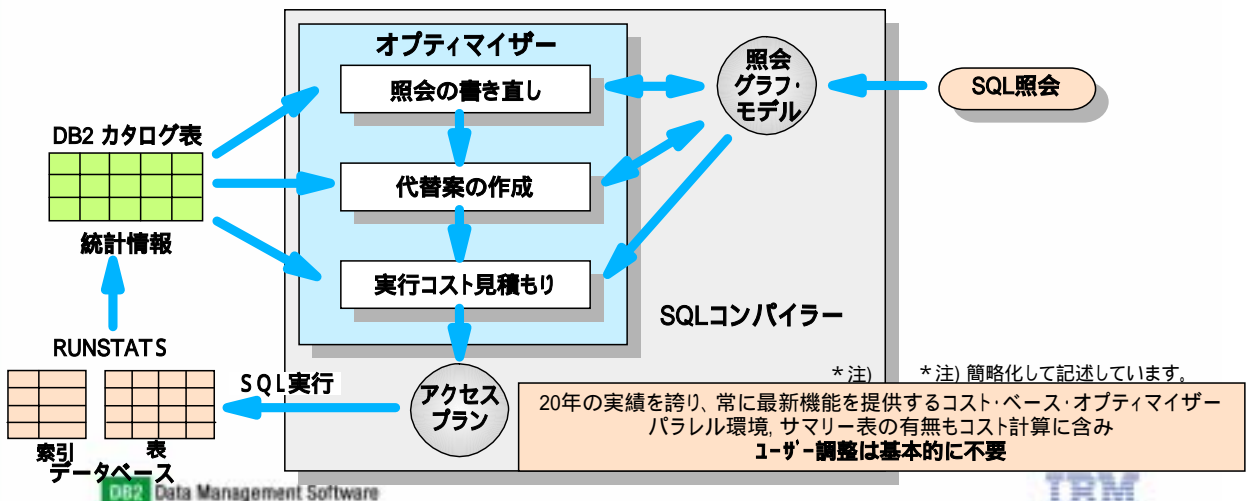
本資料に含まれている情報は正式なIBMのテストを受けていません。また、明記にしる、暗黙的にしる何ら保証もなしに配布されるものです。この情報の使用またはこれらの技術の実施は、いずれも使用先の責任において行われるべきであり、また、それらを評価し、実際に使用する環境に統合する使用先の判断に依存しています。

# パフォーマンスチューニング概要



# DB2 UDBの 옵ティマイザー

- DB2 UDBはコストベース・옵ティマイザーを実装。
  - ▶ 옵ティマイザーの目的
    - SQL照会の実行のために「最適な」アクセスパスを選択。
  - ▶ 最適アクセスパスの決定方法
    - 照会グラフ・モデル(構文解析、意味の検査、の結果)を入力として、多数の代替実行プランを生成。
    - 各代替プランの実行コストを見積もり、実行コストの見積もりが最も小さいプランを選択。
    - DB2カタログ統計情報に加えて、システム・リソース(CPU、ディスク、メモリーと通信速度)も考慮。



## DB2におけるパフォーマンスチューニングのポイント

1. Runstats !  
何はなくともRunstatsは必要。  
Runstatsユーティリティとは統計情報を収集するユーティリティでLOAD時STATISTICS YES指定で取得も可能。  
オプティマイザーの重要な情報源  
アクセスパスだけでなくデータ容量に合わせたHEAPSIZEなどの決定要因にもなる為必須。  
特にサービスイン直前のデータ移行作業項目で忘れないようにすること。
2. 物理設計
3. 構成パラメーター  
パフォーマンス構成ウィザード(初期設定時使用)
4. SQLチューニング  
Visual Explain  
インデックスアドバイザー

