

Oracle9i Real Application Clusters

配置およびパフォーマンス

リリース 2 (9.2)

2002 年 7 月

部品番号 : J06274-01

ORACLE®

Oracle9i Real Application Clusters 配置およびパフォーマンス, リリース 2 (9.2)

部品番号 : J06274-01

原本名 : Oracle9i Real Application Clusters Deployment and Performance, Release 2 (9.2)

原本部品番号 : A96598-01

原本著者 : Mark Bauer

原本協力者 : David Austin, Kotaro Ono, Stefan Pommerenk, Joao Rimoli, Michael Zoll, Wilson Chan, Sashikanth Chandrasekaran, Mitch Flatland, Rick Greenwald, Bill Kehoe, Merrill Holt, Raj Kumar, Neil MacNaughton, Vinay Srihari, Tak Wang, Valarie Moore

Copyright © 1999, 2002, Oracle Corporation. All rights reserved.

Printed in Japan.

制限付権利の説明

プログラム (ソフトウェアおよびドキュメントを含む) の使用、複製または開示は、オラクル社との契約に記載された制約条件に従うものとします。著作権、特許権およびその他の知的財産権に関する法律により保護されています。

当プログラムのリバース・エンジニアリング等は禁止されています。

このドキュメントの情報は、予告なしに変更されることがあります。オラクル社は本ドキュメントの無謬性を保証しません。

* オラクル社とは、Oracle Corporation (米国オラクル) または日本オラクル株式会社 (日本オラクル) を指します。

危険な用途への使用について

オラクル社製品は、原子力、航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションを用途として開発されておりません。オラクル社製品を上述のようなアプリケーションに使用することについての安全確保は、顧客各位の責任と費用により行ってください。万一かかる用途での使用によりクレームや損害が発生いたしましても、日本オラクル株式会社と開発元である Oracle Corporation (米国オラクル) およびその関連会社は一切責任を負いかねます。当プログラムを米国防総省の米国政府機関に提供する際には、『Restricted Rights』と共に提供してください。この場合次の Notice が適用されます。

Restricted Rights Notice

Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

このドキュメントに記載されているその他の会社名および製品名は、あくまでその製品および会社を識別する目的のみ使用されており、それぞれの所有者の商標または登録商標です。

目次

はじめに	ix
------------	----

第 I 部 Real Application Clusters の配置の概要

1 Real Application Clusters アプリケーションの配置の概要

Oracle9i 機能の活用	1-2
Real Application Clusters での Oracle9i 機能の実装	1-2
Real Application Clusters の記憶域	1-2
Real Application Clusters での高可用性およびフェイルオーバー	1-3
Real Application Clusters での共有サーバー	1-4
接続時ロード・バランシング	1-5
Real Application Clusters での透過的アプリケーション・フェイルオーバー	1-5
Real Application Clusters での PL/SQL	1-6
Real Application Clusters での Recovery Manager (RMAN)	1-7

2 Real Application Clusters でのアプリケーションの配置

Real Application Clusters ベースのアプリケーションの配置方法	2-2
Real Application Clusters を使用した E-Commerce アプリケーションの配置	2-2
キャッシュ・フュージョンによる柔軟な実装	2-3
Real Application Clusters でのデータ・ウェアハウス・アプリケーションの配置	2-4
Real Application Clusters でのデータ・ウェアハウス・アプリケーションのスピードアップ	2-4
パラレル問合せの動的な最適化	2-5

インターネット・ベースのアプリケーションの配置	2-8
Real Application Clusters でのアプリケーションのパフォーマンス	2-8
Real Application Clusters に対するシステム拡張性の管理	2-9

第 II 部 Real Application Clusters での配置およびパフォーマンス

3 Real Application Clusters でのデータベースの配置方法

Real Application Clusters に対するデータベース配置の原理	3-2
Real Application Clusters での表領域の使用	3-2
Real Application Clusters でのオブジェクトの作成およびパフォーマンス	3-3
オブジェクトのパーティション化	3-3
Real Application Clusters での順序番号の使用	3-4
順序におけるグローバル競合の検出	3-4
データベース表を使用した順序番号の生成	3-4
Real Application Clusters のアプリケーションのチューニングに関する推奨事項	3-5
問合せのチューニングのヒント	3-5
トランザクション処理に関するヒント	3-6
アドバンスト・キューイングおよび Real Application Clusters	3-6
結論およびガイドラインの概要	3-8

4 Real Application Clusters のパフォーマンスの監視

Real Application Clusters データベースの監視の概要	4-2
最適なパフォーマンスを得るための構成に関する推奨事項	4-2
ユーザー・モードの IPC プロトコルの使用	4-2
バッファ・キャッシュと共有プールのサイズ指定	4-3
Real Application Clusters のインターコネクト設定の検証	4-3
Real Application Clusters のパフォーマンス・ビュー	4-4
Real Application Clusters のパフォーマンス統計	4-5
Real Application Clusters 統計の内容	4-5
統計の記録	4-6
Statspack と統計を使用した Real Application Clusters のパフォーマンスの監視	4-7
Real Application Clusters での Statspack の使用	4-7
GCS 統計と GES 統計の分析によるパフォーマンスの監視	4-8
待機イベントの分析	4-12
CACHE_TRANSFER ビューを使用した Real Application Clusters 統計の分析	4-15

5 Oracle Performance Manager を使用した Real Application Clusters のパフォーマンスの監視

Oracle Performance Manager の概要	5-2
以前のリリースの Oracle クラスター・ソフトウェアでの Performance Manager チャートの使用	5-2
Real Application Clusters での Oracle Performance Manager チャートの表示	5-2
Real Application Clusters 固有の Performance Manager チャートの階層	5-4
Performance Manager の Real Application Clusters 固有のチャート	5-6
Total Transfer チャート	5-6
Global Cache CR Request チャート	5-7
Global Cache Convert チャート	5-8
Library Cache Lock チャート	5-8
Row Cache Lock チャート	5-9
Global Cache Current Block Request チャート	5-9
Real Application Clusters 固有の Oracle チャート	5-10
File I/O Rate チャート	5-10
Session チャート	5-10
Users チャート	5-10
Real Application Clusters Top Sessions チャート	5-10
Real Application Clusters Database Health Overview チャート	5-11

第 III 部 Real Application Clusters のリファレンス情報

A マルチブロック・ロックの構成 (オプション)

GCS および GES のリソース制御メカニズムの上書き前の注意	A-2
GCS 処理および GES 処理の上書きの判断	A-2
ロックを使用する場合	A-3
GC_FILES_TO_LOCKS の設定	A-4
GC_FILES_TO_LOCKS 構文	A-4
ロックの割当て例	A-5
GC_FILES_TO_LOCKS の設定に対する追加の考慮点	A-10
データ・ファイルの拡張または追加	A-10
GC_FILES_TO_LOCKS 設定に含めないファイル	A-10
Real Application Clusters でのパラレル実行のチューニング	A-11
Real Application Clusters の I/O 統計の分析	A-12
V\$SYSSTAT を使用した Real Application Clusters の I/O 統計の分析	A-12

