

ORACLE 9i

ハイパフォーマンスチューニングSTATSPACK編

ORACLE 9i High-Performance Tuning with STATSPACK

Donald K. Burleson / 著 株式会社トップスタジオ / 訳 日本オラクル株式会社 / 監修

本書内容に関するお問い合わせについて

このたびは翔泳社の書籍をお買い上げいただき、誠にありがとうございます。弊社では、読者の皆様からのお問い合わせに適切に対応させていただくため、以下のガイドラインへのご協力をお願いしております。下記項目をお読みいただき、手順に従ってお問い合わせください。

●ご質問前に

弊社 Web サイトの「Q&A コーナー」(<http://www.shoeisha.com/info/help.asp>)をご参照ください。これまで受けたご質問への回答 (FAQ) や、的確な質問方法に関する情報を掲示しています。

そって、折り返し質問シートをお送りいたします。シートに必要な事項を漏れなく記入し、“編集部読者サポート係”まで FAX または郵便にてご返送ください。

●ご質問方法

弊社 Web サイトの書籍専用質問フォーム (<http://www.shoeisha.com/book/qa/>) をご利用ください。記載漏れ、お電話や電子メールによるお問い合わせ、本書にはさみ込まれたアンケートはがきなど別紙に記入されたご質問は、お受けしておりません。

●回答について

回答は、ご質問いただいた方法にそってご送付いたします。ご質問の内容によっては、回答に数日ないしはそれ以上の期間を要する場合があります。

※質問専用シートのお取り寄せについて

Web サイトにアクセスする手段をお持ちでない方は、ご氏名、ご送付先 (ご住所 / 郵便番号 / 電話番号または FAX 番号 / 電子メールアドレス) および「質問専用シート送付希望」と明記のうえ、電子メール (qaform@shoeisha.com)、FAX、郵便 (80 円切手同封) のいずれかにて“編集部読者サポート係”までお申し込みください。お申し込み方法に

●ご質問に際してのご注意

本書の対象を越えるもの、記述個所を特定されていないもの、また読者固有の環境に起因するご質問等にはお答えできませんので、あらかじめご了承ください。

●郵便物送付先および FAX 番号

送付先住所 〒160-0006 東京都新宿区舟町 5
FAX 番号 03-5362-3818
宛先 (株)翔泳社 出版局 編集部読者サポート係

本書に記載された URL 等は予告なく変更される場合があります。

本書の出版にあたっては正確な記述に努めましたが、著者、監修者、及び出版社のいずれも、本書の内容に対してなんらかの保証をするものではなく、内容やサンプルに基づくいかなる運用結果に関してもいっさいの責任を負いません。

本書に掲載されている画面イメージなどは、特定の設定に基づいた環境にて再現される一例です。

Oracle は、米国カリフォルニア州レッドウッド市のオラクル・コーポレーションの登録商標です。その他、本書に掲載されている会社名、商品名、製品名などは、一般に各社の商標もしくは登録商標です。本書では™、®、© は割愛させていただいております。

Oracle9i High-Performance Tuning with STATSPACK
by Donald Keith Burleson
ISBN 0-07-222360-X

Original edition copyright 2002 by The McGraw-Hill Companies, Inc. (Publisher). All rights reserved.

Japanese edition copyright 2002 by SHOEISHA Co., Ltd. All rights reserved.

Japanese translation rights arranged with The McGraw-Hill Companies, Inc. through Japan UNI Agency, Inc., Tokyo.

CONTENTS

献辞	xiii
はじめに	xiv
謝辞	xvii

PART I メソッドとツールの概要

CHAPTER 1 Oracleのチューニングの概要	3
全体的なチューニング	4
サーバーのチューニング	8
ネットワークのチューニング	9
ディスクのチューニング	10
インスタンスのチューニング	11
オブジェクトのチューニング	13
SQLのチューニング	14
Oracleパラレル問合せのチューニング	14
Real Application Clusters (RAC) のチューニング	15
OracleとSTATSPACK	15
まとめ	18

CHAPTER 2	STATSPACK の概要	19
	STATSPACK のアーキテクチャ	21
	STATSPACK のデータ収集方法	21
	STATSPACK 表の構造	24
	STATSPACK コントロール表	25
	STATSPACK パラメータ表	27
	STATSPACK 情報の使用	28
	STATSPACK でのデータベースチューニング	29
	リソース計画	30
	予測モデルの作成	31
	まとめ	32
CHAPTER 3	STATSPACK のインストールと構成	33
	STATSPACK スクリプトの概要	34
	Oracle8i R8.1.6 以前の STATSPACK スクリプト	35
	Oracle8i R8.1.7 以降の STATSPACK スクリプト	37
	ステップ 1 : PERFSTAT ユーザーのデフォルト表領域の作成	38
	ステップ 2 : インストールスクリプトの実行	39
	インストールの前提条件	39
	ステップ 3 : STATSPACK インストール後のテスト	40
	ステップ 4 : 自動 STATSPACK データ収集のスケジュール設定	41
	STATSPACK の構成とメンテナンス	43
	STATSPACK スナップショットの表示	43
	STATSPACK 収集しきい値の調整	45
	スナップショットのレベル	45
	スナップショットのしきい値	46
	Oracle8 ~ Oracle8i R8.1.5 での STATSPACK の使用	48
	古い STATSPACK スナップショットの削除	48
	スナップショットを sppurge.sql スクリプトで削除	48
	スナップショットを手動で削除	49
	スナップショットをバッチモードで削除	49
	スナップショットを簡単な UNIX スクリプトで削除	50
	スナップショットをインテリジェント UNIX スクリプトで削除	52
	便利な STATSPACK シェルスクリプト	57
	経過時間 STATSPACK クイックスクリプト	57
	STATSPACK 表のエクステントの監視	63
	まとめ	65
CHAPTER 4	STATSPACK 表のデータ	67
	STATSPACK で取得できない情報	68
	STATSPACK 表の構造	68

STATSPACK 集計表	69
Oracle9iで STATSPACK 表に加えられた変更	70
stats\$latch_misses_summary 表	72
stats\$sgastat_summary 表 (Oracle8iのみ)	73
stats\$sql_summary 表	75
stats\$parameter 表	77
STATSPACK システム表	78
stats\$rollstat 表	79
stats\$latch 表	80
stats\$latch_children 表	81
stats\$librarycache 表	81
stats\$waitstat 表	82
stats\$enqueustat 表	83
stats\$sysstat 表	85
stats\$sesstat 表	87
stats\$sgastat 表	88
STATSPACK トランザクション表	88
stats\$buffer_pool 表 (Oracle9iより前のみ)	89
stats\$buffer_pool_statistics 表 (Oracle9i以降)	89
stats\$filestatxs 表	91
STATSPACK イベント表	93
stats\$system_event 表	94
stats\$session_event 表	96
stats\$bg_event_summary 表	97
stats\$idle_event 表	97
Oracle Parallel Server 表 (Real Application Clusters)	98
stats\$rowcache_summary 表	98
stats\$sgaxs 表	99
まとめ	99

PART II STATSPACK によるデータベースの 外部環境チューニング

CHAPTER 5 STATSPACK 拡張によるサーバー統計の収集 .. 103

vmstat ユーティリティの概要	104
vmstat での言語の違い	105
vmstat の内容	106
vmstat での CPU ボトルネックの確認	107
vmstat での CPU 使用量の確認	110
メモリーボトルネックの識別	111
UNIX でのメモリーページング	111
STATSPACK 内へのサーバーパフォーマンスデータの取得	118
vmstat 情報を取得するスクリプト	118
vmstat 取得スクリプトの仕組み	122

	他の Oracle サーバーに関する vmstat 情報のレポート	123
	まとめ	124
CHAPTER 6	サーバー環境のチューニング	125
	データベース管理者とシステム管理者の関係	126
	オンラインサーバー監視ツール	127
	glance を使用したサーバー監視	127
	top を使用したサーバー監視	128
	sar を使用したサーバー統計監視	130
	サーバー CPU 消費の監視	133
	サーバー全体のアップグレード	135
	CPU プロセッサの追加	135
	サーバータスクのロードバランシング	136
	nice 及び priocntl を使用した実行優先順位の変更	139
	サーバーメモリー消費の監視	141
	サーバーメモリー設定	142
	大規模メモリーと Oracle	142
	Oracle メモリーのスワップ不可能への変更	143
	サーバー統計のレポート	144
	サーバー例外レポート	144
	毎日の vmstat 傾向レポート	148
	長期間にわたるサーバーの傾向分析	152
	まとめ	152
CHAPTER 7	ネットワーク環境のチューニング	153
	Oracle Net 構成の最適化	154
	protocol.ora の tcp.nodelay パラメータ	155
	sqlnet.ora の automatic_ipc パラメータ	156
	tnsnames.ora の SDU と TDU パラメータ	156
	listener.ora の queuesize パラメータ	158
	sqlnet.ora の break_poll_skip パラメータ	158
	sqlnet.ora の disable_oob パラメータ	159
	epc_disabled 環境変数	159
	ネットワーク動作に影響するその他の Oracle 機能	160
	配列フェッチによるネットワークスループットの向上	160
	マルチスレッドサーバーの使用	161
	接続プーリングとネットワークのパフォーマンス	162
	ODBC とネットワークパフォーマンス	164
	Oracle レプリケーションのチューニング	165
	Oracle STATSPACK からのネットワークパフォーマンスの監視 ...	168
	分散ネットワークのチューニング	170
	netstat によるネットワークアクティビティの監視	171
	まとめ	173

