

富士通 SPARC Enterprise による
Oracle Database 11g データ・ウェアハウス検証
～データ・セグメント圧縮と ASM を活用した性能向上～

Creation Date: May 14, 2008

Last Update: June 20, 2008

Version: 1.2

ORACLE

FUJITSU

THE POSSIBILITIES ARE INFINITE

A.はじめに

日本オラクル株式会社（以降、日本オラクル）と富士通株式会社（以降、富士通）は 1989 年に OEM 契約を締結して以来、お客様に安心してご利用いただけるソリューションを提供するため、システム構築、共同検証、導入後のサポートなど、様々なアライアンス活動を行ってまいりました。

日本オラクルは 2006 年 11 月、企業のシステム基盤の最適化を実現する次世代のビジネス・ソリューションを構築するため、先鋭の技術を集結した「Oracle GRID Center（オラクル・グリッド・センター）」（http://www.oracle.co.jp/solutions/grid_center/index.html）を開設しました。富士通は Oracle GRID Center 開設に賛同し、富士通のもつサーバ、ストレージ製品を使用して Oracle GRID Center にて共同で技術検証を行っております。

このたび、両社は富士通の最新の UNIX サーバ、SPARC Enterprise シリーズを使用し、データ・ウェアハウスシステムでの性能改善手法として、Oracle Database 11g のデータ・セグメント圧縮機能と、Automatic Storage Management のリバランス機能の有効性を Oracle GRID Center にて検証しました。その成果をここに報告致します。

B.目次

A. はじめに	2
B. 目次.....	3
C. 検証目的	4
D. 富士通 SPARC Enterprise.....	5
1. SPARC Enterprise M5000.....	5
2. SPARC Enterprise T2000.....	6
3. ストレージシステム ETERNUS (エターナス)	7
3.1. ETERNUS4000 モデル 500.....	8
E. 本検証で使用了 Oracle Database の機能紹介	9
1. データ・セグメント圧縮	9
2. パラレルクエリー	10
3. Automatic Storage Management	10
4. 外部表.....	10
F. 検証環境.....	11
1. システム構成.....	11
1.1. データベースサーバ.....	12
1.2. ストレージ.....	12
1.3. クライアント.....	12
1.4. FC スイッチ.....	12
2. ストレージ構成	13
3. スキーマ構成.....	15
3.1. スキーマ構成概要.....	15
3.2. 検証クエリ.....	16
G. 検証内容	17
1. 表へのデータロード	17
2. クエリの実行	17
3. データ・セグメント圧縮	17
4. ASMによる動的なディスク追加およびデータのリバランス	17
H. 検証結果.....	18
1. データ・セグメント圧縮	18
1.1. 圧縮によるセグメントサイズの差異.....	18
1.2. 表へのデータロード.....	21
1.3. クエリの実行.....	24
1.4. セグメント圧縮によるクエリ実行時のオーバーヘッド.....	26
1.5. データ・セグメント圧縮総括	27
2. ASMによる動的なディスク追加およびデータのリバランス	28
2.1. ASM ディスク追加の効果	28
2.2. ASM のリバランスコスト	31
2.3. ASM による動的なディスク追加およびデータのリバランス総括	31
3. データ・セグメント圧縮とディスク追加の併用.....	33
I. 総括.....	35
J. 参考情報	36

C.検証目的

本検証は、富士通の最新 UNIX サーバである SPARC Enterprise を用い、Oracle Database 11g の持つデータ・セグメントの圧縮機能と、Automatic Storage Management（以降 ASM）による動的なディスク追加およびデータのリバランスが、データ・ウェアハウス（以降 DWH）システムのパフォーマンス改善手法として有効であることを証明するために行われました。

Oracle 9i Database Release 2 より提供されているデータ・セグメント圧縮機能は、DWH 環境における膨大なデータを圧縮し、データ量を減らすことでストレージの有効活用が期待できる機能です。また、圧縮によりデータ容量が縮小することで、一般的なデータベース処理の中で最もコストが高いディスク I/O も減少し、クエリ性能の改善も期待できます。一方で、圧縮にかかる CPU オーバーヘッドの影響も考えられます。データ・セグメント圧縮検証では、圧縮の CPU オーバーヘッドによる性能への影響と、ディスク I/O の削減によるクエリ性能の変化を確認します。

データベースへのディスク追加は、増加するデータ量に対応することと、追加したディスクに対してデータを分散させパフォーマンスの向上を図るという2つの目的があります。ディスク追加作業は、データの退避、ディスクの追加／構成、データの再ロード、索引の再作成、統計情報の再収集など、手順は複雑で運用コストも高く、また、業務データベースを停止せずに実施することは困難です。Oracle Database のディスク管理層である ASM は、業務データベースを停止することなく、簡単な手順で格納されているデータを追加したディスクに対して再分散（リバランス）させることが可能です。ASM リバランス検証では、ディスクの追加、リバランスを行うことで、クエリ性能が改善されることを確認すると共に、ディスク追加時に発生する工数の削減という観点からも検証を実施し、ASM リバランス機能の有効性を確認します。

検証では DWH の大量データのロード、複雑で高負荷なクエリを繰り返し長期にわたり実施します。システムは常に高負荷な状態が続くため、本検証は、SPARC Enterprise と Oracle Database 11g の組み合わせに対するストレステストとしての意味も持ちます。SPARC Enterprise は、これまで PRIMEPOWER で培われてきた Oracle Database と富士通の UNIX プラットフォームの親和性・信頼性を受け継いでおり、今回の検証を問題なく終えることは、最新の Oracle Database 11g の組み合わせにおいても、引き続き高い親和性・信頼性をもつことの証明になります。

D.富士通 SPARC Enterprise

現代の企業の情報システムは、サーバ 3 階層（データベースサーバ、アプリケーションサーバ、Web サーバ）を基本として、複数サーバをそれぞれ役割分担して配置した、多階層モデルが主流となっています。

SPARC Enterprise は、このようなお客様の要求に対応し、Web フロントやバックエンドなど、サーバを役割ごとに配置する現代の企業システムに最適化した UNIX サーバとして開発されました。SPARC Enterprise は、富士通の「PRIMEPOWER」とサンマイクロシステムズ社「Sun Fire」の技術を融合して開発した、新しい UNIX サーバのスタンダードです。富士通のメインフレーム開発で培った高信頼・高可用技術と、サンマイクロシステムズ社の優れたネットワークコンピューティング技術のさらなる強化により、優れた性能・業務継続性・仮想化機能を提供します。

SPARC Enterprise では最新の SPARC プロセッサ「SPARC64 VI」「UltraSPARC T1」「UltraSPARC T2」を搭載しており、オペレーティングシステムには世界標準の UNIX OS である「Solaris Operating System」を採用しています。従来の Solaris バージョンとのバイナリ互換を確保し、PRIMEPOWER および Sun Fire から SPARC Enterprise へシームレスに移行することができます。

スピード経営、業務の継続性、TCO 削減等、企業が抱える様々な経営課題に応えるために、SPARC Enterprise は大規模データベースや基幹業務等、バックエンドに適した M4000, 5000, 8000, 9000 から、Web フロントに適した T1000, T2000, T5120, T5220 まで、業務に最適化したラインナップをご用意しています。

1.SPARC Enterprise M5000

SPARC Enterprise M5000 は、従来のハイエンドサーバでサポートされていた高性能・高信頼・仮想化技術等の機能を、ミッドレンジクラスに凝縮したサーバです。メインフレームの信頼性を継承し、さらに性能向上した「SPARC64 VI」プロセッサを採用しており、1 台あたり最大 16 コア、32 のスレッドを実行することができます。

データベースやバッチ処理等のバックエンド業務は、1 つのトランザクション当たりの負荷が大きく、トランザクションを 1 つずつ処理するという特徴があります。そのため、負荷の大きいトランザクションを高速に処理実行するために、新たに高性能プロセッサ「SPARC64 VI」を開発しました。強力な命令並

列処理能力や高精度の命令分岐予測等の技術に加え、新たにマルチコア、マルチスレッド技術を採用することにより、高性能を実現しています。さらに、システムバスの強化や、最新 I/O インターフェースを採用するなど、システム全体で高性能化が図られています。

また、データベースやバッチ処理といった業務は、停止してしまうとビジネスに大きな影響を与えてしまいます。SPARC Enterprise は、メインフレームの設計思想をもとに開発しています。サーバの故障や不具合がお客様業務を止めることのないよう、エラーの発生を未然に防止する機能、トラブルが起こった際にも訂正や縮退を行い稼働し続ける機能、コンポーネント冗長化・活性交換機能など、システムダウンを最小限に抑えるテクノロジーを多数取り入れています。

日中・夜間・月末・月初など、時間によってサーバへの負荷を与える業務は異なります。従来は、業務負荷のピークにあわせて個別にサーバを用意していました。SPARC Enterprise M5000 ではパーティショニング機能や Dynamic Reconfiguration 機能を使って筐体内のリソースを分割し動的に再構成することで、必要な資源を必要な時に追加・削除することができます。これにより、このような変化のある業務に柔軟に対応することができます。

【SPARC Enterprise M5000 の特徴】

- ・メインフレームの信頼性を継承し、さらに性能向上した「SPARC64 VI」プロセッサを採用
- ・マルチコア・マルチスレッド対応により、プロセッサあたり 2 コア、4 スレッドを実現
- ・バス帯域幅の強化や最新 I/O インターフェースの採用により高性能を実現
- ・徹底的なデータ保護・冗長化により、装置単体の可用性を向上
- ・パーティショニング機能や DR 機能、COD 機能により、フレキシブルなサーバ運用が可能

2.SPARC Enterprise T2000

SPARC Enterprise T2000 は、プロセッサあたり 8 コア、32 スレッド実行可能な UltraSPARC T1 プロセッサを搭載しています。

Web フロント業務は、負荷の小さいトランザクションを 1 度に複数処理するという特徴があります。そのため、これらの業務を行うサーバには、数多くの

トランザクションをスピーディーに処理することが可能なスループット性能が求められます。

数多くのトランザクションを処理するために、高スループットプロセッサ「UltraSPARC T1」を開発しました。最大 32 スレッドで効率よく命令処理を行うことができるので、Web フロント業務のような、複数処理を並列で実行する用途に最適です。

また、SPARC Enterprise T2000 は、省スペースによるフロアスペースの縮小、省電力による電気代削減など、ランニングコストの低減を可能にします。従来までの UNIX サーバより高密度なシステムを構築することが可能になったことで、スペースあたり・消費電力あたりの処理性能を向上しています。同等クラスの他社製 UNIX サーバと比べると、消費電力で 1/2、スペースで 1/4 という、圧倒的な省電力、省スペースを実現しています。また、Solaris コンテナを利用して業務を集約することで、増大するサーバを集約し、TCO 削減を実現します。