不適切な強制書込みの検出によるマルチブロック・ロック使用量の監視	A-14
ロック名の書式	A-16
ロック名の書式	A-16
ロックの名前	A-17
ロックのタイプと名前	A-17

B Real Application Clusters での空きリストと空きリスト・グループの使用 (オプション)

空きリスト・グループを使用した複数ノードからの同時挿入	B-2
空きリストと空きリスト・グループの目的	B-2
空きリスト・グループを使用したデータベース・オブジェクトの作成	B-2
FREELIST GROUPS 再編成の必要性の判断	B-3
FREELISTS および FREELIST GROUPS を使用した表、クラスタおよび索引の作成	B-4
インスタンスおよびユーザー・セッションと空きリスト・グループの対応付け	B-6
エクステンツ管理	B-7
エクステンツの事前割当て	B-7
エクステンツ管理およびローカル管理表領域	B-9

用語集

索引



5-1	Oracle Performance Manager の Real Application Clusters 固有のチャート	5-3
5-2	Real Application Clusters 固有の Enterprise Manager チャートの階層	5-5
5-3	Real Application Clusters Database Health Overview チャート	5-12
A-1	データ・ブロックへのロックのマッピング	A-6
A-2	GC_FILES_TO_LOCKS の例 5	A-8
A-3	GC_FILES_TO_LOCKS の例 6	A-8
A-4	GC_FILES_TO_LOCKS の例 7	A-9
A-5	GC_FILES_TO_LOCKS の例 8	A-9

表

5-1	Real Application Clusters 固有のチャートの階層	5-4
A-1	ロックを使用する場合	A-3
A-2	GC_FILES_TO_LOCKS の変数とその意味	A-4
A-3	強制書き込み率の解析	A-14
A-4	ロック名の書式で使用する変数	A-16
A-5	ロックのタイプと名前	A-17

はじめに

このマニュアルでは、Oracle9i Real Application Clusters データベース上でアプリケーションを実装する際の配置に関する考慮事項について説明します。また、配置後に Real Application Clusters データベースのパフォーマンスを監視するための情報についても説明します。

このマニュアルに記載されている内容は、すべてのオペレーティング・システムで実行する Real Application Clusters に適用されます。必要に応じて、プラットフォーム固有のドキュメントを参照してください。

この章の内容は次のとおりです。

- [対象読者](#)
- [このマニュアルの構成](#)
- [関連文書](#)
- [表記規則](#)

対象読者

このマニュアルは、Real Application Clusters を操作するデータベースの専門家を対象としています。

このマニュアルは、『Oracle9i Real Application Clusters 概要』に記載されている Real Application Clusters での処理の概念、ソフトウェアおよびハードウェア・コンポーネントの概念を理解していることを前提としています。また、『Oracle9i Real Application Clusters セットアップおよび構成』およびプラットフォーム固有の関連ドキュメントを参照して、Real Application Clusters をインストールしておく必要があります。

Oracle のシングル・インスタンスの配置とパフォーマンスのすべての方法論は、Real Application Clusters にも適用されます。したがって、『Oracle9i データベース・パフォーマンス・チューニング・ガイドおよびリファレンス』および『Oracle9i データ・ウェアハウス・ガイド』についてもよく理解している必要があります。

このマニュアルの構成

このマニュアルは、次の 3 部で構成されています。

第 I 部 「Real Application Clusters の配置の概要」

第 I 部では、Oracle9i の機能を活用する方法を説明し、Real Application Clusters におけるアプリケーションの配置の概要を説明します。E-Commerce およびデータ・ウェアハウス環境におけるインターネット・ベースのアプリケーションの配置についても説明します。

第 1 章 「Real Application Clusters アプリケーションの配置の概要」

この章では、Oracle9i の機能に関連した Real Application Clusters 環境における配置の考慮点について説明します。

第 2 章 「Real Application Clusters でのアプリケーションの配置」

この章では、Real Application Clusters を使用したオンラインの E-Commerce およびデータ・ウェアハウス・アプリケーションの配置について説明します。

第 II 部 「Real Application Clusters での配置およびパフォーマンス」

第 II 部では、Real Application Clusters の配置とパフォーマンスの監視について説明します。

第 3 章 「Real Application Clusters でのデータベースの配置方法」

この章では、Real Application Clusters のデータベースの配置について説明します。

第 4 章 「Real Application Clusters のパフォーマンスの監視」

この章では、Real Application Clusters のパフォーマンスの監視に関する一般的な推奨事項について説明します。

第 5 章 「Oracle Performance Manager を使用した Real Application Clusters のパフォーマンスの監視」

この章では、Oracle Enterprise Manager を使用した Real Application Clusters の監視の方法について説明します。

第 III 部 「Real Application Clusters のリファレンス情報」

第 III 部では、Real Application Clusters の配置およびパフォーマンスのリファレンス情報を示します。

付録 A 「マルチブロック・ロックの構成（オプション）」

この付録では、Real Application Clusters のデフォルトのリソース制御メカニズムをオーバーライドする方法を説明します。

付録 B 「Real Application Clusters での空きリストと空きリスト・グループの使用（オプション）」

この付録では、空きリストおよび空きリスト・グループの使用方法について説明します。

用語集

用語集では、このマニュアルで使用する用語の定義を示します。

関連文書

詳細は、次の Oracle マニュアルを参照してください。

- 『Oracle9i Real Application Clusters 概要』
- 『Oracle9i Real Application Clusters セットアップおよび構成』
- 『Oracle9i Real Application Clusters 管理』
- 『Oracle9i Real Application Clusters Real Application Clusters Guard I - Concepts and Administration』
- 『Oracle9i データベース・パフォーマンス・チューニング・ガイドおよびリファレンス』
- 『Oracle9i データ・ウェアハウス・ガイド』

インストレーション・ガイド

- 『Oracle9i for UNIX Systems インストレーション・ガイド』（AIX、Compaq Tru64、HP 9000 Series HP-UX、Linux Intel、Sun Solaris 共通）
- 『Oracle9i Database for Windows インストレーション・ガイド』

オペレーティング・システム固有の管理ガイド

- 『Oracle9i for UNIX Systems 管理者リファレンス』 (AIX、Compaq Tru64、HP 9000 Series HP-UX、Linux Intel、Sun Solaris 共通)
- 『Oracle9i Database for Windows 管理者ガイド』
- 『Oracle9i Real Application Clusters Real Application Clusters Guard I - Concepts and Administration』