CHAPTER 8	STATSPACKでのディスクI/Oサブシステムのチューニング	175
	ディスクI/Oに影響するOracleチューニング要素	177
	Oracleの内部機能とディスクI/O	178
	Oracleファイル編成テクニック	178
	一時的なディスクホットスポット	179
	Oracleディスクアーキテクチャのマッピング	186
	複数のメモリーバッファに関する問題	187
	Oracleでのファイルのストライプ化	188
	OracleでのRAIDの使用	189
	RAWデバイスでのOracleの使用	190
	Oracleデータベースとのディスクロードバランシング	191
	表領域とデータファイルの構成	192
	Oracleファイル-ディスクアーキテクチャの構築	194
	Oracleディスクアーキテクチャのレポート	195
	OracleデータファイルのSTATSPACKレポート	196
	STATSPACKのディスクとファイルI/Oの詳細	203
	特定のI/Oアクティビティに関するSTATSPACKレポート	204
	ホットデータファイルを識別するSTATSPACKスクリプト	206
	ホットディスクの検索	207
	STATSPACK拡張によるディスクI/Oデータ取得	208
	基本iostatユーティリティ	208
	STATSPACK表の定義	211
	iostat情報の取得	211
	iostatレポートの作成	213
	STATSPACKでのI/Oの流れの表示	217
	まとめ	222

PART III STATSPACKによる データベース内部のチューニング

CHAPTER 9	インスタンスのチューニング	225
	Oracleデータベースインスタンスの概要	226
	SGA	227
	Oracleバックグラウンドプロセス	232
	ブロックサイズとディスクI/O	234
	db_file_multiblock_read_countとOracle	234
	データバッファのチューニング (Oracle7～Oracle8i)	235
	データブロックキャッシングの概要	235
	フルデータキャッシング	238
	データバッファヒット率	239
	データバッファプールの内部	241
	STATSPACKによるデータバッファプールの監視	244

Oracle8iのデータプールの概要	247
KEEP プールに割り当てる表と索引の特定	249
RECYCLE プールのチューニング	255
RECYCLE プールへの割り当て候補の特定	256
STATSPACKによるデータバッファヒット率の傾向レポート ...	260
1時間ごとの平均データバッファヒット率	260
1日ごとの平均データバッファヒット率	263
Oracle9iデータバッファプールのチューニング	265
七つのデータバッファヒット率	266
Oracle8 データベースライタープロセスのチューニング	272
Oracle8 のパラメータ	273
STATSPACK によるデータベースライターの監視 (R8.1.5 より前) ...	274
Oracle8i/9iにおけるデータベースライターの競合の監視	281
共有プールのチューニング	282
cursor_sharing と共有プール	283
イベントの待機と STATSPACK	285
STATSPACK によるバックグラウンドイベントの監視	287
ライブラリキャッシュのチューニング	289
ライブラリキャッシュのミス率の監視	290
STATSPACK によるライブラリキャッシュ内部のオブジェクトの監視	292
ディクショナリキャッシュのチューニング	297
ソートのチューニング	303
UNDO レコード (ロールバックセグメント) のチューニング	309
専用接続の監視	313
UNIX とマルチスレッドサーバーの相互作用	317
MTSの使用に関する前提条件	317
Oracle9i動的メモリーと UNIX	326
Oracle9iと UNIX グラニクル	328
SGA コンポーネントと PGA コンポーネントの動的変更	329
専用接続に対する PGA メモリーの割り当て	330
Oracle9iにおける自動メモリー管理	331
Oracle9iに見られる自動チューニングへの取り組み	338
まとめ	347

CHAPTER 10 表と索引のチューニング

Oracle9iの自動領域管理	350
ビットマップデータ構造に向けた取り組み	350
Oracle9iの空きリスト管理	352
ビットマップセグメント管理	353
Oracle9iの空きリスト内部機能	354
Oracle9iの自動領域管理ツール	358

従来のOracle 記憶域パラメータとパフォーマンス	361
pctfree 記憶域パラメータ	361
pctused 記憶域パラメータ	362
freelists 記憶域パラメータ	363
OPS 用 freelist groups 記憶域パラメータ	363
記憶域パラメータに関する法則の概要	363
従来の空きリスト管理と Oracle オブジェクト	364
空きリストとのリンクとリンク解除	366
空きリストの再リンクの低減	369
表の内部と空きリスト	369
長いデータ型の列と空きリストの動作	371
平均行長に基づいた pctfree と pctused の設定	373
buffer busy waits と空きリストの競合	375
STATSPACK を使用した待機競合の検出	375
STATSPACK による buffer busy waits の検出	378
表の再編成	387
CTAS を使った表の再編成	389
連鎖行を持つ表の検出	395
長い行を持つ表の検出	399
疎表の検出	401
表の行の並べ替えによるパフォーマンスの向上	403
索引再構築のテクニック	406
索引の再構築が必要な状況	412
索引の再構築の自動化	412
Oracle9i における未使用索引の検出	414
索引を監視するセッションのサンプル	414
隠しパラメータを用いた索引競合のチューニング	415
STATSPACK による表と索引の監視	416
STATSPACK 拡張表の割り当て	416
表と索引の STATSPACK スナップショットの収集	418
表と索引に関するレポート	421
電子メールによる表レポートの配信	439
まとめ	441
CHAPTER 11 SQL のチューニング	443
SQL チューニングの目標	444
宣言型 SQL 構文の問題	445
SQL オプティマイザ	447
optimizer_mode = RULE	447
optimizer_mode = FIRST_ROWS	447
optimizer_mode = ALL_ROWS	447
optimizer_mode = CHOOSE	449

ルールベースの最適化によるチューニング	451
ルールベース駆動表の変更	451
ルールベースオブティマイザが正しい索引を使用しない場合	452
コストベースの最適化によるチューニング	453
コストベースオブティマイザの呼び出し	455
CBO用の統計情報の収集	455
optimizer_mode のデフォルトの決定	457
チューニングのさまざまなテクニック	458
CBO SQL ヒントによるチューニング	459
SQL 副問合せのチューニング	459
リテラル SQL 文の問題	460
Oracle8iのcursor_sharingの使用	464
一時表による SQL のチューニング	464
索引の追加による SQL のチューニング	465
効率的な SQL を記述するための一般則	466
SQL チューニングのプロセス	467
ステップ1: ライブラリキャッシュ内の影響の大きいSQLの特定 ..	468
ステップ2: SQL 文の抽出と解釈	478
ステップ3: SQL 文のチューニング	484
実際の SQL チューニングのケーススタディ	484
フルテーブルスキャンレポートの取得	485
SQL 実行計画の高度な分析	487
表のヒストグラム	491
チューニング済み SQL の変更の永続化	492
SQL のストアアウトラインの使用	492
v\$sql_plan ビューと v\$sql_workarea ビューの使用	493
まとめ	496

CHAPTER 12 Oracle のパラレル化機能によるチューニング ... 497

Oracle Parallel Query の使用	499
パラレル問合せのパラメータ	500
最適な並列度の設定	501
パラレル問合せヒントの利用	505
Oracle Parallel Query の監視	506
STATSPACKによるOracle Parallel Queryの監視	506
v\$ビューによるOracle Parallel Queryの監視	508
パラレル問合せと分散オブジェクト	508
Oracle Parallel Queryの候補となる表の特定	509
パラレルDML の使用	511
パラレルDMLによるチューニング	512
表の再編成のパラレル化	513
索引の再作成のパラレル化	513