【SPARC Enterprise T2000 の特徴】

- ・ マルチコアプロセッサ「UltraSPARC T1」を搭載
- ・ 省スペース (2U)、省電力なエコサーバ
- ・ PCI Express/SAS ディスク等、最新 I/O インターフェースを採用
- ・ ディスク、電源、ファンは冗長構成・活性交換が可能
- ・ LAN 経由で遠隔地からサーバの監視・制御が可能

3.ストレージシステム ETERNUS (エターナス)

富士通は、SAN 対応ディスクアレイとして、2.04PB (ペタバイト) の世界最大の容量を実現したエンタープライズディスクアレイ「ETERNUS8000」と、コストパフォーマンスに優れたミッドレンジディスクアレイ「ETERNUS4000」、省スペース・省電力・静音設計なエントリーディスクアレイ「ETERNUS2000」を提供し、幅広い製品ラインナップであらゆるニーズに応えます。

部品レベルからシステムレベルまで徹底した信頼性を追求し、重要なデータを確実に保管します。またマルチプラットフォーム環境に対応する、優れた接続性を備え、SAN を利用したストレージ統合を実現。さらに、オンラインの高速バックアップを実現するアドバンスト・コピー機能を採用し、バックアップに要する時間を大幅に削減します。

企業の重要なデータを確実かつ効率的に保管し、コンプライアンス対応の強化

など、企業の抱える様々な課題に対して最適なストレージソリューションを提供します。

3.1.ETERNUS4000 モデル 500

ミッドレンジディスクアレイ「ETERNUS4000 モデル 500」は、2GHz の高速デュアルコアプロセッサと 4Gbps のファイバチャネル・インターフェースを採用し、高い処理能力を実現。また、最大 313TB の優れた拡張性を備え、大規模なストレージ統合やデータ量の増加にも対応します。

コントローラーや電源、ファン、バッテリー等の主要コンポーネントを冗長化しており、停電発生時にも内蔵バッテリーでキャッシュ内のデータをディスクに退避させ、保持日数の制限なく保護します。さらに、RAID グループ内にパリティディスクを2つ持つ RAID6 を利用可能で、二重ディスク障害時にもデータを保護。高信頼設計で 24 時間 365 日無停止稼動をサポートします。

装置のアドバンスト・コピー機能とソフトウェア「ETERNUS SF AdvancedCopy Manager」との連携により、業務運用中に高速コピー/バックアップを行うことができます。バックアップデータは複数世代持つことができ、バックアップの世代管理に対応。さらに、ディスクアレイ間での遠隔地へのデータ転送も可能で、万が一の災害時にもデータを保護します。

【ETERNUS4000 モデル 500 の特長】

- ・ 最大 313TB の優れた拡張性
- ・ 2GHz の高速デュアルコアプロセッサと 4Gbps のファイバチャネル・インターフェースで高い処理能力を実現
- ・ 24 時間 365 日無停止稼動に対応する高信頼設計
- ・ アドバンスト・コピー機能で業務運用中に高速バックアップが可能
- ・ ディスクアレイ間で遠隔地へデータ転送が可能で、災害時にもデータを保護

E.本検証で使用した Oracle Database の機能紹介

本検証にて主に使用した Oracle Database の機能について説明します。

1.データ・セグメント圧縮

データ・セグメント圧縮機能は、データブロック内に存在する重複値を排除することでセグメントを圧縮する機能です。ブロック内の重複値はブロックの先頭にあるシンボル表へ格納され、各レコード内の重複していた値は、シンボル表内の実値を参照するサイズの小さなポインタに置き換わります。これにより、レコード長が短くなり、1つのブロックに格納されるレコード数が増加するため、データ・セグメントが圧縮されます。つまり本機能によりデータを圧縮することで、DWH環境でボトルネックとなりがちなディスク I/O を削減し、その結果クエリレスポンスの改善が期待できます。

データ・セグメント圧縮は、Oracle Database 10g Release2 までは主に DWH 向けとして、データのダイレクトロード時に圧縮を実行する機能でした。しかし Oracle Database 11g Enterprise Edition のオプション機能である Oracle Advanced Compression によって、通常の DML 文でも圧縮されるよう機能拡張が行われました。これにより、データのロード方法を問わずに、圧縮のメリットを享受できるようになりました。

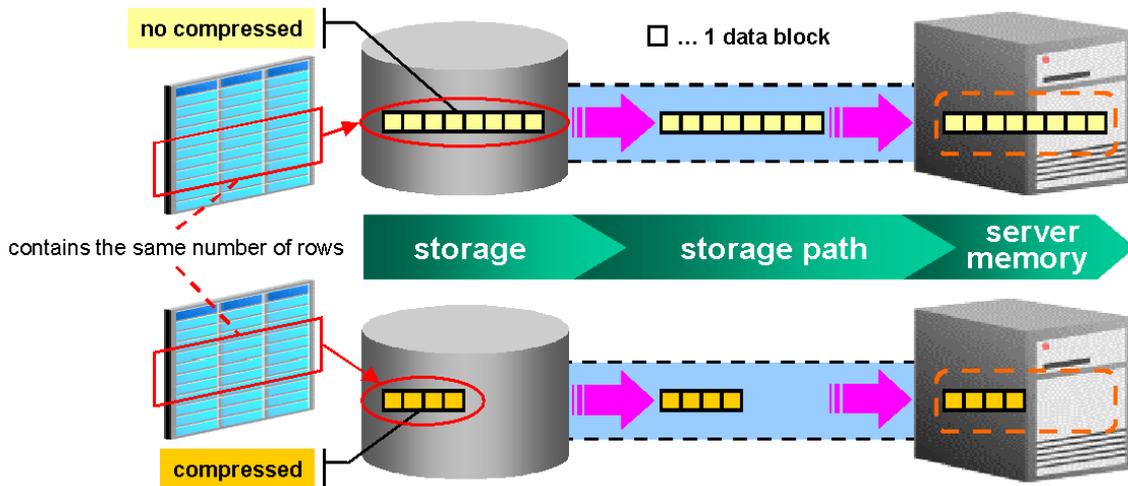


図 1データ・セグメント圧縮

2.パラレルクエリー

パラレルクエリーは1つの SQL の実行要求を複数のプロセスによって並列に実行する機能です。表のフルスキャンや索引のフルスキャン、ソート処理などを並列して実行できるため、大規模な表にアクセスする DWH 環境には特に有効です。近年のサーバマシンは搭載される CPU のマルチコア化、マルチスレッド化が進み、多重プロセスでの性能向上が著しいため、パラレルクエリーを使用することで、サーバのリソースを有効利用し、クエリレスポンスを大きく改善できます。

3. Automatic Storage Management

ASM は、Oracle Database 10g で実装された、データベースで使用されるディスクを管理する機能です。複数のディスクへのストライプやミラーリングといった機能に加え、ダイナミックにディスクの構成を変更できるリバランス機能を備えています。これにより、業務データベースを停止することなくディスクを追加することができ、ディスクの管理コスト削減や、ディスクの負荷分散による性能改善が期待できます。

4.外部表

テキストファイルをソースとするデータロードは、従来は SQL*Loader によるローディングが一般的でした。

外部表は、CSV ファイル等の OS ファイル内のデータに対して、データベース内の表データのように、SQL による検索を提供する機能です。外部表に対しては SELECT 文のみが実行でき、DML 文を実行できない、索引を作成できないといった制限があります。しかし通常の表と同様に SELECT 文による選択、結合、ソート処理やパラレルクエリーの実行も可能です。すなわち外部表を利用することで、SQL*Loader では実現できなかった高度な変換処理をしながらのロードが可能になります。

F.検証環境

本検証はデータベースクライアント 1 台、データベースサーバ 1 台、ストレージ 1 台の構成で検証を実施しました。各検証の SQL 等の処理自体はデータベースサーバ上で実行しています。

1.システム構成

[図 2]にシステム構成の概略を示します。クライアントとデータベースサーバ間は 1000Base-T、データベースサーバとストレージ間は FC スイッチを介して 4 本の 4Gbps ファイバチャネルを使用して接続しています。データベースサーバとストレージ間は ETERNUS マルチパスドライバを使用し、4 本のパスに対して負荷分散を行っています。

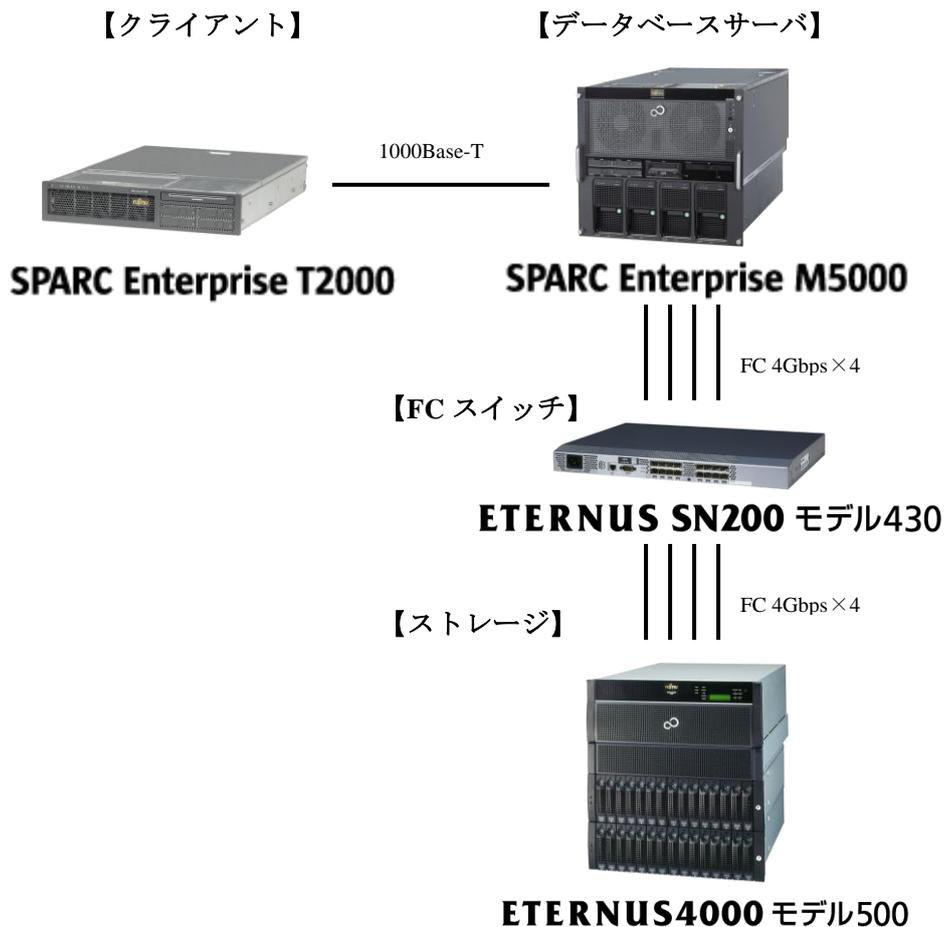


図 2 システム構成概略

各機器のスペック、使用したソフトウェア等を示します。

1.1.データベースサーバ

ハードウェア

モデル	富士通 SPARC Enterprise M5000
CPU	SPARC64 VI 2.15GHz/5MB キャッシュ 4CPU/8 コア/16 スレッド
メモリ	32GB
内蔵 HDD	73GB (10,000rpm) SAS Disk×2
I/O ボード	2
FC カード	デュアルチャンネル 4GbpsFC カード×2