Oracle9i Real Application Clusters の管理ガイド

- 『Oracle9i Real Application Clusters 管理』
- 『Oracle Enterprise Manager 管理者ガイド』
- 『Oracle Enterprise Manager Oracle Diagnostics Pack スタート・ガイド』

共通マニュアル

- 『Oracle9i データベース概要』
- 『Oracle9i Net Services 管理者ガイド』
- 『Oracle9i データベース・リファレンス』
- 『Oracle9i データベース新機能』

このマニュアルの多くの例で、Oracle のインストール時にデフォルトとしてインストールされる **シード・データベース** のサンプル・スキーマを使用しています。これらのスキーマがどのように作成されているか、およびその使用方法については、『Oracle9i サンプル・スキーマ』を参照してください。

リリース・ノート、インストレーション・マニュアル、ホワイト・ペーパーまたはその他の関連文書は、OTN-J (Oracle Technology Network Japan) に接続すれば、無償でダウンロードできます。OTN-J を利用するには、オンラインでの登録が必要です。次の URL で登録できます。

<http://otn.oracle.co.jp/membership/>

OTN-J のユーザー名とパスワードを取得済みであれば、次の OTN-J Web サイトの文書セクションに直接接続できます。

<http://otn.oracle.co.jp/document/>

表記規則

このマニュアル・セットの本文とコード例に使用されている表記規則について説明します。次の項目について説明します。

- 本文の表記規則
- コード例の表記規則
- Windows オペレーティング・システムの表記規則

本文の表記規則

本文中には、特別な用語が一目でわかるように様々な表記規則が使用されています。次の表に、本文の表記規則と使用例を示します。

規則	意味	例
太字	太字は、本文中で定義されている用語または用語集に記載されている用語（あるいはその両方）を示します。	この句を指定すると、 索引構成表 が作成されます。
固定幅フォントの大文字	大文字の固定幅フォントは、システムが提供する要素を示します。このような要素には、パラメータ、権限、データ型、Recovery Manager キーワード、SQL キーワード、SQL*Plus またはユーティリティ・コマンド、パッケージおよびメソッドが含まれます。また、システムが提供する列名、データベース・オブジェクト、データベース構造、ユーザー名およびロールも含まれます。	NUMBER に対してのみ、この句を指定できます。 BACKUP コマンドを使用して、データベースのバックアップを取ることができます。 USER_TABLES データ・ディクショナリ・ビューの TABLE_NAME 列を問い合わせます。 DBMS_STATS.GENERATE_STATS プロシージャを使用します。
固定幅フォントの小文字	小文字の固定幅フォントは、実行可能ファイル、ファイル名、ディレクトリ名およびユーザーが提供する要素のサンプルを示します。このような要素には、コンピュータ名、データベース名、ネット・サービス名および接続識別子が含まれます。また、ユーザーが提供するデータベース・オブジェクトとデータベース構造、列名、パッケージとクラス、ユーザー名とロール、プログラム・ユニットおよびパラメータ値も含まれます。 注意: 大文字と小文字を組み合わせて使用するプログラム要素もあります。これらの要素は、記載されているとおりに入力してください。	sqlplus と入力して、SQL*Plus をオープンします。 パスワードは、orapwd ファイルで指定します。 データ・ファイルと制御ファイルのバックアップを /disk1/oracle/dbs ディレクトリに作成します。 hr.departments 表には、department_id、department_name および location_id 列があります。 QUERY_REWRITE_ENABLED 初期化パラメータを TRUE に設定します。 oe ユーザーとして接続します。 JRepUtil クラスが、次のメソッドを実装します。

規則	意味	例
固定幅フォントの 小文字の イタリック	固定幅フォントの小文字のイタリックは、 プレースホルダまたは変数を示します。	<i>parallel_clause</i> を指定できます。 <i>old_release</i> .SQL を実行します。ここで、 <i>old_release</i> とはアップグレードの前にインス トールしたリリースを示します。

コード例の表記規則

コード例は、SQL、PL/SQL、SQL*Plus またはその他のコマンドラインを示します。次のように、固定幅フォントで、通常の本文とは区別して記載しています。

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

次の表に、コード例の記載上の表記規則と使用例を示します。

規則	意味	例
[]	大カッコは、任意に選択する 1 つ以上の項目を囲みます。大カッコは入力しないでください。	DECIMAL (<i>digits</i> [, <i>precision</i>])
{ }	中カッコは、2 つ以上の項目を囲み、そのうち 1 つの項目は必須です。中カッコは入力しないでください。	{ENABLE DISABLE}
	縦線は、大カッコまたは中カッコ内の 2 つ以上のオプションの選択項目を表します。オプションのうちの 1 つを入力します。縦線は入力しないでください。	{ENABLE DISABLE} [COMPRESS NOCOMPRESS]
...	水平の省略記号は、次のいずれかを示します。 <ul style="list-style-type: none"> ■ 例に直接関連しないコードの一部が省略されている。 ■ コードの一部を繰り返すことができる。 	CREATE TABLE ...AS <i>subquery</i> ; SELECT <i>col1</i> , <i>col2</i> , ..., <i>coln</i> FROM employees;

規則	意味	例
.	垂直の省略記号は、例に直接関連しない複数の行が省略されていることを示します。	<pre>SQL> SELECT NAME FROM V\$DATAFILE; NAME ----- /fsl/dbs/tbs_01.dbf /fsl/dbs/tbs_02.dbf . . . /fsl/dbs/tbs_09.dbf 9 rows selected.</pre>
その他の表記	大カッコ、中カッコ、縦線、および省略記号以外の句読点は、表示されているとおりに入力する必要があります。	<pre>acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;</pre>
イタリック	イタリック体は、特定の値を指定する必要があるプレースホルダや変数を示します。	<pre>CONNECT SYSTEM/system_password DB_NAME = database_name</pre>
大文字	大文字は、システムが提供する要素を示します。これらの用語は、ユーザー定義の用語と区別するために大文字で示されます。用語が大カッコ内にかぎりで表示されているとおりの順序および綴りで入力します。ただし、これらの用語は大文字 / 小文字が識別されないため、小文字でも入力できます。	<pre>SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;</pre>
小文字	小文字は、ユーザー定義のプログラム要素を示します。たとえば、表名、列名、ファイル名などです。 注意: 大文字と小文字を組み合わせるプログラム要素もあります。これらの要素は、記載されているとおりに入力してください。	<pre>SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjones IDENTIFIED BY ty3MU9;</pre>