SQL文自体の修正を  
ともなうチューニング  
は最後

## ボトルネックの探索 ~ チューニング

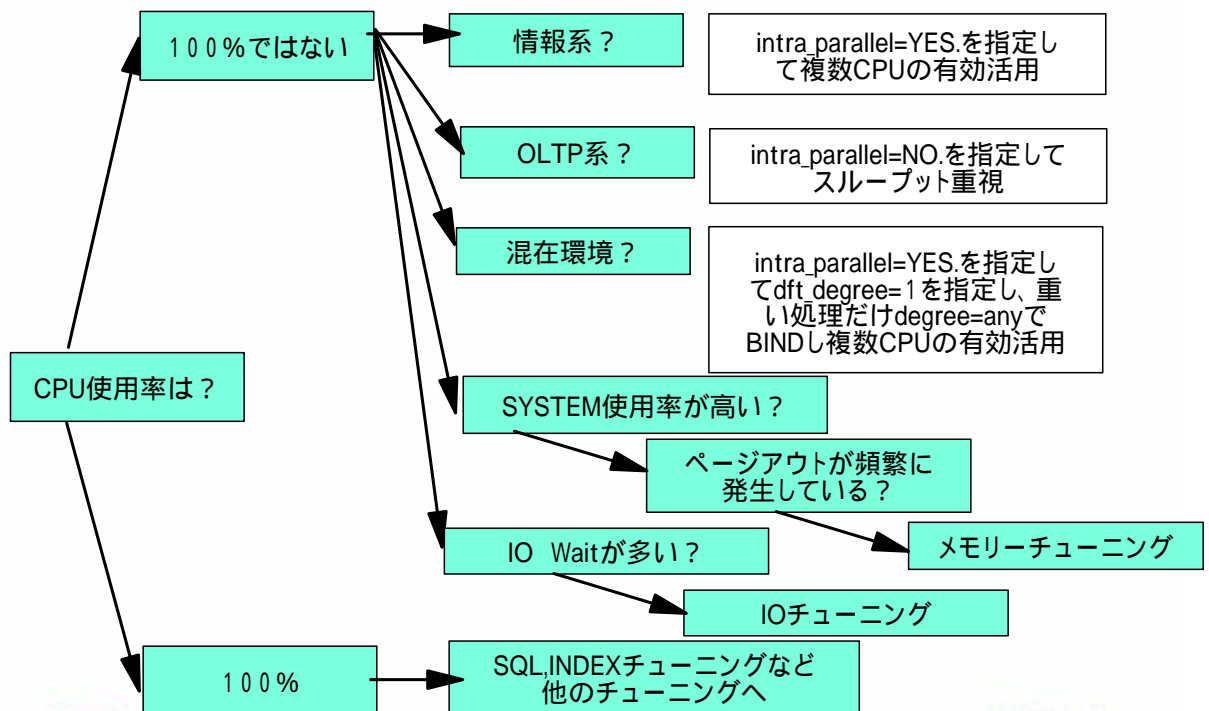
- ボトルネックの探索
  - ▶ バッチ処理や情報系の複雑なSQLを単体で実行している場合は、システム状況をモニターする。
    - CPU使用率は？
    - メモリーの使用率、ページングは？
    - IO待ちなどが発生していないか？
  - ▶ 個別SQLのチューニングが必要な場合、Event Monitorにて、処理されている全SQL(あるいは静的パッケージのセクション)の実行時間、CPU使用率などの情報を取得できる
  - ▶ Event Monitorで出力された内容をDB2にロードしてSQLで分析可能

PACKAGE	OPERATION	CPU	EXEC	COUNT
XXSEUP	Execute	293.718	4675.441	5312
XXSYUU	Execute	4.051	31.213	5364
XXSESE	Fetch	3.638	6.217	4028
XXKOUP	Execute	2.428	8.279	4205
XXIKKA	Close	1.661	2900.147	444
XXKICU	Close	0.335	3288.525	90

## ボトルネックの探索 ~ チューニング

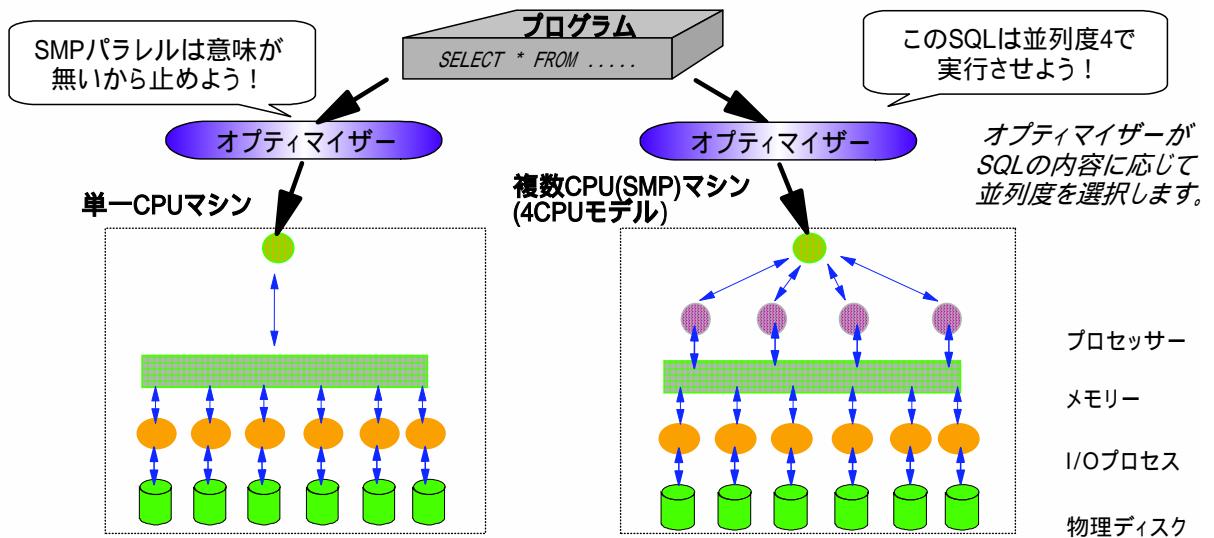
- 該当SQLのチューニング
  - ▶ アクセス・パスの検証
    - Visual Explain
    - db2exfmt
    - 簡易的なdb2expln(静的SQL用)/dynexpln(動的SQL用))
  - ▶ index使用の確認 => index定義(インデックスアドバイザー)
  - ▶ sortの確認 => DB構成パラメーター
  - ▶ db2exfmtではオプティマイザrewrite後のSQLの確認可能
- チューニング後に再度explainツールで確認する
  - ▶ 期待したアクセス・プランになっているか?
  - ▶ SQLコストが下がっているか?
- 期待どおりなら実行して検証