ソフトウェア

OS	Solaris 10 OS 8/07 (SunOS 5.10 Generic_127111-03)
ストレージ・ソフトウェア	ETERNUS マルチパスドライバ 2.0.3
データベース	Oracle Database 11g (11.1.0.6)

1.2.ストレージ

モデル	富士通 ETERNUS 4000 モデル 500
コントローラー (CM)	2
メモリ	16GB (8GB/CM)
チャンネルアダプター	4Gbps 2Port FCCA×2 (1FCCA/CM)
ドライブエンクロージャー	4
ディスクドライブ	300GB (15,000rpm) × 60

1.3.クライアント

ハードウェア

モデル	富士通 SPARC Enterprise T2000
CPU	Ultra SPARC T1 1.2GHz/3MB キャッシュ 1CPU/8 コア/32 スレッド
メモリ	32GB
内蔵 HDD	73GB (10,000rpm) SAS Disk×2

ソフトウェア

OS	Solaris 10 OS 8/07 (SunOS 5.10 Generic_127111-03)
データベースクライアント	Oracle Database 11g (11.1.0.6)

1.4.FC スイッチ

モデル	富士通 ETERNUS SN200 M430
転送能力	4Gbps
ポート数	8

2.ストレージ構成

本検証で用いたストレージの RAID グループ、ボリュームの構成、ASM ディスクグループの構成について説明します。

[図 3]のストレージ構成図に示すように、RAID 1+0 (ディスクドライブ 4 本) を 1 つの RAID グループとして構成し、合計 14 の RAID グループを作成しました。ETERNUS のシステム領域を必要とする RAID グループ#0000 のみ 126GB の論理ボリュームを 4 つ作成し、その他の各 RAID グループには 91GB の論理ボリューム 6 つを作成しました。

ASM ディスクグループの構成は表 A の通りです。

表 A ASM ディスクグループの構成

ASM ディスクグループ	RAID グループ	用途
SYS_DG	#0000	SYSTEM、SYSAUX、UNDO 表領域 REDO ログ、制御ファイル
TEMP_DG	#0001～#000D	一時表領域
COMP_DG	#000A～#000D	圧縮データ用表領域
NOCO_DG	#0006～#0009	非圧縮データ用表領域

本検証では、負荷の小さい SYSTEM、SYSAUX、UNDO 表領域、REDO ログファイル、制御ファイルを全て RAID グループ#0000 へ格納しました。

一時表領域はソート、表の結合、索引作成がメモリ内で完了しなかった場合に高い I/O 速度が求められるため、#0000 以外の RAID グループから論理ボリュームを 1 つずつ使用し、ディスク負荷を分散させる構成にしました。

通常、ASM のディスクグループは、管理を容易にし、より良い性能を得るために、全 RAID グループを 1 つのディスクグループとして構成します。この検証では、圧縮、非圧縮の検証データを格納する ASM ディスクグループ COMP_DG および NOCO_DG が、それぞれの場合で同一の構成となるよう、4 つの別々の RAID グループを使用しました。圧縮と非圧縮で ASM ディスクグループをあえて分けたのは、ディスクの内外周による性能差、および使用コントローラーの片寄りが生じない構成で各 ASM ディスクグループへ RAID グループを割り当て、同条件下で性能を比較するためです。

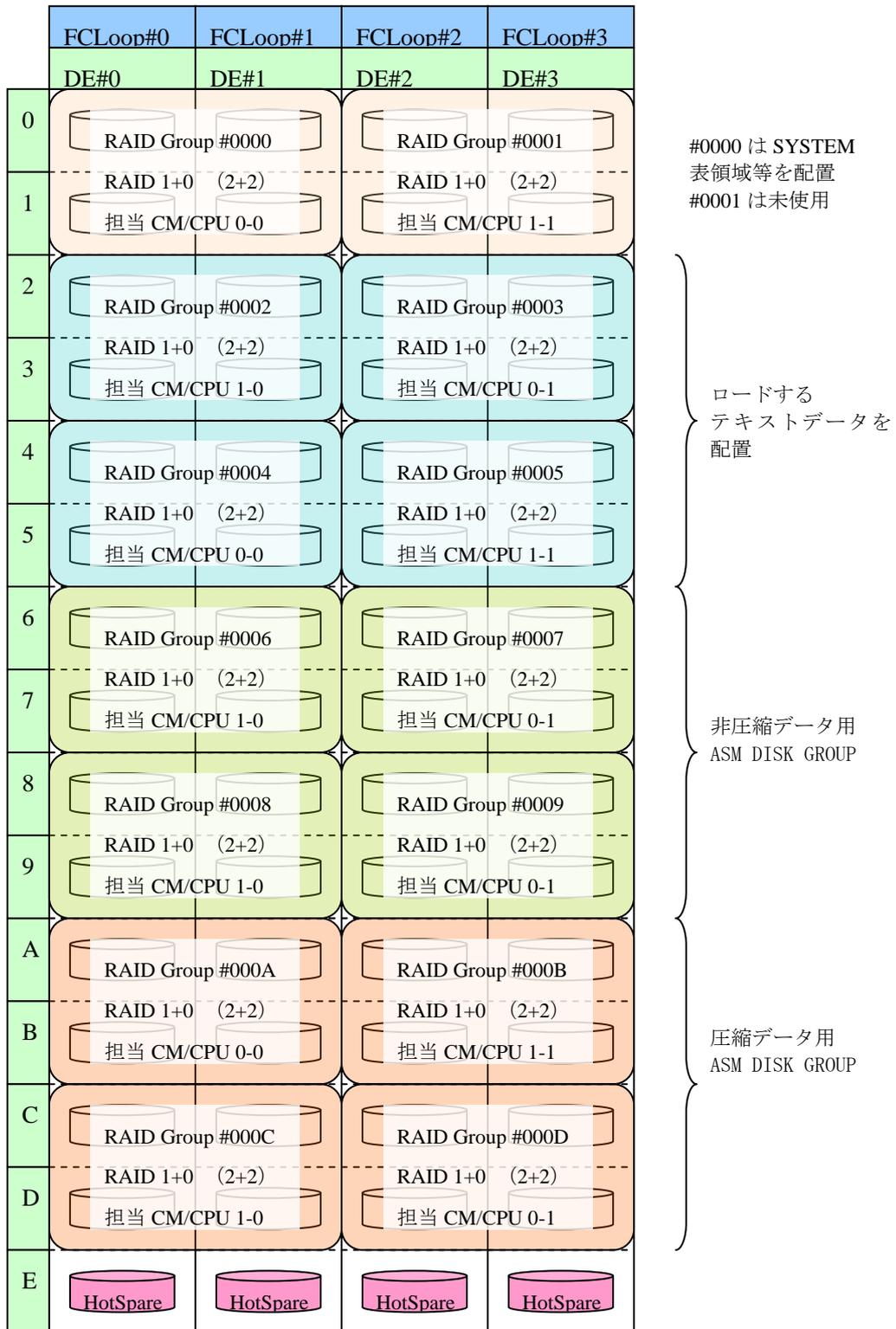


図 3ストレージ構成図

3.スキーマ構成

本検証で使用した検証モデルのスキーマ構成は、次の通りです。

3.1.スキーマ構成概要

本検証では、DWH を想定したスキーマを使用しています。表および、表の行数、非圧縮時のデータサイズは[表 B]の通りです。

表 B 検証に用いた表

表名	行数	サイズ
T1	1,200,000,000	153GB
T2	160,000,000	22.8GB
T3	40,000,000	6GB
T4	5	192KB
T5	25	192KB
T6	2,000,000	352MB
T7	30,000,000	4.75GB
T8	300,000,000	34.48GB

3.2.検証クエリ

検証では DWH を想定した 18 種類のクエリを使用しました。各クエリの概要は表 C の通りです。

表 C 検証クエリ

クエリ番号	クエリの概要
1	ある 2 か月間における返品フラグ、注文ステータスごとの売上金額や割引額等を計算する。
2	ある地域において、ある商品のサプライコストが最小のサプライヤを検索する。
3	ある顧客による注文 1 日分の合計収益を求める。
4	ある 3 か月間のなかで、取引確定日より領収日が後になっている注文について、注文の優先度ごとに注文数を求める。
5	ある地域における国ごとの収益を計算する。
6	ある 1 年間に於いて、1 回の注文で 25 個未満かつ、割引率が 0.3 から 0.5 で取引された品目の合計の収益を計算する。
7	サプライヤと顧客のどちらかが A 国、かつどちらかが B 国であった注文について 2 年間の収益を計算する。
8	ある商品のある地域における C 国でのマーケットシェアを計算する。
9	キーワード検索（中間一致の like 検索）に合致した商品の利益を計算する。
10	ある 3 か月間で返品された商品の合計金額を計算する。
11	ある国におけるサプライコストの合計が、全体に対して 0.00002% 以上の商品と、その商品のサプライコストの合計金額を計算する。
12	ある 1 年間に於いて、優先度の高い注文の数、低い注文の数を求める。
13	通常のパッケージ商品ではない商品の注文がある顧客について、顧客ごとに、パッケージ商品ではない商品を注文した回数を求める。
14	商品タイプにてキーワード検索（前方一致の like 検索）でヒットした商品について、1 年間の売り上げを計算する。
15	ある 3 か月間のサプライヤごとの収益を計算したビューを作成し、そのビューから収益が最も多いサプライヤを検索する。
16	1 回の注文で 1 つの商品を 300 個以上購入した顧客を検索する。
17	ある国の各サプライヤについて、ステータスが待機中となっている注文の数を求める。
18	国別平均の未払い金額を上回る未払いのある顧客について、未払い金額の合計を求める。

G. 検証内容

本検証では、表へのデータロードおよびクエリの実行をもって、データ・セグメント圧縮と ASM による動的なディスク追加およびデータのリバランスの性能測定を行いました。表へのデータロードおよびクエリの実行の概要は、次の通りです。

1. 表へのデータロード

スキーマ構成の通りの空表と、テキストデータを参照する外部表を作成し、外部表を元に空表に対してデータの平行ダイレクトロードを行いました。外部表が参照するテキストデータは別途アプリケーションで生成し、データファイルが使用する RAID グループとは別の、4つの RAID グループへ格納しました。

本検証環境が 4CPU 8 コア 16 スレッドであることから、平行度は 16 としました。なおブロックサイズは 32KB です。

2. クエリの実行

表 C検証クエリにある 18 種類のクエリを、平行度 16 で実行しました。この平行度は表へのデータロードと同じく、検証環境が 4CPU 8 コア 16 スレッドであることから決定しました。