まとめ	515
CHAPTER 13 Oracle Parallel Server環境のチューニング	517
Oracle クラスターサーバーアーキテクチャの概要	518
RACのためのデータのパーティショニング	521
統合分散ロックマネージャ (Oracle7からOracle8iまで)	522
Oracle8iのOPS環境内でのIDLMの構成	523
OPS使用時の表の設定	525
RAC環境のチューニング	528
RAC監視用のSTATSPACK表	530
stats\$rowcache_summary表	530
stats\$sgxs表	530
stats\$sysstat表	531
STATSPACKの拡張によるOPS情報の取得	533
RAC用のv\$ビュー問合せ	536
Oracle Parallel ServerとReal Application Clusters	538
Transparent Application Failoverの監視	540
まとめ	541
PART III STATSPACKによるデータベースのレポート ...	223
CHAPTER 14 Oracle Parallel Server環境のチューニング	517
標準STATSPACKレポート	546
先頭セクション	546
待機イベントのセクション	548
SQLセクション	550
表領域アクティビティのセクション	555
アラートスクリプトの概要	565
DBA用にカスタマイズされた例外アラートレポート	566
デイリーSTATSPACKアラートレポート	567
デイリーサーバーアラートレポート	577
Oracleの問題のリアルタイムチェック	579
ウィークリーオブジェクト拡大レポート	580
トレースアラートレポート	583
Webサーバーアラートレポート	586
Buffer Busy Waitsアラート	587
STATSPACKリアクティブレポート	590
Oracleアラートレポートのスケジューリングとカスタマイズ	595
UNIXのcrontabによるOracleレポートとアラートのスケジューリング例	595
まとめ	596

CHAPTER 15 STATSPACKによる傾向分析	597
Microsoft Excelを使ったSTATSPACKデータのグラフ化	598
ExcelのグラフウィザードによるSTATSPACKデータのグラフ化	598
ステップ1：STATSPACKレポートのカスタマイズ	599
ステップ2：SQL*Plusでのレポート実行	600
ステップ3：結果の選択とコピー	601
ステップ4：Excelの起動とデータの貼り付け	602
ステップ5：データの複数列への分割	602
ステップ6：データを列に区切る形式の指定	602
ステップ7：グラフウィザードの起動	604
ステップ8：折れ線グラフの選択	604
ステップ9：グラフウィザードの終了	604
ステップ10：近似曲線の追加	605
予測に活用できるSTATSPACKレポート	606
傾向分析の基本的なSTATSPACKメトリック	607
STATSPACK 拡張によるデータベースサーバーの傾向分析	607
サーバー傾向の時間ごとの調査	611
曜日単位でのサーバー統計のグラフ化	613
STATSPACKデータのWebベースのグラフ化	614
RRDtool製品	614
その他のWebベースのグラフ作成ツール	615
STATSPACK Viewer	617
まとめ	622
 INDEX	 623

執筆中、愛情を持って私を支えてくれた妻ジャネットにこの本を捧げます。

まえがき

STATSPACKユーティリティが最初にリリースされて以来、STATSPACKデータを使用してデータベースのチューニングを支援する方法は大きな関心事となってきました。STATSPACKは他のOracle製品のように大々的に鳴り物入りで発表されたわけではないので、Oracleプロフェッショナルのほとんどがこの強力な新ツールに気付いておらず、また、STATSPACKを使ったデータベースのチューニング法も全く知られていませんでした。

STATSPACKは、Oracleの統計スナップショットを表に取得するシンプルなツールです。UTLBSTAT/UTLESTATユーティリティの後継にあたります。このユーティリティは開始時及び終了時のスナップショットを取得し、この二つのスナップショット間のすべてのデータベースアクティビティを示すレポートを作成するものでした。

Oracleのパフォーマンス統計を表に取得するというニーズは常にありました。UTLESTAT機能を拡張してOracleデータを表に取得するユーティリティを作成し、1996年に雑誌で発表して以来、私はOracleデータベースをリモートで監視し、そのパフォーマンスをレポートする独自のユーティリティ/STATSPACKの前身に相当するものを作成してきました。STATSPACKを導入すると、だれもがOracle統計を表に取得する標準メカニズムを利用できます。そして、統計のアーカイブ履歴を使って、指定した期間のパフォーマンスについて比類ない分析が可能です。更に重要なのは、データベースの今後の動作を予測する傾向分析レポートを作成できるという点です。

STATSPACKの基本インストールと設定は実にシンプルです。私は統計を分析してきた長年の

経験から、STATSPACK表のデータを読み込む既存のスクリプトを変更して、とても便利なレポートを作成しました。

これらのスクリプトだけでも本書の価値は十分ですが、STATSPACKの出力の解析技術の詳細についても取り上げており、更に、STATSPACKレポートを使用してデータベースを適切にチューニングする方法も段階的に説明しています。

本書の目的は、私の20年のデータベースチューニングの経験を包括的論文として集大成することです。本書は、私にとってはOracleチューニングに関する3冊目の著書です。私がこれまでに収集してきた、データベースのチューニングに利用できる情報の量は相当なものです。統計量が多いと、持て余してしまうことが多いため、本書ではデータベースのパフォーマンス向上に関連する情報のみを取り上げています。

また、現在出版されているOracleチューニングに関する書籍は、事実上すべてのものが、既に発生したパフォーマンス上の問題を中心に説明していますが、本書は、STATSPACKユーティリティを使用して履歴傾向とパフォーマンスパターンを調べ、前もってデータベースをチューニングする方法に焦点を当てています。この事前チューニング方法を採用すると、DBAが問題に気が付いたときに処理を行うのではなく、あらゆる処理要件に合わせてチューニングを行うことができます。データベースにパフォーマンス上の問題が発生してからではDBAが施すことのできる処置はほとんどありません。Oracle Enterprise Managerパフォーマンスパックを使って、カスタマイズしたスクリプトをOracle v\$ビューに対して実行すれば、パフォーマンス上のボトルネックの原因を特定できるかもしれませんが、稼働中のデータベースをリアルタイムで変更して問題を修正できるわけではないのです。

本書は更に、STATSPACKユーティリティを使用して関連パフォーマンス統計を収集する方法や、Oracleコンポーネントとその相互作用についてDBAが確認できるようにする構造化する方法についても取り上げています。Oracle事前チューニングの目的は将来のパフォーマンス上の問題を回避することです。つまり、Oracleのチューニングでは、「過去を顧みることができない人々は再びそれを繰り返す」という言葉がそのまま教訓として当てはまるのです。

私は長年のデータベースチューニングの経験から、データベースのパフォーマンスを長期的に見た場合、この事前チューニングが最良の方法であると判断しました。私のデータベースでは、STATSPACKを使ってデータベースのパフォーマンスを常時監視し、異常な状態が発生したら警告する自動スクリプトを使用することに成功しました。つまり、Oracleデータベースパフォーマンスは、大部分が自動化されたタスクにできることが分かったのです。

本書のもう一つの特徴は、パフォーマンスの適切な依存順序が分かるようにその内容を整理したことです。例えば、Oracleデータベース内でチューニングを行っても、データベースサーバーに関連するパフォーマンス上の問題は軽減できません。言い換えれば、サーバーのパフォーマンスは、データベースサーバーで稼働しているOracleデータベースのパフォーマンスに直接影響を与えることになります。これらの依存関係は、データベースサーバー、ディスクI/Oサブシステム、及びネットワークに関連するところで存在します。Oracleデータベースを適切に調整するには、デー

データベースのチューニングを開始する前にこれらの外的要因を調整しなければなりません。

そのほかに、Oracle チューニングは順次実行するべきものではないことを把握することも重要です。例えば、SQL文を個別にチューニングすると、チューニングしたSQL文のパフォーマンスには大きな影響を及ぼします。しかし、個々のSQL文のメモリー使用量とデータベース全体のパフォーマンスにはトレードオフがあります。例えば、SQL文の内部記憶域が多ければ、その文にとっては好都合ですが、そのデータベース内で実行中のほかのすべてのSQL文を犠牲にします。

データベース設計に関する問題については触れていません。DBAは多くの場合、表を再設計できないからです。現在の開発環境に使用されているサードパーティ製のアプリケーションや社内のアプリケーション用のものとなればなおさらです。しかし、Oracleデータベースを適切に設計することが、Oracleパフォーマンスを向上させる最も重要な要因であるのは間違いありません。

なお、データベースサーバー、ディスクI/Oサブシステム、表の関連パフォーマンスメトリックを取得するには、STATSPACKユーティリティを拡張します。本書はこの拡張方法も説明しています。これらの要素を追加すれば、DBAは、データベースパフォーマンスの全体像を俯瞰できます。本書がほかのOracleチューニング関連の書籍と異なる点を次に示します。

- STATSPACKユーティリティの包括的チューニング方法を示す最初の書籍です。
- Oracleの事前チューニング方法について検証します。事後チューニング方法論を採用しているほかのOracleチューニング書籍とは異なり、まったく新しい角度からチューニング方法を捕らえています。事前チューニング方法を使用すると、DBAは過去のデータベースパフォーマンスの詳細を調べ、使用しているOracleシステムの処理特性に応じて最適なチューニングを立案できます。
- Oracleパフォーマンスに影響を及ぼす要因をすべて検証しています。他書の説明はOracleデータベース内部に関する内容に限定されていますが、本書は、データベースサーバー、ネットワーク、ディスクI/Oなど、関連する周囲の問題についても検証しています。

本書はOracleデータベースのチューニングの完璧な方法論を展開しており、重要な要因を要約した、すぐに利用できる多くのSTATSPACKスクリプトを紹介しています。従って、初心者でも素早くパフォーマンス上のボトルネックを発見できます。