## チューニング・ステップ1 (CPU)



# ノード内 (SMP or 単一CPU) 並列処理の最適化

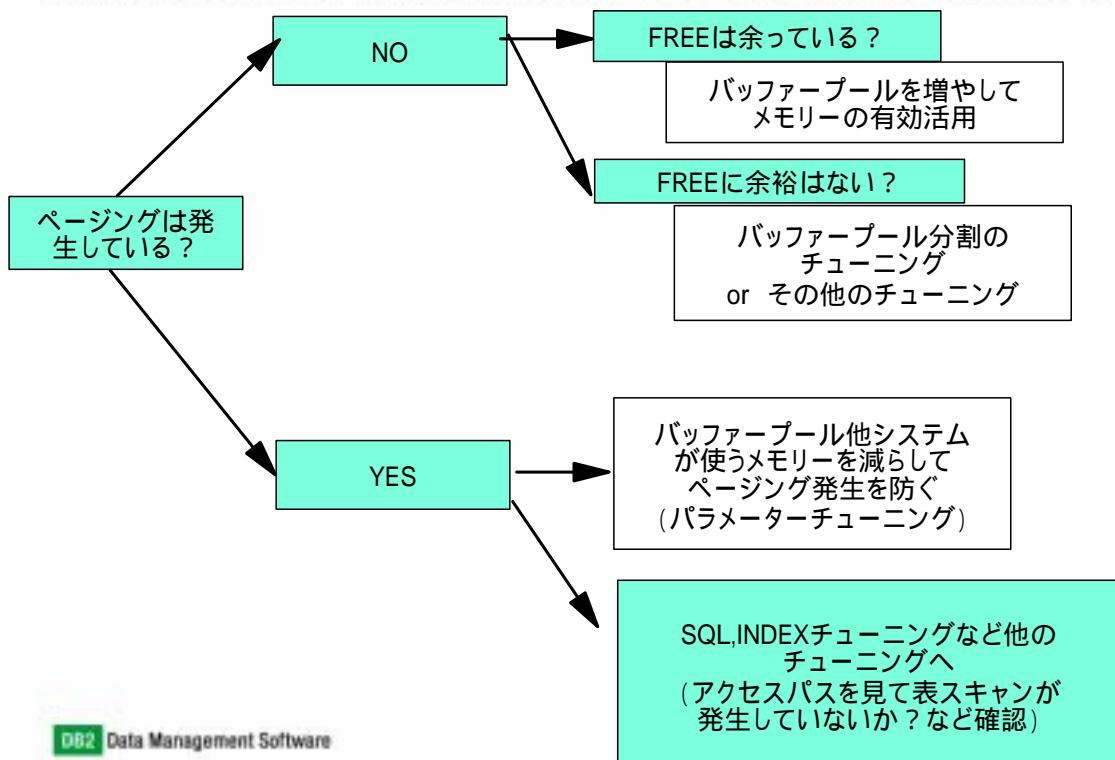
- オプティマイザーによりSQLの並列処理を自動的に最適化。



- SQL処理の並列度はオプティマイザーが自動的に設定します。
  - ▶ ユーザーによるSMPパラレル処理を意識した運用、コーディングが不要。
  - ▶ CPU追加による処理時間短縮を極めて容易に実現可能。