データ・セグメント圧縮および ASM リバランスの検証内容については次の通りです。

3. データ・セグメント圧縮

検証で用いる全ての表を、CREATE TABLE 文の圧縮属性である [COMPRESS FOR ALL OPERATIONS]を指定して作成しました。[COMPRESS FOR ALL OPERATIONS]を用いることにより、Oracle のデータ・セグメント圧縮にある通り、データの挿入方法を選ばずにセグメント圧縮が実行されます。検証では圧縮属性の有無により、セグメントサイズだけでなくデータのロードおよびクエリの実行時間に違いがあるかも比較します。

4. ASM による動的なディスク追加およびデータのリバランス

Automatic Storage Managementで説明したとおり、ASM ディスクグループへディスクを追加および削除するとき、ASM ディスク間でデータのリバランスが行われます。本検証では 1つの RAID グループへロードしたデータを、RAID グループ 2つ、RAID グループ 4つへリバランスさせ、I/O 性能を向上させると、クエリ性能が変化するかを検証しました。

H.検証結果

各検証項目の結果は次の通りです。

1.データ・セグメント圧縮

1.1.圧縮によるセグメントサイズの差異

全表の合計サイズの相対値（非圧縮時のサイズ=1）を[図 4]に示します。本検証で使用したデータでは、圧縮を行うと非圧縮時の 62%程度のサイズとなりました。なお、圧縮された表の相対サイズについては、列の定義やデータの内容に強く依存します。この検証で使用したデータは乱数を使用して合成したものであり、全体の傾向として圧縮によるサイズ縮小がわかりにくくなっています。

以降では圧縮される表と圧縮されない表の違いについて考察します。本検証で用いた表の中で、最も圧縮された T1 表と、最も圧縮されなかった T2 表の相対サイズは、非圧縮時を 1 とした場合、[図 5]のようになります。

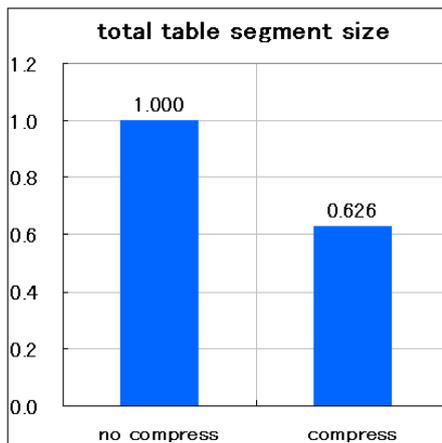


図 4 全表合計サイズの相対値

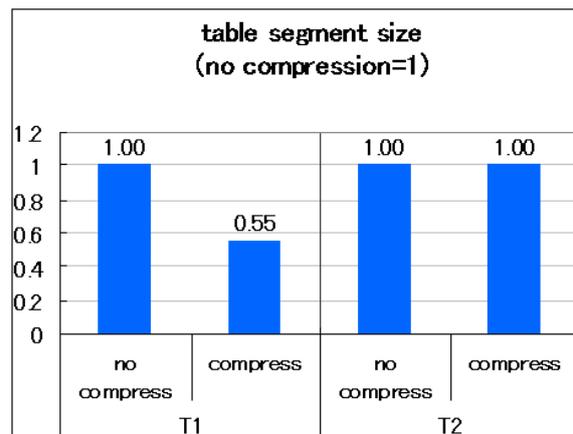


図 5 T1 表、T2 表の相対サイズ

各表のカラム定義は次の通りです。

・ T1 表

カラム番号	型	制約
c1	DATE	
c2	NUMBER	NOT NULL
c3	NUMBER	NOT NULL
c4	NUMBER	NOT NULL
c5	NUMBER	NOT NULL
c6	NUMBER	NOT NULL
c7	CHAR (1)	
c8	NUMBER	NOT NULL
c9	CHAR (1)	
c10	NUMBER	NOT NULL
c11	DATE	
c12	DATE	
c13	CHAR (10)	
c14	NUMBER	NOT NULL
c15	CHAR (25)	
c16	VARCHAR2 (44)	

・ T2 表

カラム番号	型	制約
c1	NUMBER	NOT NULL、PK
c2	NUMBER	NOT NULL、PK
c3	NUMBER	NOT NULL
c4	NUMBER	
c5	VARCHAR (199)	

データセグメント圧縮は、ブロック内の重複値をシンボル表とポインタへ置換します。重複値は同じ列内である必要はなく、列が異なっても値が同一であれば、シンボル表内の同じエントリを参照するポインタへ置き換わります。

本検証においてはデータの傾向を調べるために、最も圧縮された T1 表と最も圧縮されなかった T2 表の、各列における重複値の種類を示すカーディナリティを比較しました。カーディナリティのみで圧縮効果が決まるわけではありませんが、これは 1 つの要素となりえます。

まず、最も圧縮された T1 表について考察します。ブロック数および総行数から算出した結果、T1 表は圧縮しない状態では 1 ブロックに平均 239 行格納されたのに対し、圧縮を行うと 1 ブロックに平均 438 行格納されました。データの傾向を調べるために、圧縮した T1 表の平均 1 ブロック分に相当する 438 行目までに、各カラムで重複値がないかチェックした結果は次の通りです。

```
select
count(*) count,
(省略、各列にて count(distinct 列名))
from T1
where ...
```

COUNT	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16
438	388	111	11	438	438	50	3	438	2	9	381	391	7	7	4	434

438 行中ほとんどのカラムの値が 438 種類未満です。つまり重複値を含有していることとなります。特に C13 列および C15 列は 438 行中、値が 7 種類ないしは 4 種類しかなく、列データの種類の数を示すカーディナリティが相対的に低いといえます。C13 列および C15 列はそれぞれ CHAR (10) / CHAR (25) の固定長であるため、非圧縮表では (10+25) *438 行分の領域を占めていたものが、シンボル表内に格納された実データ (10*7+25*4) とポインタで事足りることになります。

このように表の中で相対的に列長が長いカラムのカーディナリティが低いと、圧縮によってサイズがより小さくなる傾向があります。

次に、最も圧縮されなかった T2 表について考察します。ブロック数および総行数から算出した結果、T2 表は圧縮、非圧縮ともに 1 ブロック平均 212 行が格納されました。圧縮した T2 表の平均 1 ブロック分に相当する 212 行目までに、各カラムで重複値がないかチェックした結果は次の通りです。

```
select
count(C1) count,
(省略、各列にて count(distinct 列名))
from T2
where ...
```

COUNT	C2	C3	C4	C5
212	212	212	208	212

この 212 行中、各カラムの値の種類は C4 列が 208 種類であり、重複しているのは 4 行だけになります。残りの 4 列は 212 行に 212 種類の値が格納されており、これは重複がないことを意味します。つまり全カラムにおいてカーディナリティが高いといえます。以上より本検証においては、圧縮してもサイズが小さくならない表は、カーディナリティが高いことが確かめられました。

1.2.表へのデータロード

本検証で用いた全ての表へのデータロード所要時間を、データ・セグメント圧縮を行う場合と行わない場合で比較します。

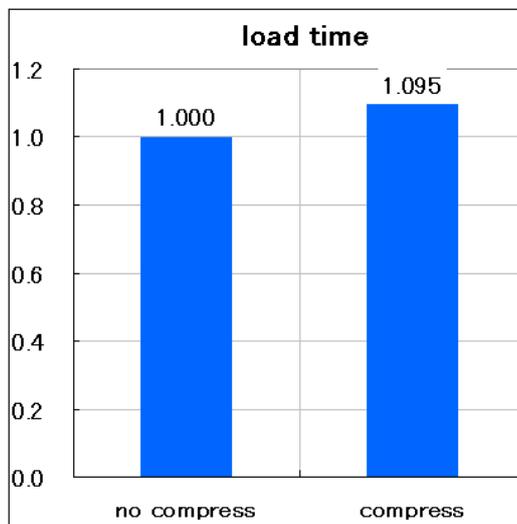


図 6 データロード合計時間相対値

ソースデータとなるテキストファイルを参照する外部表を作成し、外部表を元にデータを変換しながら空の表に対してパラレルダイレクトロードを行いました。このとき、表が格納されている ASM ディスクグループは、1つの RAID グループ（ディスクドライブ 4 本で RAID 1+0）のみで構成されています。

データロード所要時間の相対値（非圧縮時の所要時間を 1.000）は[図 6]の通りです。

本検証で用いた表のうち、最もサイズが大きい T1 表（非圧縮時 約 153GB）をロードした際の、CPU 使用率の時間推移は[図 7]の通りです。このデータロードは SQL*Loader のように単純なロードではなく、外部表から INSERT ... SELECT 文で行データを読み取り、行データの変換処理をしてダイレクト INSERT を行っています。前半（[図 7]中の[TRANSFORMATION]が示す時間）は非圧縮、圧縮ともに 60%前後の使用率ですが、ここで先述の通り外部表を介して読み込んだテキストデータの変換を行っています。変換処理部分の所要時間は[図 7]が示す通り非圧縮時、圧縮時ではほぼ同じでした。