Windows オペレーティング・システムの表記規則

次の表は、Windows オペレーティング・システムの表記規則と使用例を示しています。

規則	意味	例
「スタート」 →	プログラムの起動方法を表します。	Oracle Database Configuration Assistant を起動するには、「スタート」 → 「プログラム」 → 「Oracle - HOME_NAME」 → 「Configuration and Migration Tools」 → 「Database Configuration Assistant」を選択します。
ファイル名とディレクトリ名	ファイル名とディレクトリ名には、大 / 小文字区別はありません。特殊文字のうち、左山カッコ (<)、右山カッコ (>)、コロン (:)、二重引用符 (")、スラッシュ (/)、パイプ () およびハイフン (-) は使用できません。特殊文字である円記号 (¥) は、二重引用符で囲まれている場合も要素セパレータとして扱われます。¥¥ で始まるファイル名は、Windows では汎用命名規則を使用するものとみなされます。	C:¥winnt "¥"system32 は C:¥WINNT¥SYSTEM32 と同じです。
C:¥>	現行のハード・ディスク・ドライブの Windows コマンド・プロンプトを表します。コマンド・プロンプト内のエスケープ文字はカレット (^) です。プロンプトには、作業中のサブディレクトリが反映されます。このマニュアルでは、コマンド・プロンプトと呼ばれます。	C:¥oracle¥oradata>
特殊文字	特殊文字のうち円記号 (¥) は、Windows のコマンド・プロンプトで二重引用符 (") のエスケープ文字として必要な場合があります。カッコと一重引用符 (') には、エスケープ文字は不要です。エスケープ文字と特殊文字の詳細は、Windows オペレーティング・システムのマニュアルを参照してください。	C:¥>exp scott/tiger TABLES=emp QUERY=¥"WHERE job='SALESMAN' and sal<1600¥" C:¥>imp SYSTEM/password FROMUSER=scott TABLES=(emp, dept)
HOME_NAME	Oracle ホーム名を表します。ホーム名は、英数字で 16 文字以内です。ホーム名に使用できる特殊文字は、アンダースコアのみです。	C:¥> net start OracleHOME_NAME\$TNSListener

規則	意味	例
ORACLE_HOME と ORACLE_BASE	<p>Oracle8 リリース 8.0 以前では、Oracle コンポーネントをインストールすると、すべてのサブディレクトリはデフォルトで次のいずれかの名前のトップレベルの ORACLE_HOME ディレクトリに置かれていました。</p> <ul style="list-style-type: none"> ■ Windows NT では、C:¥orant ■ Windows 98 では、C:¥orawin98 <p>このリリースは、Optimal Flexible Architecture (OFA) のガイドラインに準拠しています。すべてのサブディレクトリがトップレベルの ORACLE_HOME ディレクトリにあるとはかぎりません。ORACLE_BASE というトップレベル・ディレクトリがあり、デフォルトでは C:¥oracle です。他の Oracle ソフトウェアがインストールされていないコンピュータに最新の Oracle リリースをインストールする場合、最初の Oracle ホーム・ディレクトリのデフォルト設定は C:¥oracle¥orann で、nn は最新のリリース番号です。Oracle ホーム・ディレクトリは、ORACLE_BASE の直下にあります。</p> <p>このマニュアルに記載されているディレクトリ・パスの例は、すべて OFA の表記規則に準拠しています。</p>	%ORACLE_HOME%¥rdbms¥admin ディレクトリにアクセスします。

第 I 部

Real Application Clusters の配置の概要

第 I 部では、Real Application Clusters のアプリケーションの配置について説明します。また、Real Application Clusters 環境で、Oracle の高可用性機能と拡張性機能を活用する方法を説明します。第 I 部に含まれる章は、次のとおりです。

- 第 1 章「Real Application Clusters アプリケーションの配置の概要」
- 第 2 章「Real Application Clusters でのアプリケーションの配置」

Real Application Clusters アプリケーション の配置の概要

この章では、Real Application Clusters の拡張性機能と高性能な機能を活用する場合の留意点を説明することで、**Oracle Real Application Clusters** アプリケーションの配置について説明します。内容は次のとおりです。

- [Oracle9i 機能の活用](#)
- [Real Application Clusters での Oracle9i 機能の実装](#)

Oracle9i 機能の活用

Oracle9i Real Application Clusters でアプリケーションを最適に配置するには、この章で説明する Oracle 機能の問題を考慮する必要があります。これらの機能を正しく実装することによって、配置の問題を最小限に抑えることができます。また、システムで、[キャッシュ・フュージョン](#)・テクノロジーおよび Real Application Clusters の高性能な機能を十分に活用できます。

この章では、Real Application Clusters アプリケーションの配置に関する概要を説明します。この配置を使用できるのは、シングル・インスタンスの配置方法を実装した後です。

Real Application Clusters での Oracle9i 機能の実装

この項で説明する考慮点は、Oracle 固有の機能、および Real Application Clusters のパフォーマンスを向上させる機能に関連しています。この項で説明する機能は次のとおりです。

- [Real Application Clusters の記憶域](#)
- [Real Application Clusters での高可用性およびフェイルオーバー](#)
- [Real Application Clusters での共有サーバー](#)
- [接続時ロード・バランシング](#)
- [Real Application Clusters での透過的アプリケーション・フェイルオーバー](#)
- [Real Application Clusters での PL/SQL](#)
- [Real Application Clusters での Recovery Manager \(RMAN\)](#)

Real Application Clusters の記憶域

Oracle9i の高度な記憶域機能は、優れたパフォーマンスを Real Application Clusters に提供します。この機能には、自動セグメント領域管理、自動 UNDO 管理およびサーバー・パラメータ・ファイルが含まれます。また、特定のプラットフォームでは、[RAW デバイス](#)を使用しないで、Real Application Clusters のファイルをクラスタ・ファイル・システムのファイルに格納できます。

Real Application Clusters でのクラスタ・ファイル・システム

クラスタ・ファイル・システムを使用すると、Real Application Clusters のインストールおよび管理が容易になります。また、RAW デバイスを管理する必要がなくなります。さらに、スケーラブルで待機時間の短い、高リジリエンスの記憶域が提供されるため、コストが大幅に削減されます。クラスタ・ファイル・システムの実装方法の詳細は、ベンダーのマニュアルを参照してください。

Real Application Clusters での高可用性およびフェイルオーバー

高い可用性を備えた構成には、障害の発生時に操作を維持し、単一の障害箇所を回避する冗長的なハードウェアとソフトウェアが含まれています。障害が発生すると、フェイルオーバーによって、障害コンポーネントが実行していた処理がバックアップ・コンポーネントに移されます。Oracle のフェイルオーバー・プロセスによって、リソースが早急に再マスター化され、トランザクションの一部や障害トランザクションがリカバリされて、システムが迅速にリストアされます。

多くの Oracle 製品と機能を組み合わせることで、信頼性の高いコンピューティング環境を構築できます。そのためには、容量と冗長性を計画する必要があります。さらに、システム全体のコストおよび投資効率も考慮する必要があります。また、適切なハードウェアの選択や、高可用性構成の一部であるアイドル状態のマシンを使用するかどうかの決定など、その他にも現実的な考慮点があります。