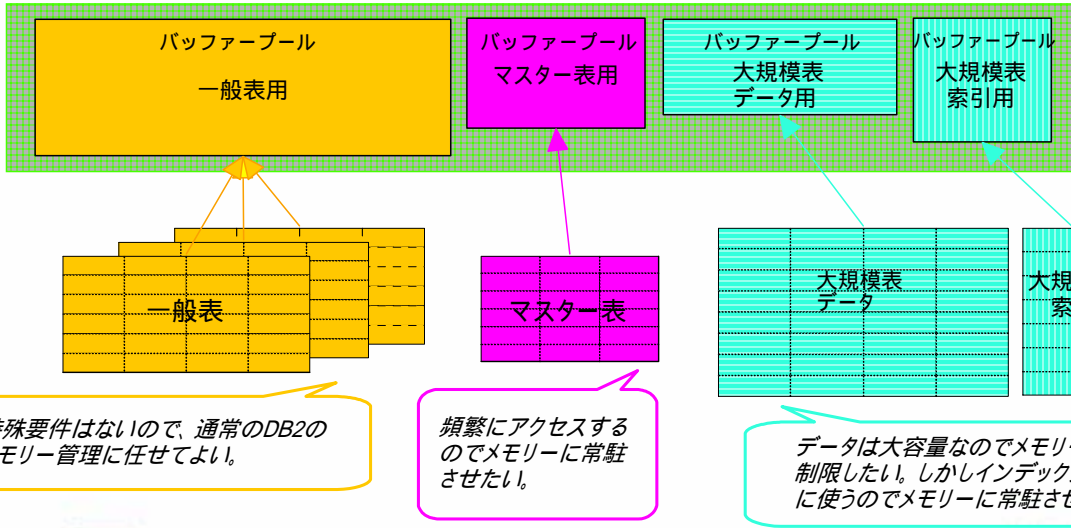


# チューニング・ステップ2 (メモリー)

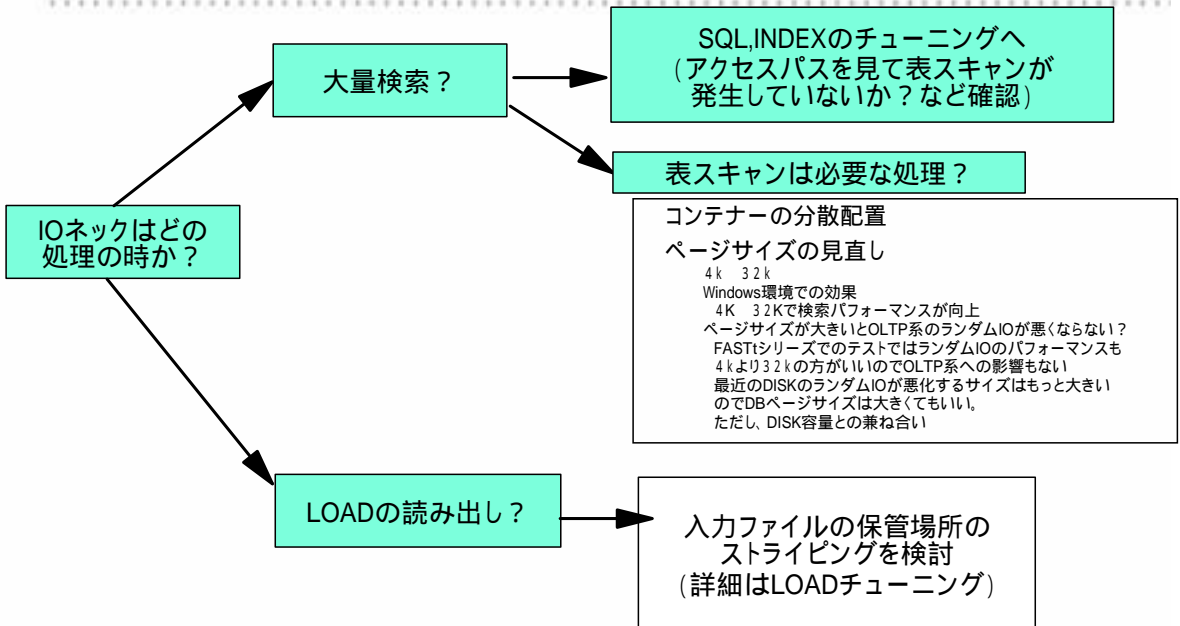


# バッファープール分割例

- バッファープールの効果的な利用によりハイパフォーマンスを実現。
  - ▶ バッファープールは、テーブル・スペースと関連づけて複数定義可能。
    - DB2 UDBでは、DB2自身で管理しているI/O Cacheメモリーをバッファープールと呼びます。
    - バッファープールは、テーブル・スペースと関連づけて複数定義できる(1:Nの関係)
    - 表および表アクセスの特徴に応じて効果的なI/O Cache処理を実現できます。