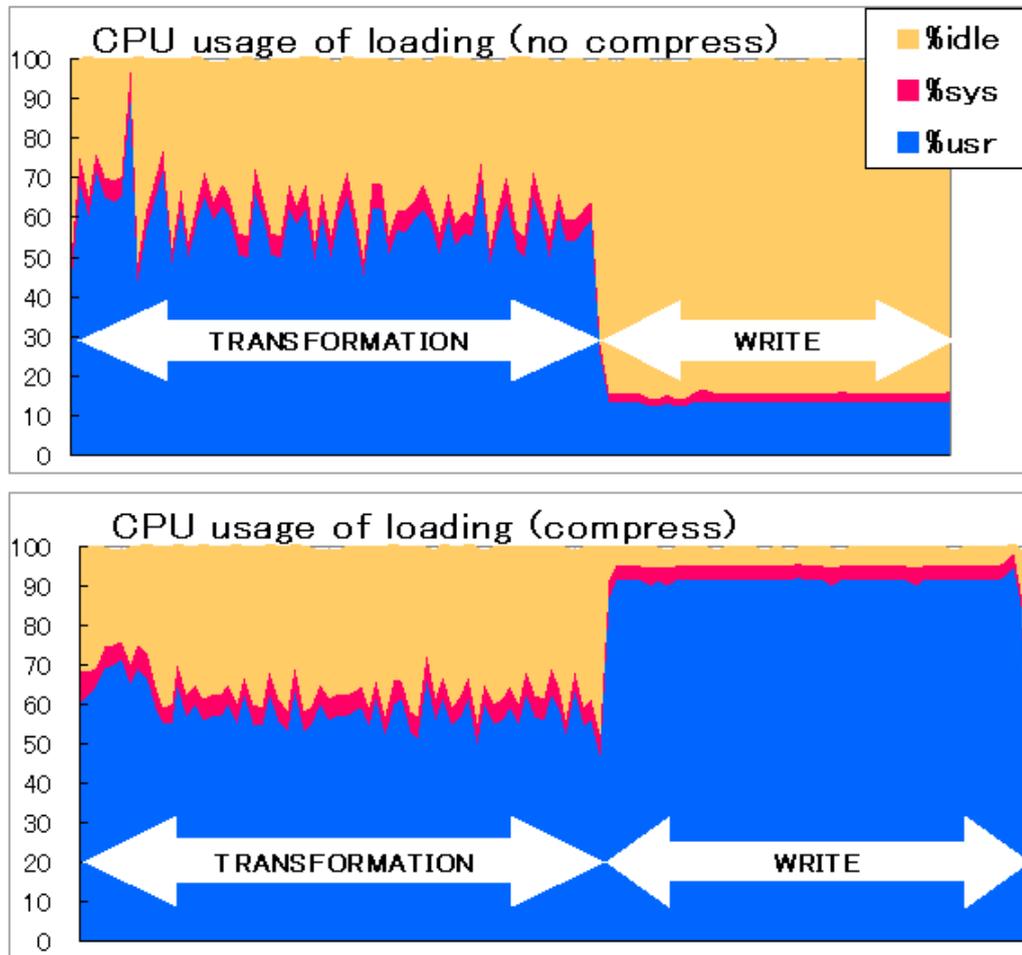


図 7 T1 表ロード時の CPU 使用率時間推移

この検証での圧縮と非圧縮のロード時間差は、データ変換処理後のデータファイルへの書き込み等([図 7]中の[WRITE]が示す時間)で発生したと考えられます。

次に、データロードの所要時間のみの差を示すために、本検証で用いた表の 1 つをデータ変換せずにロードした際の CPU 使用率の時間推移を[図 8] (RAID グループ 1 つ)、[図 9] (RAID グループ 2 つ) に示します。上段が非圧縮時、下段が圧縮時の結果です。

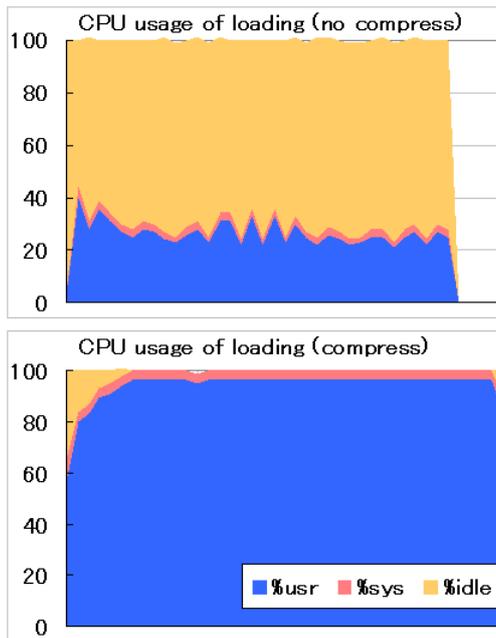


図 8 CPU 使用率の時間推移 (RAID グループ 1 つ)

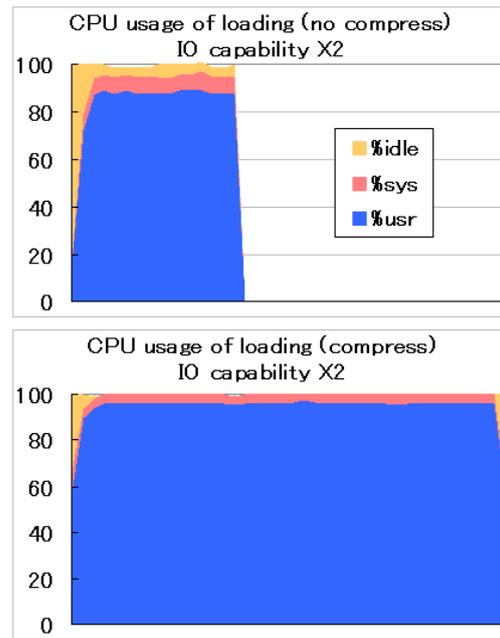


図 9 CPU 使用率の時間推移 (RAID グループ 2 つ)

[図 8]より、圧縮を行うと CPU 使用率が高くなるのが分かります。一般的にデータロードは CPU 負荷が高い処理ですが、非圧縮の場合の CPU 使用率が 30%前後であることから、ボトルネックはディスク I/O であると推測できます。

そこで、用いるディスクの数を 2 倍にし、すなわちディスク I/O 性能を 2 倍にして、データ変換を行わないロードを実施しました。結果は[図 9]の通りです。この場合では非圧縮時においてボトルネックになっていたディスク I/O が解消され、CPU 使用率が高くなっているのが分かります。処理時間に着目すると、[図 8]よりも[図 9]の方が、圧縮の有無による差が大きいことが分かります。

以上より、圧縮の有無によるデータロード所要時間の差は、ディスク I/O がボトルネックとなっている環境では比較的小さく、ディスク I/O 以外、例えば CPU 性能がボトルネックとなっているような環境では比較的大きくなると言えます。

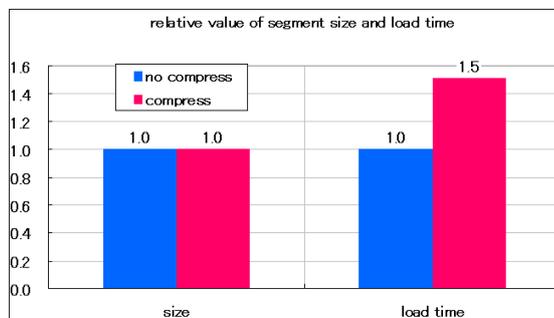


図 10 T2 表のロード時間および相対サイズ

次に、最も圧縮されなかった T2 表のロード時間および相対サイズに着目します。[図 10]は非圧縮時を 1 としたこれらの相対値です。[図 10]の示すとおり、非圧縮時と比較すると、圧縮属性のついた表へのロードの方が、所要時間が長くなっています。

これは圧縮されないデータであっても、ロード時に重複値のチェックは行われるため、その分オーバーヘッドが生じるためです。

したがって、セグメント圧縮の適用について、ロード時間という観点では対象表の各ブロック内にどの程度重複値が含まれるのかを考慮して適用を検討する必要があります。

1.3.クエリの実行

ここからは、圧縮の有無がクエリ性能に与える影響について調べます。

検証に用いた 18 種類のクエリの実行時間は、[図 11]の通りです。なお表が格納されている ASM ディスクグループは、1 つの RAID グループのみで構成されています。クエリの処理内容によって実行時間は様々ですが、圧縮している表に対するクエリの方が、圧縮していない表に対するクエリよりも、全体的に実行時間が短いことが分かります。

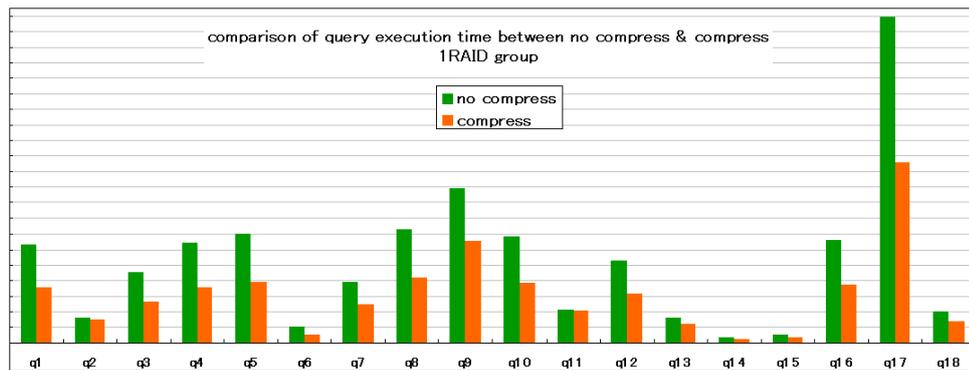


図 11 クエリ実行時間 (RAID グループ 1 つ)

各クエリでの圧縮の効果を調べるために、各クエリの実行時間の非圧縮時の実行時間を 1 とすると、圧縮時の実行時間の相対値は[図 12]のようになります。

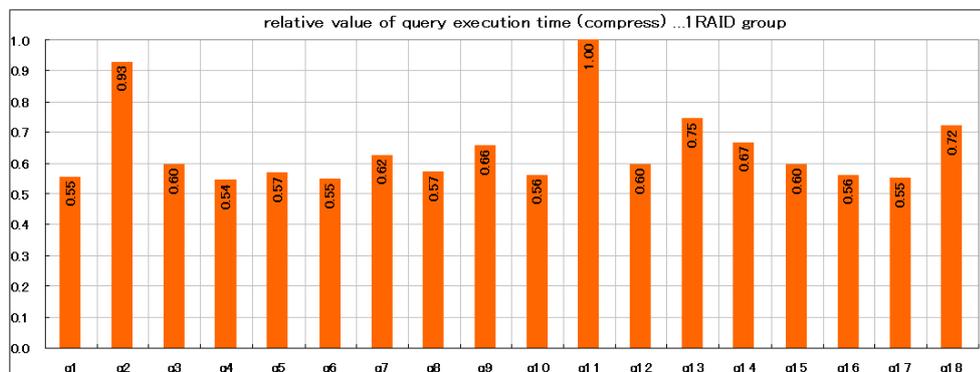


図 12 クエリ実行時間相対値 (非圧縮時=1.00、RAID グループ 1 つ)

次に、実際に各クエリで行われたディスク I/O 量を比較します。圧縮表に対するクエリ実行時の物理読み込み数の相対値を、非圧縮表へのクエリ実行時の物理

読み込み数を 1 として、[図 12]に加えると[図 13]のようになります。

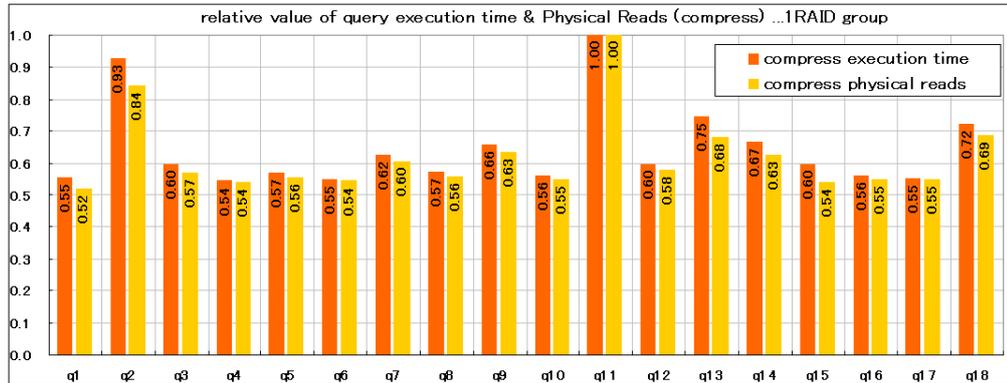


図 13 クエリ実行時間および物理読み込み数の相対値（非圧縮時=1.00）

クエリ実行時間および物理読み込み数の相対値が似た傾向を示し、2 者の相関係数は 0.98724 と高い値になりました。したがってクエリ実行時間の短縮は、圧縮によって I/O 回数が削減された点に起因する部分が大いと考えられます。これはデータ・セグメントを圧縮したことで 1 ブロックに含まれる行数が多くなり、同じ行数を読み込む場合でも、非圧縮時と比較して圧縮時の方が、少ないブロックの読み込みで済むためです。