最も素晴らしいのは、何と言ってもOracleチューニングを簡略化でき、統計の適用範囲を拡大している点でしょう。これは、Oracleチューニングテクニックに修正を加えただけの本とは異なります。STATSPACKユーティリティを導入すればOracleのパフォーマンス上の問題を収集、解析、修正することができ、チューニング方法も変わってきます。

本書により、日常のOracleチューニング業務のほかに、アラートメカニズム、傾向分析、長期リソース計画に関する進行中のパフォーマンス情報を収集する方法を習得することもできます。本書がOracleチューニングツールとして、皆さまにとって必要不可欠な武器になることを切に願っております。

謝辞

多くの技術書がそうであるように、この本も関係者各位のご厚意とご協力に支えられて、こうして完成にたどり着くことができました。構想やアイデアを一冊の本として結実させることができたのは、Lisa McClain氏とAthena Honore氏のおかげです。監修をお願いしたMike Ault氏、そして、編集の取りまとめ役でお骨折りいただいたBetsy Manini氏、その他組み版チームのメンバーのみなさんに心より感謝いたします。

サンプルコードの入手先

Oracle Press シリーズのサンプルコードは、原書出版社 McGraw-Hill/Osborne 社の Web サイトからダウンロードできます。コメントなどは日本語化されておりません。

<http://shop.osborne.com/cgi-bin/oraclepress/downloads.html>

- (1) 本書の原書『Oracle9i High-Performance Tuning with STATSPACK』のサンプルスクリプト:

http://www.osborne.com/products/007222360X/007222360X_code.zip

- (2) 『Oracle High-Performance Tuning with STATSPACK』—— Oracle8i に基づいて書かれた前の版のサンプルスクリプト:

http://www.osborne.com/products/0072133783/0072133783_code.zip

(1) のファイルは、(2) の `statspack.zip` を改訂したファイルです。従って、本書のサンプルスクリプトとしては、(2) の `statpack.zip` 以外のファイルと (1) のファイルをご利用ください。

The background of the entire page is a textured, marbled paper in shades of grey and beige, with irregular, vein-like patterns.

PART I

メソッドとツールの概要

CHAPTER 1

Oracleのチューニングの概要

チューニングは、Oracle データベースの管理の中でも最も難しい作業とされてきました。Oracle は柔軟性に優れているだけに内部ソフトウェアが非常に複雑で、パフォーマンス上の問題を発生させる原因を多くはらんでいるからです。その複雑さゆえ、適切にチューニングされているかどうかの確認がなされないまま運用されているデータベースも少なくありません。

Oracle というソフトウェアの複雑さのほかに、Oracle を利用するアプリケーションが動的であるという問題もあります。Oracle データベースは絶えず変化し続けており、全く同じ状態になることは二度とありません。Oracle 環境は動的なので、データベース内部の動作を掌握するのは非常に困難です。Oracle データベースのチューニングは、高速道路上で自動車を時速100キロで運転しながら、同時にその自動車の整備点検をするようなものなのです。

絶えず変化し続けるという Oracle データベースの性質はやっかいなものですが、そのチューニングにはやはり定石があり、多くの Oracle エキスパートもそれに従って作業を行っています。Oracle データベースのチューニングは、トップダウン方式で行うことが重要です。つまり、Oracle 環境全体を見渡せる上位レベルから開始し、データベースエンジン内の個々のコンポーネントをチューニングしながら順々にドリルダウンしていくのです。ここで言うトップダウン方式とは、サーバー、インスタンス、オブジェクト、最後にSQLという順番で検証していくことを意味します。

本書の目的は、Oracle データベースを最適レベルで動作させるために必要なSTATSPACKツールと診断方法について説明することです。本書を読めば、Oracle チューニング上の問題について深く理解し、STATSPACK 問合せを実行してパフォーマンスの尺度となる統計情報を取得する方法を知ることができます。

全体的なチューニング

Oracle データベースのチューニングには、万能の解決策はありません。しかし、包括的な方法を使えば基本的事項についてはすべてカバーすることができ、重要事項を見落とすことはありません。まず外部環境を見て全体像をとらえ、それから順に内部へと焦点を絞っていきます(図1-1を参照)。この手法をドリルダウンと言います。

ドリルダウンは Oracle チューニングにとっては非常に重要な考え方です。最初はごく広範なレベルで環境全体を検討し、CPU、メモリー、またはディスク構成に関する問題がデータベースサーバーに発生していないかどうかを確認します(図1-2を参照)。リソースが不足しているときは、Oracle データベースをいくらチューニングしても意味はありません。

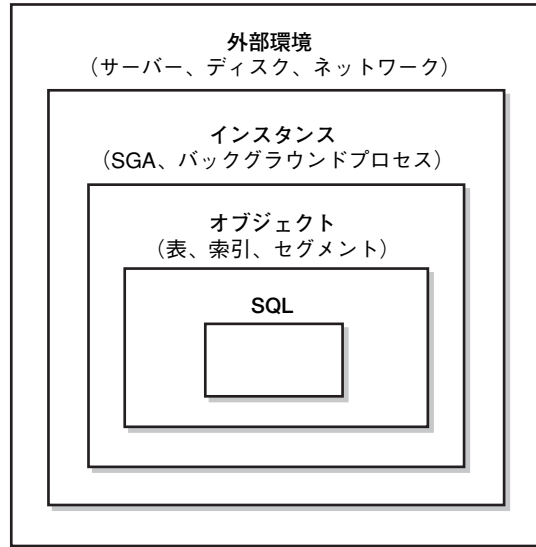


図 1-1 Oracleデータベースのチューニング階層

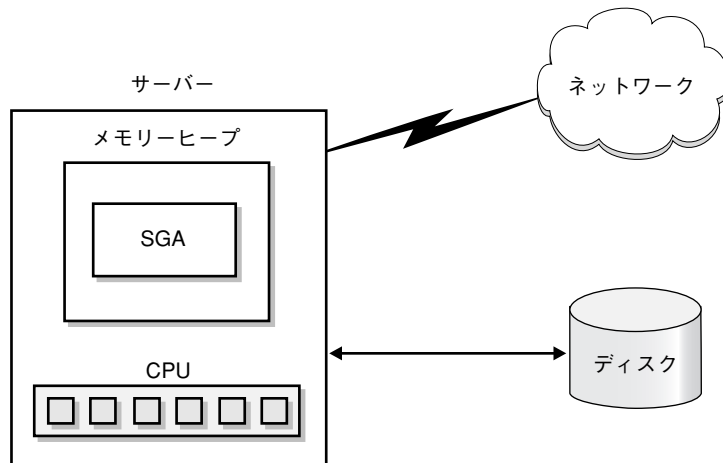


図 1-2 サーバー環境のチューニング

サーバー環境のチューニングが完了したら、データベース(インスタンス)に影響を及ぼすグローバルパラメータを調べます。Oracleデータベースを調べるときは、データベースを全体的にとらえ、SGA構成とデータベース全体の動作を管理する初期化パラメータに注目します(図1-3を参照)。

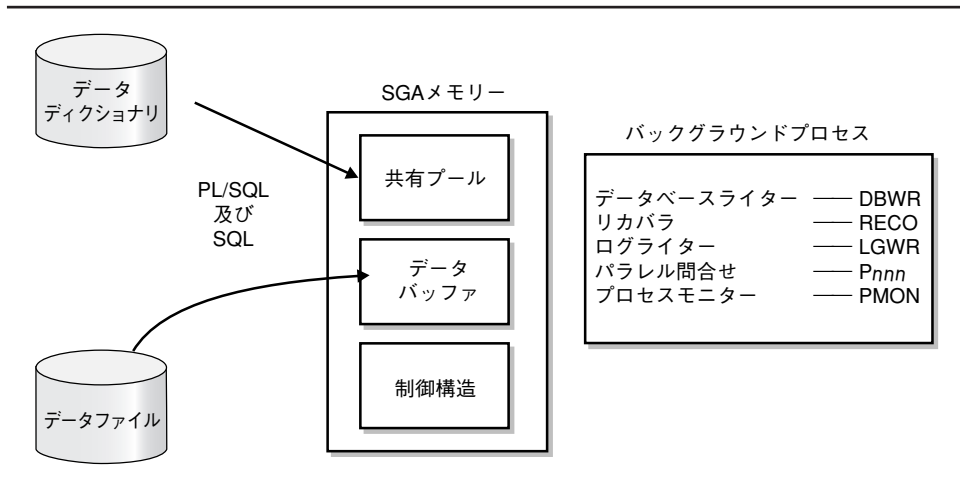


図 1-3 インスタンスのチューニング

インスタンス内のデータベースサーバーをチューニングしたら、次はデータベース内の個々の表や索引に注目します。このフェーズでは、表の動作を左右する記憶域の設定を確認し、その設定により個々のオブジェクトのニーズがどの程度満たされているかを調べます(図 1-4 を参照)。