# チューニング・ステップ3 (DISK)



## DB2 UDBの並列I/O処理

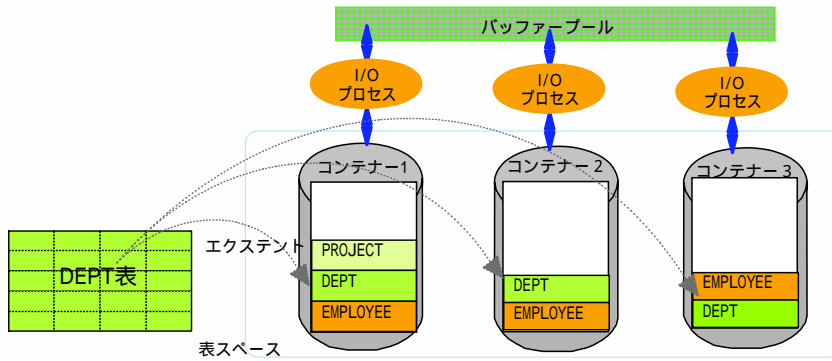
- 効率のよい並列I/O処理を行う為のデータ格納方法

### DB2 UDBによるストライピング

- ラウンド・ロビン方式でコンテナを均等に使用し、コンテナごとに並列にI/Oを行う

### 非同期データ読み込み

- 複数のI/O用プロセスが非同期に入出力



#### ページ:

1ページのサイズは 4KB, 8KB, 16K, 32K のいずれか

#### エクステント:

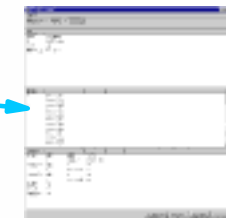
ページの集まり、デフォルト32ページ、表スペース単位で設定

#### コンテナ:

表スペースを構成する単位、表スペースのタイプによって、ディレクトリー、ファイル、デバイスとなる

## チューニング・ステップ4 (VisualExplain)

- アクセスパスの検証
  - ▶ Explain 表からのグラフィックに表示

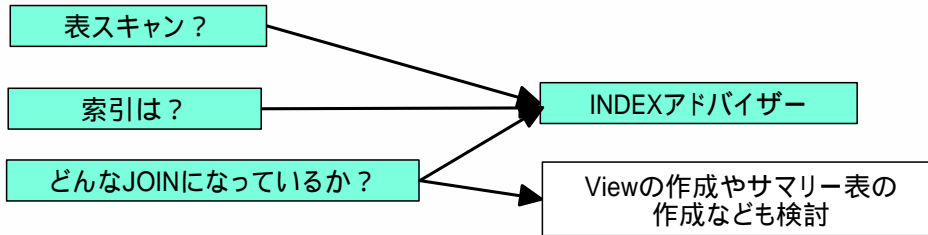


#### 詳細情報

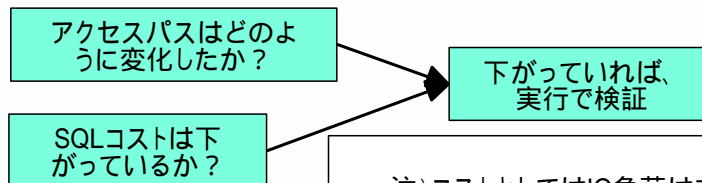
- ・CPUコスト
- ・I/Oコスト
- ・バッファプール使用ページ数
- ・結合タイプ など

## チューニング・ステップ4 (VisualExplain)

### ■ アクセスパスの検証



### 索引作成後



注)コストとしてはIO負荷は非常に高いものとして計算される  
 表SCANが非常に早い構成の場合、索引経由でない方が早いケースもある。  
 ただし、ロックや多重度の問題から情報系でユーザー数が少ないケースを除いて表スキャンは避ける

## チューニング・ステップ5 (INDEXアドバイザー)

### ■ 有効な索引がないか？

- ▶ SQLが判っている場合
  - SQL文と制限を入力に推奨索引を確認
  - 推奨索引が出たら有効かどうか判断して作成
  - 推奨なし、または、推奨索引が更新処理や他への影響を考慮して作成不可 他のチューニング 注) 特定SQLしか入力しないと他への影響が判断できない為、有効な索引かどうか人手の判断加味が必要
- ▶ 動的SQLのトランザクション全体
  - パッケージキャッシュと制限を入力にアドバイザー利用
  - 推奨索引だけでなく、不要や統合対象の索引も含め検討





## チューニング・ステップ6 (スナップショット)

- スナップショットとは特定時点の状況を照会
- 取得手順
  - ▶ 取得したい対象のMONITOR SWITCHESの更新
    - BUFFERPOOL (バッファープールのヒット率など)
    - LOCK (各アプリケーションの具体的なロック取得情報)
    - SORT
    - STATEMENT
    - TABLE
    - UOW
  - ▶ 必要なスイッチをONにした後、get snapshot for xxx (取得対象) で取得
- 実行中のDB2のリソース使用状況・内部情報などを取得
  - ▶ シェルなどでSleepコマンドと合わせて使用することにより、パフォーマンスの確認も可能。(差分で確認)

## チューニング・ステップ7 (イベントモニター)

- イベントモニター (長時間にわたって特定の活動状態を照会)
  - ▶ イベントタイプ:
    - データベース
    - 接続
    - テーブル
    - ステートメント
    - トランザクション
    - デッドロック
    - 表スペース
  - ▶ イベント出力先:
    - create event monitor event名 for xxx write to file yyyy
  - ▶ 出力結果の調査:
    - db2eva (イベントアナライザー GUIツール)
    - db2evmonツール

## ユーティリティー・チューニング (LOAD)

- LOADユーティリティーを早くするには？
  - ▶ INPUT ファイルの読み出しを早くする
    - DISKのストライピング
      - UNIX環境ではストライピングするDISK個数を多く、IOを分散させる
      - Windows環境では早いDISK又は、書き込みとは別系列に分散させる
  - ▶ SORTサイズを大きく取得
    - 索引作成時、索引毎に同時に取得されるので、とれるだけ大きくする
      - ただし、とれなければ半分づつとれるまで自動的に小さくなるので注意
  - ▶ バッファプールサイズを大きく取得
    - INDEXデータをLOADしながら保存している為、物理IOを減らすのに効果的
    - それほどBPが取得できない場合は一時表スペースも複数分散しておく
  - ▶ CPUの並列処理はオプティマイザーにまかせる。(CPUが振り切れていない場合手動で試行錯誤してみるのも手だが、有効かどうかは？)
  - ▶ 物理設計上、LOAD入力ファイルとコンテナ配置などそのDISK構成にあったものをテストして取り入れていく