次に、圧縮によって同一行数を読み込むのに必要なブロック数が減少し、I/O 効率が向上したことによって、クエリ実行中の CPU 使用率がどのように変化するかを調べます。例としてクエリ q1,q13 実行中の CPU 使用率の時間推移を[図 14](q1)、[図 15] (q13) に示します。なお物理読み込み数の比率は q1:q13=4:1 程度です。

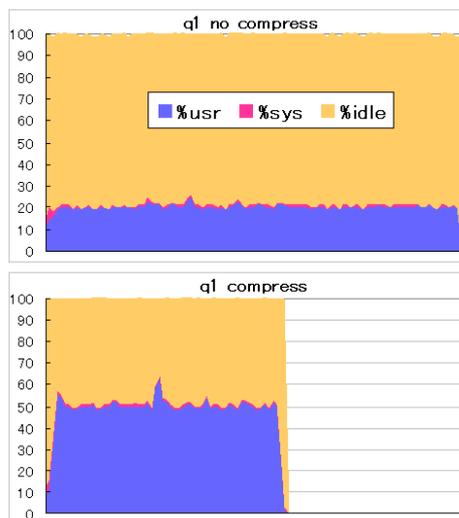


図 14 q1 実行時の CPU 使用率の時間推移 (上段：非圧縮、下段：圧縮)



図 15 q13 実行時の CPU 使用率の時間推移 (上段：非圧縮、下段：圧縮)

どちらのクエリにおいても、圧縮を行っている場合にはクエリ実行時間が減少し、CPU 使用率が増加しています。圧縮によってディスク I/O の待機時間が減少し、CPU が効率的に使われるようになったことが分かります。

1.4.セグメント圧縮によるクエリ実行時のオーバーヘッド

これまで述べてきた検証に用いた表データの格納先は、圧縮、非圧縮の場合ともに 1 つの RAID グループ上に配置されており、ディスク I/O 性能がボトルネックとなる傾向がありました。そこでディスク I/O 性能を強化するために 4 つの RAID グループへデータを格納し、RAID グループ 1 つの場合と同様に圧縮の有無によるクエリ性能を比較する検証を行いました。4 つの RAID グループでのクエリ実行時間は[図 16]の通りです

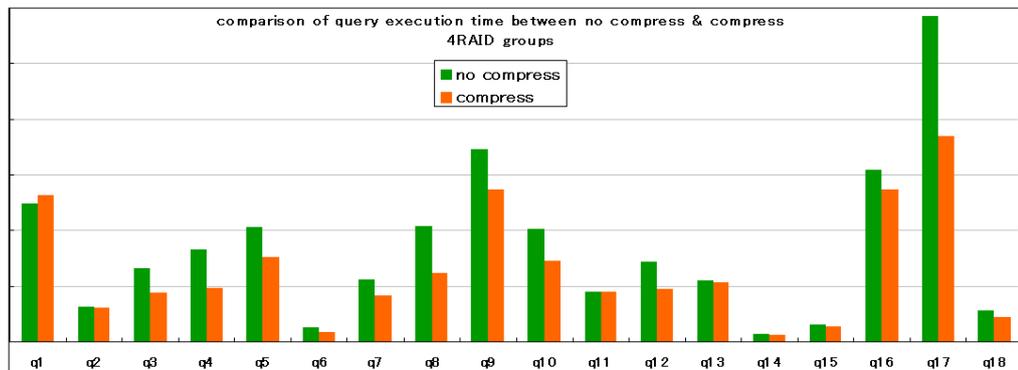


図 16 クエリ実行時間 (非圧縮時=1.00、RAID グループ 4 つ)

RAID グループ 1 つの場合の[図 13]と同様に、非圧縮時を 1 としたときの、圧縮時のクエリ実行時間と物理読み込み数の相対値を[図 17]に示します。

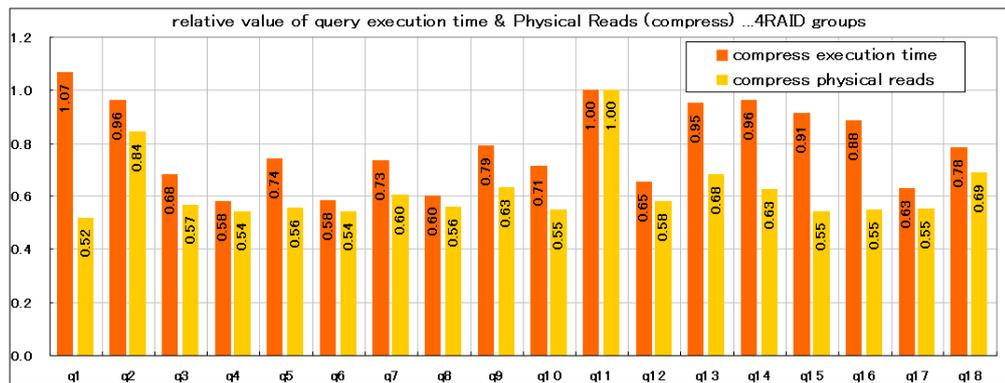


図 17 クエリ実行時間および物理読み込み数の相対値 (非圧縮時=1.00、RAID グループ 4 つ)

RAID グループ 1 つを用いた場合の[図 13]と比較すると、クエリ実行時間と物理読み込み数の相対値間に差が見られます。2 者の相関係数を算出すると 0.4893 であり、RAID グループ 1 つの場合の相関係数 0.98724 より低い値となりました。つまり RAID グループ 1 つの場合よりもディスク I/O 性能の高い 4 つの RAID グループ

プの場合、クエリ実行時間に占めるディスク I/O 時間の割合が低下したと言えます。両者の差が特に大きいクエリについては、ディスク I/O 性能から他の、たとえば CPU 性能などへボトルネックが移ったものと考えられます。

[図 16] を見ると、RAID グループが 4 つの場合、q1 についてのみ、圧縮時の実行時間が非圧縮時の実行時間をわずかに上回っています。これには 2 つの理由があります。1 つ目はポインタからシンボル表をたどるための、CPU オーバーヘッドの影響です。圧縮によってシンボル表へのポインタに置き換わった列データは、参照の際にポインタからシンボル表を参照します。この処理によるオーバーヘッドは極めて小さく、他のクエリの結果が示す通り、通常はこれが問題となることはありません。しかし q1 は合計値や平均値などを計算するクエリであり、値を何度も参照します。1 回あたりの CPU オーバーヘッドはわずかでも、q1 のように値を何度も参照するようなクエリでは、CPU オーバーヘッドの影響が表面化すると考えられます。2 つ目は、q1 が演算主体のクエリであることです。演算主体のクエリであるため、他のクエリに比べて、もともと CPU に対する負荷が高いと言えます。他のクエリでは CPU オーバーヘッドに対して I/O の削減によるレスポンス改善の割合が大きいため、CPU オーバーヘッドは問題になりません。以上の 2 つの要因によって、q1 では、RAID グループを 4 つにしたことで、ボトルネックが I/O から CPU へ移り、ポインタからシンボル表をたどる CPU オーバーヘッドが表面化し、レスポンスが逆転したと考察されます。

また、[図 13]を見ると、q11 については圧縮時、非圧縮時で実行時間に差がありません。q11 は圧縮されなかった T2 表に対するクエリです。q1 の例で示された通り、圧縮表へのクエリのオーバーヘッドはポインタをたどる際に生じます。圧縮されなかった T2 表については、表に圧縮属性がついていてもシンボル表およびポインタがないため、データに対するアクセス方法は非圧縮表に対するクエリのとおりと同じと言えます。以上より、セグメント圧縮によりサイズが小さくならない表を圧縮した場合でも、クエリ性能は非圧縮時と変わらないと言えます。

1.5. データ・セグメント圧縮総括

本検証では、データ・セグメント圧縮機能を使うことにより、セグメントサイズが全体で約 4 割削減されました。この値は列の定義やデータ内容に強く依存します。

データセグメント圧縮は列単位でなくブロック単位の重複値をポインタ化するため、カーディナリティが圧縮効果の唯一の決定要素ではありません。しかし本検証においては、圧縮によりサイズが著しく小さくなる表は、カーディナリティが低いカラムを多く含む、カーディナリティの低いカラムのサイズが大きい、などの傾向があることが確かめられました。この検証で使用したデータは乱数を使

用して合成したものであり、全体の傾向として圧縮によるサイズ縮小が比較的かかりにくくなっています。実システムのデータセットでは、圧縮によるサイズ縮小の効果はより大きなものが期待できます。

また、表を圧縮する場合、データロード時に CPU オーバーヘッドが生じます。データロード時の CPU オーバーヘッドの原因は、同一ブロック内の重複値のチェック、シンボル表の作成など圧縮のための作業が考えられます。

他方で圧縮を行うと 1 ブロックに含まれる行数が多くなるため、より少ないディスク I/O 回数で大量の行を読み込めます。これによりクエリ実行時にディスク I/O を待機する時間が減少し、効率的に処理できるようになるため、実行時間が短縮されることが確認できました。

データ・セグメント圧縮の利用は、クエリ実行時間を短縮できるため、データロード時のオーバーヘッドの影響を考慮しても、業務ユーザーの要求にこたえるための非常に有効な手段であるといえます。

2.ASM による動的なディスク追加およびデータのリバランス

2.1.ASM ディスク追加の効果

DWH 環境のクエリ性能改善手法の 1 つとして、ディスクドライブの増設による I/O 分散により、ディスク I/O 性能を向上させる方法があげられます。ここではデータ格納に使用する RAID グループを、1 つから 2 つ、4 つへ増加させ、データがリバランスされた結果、クエリ実行時間がどのように変化するかを調べます。

[図 18]に示すように、1 つの RAID グループで構成される ASM ディスクグループに、RAID グループを追加することにより、ASM ディスクグループが RAID グループ 2 つ、4 つを用いるよう構成変更し、それぞれのクエリ性能を測定します。