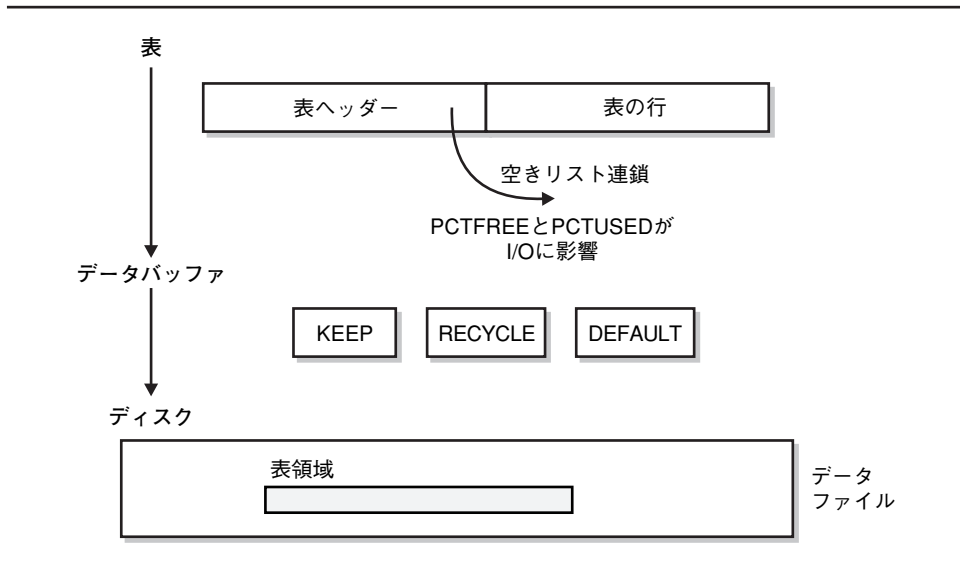


図 1-4 オブジェクトのチューニング

オブジェクトをチューニングしたら、次は、Oracleデータベースに対して発行される個々のSQL問合せをチューニングします。これは、チューニング作業の中で最も大変な作業の一つです。動作の活発なOracleデータベースに対しては数千ものSQL文が発行されるからです。

そこで考えられるのが、使用頻度の高いSQL文を識別し、各種ツールを使ってそれらをチューニングし、最適な実行計画が選択されるようにするという方法です(図1-5を参照)。後の章では、静的バインディングについて詳しく説明し、SQLのバインド計画を作成して処理効率を向上させる方法に注目します。

Oracleチューニング作業は以下の手順で行います。最初のステップが内容的には最も全般的で幅広く、次のステップに進むたびにその作業内容は絞られてきます。

1. サーバー、ネットワーク、ディスクをチューニングします。

Oracleサーバーに問題(CPUの過負荷、メモリスワッピングの大量発生、ディスクI/Oのボトルネックなど)がある場合は、Oracleデータベースをいくらチューニングしてもパフォーマンスは向上しません。Oracleのプロフェッショナルたらんとするのであれば、まず、サーバー、ディスク、ネットワーク環境を調べるのが先決です。

2. インスタンスをチューニングします。

SGAをチューニングし、すべての初期化パラメータを見直して、データベースが適切に構成されているかどうかを確認します。このフェーズでは、db_cache_size、shared_pool_size、pga_aggregate_targetのリソース不足に注目します。また、optimizer_modeなどの主要デフォルトパラメータの値も調べます。

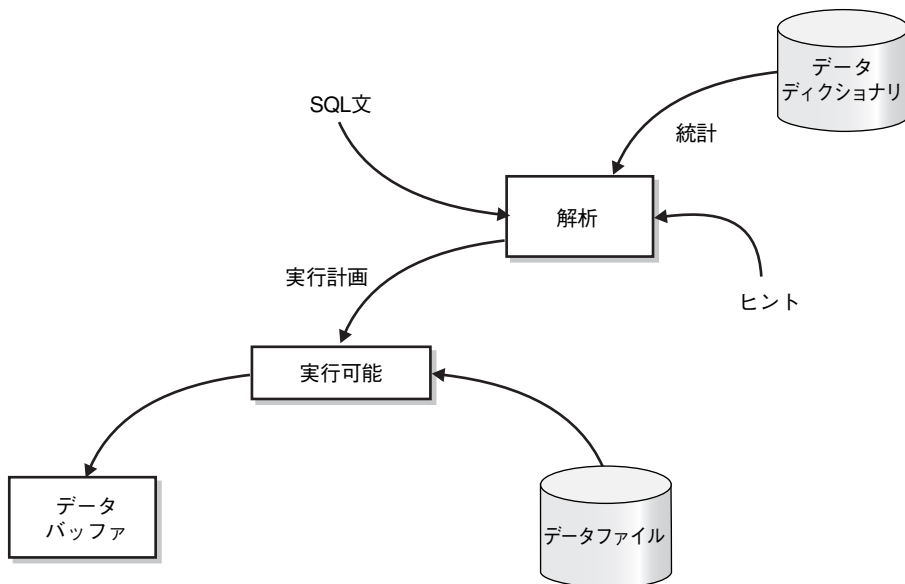


図1-5 SQLのチューニング

3. オブジェクトをチューニングします。

表や索引の設定を調べます。pctfree、pctused、freelistsなどの設定はOracleのパフォーマンスに大きく影響します。なお、記憶域を適切に設定すれば各オブジェクトがその恩恵を受けることとなります。

4. SQL文をチューニングします。

OracleデータベースにアクセスするSQL文は数千の単位にのぼることがあり、これはチューニング作業の中でも特に時間のかかるフェーズです。まず、使用頻度の高いSQL文を特定し、それらをチューニングします(特定したSQL文の実行計画を調べ、Oracleヒントに従って実行計画を修正します)。

オプティマイザプランスタビリティ機能を使うのも有効です。この機能を使うと、SQL文の実行計画をいつでも使えるよう保存しておくことができ、パフォーマンス向上につながります。OUTLINEパッケージでオプティマイザプランスタビリティ機能を実装し、SQL文に合わせて実行計画を変更するのもよいでしょう。これは、ソースコードを変更できないベンダー製SQLを使っている場合に特に便利です。例えば、Oracle ApplicationsとSAPでは、SQLを修正することはできません。しかし、オプティマイザプランスタビリティを使うと、保存されているSQL実行計画を修正することでSQLをチューニングできます。

5. 設計をチューニングします。

アプリケーションの設計はOracleのパフォーマンスを大きく左右する要因の一つです。しかし、設計がお粗末なものであったとしても、それをDBAが修正できることは不幸にしてあまりありません。ソースコードが公開されていないソフトウェアが使われていたり、その設計が既に実働環境で実装されていたりするからです。

Oracleのチューニングを成功させるには、作業を正しい順序で行うことが大切です。経験の浅いDBAは、SQLが実行されている環境を見ずにいきなりSQL文をチューニングしようとします。このような誤った方法を取ると本質的な問題を見逃し、無駄な努力を重ねることになるおそれがあります。

ここではまず、チューニングのさまざまな領域に一通り目を通します。個々の領域の詳細とそれがOracleのパフォーマンスに与える影響については、第2章以降で学習することになります。

サーバーのチューニング

サーバー環境をチューニング対象として認識していないOracleプロフェッショナルが多くいますが、それは彼らがOracleデータベースとサーバーの対話(やり取り)の仕組みを十分に理解していない証拠です。これは経験上言えることですが、パフォーマンス上の問題が発生した場合に第一に確認すべきなのがサーバー環境なのです。データベースサーバーのチェックでは、OSのカーネルパラメータの設定を逐一調べ、メモリーとCPUの使用量を監視する必要があります(図1-6を参照)。ディスクまたはネットワークがボトルネックになっている場合、あるいはハードウェアリソースが不足している場合は、Oracleをいくらチューニングしても事態は改善しません。

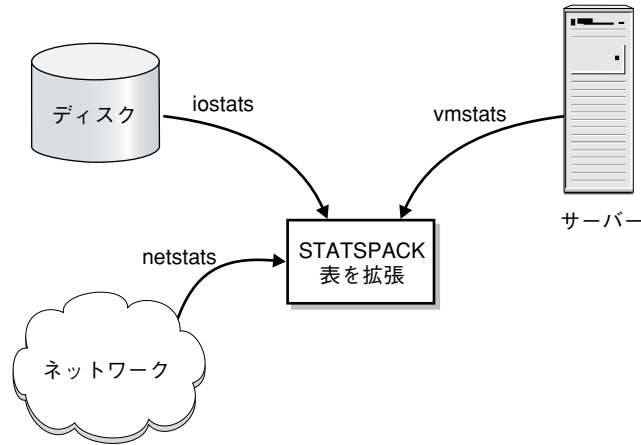


図 1-6 サーバートューニングの詳細

サーバー上の CPU 及びメモリーのおおよその消費状況が分かったら、次は STATSPACK ユーティリティを活用してサーバー情報を取得する方法について考えます。UNIX の `vmstat` コマンドを使ってコンピュータ内の CPU とメモリーの使用量に関する情報を取得し、それらの情報を STATSPACK 表に格納する方法に注目します。STATSPACK からアラートレポートを生成して、Oracle データベースのパフォーマンスに影響を及ぼすサーバー関連のすべての問題を管理者に通知する方法についても学習します。