プライマリ/セカンダリ・インスタンス構成およびアクティブ/アクティブ・インスタンス構成

プライマリ/セカンダリ・インスタンス構成は、構成が最も簡単な高可用性構成です。また、管理が最も簡単な構成でもあります。たとえば、この構成におけるプライマリ・データベースの管理オーバーヘッドは、シングル・インスタンス構成のオーバーヘッドと同じです。

プライマリ/セカンダリ構成では、セカンダリ・インスタンスをアイドル状態にしておく必要はありません。たとえば、セカンダリ・インスタンスを読取り専用操作に使用することができます。プライマリ/セカンダリ構成の拡張性は特に優れていませんが、高可用性は備えています。

一方、アクティブ/アクティブ・インスタンス構成は、構成方法がより複雑でした。しかし、Real Application Clusters Guard II によって、完全なアクティブ構成を簡単に管理できるようになります。

Real Application Clusters Guard II

Real Application Clusters Guard II では、2 ノードのアクティブ/アクティブ・クラスタから n ノードの完全にアクティブなクラスタに概念が拡大され、すべてのインスタンスがクラスタ・データベース内のサービスをサポートできます。

関連項目： Real Application Clusters Guard II の詳細は、『Oracle9i Real Application Clusters Guard II Concepts, Installation, and Administration』を参照してください。

Oracle Real Application Clusters Guard I

Real Application Clusters の拡張構成である Oracle Real Application Clusters Guard I は、プラットフォームの **クラスタ・データベース**・フレームワーク内で、Oracle の高度なりカバリ機能を緊密に統合します。

関連項目： Real Application Clusters Guard I の概要については、『Oracle9i Real Application Clusters Real Application Clusters Guard I - Concepts and Administration』を参照してください。

Data Guard

Oracle9i Data Guard はスタンバイ・データベースとともに機能し、データベース破壊の原因となるエラー、障害および破損からデータを保護します。Data Guard は、スタンバイ・データベース環境での作成、管理および監視を自動化して、重要なデータを保護します。また、Data Guard によって、トランザクション上一貫した Oracle データベースのコピーを、本番データベースの損害または破損からリカバリして保持するための手動処理が自動化されます。

関連項目： Data Guard の詳細は、『Oracle9i Data Guard 概要および管理』を参照してください。

Real Application Clusters での共有サーバー

共有サーバーの機能を持つ Real Application Clusters は、何千ものデータベース同時接続ユーザーを処理できます。共有サーバーは、多数のユーザーの接続ロードを効率的に管理し、**トランザクション・モニター**と同様に機能します。

共有サーバーを備えた Real Application Clusters によって、2 つ以上の小規模なコンピュータ上で実行するアプリケーションのパフォーマンスが大幅に向上します。共有サーバーは、アプリケーションを再作成しなくても使用できます。実際、アプリケーションによっては、共有サーバーを使用しない場合より使用した場合の方がパフォーマンスが向上する場合があります。

共有サーバー構成では、ユーザー・プロセスは**ディスパッチャ**に接続します。接続されたディスパッチャは、複数のネットワーク・セッションの受信要求を共通キューに送ります。サーバー・プロセスが使用可能になると、ディスパッチャは、受信要求をアイドル状態のサーバー・プロセスに接続します。その接続が不要になると、サーバー・プロセスは他の要求で使用可能になります。そのため、少数のサーバー・プロセスで、多数のクライアントに対応できます。

接続時ロード・バランシング

接続時ロード・バランシング機能は、アクティブ・インスタンス間で接続を自動的に分散させます。接続時ロード・バランシングでは、この分散をクラスタ内の各**ノード**およびインスタンスのワークロードに基づいて行います。**接続時ロード・バランシング**は、共有サーバー環境と**専用サーバー**環境の両方で使用できます。Real Application Clusters とキャッシュ・フュージョンを接続時ロード・バランシングと組み合わせることで、アプリケーションまたはデータをパーティション化せずに、すべての種類のアプリケーションを実行できます。

注意： 共有サーバーと**ロード・バランシング**機能を使用するには、Oracle Net をインストールする必要があります。

Real Application Clusters での透過的アプリケーション・フェイルオーバー

透過的アプリケーション・フェイルオーバー (TAF) 機能は、接続に失敗した場合に、自動的にアプリケーションをデータベースに再接続します。再接続は OCI ライブラリ内で自動的に行われるため、TAF を使用するようにクライアント・アプリケーションを変更する必要はありません。

ほとんどの TAF 機能はクライアント側のネットワーク・ライブラリ (OCI) に実装されるため、クライアントが TAF 機能を活用するには、Oracle Net の OCI ライブラリを使用する必要があります。そのため、Real Application Clusters で TAF を実装するには、PL/SQL パッケージではなく JDBC OCI を使用してください。

注意： Real Application Clusters は Thin JDBC および JDBC OCI の両方をサポートしますが、TAF は JDBC OCI でのみサポートされます。

TAF は、プライマリ / セカンダリ・インスタンス構成でも使用できます。通常は、プライマリ・インスタンスに接続しますが、セカンダリ・インスタンスに接続するには、接続記述子の Connect Data 部分に `INSTANCE_ROLE` パラメータを使用して、明示的にセカンダリ・インスタンス接続を設定します。

関連項目： Real Application Clusters での `INSTANCE_ROLE` パラメータの使用については、『Oracle9i Real Application Clusters 管理』を参照してください。また、このパラメータの詳細と例については、『Oracle9i Net Services 管理者ガイド』を参照してください。

TAF を使用するには、**Oracle9i Enterprise Edition** のライセンスが必要です。TAF は Real Application Clusters 用に設計されているため、Real Application Clusters 環境用には非常に簡単に構成できますが、Real Application Clusters 以外の環境でも使用できます。TAF は、シングル・インスタンスの Oracle でも使用できます。また、次のようなシステムでも使用できます。

- Oracle Real Application Clusters Guard
- レプリケート・システム
- Data Guard

関連項目：

- Real Application Clusters における TAF の概要は、『Oracle9i Real Application Clusters 概要』を参照してください。
- TAF の詳細は、『Oracle9i Net Services 管理者ガイド』および『Oracle Call Interface プログラマーズ・ガイド』を参照してください。
- Oracle Replication の詳細は、『Oracle9i レプリケーション』を参照してください。

Real Application Clusters での PL/SQL

PL/SQL は、Oracle が提供する、SQL のプロシージャ拡張です。PL/SQL は、データのカプセル化、オーバーロード、コレクション型、例外処理、情報隠蔽などの機能を提供する高度な 4GL プログラミング言語です。また、PL/SQL は、透過的な SQL アクセス、Oracle サーバーおよびツール製品との緊密な統合、移植性およびセキュリティも提供します。