なお、ディスク I/O 性能とクエリ実行時間の関係に着目するため、ここではデータ・セグメント圧縮は行いません。

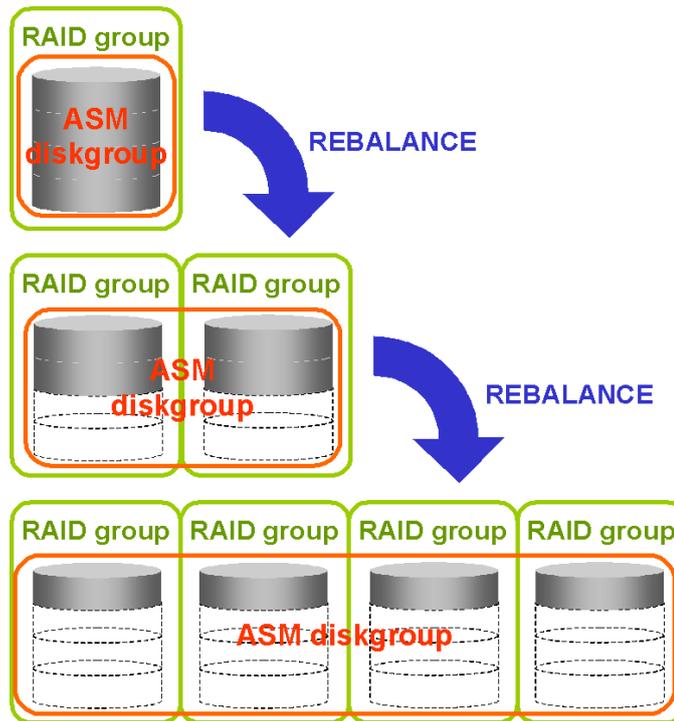


図 18 ASM ディスク追加とデータのリバランス

各ディスク構成のクエリ実行時間は[図 19]の通りです。全てのクエリにおいて用いる RAID グループが多いほどクエリ実行時間が短い傾向にあることが分かります。

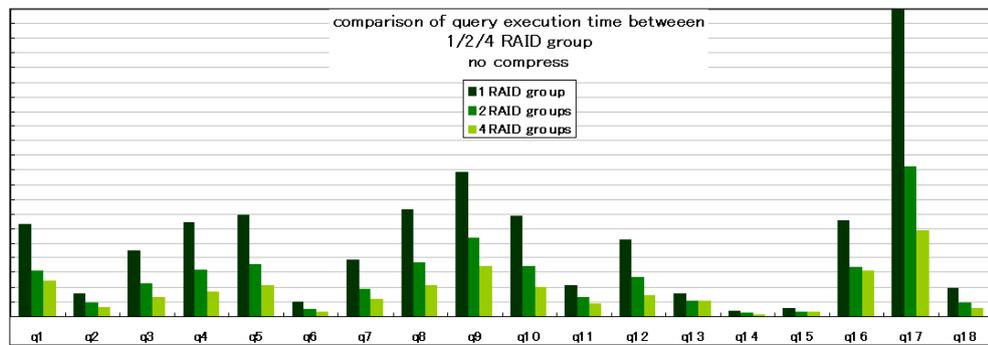


図 19 クエリ実行時間

各クエリについて、RAID グループ 1 つのときのクエリ実行時間を 1 とし、RAID グループ 2 つ、RAID グループ 4 つのクエリ実行時間の相対値を次に示します。

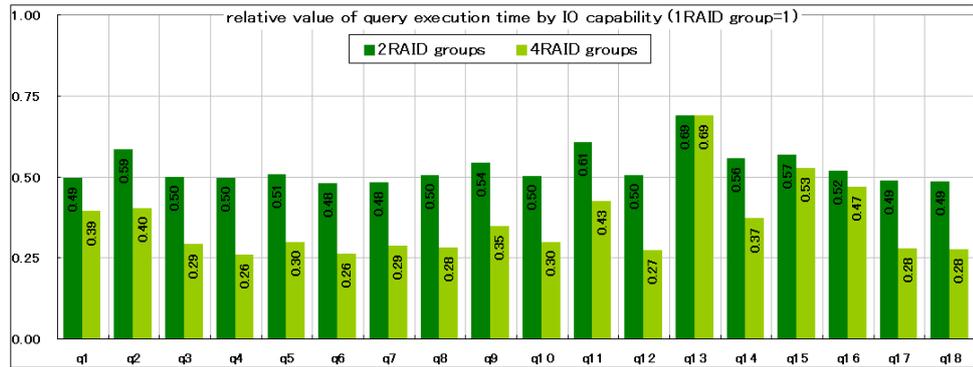


図 20 クエリ実行時間相対値

RAID グループ 1 つの場合に比べ、RAID グループ 2 つの場合は全クエリにおいて実行時間が改善されました。用いる RAID グループ数が 2 倍になったことでディスク I/O 性能が 2 倍となり、その結果クエリの実行時間が短縮されたと言えます。一方で RAID グループ 2 つの場合と RAID グループ 4 つの場合を比較すると実行時間短縮の度合に、各クエリ間でばらつきがあります。

性能改善の有無の原因を分析するために、実行時間が RAID グループ 1 つ、2 つ、4 つでほぼ線形に改善された q8 と、RAID グループ 2 つと 4 つの間で改善が見られなかった q13 を比較します。q8 と q13 を実行した際の CPU 使用率の時間推移は次の通りです。

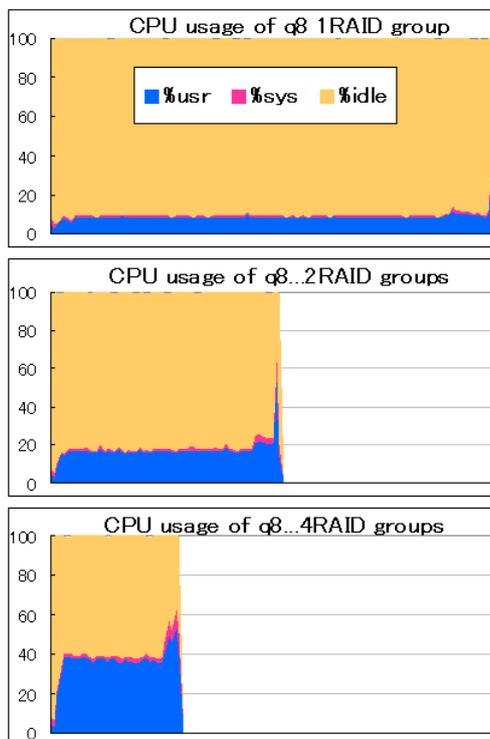


図 21 q8 の CPU 使用率時間推移 (上から RAID グループ 1 つ、2 つ、4 つ)

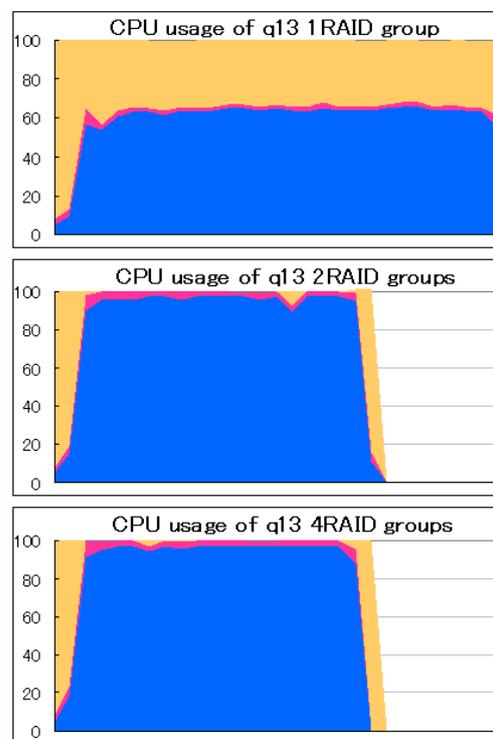


図 22 q13 の CPU 使用率時間推移 (上から RAID グループ 1 つ、2 つ、4 つ)

q8 の CPU 使用率の時間推移は[図 21]が示す通りで、RAID グループが 1 つから

2つに追加されたのに伴い、CPU使用率もほぼ倍になっていることが分かります。また RAID グループ 4 つでもまだ CPU 使用率に余裕があるため、更に RAID グループを追加することでクエリ性能向上が見込めます。

他方で[図 22]が示す q13 の CPU 使用率は、RAID グループ 1 つで 60%程度であり、RAID グループ 2 つの時点ではほぼ 100%に達していることが分かります。

したがって q13 においては、RAID グループ 1 つのときはディスク I/O 性能がクエリ実行のボトルネックだったものが、RAID グループ 2 つ以上ではボトルネックが CPU 性能へ移ったため、2 つから 4 つへの RAID グループの追加によるクエリ性能の向上が得られなかったと考えられます。

2.2.ASM のリバランスコスト

RAID グループ 1 つの環境から RAID グループ 2 つの環境へ移行するには、ASM ディスクの格納データをリバランスする以外に、RAID グループ 2 つの環境を用意した後にデータを再ロードするという方法が考えられます。以降では ASM のリバランスと再ロードの時間的コストを比較します。

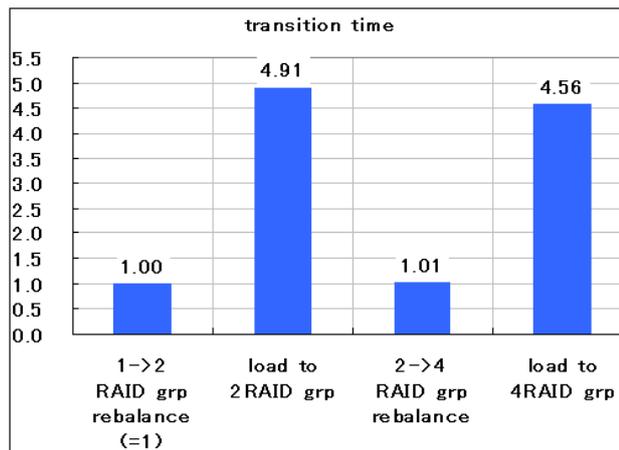


図 23 リバランスおよび再ロードの所要時間相対値

[図 23]に RAID グループ 1 つから RAID グループ 2 つへのリバランス所要時間を 1 とした、リバランスおよび再ロード所要時間の相対値を示します。データ・セグメント圧縮は行っていません。再ロードの所要時間には、空表へのデータのダイレクトロード、索引作成時間を含みます。

実際に既存環境からディスク増設後の環境へ再ロードで移行する場合、上記 2 つに加えて既存環境上のデータをアンロードする時間も生じます。なお[図 23]が示す通り、本検証環境においては RAID グループ 1 つから 2 つへのリバランスと、RAID グループ 2 つから 4 つへのリバランスの所要時間はほぼ同じでした。

2.3.ASM による動的なディスク追加およびデータのリバランス総括

一般的に、データベースシステムにディスクを増設する動機は容量不足の解消

であることが多いと想定されます。ASMではディスクを増設すると、既存データのリバランスも行われるため、容量の拡張とともにディスク I/O 性能も向上します。本検証から、ディスク I/O 性能がボトルネックのクエリについては、ASMによる動的なディスク追加およびデータのリバランスを行って RAID グループを増やすことにより、ほぼ線形に性能を向上できることが確認できました。