更に、STATSPACK を使って Oracle エンタープライズ環境内の他のサーバーを監視する方法や、`vmstat` コマンドをリモートで実行して Oracle データベース環境内の Web サーバー及びアプリケーションサーバーの動作を監視する方法についても注目します。

ネットワークのチューニング

地理的に離れた環境で複数の Oracle データベースを共有している場合は、データベースのパフォーマンスに対するネットワーク通信の影響を認識することが大切です。ご存じかもしれませんが、Oracle では Transparent Network Substrate (TNS) によりデータベース間で分散通信が行えます。TNS は、リモートシステム間の透過的データベース通信を可能にする分散プロトコルで、Oracle の論理データ要求と分散サーバー間の物理通信を隔離する役割を果たしています。TNS により Oracle の論理データ要求とネットワークの内部動作が隔離されているため、ネットワークパフォーマンスチューニングのほとんどの作業はネットワーク管理者の手に委ねられます。つまり、データベース全体のパフォーマンスに影響を及ぼすネットワーク構成(図 1-7 を参照)を DBA が直接制御できることはほとんどありません。

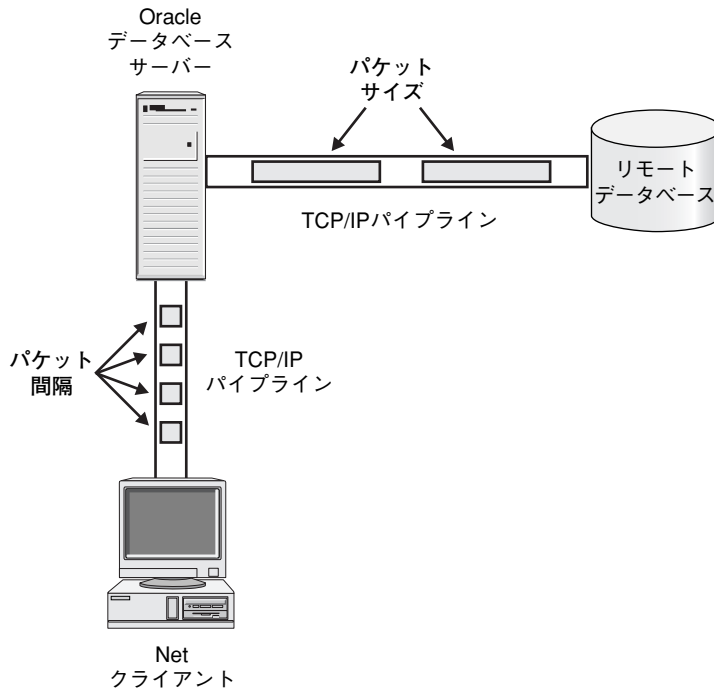


図 1-7 ネットワークのチューニング

しかし、分散トランザクションのパフォーマンス向上につながる重要な設定項目がいくつかあります。この章では、分散通信に関連する初期化パラメータと、`tcp.nodelay`などのTCPパラメータについて検証します(TCPパラメータにより、Oracleデータベース内の基本パケット転送メカニズムを変更できます)。

また、`sqlnet.ora`、`tnsnames.ora`、`protocol.ora`ファイルに定義されている、TCPパケットのサイズと構成に関する各種パラメータについても見ていきます。パラメータネットワーク/トランスポート層に大きな影響を及ぼすこれらのパラメータを変更することで、すべてのトランザクションのスループットを改善できることがあります。

ディスクのチューニング

Oracleの応答時間を左右する最大の要因がディスクI/O(入出力)です。いかなる方法であろうとディスクI/Oを削減できれば、確実にデータベースのパフォーマンス向上につながります。本書の中で、初期化パラメータの変更やSQL文のチューニングについて触れている部分がありますが、それらはすべてディスクI/Oを削減し、ひいてはデータベースのパフォーマンスを向上させることを目的としたものです。

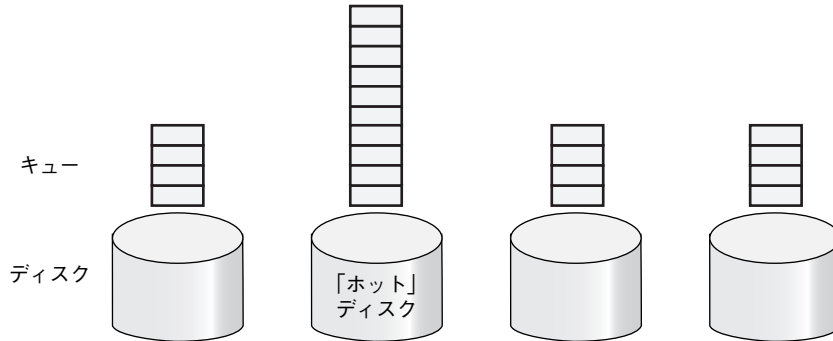


図1-8 ディスクI/Oのチューニング

Oracle データベースのチューニングの要点がディスク I/O にあるということは、即ち、ディスク I/O サブシステムの内部動作を完全に理解する必要があるということです。しかし、EMC などのディスクキャッシュ記憶デバイスが普及した現在、Oracle データベースとディスク I/O サブシステムのやり取り (図1-8を参照) にあまり注意を払っていないDBA もいるようです。また、ディスクアレイが独自のメモリキャッシュを持つことが多く、Oracle からの I/O 要求が必ずしも物理ディスク I/O として扱われるわけではないことも、話をややこしくしている問題の一つです。

ディスク I/O サブシステムを構成する方法は多数あります。そこで、第8章では、代表的なディスク配置技術と、RAID の使用方法、さらには論理マウントポイントから物理ディスクにディスクをマッピングする方法について考察します。物理ディスク内の Oracle データファイル間の対応を完全に把握できたら、次はディスク I/O サブシステムの動作を監視し、ファイルを移動してディスク I/O のボトルネックを解消することのできるツールの作成にとりかかります。

STATSPACK の使用という本来のテーマを念頭に置きつつ、第8章では STATSPACK 表を拡張して、UNIX の汎用的な `iostat` コマンドを使って I/O 統計を入手する方法に注目します。`iostat` コマンドを使ってディスクの動作を常時監視する体制が整ったら、次は、I/O サブシステムに競合が発生したことを自動的に警告する例外レポートを作成します。

また、第8章では、ディスク I/O のボトルネックを解消する方法について深く掘り下げ、さらに、すべての物理ディスク間で負荷を均等に分けるための手法として使われるファイルのストライプ化やその他の RAID 技術にも注目します。

インスタンスのチューニング

インスタンスのチューニングは、Oracle チューニングの中で最も誤解されやすい分野の一つです。事実、SGA に過大な負荷がかかるとパフォーマンス上深刻な問題が発生する可能性があります。しかし、この SGA に関しては1回チューニングしさえすれば、後はさほど注意を払う必要はありません。このセクションでは、STATSPACK の表から SGA 情報を直接取得し、Oracle のライブラリキャッ

シュ、共有プール、データブロックバッファの領域不足をDBAに自動的に警告するメカニズムについて考えてみましょう(図1-9を参照)。

インスタンスをチューニングするには、Oracleデータベースのすべての初期化パラメータをチェックする必要があります。ご存じのように、初期化パラメータは、Oracleデータベースが多機能化するほど複雑になっていきます。第9章では、個々の初期化パラメータの詳細と、最適なパフォーマンスを得るためにこれらの初期化パラメータをどう変更すべきかというガイドラインに注目します。

インスタンスにはデータバッファキャッシュが関与していますが、このデータバッファプールの動作をSTATSPACKで測定する方法についても学習します。

プールの種類には、DEFAULTプール、KEEPプール、RECYCLEプール、そして各種サイズのデータバッファ(db_2k_cache_size、db_4k_cache_size、db_8k_cache_size、db_16k_cache_size、db_32k_cache_size)などがあります。STATSPACKを使うと、これらのデータ記憶域バッファプールのサイズを拡大すべきタイミングを計り、最小公分母的なバッファプールサイズをチューニングすることができますが、その方法についても学習します。

SGA内の共有プールとライブラリキャッシュのチューニングにも注目します。更に、SGAの動作に関するSTATSPACK情報を収集する方法や、関連する初期化パラメータを調整してOracle共有プール内のオブジェクトの動作を最適化する方法について学習します。