関連項目： PL/SQL の詳細は、『PL/SQL ユーザーズ・ガイドおよびリファレンス』を参照してください。

Real Application Clusters での Recovery Manager (RMAN)

Recovery Manager (RMAN) は Oracle のツール製品の 1 つです。Recovery Manager を使用すると、データ・ファイル、**制御ファイル**およびアーカイブ REDO ログをバックアップ、コピー、リストアおよびリカバリできます。Recovery Manager は、コマンドライン・ユーティリティとして起動するか、**Oracle Enterprise Manager** を介して使用できます。

Recovery Manager は、多くのバックアップおよびリカバリ・タスクを自動化します。たとえば、各データ・ファイルの適切なバックアップの場所を自動的に特定し、正しいコピー先にコピーします。これによって、同じタスクを実行するために、オペレーティング・システム・コマンドを使用して、エラーが発生しやすい操作を手動で行う必要がなくなります。

Recovery Manager は、すべてのインスタンスがクラスタ内のすべてのアーカイブ・ログにアクセスできるように構成する必要があります。あるインスタンスに障害が発生すると、リカバリを実行する、障害が発生していないインスタンスは、障害が発生したインスタンスのアーカイブ・ログにアクセスする必要があります。

関連項目： Real Application Clusters で Recovery Manager を使用するように構成する方法は、『Oracle9i Real Application Clusters 管理』を参照してください。また、Recovery Manager の詳細は、『Oracle9i Recovery Manager ユーザーズ・ガイド』を参照してください。

第 2 章では、オンライン E-Commerce および意思決定支援システム用のアプリケーション開発について説明します。このマニュアルの後半では、Real Application Clusters データベースを使用するアプリケーションの配置とパフォーマンスについて説明します。

Real Application Clusters での アプリケーションの配置

この章では、**Oracle Real Application Clusters** 環境でのオンライン E-Commerce (OLTP) アプリケーションの配置およびデータ・ウェアハウス・アプリケーションの使用方法について説明します。また、アプリケーションのパフォーマンス監視についても説明します。内容は次のとおりです。

- [Real Application Clusters ベースのアプリケーションの配置方法](#)
- [Real Application Clusters を使用した E-Commerce アプリケーションの配置](#)
- [Real Application Clusters でのデータ・ウェアハウス・アプリケーションの配置](#)
- [インターネット・ベースのアプリケーションの配置](#)

Real Application Clusters ベースのアプリケーションの配置方法

シングル・インスタンス・アプリケーションのすべての開発方法と配置方法が、Real Application Clusters にも適用されます。シングル・インスタンスの Oracle データベースで効率的に実行できるアプリケーションは、Real Application Clusters でも効率的に実行できます。

Real Application Clusters を使用した E-Commerce アプリケーションの配置

キャッシュ・フュージョンによって、Real Application Clusters データベースは、オンライン E-Commerce アプリケーションに最適な配置サーバーになります。これは、これらのアプリケーションには次の要件があるためです。

- 障害時の高可用性
- 増加するシステム需要に対応するための拡張性
- 需要変動に応じたロード・バランシング

Oracle および Real Application Clusters の高可用性機能によって、処理を中断することなく、ワークロードを障害が発生していないインスタンスに再分散してロード・バランスをとることができます。Real Application Clusters は、優れた拡張性も提供します。これによって、**ノード**を追加または置換した場合、Oracle は、再構成またはアプリケーションの再パーティション化を行うことなく、リソースを再マスター化してワークロードを再分散します。

Real Application Clusters は、アプリケーションのワークロードを非常に動的に管理します。また、ビジネス要件の変化に基づいて、リアルタイムでワークロードを変更できます。これは、管理オーバーヘッドが最小限に抑えられた管理環境で行われます。このキャッシュ・フュージョン・アーキテクチャの動的リソース割当て機能によって、オンライン・アプリケーションのパフォーマンスが最適化され、配置に高い柔軟性が得られます。

キャッシュ・フュージョンによる柔軟な実装

E-Commerce（特に、オンライン・トランザクション処理システム）の要件には、頻繁に変化するワークロードがあります。この要件を満たすため、Real Application Clusters は、システム負荷とシステム可用性の変化に対して柔軟にかつ動的に対応します。Real Application Clusters は、次のような原因で変動する幅広いサービス・レベルに対応します。

- ユーザーの需要の変化
- 取引の集中（大量取引の発生）など、ピーク時の拡張性の問題
- システム・リソースの可用性の変化

これらの要件を満たすためにパッケージ E-Commerce アプリケーションをパーティション化することは、実際的でなく、多くの場合複雑すぎます。このようなアプリケーションは、一度配置されると、何百、何千もの表にアクセスする可能性があります。このようなサービス・レベルへの一般的な対応策としては、より大規模なサーバーの購入が推奨されます。または、個別のデータベース間でアプリケーションまたはアプリケーション・モジュールをセグメント化することが推奨される場合もあります。ただし、アプリケーションをセグメント化すると、データが断片化され、グローバル企業全体での情報へのアクセスが制約される場合があります。

Real Application Clusters を使用すると、余分なハードウェアの購入が不要となり、アプリケーションのセグメント化も回避できます。さらに、Real Application Clusters は、ビジネス要件とワークロードの変化に対応するようにデータベース・リソースのマスター化を動的に移行して、これらの需要に対応します。

Real Application Clusters でのデータ・ウェアハウス・アプリケーションの配置

この項では、Real Application Clusters 環境でのデータ・ウェアハウス・システムの配置について説明します。また、共有ディスク・アーキテクチャで使用可能なデータ・ウェアハウス機能についても説明します。この項の内容は次のとおりです。

- [Real Application Clusters でのデータ・ウェアハウス・アプリケーションのスピードアップ](#)
- [パラレル問合せの動的な最適化](#)

関連項目： Real Application Clusters 環境におけるデータ・ウェアハウス・アプリケーションの実装の詳細は、『Oracle9i データ・ウェアハウス・ガイド』を参照してください。

Real Application Clusters でのデータ・ウェアハウス・アプリケーションのスピードアップ

Real Application Clusters は、Oracle のシングル・インスタンスのメリットを拡大するため、データ・ウェアハウス・アプリケーションには理想的です。Real Application Clusters は、[クラスタ・データベース](#)のすべてのノード上で使用可能な処理能力を最大限に活用して、データ・ウェアハウス・システムをスピードアップおよびスケールアップすることによって、Oracle のシングル・インスタンスのメリットを拡大します。

たとえば、E-Business では、顧客の獲得および市場の拡大のために、戦略的にデータ・ウェアハウス・システムを使用します。企業はデータ・ウェアハウス・システムを使用し、製品の宣伝活動を通じて情報を収集できます。そして企業は、ターゲットとなる顧客の統計的な特性に合わせて、顧客リストを最適な状態にカスタマイズできます。