## DB2 UDB V8新機能 (パフォーマンス・チューニングに関わるもの)

- プラットフォームとクライアント・サポート
  - ▶ 64ビットサポート
- ビジネス インテリジェンス サポート
- オンライン保守の拡張
  - ▶ オンラインユーティリティー
  - ▶ パラメーター動的変更
  - ▶ バッファプール動的変更
- データベース管理の拡張
- コントロールセンターとツールの拡張
- SQLの拡張
- 開発機能拡張

## 少ない管理コストでTCO削減 - SMARTプロジェクト

### □ SMART(Self Management & Self Tuning)プロジェクト:

- ★(自己管理(Self-Management)と資源最適化(Resource Tuning) によるDB管理コストの削減を目指した取り組みです



"eLiza" プロジェクトのソフトウェア版!



#### SMARTの範囲:

- データベース構成
- データベース管理
- パフォーマンス自己チューニング
- 問題判別と問題解決



DB2 Data Management Software

IBM

## DB2 UDB V8ではどう変わるか?

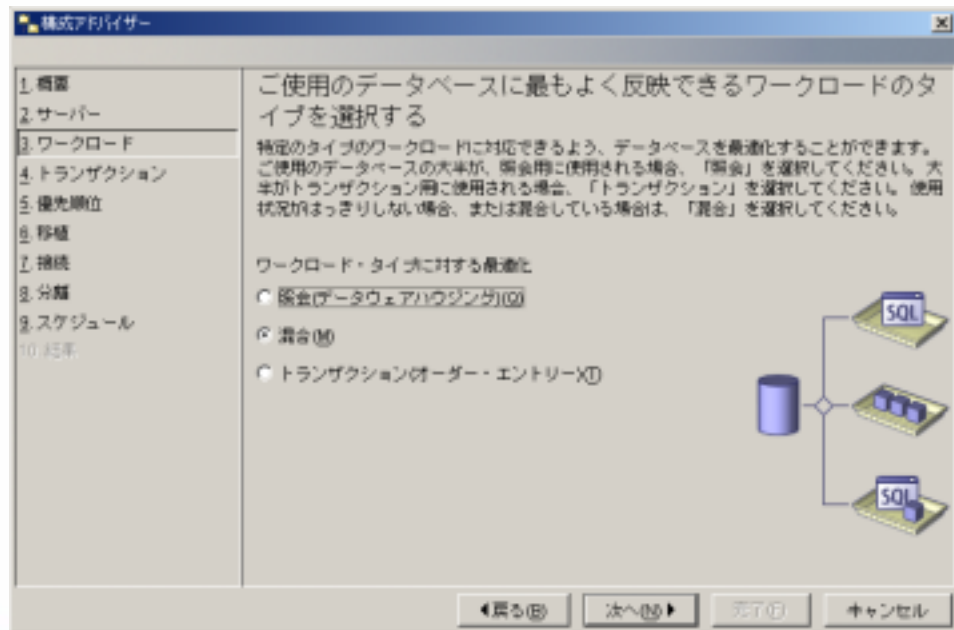
- 1. 64 Bitによるメモリーチューニングの柔軟性
  - ▶ SharedMemoryの制約からの解放 物理メモリーの有効活用
  - ▶ BufferPoolの複数可、大きくBPなど柔軟なチューニングの対象となる
  - ▶ オンラインでのBP変更、パラメーター変更が柔軟にできる
- 2. 各種アドバイザーツール充実によるチューニング作業の軽減
  - ▶ 構成アドバイザー
    - DB2エキスパートでなくても90%以上のチューニングが可能
  - ▶ 設計アドバイザー
    - Indexアドバイザーの後継さらに推奨してくれる対象が増えた
  - ▶ Storage管理ツール
    - Storage管理セットアップランチパッド
      - タスクセンターにSnapshotの設定、スケジューリングができる
      - Storage管理ビュー
    - Snapshotの分析がグラフィカルに表示ができる
  - ▶ パフォーマンスエキスパート
    - イベントモニター結果のグラフィカルな表示、ソートができる
    - 時系列データの格納、過去に遡った分析が可能