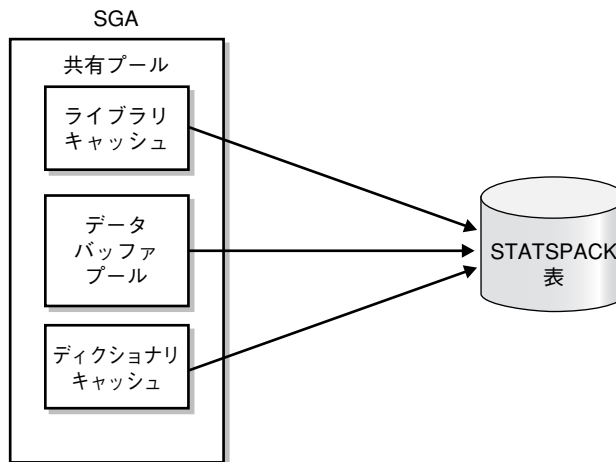


図1-9 SGAとバックグラウンドプロセスのチューニング

オブジェクトのチューニング

Oracleの表と索引の記憶域パラメータがデータベースのパフォーマンスに大きな影響を及ぼしているという事実は意外に認識されていません。このセクションでは、オブジェクト関連の記憶域パラメータを一つ一つ検証し、オブジェクトの動作に従って記憶域パラメータを設定する際のガイドラインを確認します。また、オブジェクトの断片化(図1-10を参照)にも注目し、データベースの再編成のコストを最小限に抑える方法をいくつか見ていきます。

第10章では、データブロックの内部に踏み込み、Oracleの空きリストのリンク設定とリンク解除の内部メカニズムを理解します。表での内部操作の詳細を理解すると、記憶域パラメータを最適に設定してOracleのパフォーマンスを向上させる方法が見えてきます。もちろん、この章でもSTATSPACKユーティリティに焦点を当て、データベース内の個々のOracleオブジェクトの動作に関する有用な情報を引き出す方法を学習します。また、オブジェクト上のブロック待機アクティビティを測定するSTATSPACK表を解剖し、パフォーマンスを向上させるために記憶域パラメータを調整すべきオブジェクトを見極めるスクリプトを検証します。特にPCTFREEとPCTUSEDというOracleオブジェクトパラメータに焦点を絞り、これらのパラメータの設定がSQLのINSERT及びUPDATEタスクのパフォーマンスにいかに関与するかを理解します。また、表内の空きリストの使用法に注目し、空きリストの競合が発生している表を特定する方法と、これらの表に対する大量のトランザクションのスループットを最大にするにはOracleオブジェクトの空きリストをどう設定すべきかを理解します。

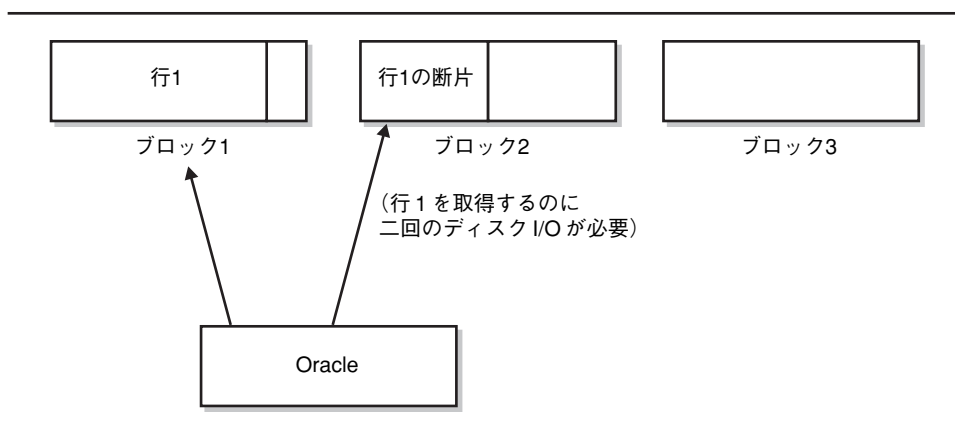


図 1-10 Oracleのオブジェクト、空きリスト、及び断片化

SQLのチューニング

個々のSQL文をチューニングしていく作業は、Oracleのチューニングの中でも最も時間のかかる部分です。時間のかかる作業ではありますが、Oracleシステム全体のパフォーマンスを向上させる最も効果的な方法ともいえます。パフォーマンスに対する影響の大きいものから順に、それぞれに適したチューニングテクニックを使ってパフォーマンスを向上させていくという手法もよく使われます。第11章では、使用頻度の高いSQL文を見極め、そのSQL文の動作を確実に最適化できる方法について学習します。

また、STATSPACKユーティリティを使ってライブラリキャッシュ内のSQLの動作を監視し、最適化されていない可能性のあるSQL文を定期的に通知する方法についても注目します。この通知は、stats\$sql_summary表のSQLソースを検証することによって実施されます。

ライブラリキャッシュに格納されているSQL文の管理についてのセクションもあります。ご存じのように、OracleインスタンスのSQL文は非常に変化しやすく、瞬間的にしかライブラリキャッシュに存在しない場合もあります。ライブラリキャッシュ内の情報は常に変化する可能性があります。ライブラリキャッシュに取り込まれたSQL文はどれも最終的にはエージアウトし、Oracleインスタンスで利用できなくなります。第11章では、SQL実行情報をライブラリキャッシュから取得する方法を学習します。この便利なテクニックを使うと、任意の時点でライブラリキャッシュに問合せを行い、Oracleライブラリキャッシュ内にあるすべての重要なSQL文の実行計画を詳細レポートにまとめることが可能です。

STATSPACKユーティリティでは、ライブラリキャッシュ情報を監視し、適切にチューニングされていないSQL文がOracleインスタンスで処理されている場合にそのことを示す自動アラームレポートを生成させることができます。

SQLのさまざまなオプティマイザモードについても注目します。コストベースオプティマイザ(CBO)の相対的なメリットとデメリットを洗い出すとともに、ルールベースのオプティマイザの機能と、それを使ってSQLのスループットを向上させる方法も検証します。さらに、コストベースオプティマイザのfirst_rowsモードとall_rowsモードの違いにも注目します。

SQLヒントを使って実行計画を変更し、SQL文を個別にチューニングする方法を考え、SQLチューニングの実例を検証します。これにより、SQL文の実行所要時間を数時間からわずか数分に短縮できるということがわかるはずですが、また、Oracleのパフォーマンスに影響する初期化パラメータに注目し、オプティマイザプランスタビリティ機能を使った静的SQLバインディングについて学習します。Oracle 8i以降に関しては、SQLのクエリライト機能と、この機能を使ってSQLパフォーマンスを向上させる方法についても検証します。

Oracle 並列問合せのチューニング

第12章では、Oracle 並列問合せに焦点を当て、大きな表に対してフルテーブルスキャンが実行されているときに並列化機能を起動するさまざまな手法に注目します。まず、大きな表でのフ

ルテーブルスキャンを検出する方法について学習し、その後で、これらの種類の間合せを並列化するときには使用するさまざまな手法を検証します。さらに、CPUの数と表のファイルに対するディスクの数に基づいて、個々の間合せのスループットと実行時間を最大限に効率化するには並列度をどのように設定すべきかについて考察します。

また、パラレルDML (Data Manipulation Language: データ操作言語) についても注目し、パラレル索引再作成などの操作によって、SQLのデータベースメンテナンスアクティビティ (UPDATE、INSERT、DELETE) の速度がいかに向上するかを理解します。第12章では、さらに、Oracleへの同時要求送信によって多数の大きなOracleタスクがどのように並列化されるかについても検証します。

Real Application Clusters(RAC)のチューニング

第13章では、Oracle7/8iのOracle Parallel Server (OPS) 環境とOracle9iのReal Application Clusters (RAC)環境に焦点を当て、この2つの環境がアプリケーションシステムに及ぼすメリットに注目します。また、ハードウェアのクラスタ化に必要な構成を考え、アプリケーションをパーティション化してOPS環境やRAC環境のスループットを向上させる方法について考察します。さらに、OPS環境とRAC環境におけるインスタンスの最適数を決定する方法を学習し、OPS用の統合分散ロックマネージャー (IDLM) に注目します。具体的には、IDLMパラメータの詳細に目を通し、STATSPACKを拡張して実働環境でのIDLMの動作情報を収集する方法を理解します。また、インスタンス間のpingを検証し、アプリケーション構造をパーティション化してpingを最小にする方法についても考察します。

チューニング方法については一通り目を通したので、次は、STATSPACKを使用してチューニング作業を円滑に行う方法を見ていきましょう。

Oracle と STATSPACK

本書の主な目的はOracleチューニングの実践的技術を紹介することですが、本文では一貫してSTATSPACKユーティリティの利用方法を示しています。パフォーマンスのチューニングを支援する手段としてSTATSPACKを使用するという概念は実に重要で、本書の主な焦点となっています。statsrep80.sqlを使えばSTATSPACKをOracle8～Oracle8i R8.1.5でも使えるようになりますが、その場合、STATSPACKにいくつか問題が発生することが分かっています。Oracle8i R8.1.6より前のリリースで作成したSTATSPACKレポートには不正確なデータが含まれていることがあり、その信頼性には不安があります。また、Oracle8i R8.1.7とOracle9iとはSTATSPACK表の構造も異なります。STATSPACKの移行手順については第3章を参照してください。