Real Application Clusters 環境での柔軟なパラレル化

Oracle の[パラレル実行](#)機能では、マルチ・プロセスを使用して1つ以上のCPUでSQL文を実行します。パラレル実行は、シングル・インスタンス・データベースおよびReal Application Clusters データベースの両方で使用可能です。

Real Application Clusters は、パラレル処理をクラスタ内のすべてのノードに分散させることによって、パラレル実行を十分に活用します。パラレル操作に使用できるプロセスの数は、各表または索引に割り当てられる[並列度](#)によって異なります。

ファンクション・.shipping 疎結合システムでは、Oracle のパラレル実行テクノロジーによって、ファンクション・shippingという方法を使用してリモート・ノード上で作業が行われます。Oracle のパラレル・アーキテクチャでは、ターゲット・データがリモート・ノード上にある場合、ファンクション・shippingが使用されます。これによって、パラレル実行が効率的に行われ、[インターコネクト](#)経由での不要なノード間データ転送が排除されません。

データ局所性の使用 シェアード・ナッシング・ハードウェア・システムが一般的だったころは、一部のハードウェア・システムには、強力なデータ局所性機能が適していました。ただし、現行のほぼすべてのクラスタ・システムでは、共有ディスク・アーキテクチャを使用しています。

シェアード・ナッシング・システムでは、各ノードはディスク・デバイスのサブセットに直接ハードウェア接続します。これらのシステムの場合、所有するノードからローカル・デバイスにアクセスする方が、より効率的です。Real Application Clusters は、ノードに対するこのデバイスのアフィニティを使用し、クラスタ構成および共有ディスク・アーキテクチャを使用して、シェアード・ナッシング・システムより高いパフォーマンスを提供します。

キャッシュ・フュージョンの他の要素と同様に、Oracle は、データをパーティション化することなく、透過的に機能します。Oracle は、ターゲット・データが存在するディスクを動的に検出し、次の 2 つの方法でデータの位置を効果的に使用します。

- 処理対象のデータが存在するノード上で、パラレル実行サーバー・プロセスを起動します。
- ローカル・データのパーティションを各サブプロセスに割り当てて、ノード間のデータ移動を排除または最小化します。

パラレル問合せの動的な最適化

Oracle のコストベース・オプティマイザは、最適な実行計画を判断する場合に、パラレル実行を検討します。オプティマイザは、プロセッサ数に基づいて、パラレル化の適切なデフォルト値を動的に計算します。

代替アクセス・パスのコスト評価（表スキャンと索引アクセスなど）では、操作に使用できる並列度（DOP）が考慮されます。これによって、Oracle は、パラレル実行用に最適化された実行計画を選択します。

また、Oracle は、Real Application Clusters 環境でノード内およびノード間でのパラレル化について適切な決定を行います。たとえば、ノード内でのパラレル化では、SQL 文に 6 つの問合せサブプロセスが必要で、6 つの CPU がローカル・ノード（ユーザーが接続しているノード）でアイドル状態である場合、その SQL 文はローカル・リソースを使用して処理されます。これによって、複数ノード間の問合せ調整のオーバーヘッドが排除されます。

前述の例で、ローカル・ノード上に 2 つの CPU しかない場合、これらの 2 つの CPU および別のノード上にある 4 つの CPU を使用して、SQL 文が処理されます。このように、Oracle では、ノード内とノード間の両方の並列性を使用して、問合せ操作のスピードアップが行われます。

異なるパラレル実行サーバー間で、SQL 文の完全なパーティション化が行われているデータ・ウェアハウス・アプリケーションは、現実には存在しません。そのため、システム内の一部の CPU が、割り当てられた作業を他の CPU よりも早く完了してアイドル状態になります。Oracle のパラレル実行テクノロジーでは、アイドル状態の CPU を動的に検出し、よりワークロードが大きい CPU の実行キューから、これらのアイドル状態の CPU に作業を割り当てることができます。このように、Oracle では、システム内のすべての CPU 間で問合せ

のワークロードが効率的に再分散されます。Real Application Clusters は、クラスタのすべてのノード間で作業を再分散させることによって、より効率的にクラスタを使用します。

複数の同時パラレル操作に対するロード・バランシング

ロード・バランシングでは、パラレル実行サーバー・プロセスが分散され、CPU およびメモリー使用がノード間で均等に分散されます。また、ロード・バランシングによって、通信とリモートの I/O も最小化されます。Oracle は、実行中のプロセス数が最も少ないノードにパラレル実行サーバーを割り当てることによって、最小化を行います。

ロード・バランシング・アルゴリズムによって、すべてのノード間でワークロードが均等に保たれます。たとえば、各ノードに CPU が 1 つずつある 8 ノード・システムで並列度 8 が要求された場合は、ロード・バランシング・アルゴリズムによって、各ノード上に 2 つのサーバーが配置されます。パラレル実行サーバー・グループ全体が 1 つのノード上に収まる場合、ロード・バランシング・アルゴリズムによって、すべてのプロセスが単一ノード上に置かれ、通信オーバーヘッドが回避されます。たとえば、各ノードに CPU が 16 ずつある 2 ノード・クラスタ上で並列度 8 を使用する場合、ロード・バランシング・アルゴリズムによって、すべてのパラレル実行サーバー・プロセスが 1 つのノード上に置かれます。

パラレル・インスタンス・グループの使用

インスタンス・グループを使用すると、どのインスタンスがパラレル実行サーバー・プロセスを割り当てるかを制御できます。これを行うには、各アクティブ・インスタンスを 1 つ以上のインスタンス・グループに割り当てます。次に、特定のインスタンス・グループをアクティブにして、どのインスタンスがパラレル処理を起動するかを動的に制御します。

INSTANCE_GROUPS 初期化パラメータに 1 つ以上のインスタンス・グループを表す名前を設定することによって、インスタンスごとにインスタンス・グループ・メンバーシップを構築します。たとえば、マーケティング部門およびセールス部門の両方が所有する 32 ノード・システムでは、インスタンス・グループ名を使用して、各部門に 32 のノードを半分ずつ割り当てることができます。これを行うには、**初期化パラメータ・ファイル**で次のパラメータ構文を使用して、ノード 1 ~ 16 をマーケティング部門に割り当てます。

```
sid|1-16|.INSTANCE_GROUPS=marketing
```

次に、パラメータ・ファイルで次の構文を使用して、ノード 17 ~ 32 をセールス部門に割り当てます。

```
sid|17-32|.INSTANCE_GROUPS=sales
```

次のコマンドを入力して、セールス部門が所有するノードをアクティブにし、パラレル実行サーバー・プロセスを起動します。

```
ALTER SESSION SET PARALLEL_INSTANCE_GROUP = 'sales';
```