DB2 Data Management Software

IBM

## 構成アドバイザー

- 質問に答えるとDB構成パラメーターの推奨値を提示、更新コマンド、スケジュール実行も可。

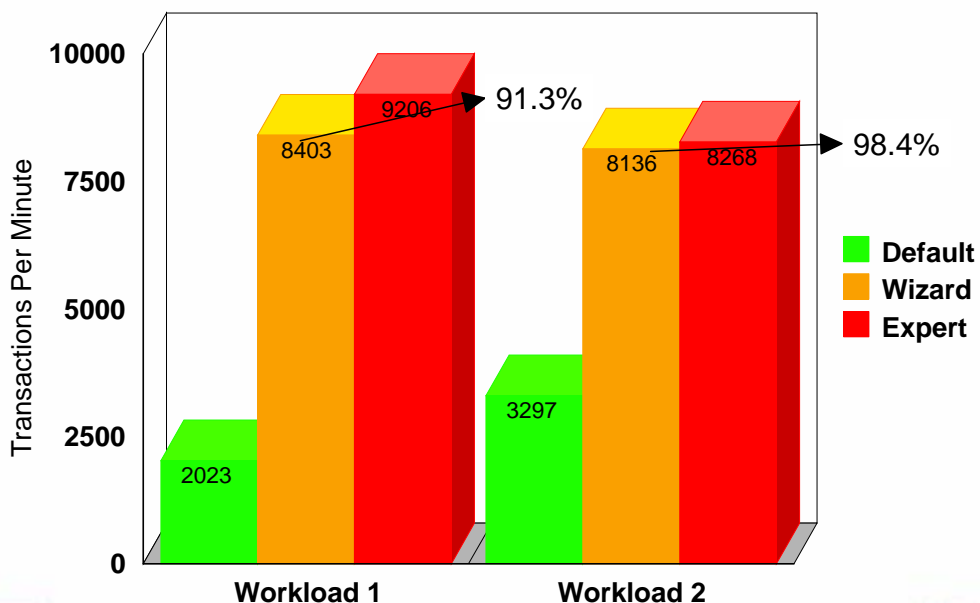


DB2



## 構成アドバイザー

- 構成アドバイザーによるチューニング効果
  - ▶ エキスパートによるチューニングの90%以上のスループットを実現



DB2 Data Management Software

IBM  
for a specific OLTP benchmark

## 設計アドバイザー

- 設計アドバイザーは、INDEXアドバイザーの機能を拡張。
- SQLワークロード、データベース構成、ディスクスペースなどから推奨される索引、MQT(マテリアライズ照会表)、テーブルのパーティショニングをアドバイス。また、既に作成されていて使用されていないものも表示。
- 推奨されたものをその場で作成したり、タスク・センターへ登録し、後で実行することが可能。



IBM

## ストレージ管理ツール

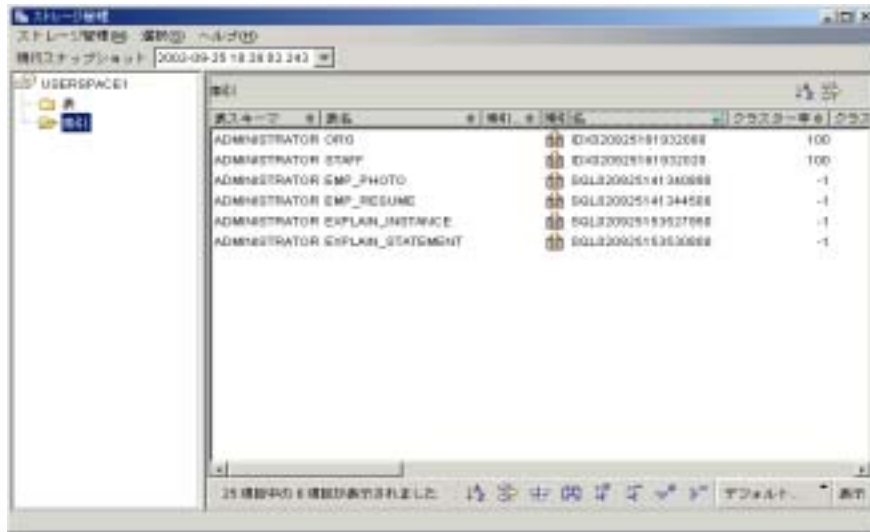
- ストレージ管理ツールは、特定のデータベースまたはデータベース・パーティションのストレージを長期間に渡って管理。各ノードのデータ配分や、ストレージ履歴を表示。
- 「ストレージ管理セットアップ」ランチパッド
  - ▶ ストレージ管理ツールを使用するためのセットアップ処理を行う
  - ▶ 管理対象、しきい値、スナップショットの設定が可能
  - ▶ スナップショットは、タスク・センターに登録して実行することも可能



## ストレージ管理ツール

### ■「ストレージ管理」ビュー

- ▶ストレージ管理ツールのグラフィカル・インターフェース
- ▶取得した**スナップショット**をもとに、各ノードのデータ配分、索引のクラスター率などをグラフィカルに表示
- ▶設定したしきい値を超えると警告アイコンを表示
- ▶スナップショットの履歴を管理し、データ量の推移などを見ることが可能



ユーザー名	表名	索引	場所	クラスター率
ADMINISTRATOR_ORIG			01:02:00025181932000	100
ADMINISTRATOR_STAFF			01:02:00025181932000	100
ADMINISTRATOR_EMP_PHOTO			01:02:00025181340000	-1
ADMINISTRATOR_EMP_RESUME			01:02:00025181344000	-1
ADMINISTRATOR_EXPLAN_INSTANCE			01:02:00025183927000	-1
ADMINISTRATOR_EXPLAN_STATEMENT			01:02:00025183930000	-1