第2章では、まず、STATSPACKユーティリティの概要を理解し、STATSPACKをインストール及び設定してOracleサーバーのパフォーマンスを測定する方法を学習します。STATSPACK表のインストール手順やSTATSPACK全体の表の構造に目を通し、サーバーのパフォーマンスを知る上で重要な手がかりとなる統計値に注目します。

STATSPACKのスナップショットにも注目し、長期間にわたって時間帯別の統計値の平均を求め、レポートとして出力する方法を学習します。この方法は、STATSPACKの情報をスライスして目的の情報を取得することのできる、Oracleプロフェッショナルにとっては不可欠のテクニックです。

さらに、STATSPACKに対する問合せの例(実際に実行可能なもの)を検証し、STATSPACKから取得した情報に基づいて、ある時点においてデータベース内で何が発生したかを識別する方法を学習します。

本書では、あらかじめ用意されているSTATSPACKスクリプトを利用していますが、Oracleプロフェッショナルにとっては、STATSPACK問合せの性質を理解し、Oracle環境で発生した問題に基づいて独自の問合せを作成できることが実に重要です。本書では多数のスクリプトを取り上げますが、これらのレポートを拡張し、そのレポートを使ってシステムの詳細を確認していくことをお勧めします。

STATSPACKユーティリティはOracleデータベースのドメインに制限されています。そこで本書では、STATSPACKを拡張して外部環境の情報を入手する方法をいくつか学習します。まず、STATSPACKを拡張してサーバーメトリックを入手する方法に注目します。第5章では、UNIXのvmstatコマンドを使ってサーバー情報を取得し、その情報をSTATSPACK拡張表に格納する方法を学習します。さらに、パフォーマンスの問題が発生した場合にこれらの表に問合せて、データベース内部で何が起こったのかを確認する方法も検証します。

STATSPACK表を拡張してディスクI/Oについての情報を入手することもできます。第8章では、ディスクI/Oサブシステムのチューニングに焦点を当て、ディスクI/Oのサブシステムの性質や、Oracleのファイル配置とディスクI/Oとの相互作用を理解するとともに、STATSPACKを拡張してI/Oサブシステム内の動作を時間経過に沿って監視する方法を学習します。

STATSPACKを拡張して長期間の統計傾向レポートを取得する方法にも注目しましょう。第14章と第15章では、重要な統計値を時間経過に沿ってプロットしたり、既存の統計を予測モデルとして使ってデータベースリソースの追加計画を立てたりすることのできる便利なスクリプトについて検証します。第15章で説明するテクニックは統計学の知識がなくても理解できますが、DBAたるもの、Oracleのパフォーマンスメトリックと、そのメトリックが時間経過に沿ってどのように変化するかについて一般的な知識を持っている必要があります。

STATSPACKを使ってOracleの傾向分析を行うときは、さまざまなディメンション(次元)に基づいて情報を集計することになります。この点を理解しておくことが重要です。例えば、曜日ごとに情報を集計するスクリプトを使用すると、システムにかかっている相対的な負荷を曜日ごとに確認できます(図1-11を参照)。

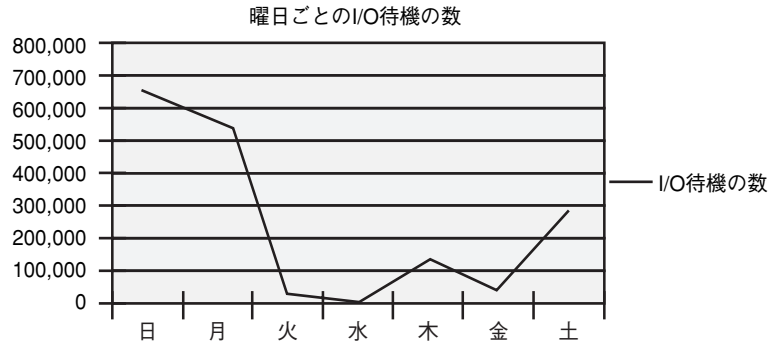


図 1-11 曜日平均レポート

また、STATSPACK情報を時間ごとに分析するスクリプトも使用できます。例えば、どの時間帯のエンドユーザーの処理がOracle データベースに負荷をかけているかを時間単位で確認できます(図 1-12 を参照)。

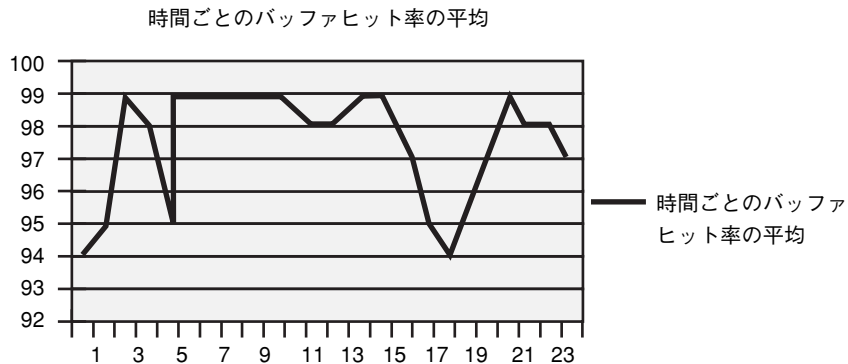


図 1-12 時間ごとのアクティビティの平均

曜日ごとあるいは時間ごとに情報を測定して、週及び月単位でのデータベース全体のパフォーマンスを推定するには、いつも同じテクニックを使用します。STATSPACKの要約情報を取得して、その情報を統計分析ルーチンに入力すれば、予測モデルを使って線形回帰を実行し、既存のメトリックに基づいてOracle データベースの今後のパフォーマンスを予測できます。

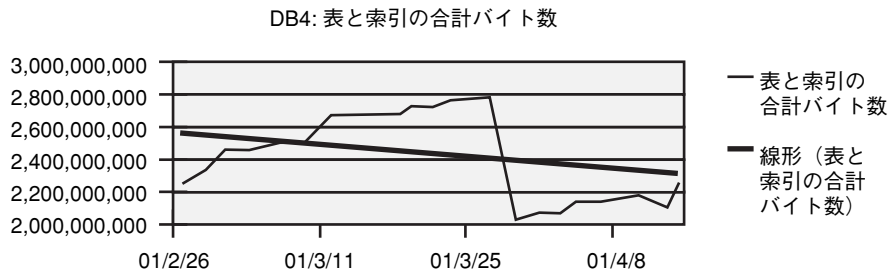


図 1-13 Microsoft Excelを使用した予測STATSPACK傾向分析の例

本書では、線形回帰のさまざまな手法を簡単に紹介します。線形回帰には、例えば指数、2次、及び3次多項式や、全体的なOracleパフォーマンスの予測に使われる、より一般的な最小二乗法などがあります。

長期間にわたる傾向レポートは、Oracleシステムに必要なハードウェアリソースの量を予測する上で特に役に立ちます。本書では、将来のディスク記憶域要件、メモリー要件、及びCPU要件を正確に予測するスクリプトを示します。これらの予測はすべて時間経過に沿ったSTATSPACK表の統計に基づいています。

STATSPACK情報をグラフで表す場合は、ほぼすべてのDBAが利用できるツールを使用することが重要です。SASやSPSSなどの強力な統計パッケージを使用すれば信じられないほど細かい統計分析ができますが、本書ではあえてMicrosoft Excelワークシートを使用することにしました。Excelを使用した例に限定することにより、データベースの動作を抽出、グラフ化、予測する方法を、ほとんどのデスクトップに用意されているツールを使って分かりやすく示しています(図1-13を参照)。

STATSPACKは、ある時点におけるOracleデータベースの情報を事実上すべて取得するように設定されています。従って、STATSPACK表で取得される有用なメトリックは何百にも上ります。これらのメトリックを一つ一つ取り上げていきたいのはやまやまですが、Oracleプロフェッショナルにとっては、Oracleシステムをチューニングする際に最も重要となるサーバーメトリックだけに着目することの方が重要でしょう。従って、本書では、システムのパフォーマンスとチューニングについての合理的な決断を素早く下すために必要な情報を提供するメトリックについてのみ言及します。

まとめ

本書では、全体にわたってさまざまなSTATSPACK表を取り上げながら、この章で説明した概念に関連する値を調査していきます。従来のOracleチューニングテクニックとSTATSPACKを組み合わせて解説するというのは本書のほかに類を見ない特徴です。

次の章では、いよいよSTATSPACKユーティリティの概要について説明していきます。