ASMによる動的なディスク追加およびデータのリバランスには、所要時間が短い、アプリケーション停止の必要がないなどの利点もあります。例えば、システムを継続して利用するにあたりデータ量の増加は不可避ですが、データ増に伴いディスク I/O 性能が限界に達し、クエリの性能が低下した場合、ASM環境ではアプリケーション無停止でディスク増設を行い、ディスク I/O 性能を向上させることでクエリ性能を改善できます。一方データファイル用記憶域として ASM 以外を使用している場合、ディスク増設後に既存データのリバランスを行うためには、多くの場合、アプリケーションを停止してのデータのアンロード、データの再ロード、統計情報の再取得、索引の再作成など多大なコストがかかります。以上を考慮すれば、ASMの利便性は極めて高いと言えます。

3.データ・セグメント圧縮とディスク追加の併用

ここまではデータ・セグメント圧縮および ASM による動的ディスク追加それぞれ単体でのクエリ性能の改善について言及してきました。これらは組み合わせることにより大きなクエリ性能の改善を見込めます。

RAID グループを1つ、2つ、4つ使用した際の、非圧縮および圧縮のクエリ実行時間を[図 24]に示します。圧縮しているか否かにかかわらず、ASM による動的なディスク追加およびデータのリバランスによって用いる RAID グループ数を増やせば、ほとんどのクエリで大きな性能改善がなされることが分かります。

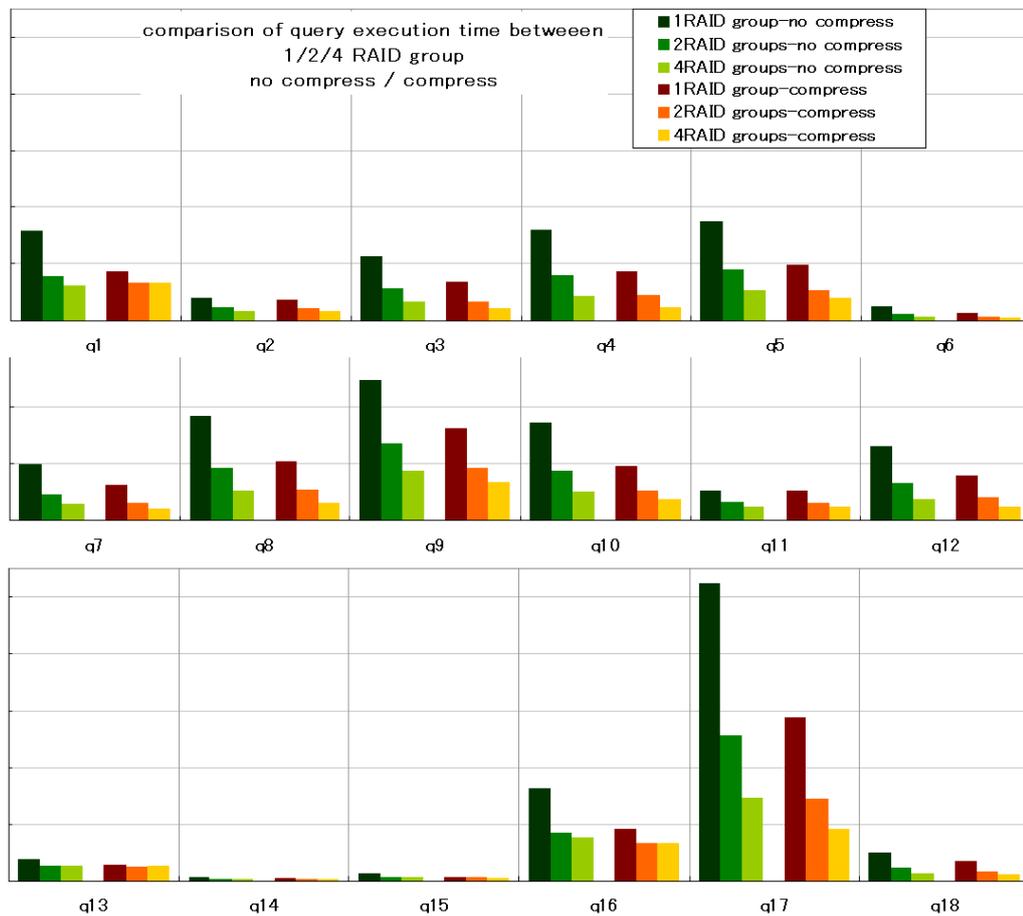


図 24 データ・セグメント圧縮とディスク追加の併用

以上の結果より、データ・セグメント圧縮と、ASM による動的なディスク追加およびデータのリバランスは、単体でもクエリ性能向上に効果がありますが、2つを組み合わせることにより大きな性能改善があると確かめられました。

ディスク I/O 性能が低い状態、例えば本検証においては RAID グループ 1 つにデータが格納されている場合、圧縮表に対するクエリは、非圧縮表に対するクエ

りの約 6 割程度の実行時間で完了しました。これより、ディスク I/O 性能が低い環境下において、データ・セグメント圧縮が性能改善に大きな効果をもたらすことも確かめられました。

他方でディスク I/O 性能が高い状態、例えば本検証においては RAID グループ 4 つにデータが格納されている場合では、圧縮によるクエリ性能改善の効果は、ディスク I/O 性能が低い状態と比較し小さくなります。それでもなお、圧縮表に対するクエリは非圧縮表に対するクエリよりも実行時間が短いことが確かめられました。

I.総括

データ・セグメントの圧縮機能を使用することにより、DWH 環境の膨大なデータサイズを削減し、ストレージの有効活用ができるだけでなく、必要なデータを読み込むための I/O 量を減少させ、クエリ性能の向上も可能であることが明らかとなりました。データロード時はデータを圧縮するためのオーバーヘッドが生じますが、クエリ性能向上の度合はこのオーバーヘッドを大きく凌駕するため、データ・セグメント圧縮によって DWH システム全体で見たとときのパフォーマンスが改善されることは明らかです。データのばらつきなどの特性を見極めた上で、圧縮効果の高い表に対して圧縮をかけるようにすることで、データロード時の CPU オーバーヘッドを最小限におさえつつ、クエリの性能向上が実現でき、システム全体のパフォーマンスがより向上します。

データベースのパフォーマンス向上を動機にディスクドライブを増設する場合、ASM による動的なディスク追加およびデータのリバランスがいかに有効かも証明されました。ASM 環境では、通常ディスクドライブ増設時に必要な、データの退避、ディスクの追加および構成、退避させたデータの再ロード、索引の再作成、統計情報の再収集などの煩雑な諸作業を行う必要がなく、Oracle Enterprise Manager もしくはコマンドラインにて ASM ディスクを追加する作業のみで、使用する全ディスクへデータのリバランスが行われ、I/O 分散によるデータベースのパフォーマンス向上が可能だからです。

データ・セグメント圧縮および ASM による動的なディスク追加およびデータのリバランスは単体でも効果がありますが、両方を用いることによって、より大きな性能向上も見込めます。本検証ではデータ・セグメント圧縮を行わなかった場合、行った場合の双方で、ASM による動的なディスク追加およびデータのリバランスによるクエリ性能向上を確認しました。使用する RAID グループ数が同じでも、圧縮を行った方が圧縮を行わない場合よりもクエリ実行時間が短いことも確認できました。

以上より、Oracle Database 11g の持つデータ・セグメントの圧縮機能、ASM による動的なディスク追加およびデータのリバランス機能が、DWH システムのパフォーマンス改善手法として有効であることが明らかとなりました。

最後に、DWH システムを想定した大量データのロード処理、多数の高負荷な検索、集計処理を繰り返し実行し、非常に高負荷な状態が続くなか、SPARC Enterprise と Oracle Database 11g は、システムを停止・再起動させることなく、安定して常に高いパフォーマンスを維持したことを報告いたします。

SPARC Enterprise M5000 は、最新のプロセッサ高速化技術を採用した「SPARC64 VI」等による高いデータベース処理性能と、徹底したデータ保護・冗長化による高い信頼性という特徴を持ちます。大容量データ処理のパフォーマンスの向上を可能にする Oracle Database 11g のセグメント圧縮、ASM 機能と組み合わせることで、増え続ける大量のデータと、それに伴って増大する処理に対応した、高性能、高信頼なシステムの実現が可能です。

J.参考情報

日本オラクル Web サイト

- Oracle Japan / Oracle GRID Center

http://www.oracle.co.jp/solutions/grid_center/

富士通 Web サイト

- UNIX サーバ SPARC Enterprise

<http://primeserver.fujitsu.com/unix/>

- ストレージシステム ETERNUS (エターナス)

<http://storage-system.fujitsu.com/jp/>

- ソフトウェア Oracle

<http://software.fujitsu.com/jp/oracle/>



日本オラクル株式会社

〒102-0094

東京都千代田区紀尾井町 4-1

ニューオータニガーデンコート

富士通株式会社

〒105-7123

東京都港区東新橋1-5-2

汐留シティセンター

Copyright © 2008 Oracle Corporation Japan. All Rights Reserved.

Copyright © 2008 FUJITSU LIMITED, All Rights Reserved

無断転載を禁ず

このドキュメントは単に情報として提供され、内容は予告なしに変更される場合があります。このドキュメントに誤りが無いことの保証や、商品性又は特定目的への適合性の黙示的な保証や条件を含め明示的又は黙示的な保証や条件は一切無いものとします。日本オラクル株式会社は、このドキュメントについていかなる責任も負いません。また、このドキュメントによって直接又は間接にいかなる契約上の義務も負うものではありません。このドキュメントを形式、手段（電子的又は機械的）、目的に関係なく、日本オラクル株式会社の書面による事前の承諾なく、複製又は転載することはできません。

本書は、Oracle GRID Center の取組みにて実施された検証結果に関する技術情報を提供するものであり、本書に記載されている内容は改善のため、予告無く変更することがあります。富士通株式会社は、本書の内容に関して、いかなる保証もいたしません。また、本書の内容に関連した、いかなる損害についてもその責任は負いません。

Oracle、JD Edwards、PeopleSoft、および Siebel は、米国オラクル・コーポレーションおよびその子会社、関連会社の登録商標です。その他の名称は、各社の商標または登録商標です。

UNIX は、米国およびその他の国におけるオープン・グループの登録商標です。

すべての SPARC 商標は、SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における登録商標です。SPARC 商標が付いた製品は、Sun Microsystems, Inc. が開発したアーキテクチャーに基づくものです。

SPARC64 は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の登録商標です。

Sun、Sun Microsystems、Sun ロゴ、Solaris およびすべての Solaris に関連する商標及びロゴは、米国およびその他の国における米国 Sun Microsystems, Inc. の商標または登録商標であり、同社のライセンスを受けて使用しています。

その他各種製品名は、各社の製品名称、商標または登録商標です。

本資料に記載されているシステム名、製品名等には、必ずしも商標表示 ((R)、TM) を付記していません。