DB2 Data Management Software



## DB2 パフォーマンス・エキスパート



### パフォーマンス管理ツール

- 全てのDB2インスタンスを集中管理
- 複数のDB2を並行モニタリング
- 以下の内容をモニター
  - ▶システム統計 - アプリケーション-End2End-ボトルネック(ロック競合、ユーザー定義の基準からの逸脱)
- 重要なパフォーマンス・データをグラフィカルに表示
- 現在の動作状況と履歴情報を見ることが可能
- 各種プラットフォームで共通のユーザー・インターフェース
- Staged delivery
  - ▶Stage3(2002年12月)で追加される機能(抜粋)
  - ▶パフォーマンス・データを選択的にパフォーマンス・ウェアハウスに保存・管理することが可能
  - ▶スループットを最適化するためのシステム・チューニングの推奨値を提供
  - ▶イベント・レポート/エキスパート・レポートを提供
  - ▶BufferPoolAnalyzer



DB2 Data Management Software





## DB2 パフォーマンス・エキスパート

- 実行SQLのモニターおよびチューニング対象アプリケーションの洗い出しが可能
- 履歴情報の保管による過去に遡った分析が可能

The screenshot shows two windows from the DB2 Performance Expert tool. The top window, 'Application Monitor', displays a table of application details. The bottom window, 'Dynamic SQL Statistics', shows a table of SQL execution statistics.

**Application Monitor Table:**

Application Name	Application Handle	Application ID	Application Status	Status Change Time	User CPU Time used by agent(s)	System CPU Time used by agent(s)
java.db2	8	*LOCAL.DB2...	UOFF waiting	03/10/01 18:28:04	0.99078	0.10014
java.db2	88	*LOCAL.DB2...	UOFF waiting	03/10/01 18:28:06	0.99300	0.11016
java.db2	83	*LOCAL.DB2...	connect completed	03/10/01 18:27:45	N/A	N/A

**Dynamic SQL Statistics Table:**

DB Name	Statement	Execution	Elapsed time	SQL	SQL per sec	Wait time	Wait per sec	Execs	CPU per stmt
...	...	...	...	...	...	...	...	...	...

**Callouts:**

- Top window: CPU時間、処理時間でのソート表示が可能
- Bottom window: 実行SQL文の表示

DB2 Data Management Software

## まとめ

### ■ V7でチューニング時の課題点

- ▶ メモリー・チューニング
  - 32 BitではSharedMemoryの制約
    - AIX 1.75GB
    - HP 1GB
    - SUN 3.35GB
  - 物理メモリーがあってもバッファプールサイズを大きくしたり、分割するには限界あった。
- ▶ GUIツールは64ビットインスタンスでは使用不可
  - Java, Windowsなどのクライアントは32ビットしかサポートがなく、64ビットインスタンスに接続できない為

### ■ V8での改良点

- ▶ 64ビットフルサポート
  - SharedMemoryからの開放
  - 32ビットクライアントからのアクセス



柔軟なバッファプールサイズ、個数などメモリーチューニングの柔軟性  
どんなインスタンスでもGUIツールの利用が可能

## まとめ

### ■ V7でチューニング時の課題点

- ▶ パフォーマンス構成ウィザードやINDEXアドバイザーの効果、適用範囲
  - 入力条件不足により精度・効果が上がらないケースがあった
  - 適用範囲が索引に限定されていた
  - EEEでは使えない

- ▶ スナップショット結果の見方にW/Lがかかる

- ▶ イベントモニター結果の見方にW/Lがかかる

### ■ V8での改良点

- ▶ 構成アドバイザーの精度向上
- ▶ 設計アドバイザーの対象増加
  - マテリアライズ照会表サポート
  - 区分のサポート

エキスパートによるチューニングと遜色のない効果がえら得る

- ▶ ストレージ管理ツールによるスナップショットのグラフィカル表示

- ▶ パフォーマンスエキスパートによるイベントモニターのグラフィカル表示

分析作業の生産性向上

## 参考資料

- 下記URLにパフォーマンス・チューニング、物理設計に有用な資料がありますので、参考にしてください。

- ▶ カンタン！DB2テクテク第1歩

- <http://www-6.ibm.com/jp/software/data/developer/library/techdoc/kantandb2.html>

- パフォーマンス・チューニング編
- スナップショット・モニター編
- Explainアクセスプラン編

- ▶ DB2 Day2001の資料

- <http://www-6.ibm.com/jp/software/data/library/db2day/trackdoc.html>

- 意外とカンタン！？DB2パフォーマンス・チューニングの世界

- ▶ OLTP アプリケーションのためのDB2(R)のチューニング・ヒント

- <http://www-6.ibm.com/jp/software/data/developer/library/techdoc/tuninghints.html>

- ▶ DB2 UDB (PC&UNIX)V7デザイン・ガイド

- <http://www-6.ibm.com/jp/software/data/db2univ/lab/ise/techlib/>

- ▶ DB2 UDB (PC&UNIX)V7運用管理ガイド

- <http://www-6.ibm.com/jp/software/data/db2univ/lab/ise/techlib/>

- パフォーマンス・モニタリング