これに対して、Oracle はパラレル実行サーバー・プロセスをノード 17～32 に割り当てます。PARALLEL_INSTANCE_GROUP のデフォルト値は、すべてのアクティブ・インスタンスです。

注意： 1つのインスタンスは、1つ以上のグループに属することができません。INSTANCE_GROUPS パラメータに、複数のインスタンス・グループ名をカンマで区切って入力してください。

ディスク・アフィニティ

ディスク・アフィニティとは、インスタンスと、そのインスタンスがアクセスするデータの関係をいいます。インスタンスによる特定のデータ・セットへのアクセス頻度が高いほど、そのデータが存在するディスクに対するインスタンスのアフィニティは大きくなります。

ディスク・アフィニティは、パラレル表スキャン、一時表領域のパラレル割当て、パラレル DML およびパラレル索引スキャンに使用されます。ディスク・アフィニティは、パラレル表作成またはパラレル索引作成には使用されません。一時表領域へのアクセスでは、ローカル・データ・ファイルが優先的に使用されます。これによって、最適な領域管理エクステンション割当てが保証されます。オペレーティング・システムによってストライプ化されたディスクは、ディスク・アフィニティによって単一ユニットとして処理されます。

ディスク・アフィニティがない場合、Oracle は、インスタンス間で均等にパラレル実行サーバーを割り当てようと試みます。ディスク・アフィニティがある場合、Oracle は、要求されたデータに最も頻繁にアクセスするインスタンス上で、パラレル表スキャン用のパラレル実行サーバーを割り当てます。

たとえば、Oracle が表 T で全表スキャンを実行し、インスタンス 1、2 および 3 にはその表に対するアフィニティがあり、インスタンス 4 にはアフィニティがない場合を考えます。

- 問合せに 2 つのインスタンスが必要な場合、1、2 および 3 のうちの 2 つのインスタンスが使用されます。
- 問合せに 3 つのインスタンスが必要な場合、インスタンス 1、2 および 3 が使用されます。
- 問合せに 4 つのインスタンスが必要な場合、すべてのインスタンスが使用されます。
- 表 T に対して 2 つの同時操作が行われ、操作ごとに 3 つのインスタンスが必要な場合（およびそれらのインスタンス上で両方の操作に対して十分なプロセスを使用できる場合）は、両方の操作でインスタンス 1、2 および 3 が使用され、インスタンス 4 は使用されません。これに対し、ディスク・アフィニティがない場合は、インスタンス 4 が使用されます。

関連項目： インスタンスのアフィニティの詳細は、『Oracle9i Real Application Clusters 概要』を参照してください。

インターネット・ベースのアプリケーションの配置

アプリケーションの拡張性、可用性およびロード・バランシング機能のテストは、インターネット・ベースのアプリケーションの配置における最も重要な課題の1つです。たとえば、変化の激しいプロトタイプ環境では、テスト用のハードウェア・プラットフォーム上で、限られた数の Real Application Clusters インスタンスを使用してサイトを内部的にテストし、ベンチマークを取得することができます。

どのアプリケーションにも、固有のハードウェアおよびソフトウェア・アーキテクチャの制約によって、サポートするユーザー数に制限があります。予想される需要に対応するために、システム上の通信負荷を見積もり、必要なノード数を判断します。また、ピーク時のワークロードも考慮する必要があります。

拡張性のテストでは、Web サイトへのユーザー・アクセスをシミュレートする必要もあります。これを行うには、通信ジェネレータを、Hypertext Transfer Protocol (HTTP) で使用された場合に擬似 get コマンドを発行するように構成します。これによって、非常に現実的な条件下でシステムのパフォーマンスおよび負荷処理能力がテストされます。

次に、通信ジェネレータを使用して、構造化された Web ベースのテストを行い、システムのストレス・テストを行います。妥協を排したパフォーマンス目標を定めて、この種のテストを実施することで、容量上の限界が明らかになります。

テスト結果に従ってさらなる拡張を行った後、サイトを初期採用者向けにプロトタイプ化します。これによって、本番環境のベンチマークを取得することが可能となります。そして、そのベンチマークに基づいて、システムのパフォーマンスをより細かく調整することができます。

注意： これらの方法は、Real Application Clusters に固有なものではありません。これらの方法は、シングル・インスタンスの Web ベースの Oracle データベース・アプリケーションを配置するためにも使用できます。

Real Application Clusters でのアプリケーションのパフォーマンス

Real Application Clusters では、シングル・インスタンスの Oracle データベースに対する方法論と同じ方法論を使用して、アプリケーションの開発とチューニングを行うことができます。シングル・インスタンスの Oracle データベースで効率的に実行できるアプリケーションは、Real Application Clusters でも効率的に実行できます。

アプリケーションでホット・スポットがある場合は、必要に応じて、[第3章「Real Application Clusters でのデータベースの配置方法」](#)で説明する配置方法を参照してください。また、[第4章「Real Application Clusters のパフォーマンスの監視」](#)で説明するパフォーマンス監視の推奨事項も参照してください。

Real Application Clusters に対するシステム拡張性の管理

要件の増加に対応したり、障害ハードウェアを交換するために、配置前または本番中にノードを追加する必要がある場合があります。Real Application Clusters 環境にノードを追加するには、主に次の 2 つの手順を実行する必要があります。

- クラスタウェア・レイヤーでのノードの追加
- Oracle レイヤーでのノードの追加

ノードを追加するには、ベンダーのインストール手順に従ってハードウェアを接続します。次に、[オペレーティング・システム依存 \(OSD\) のクラスタウェア](#)をインストールします。Oracle Universal Installer (OUI) および Database Configuration Assistant (DBCA) を使用して、新しいノードにインスタンスをインストールします。一部のプラットフォームでは、ノードとインスタンスを動的に追加できます。

関連項目： ノードおよびインスタンスの追加手順の詳細は、『Oracle9i Real Application Clusters 管理』を参照してください。

第 II 部では、Real Application Clusters データベースのアプリケーションの配置とパフォーマンス監視に関する考慮点について説明します。

第 II 部

Real Application Clusters での配置およびパフォーマンス

第 II 部では、Real Application Clusters でのアプリケーションの配置に関する考慮点について説明します。また、第 II 部では、統計を監視する方法を説明し、必要な場合にパラメータを調整して Real Application Clusters のパフォーマンスを向上させる方法を説明します。さらに、Oracle Enterprise Manager を使用して Oracle Real Application Clusters データベースのパフォーマンスを監視する方法も説明します。

注意： シングル・インスタンスの Oracle データベース上でアプリケーションが効率的に実行されている場合は、Real Application Clusters での配置についてアプリケーションをチューニングする必要はありません。

第 II 部に含まれる章は、次のとおりです。

- 第 3 章「Real Application Clusters でのデータベースの配置方法」
- 第 4 章「Real Application Clusters のパフォーマンスの監視」
- 第 5 章「Oracle Performance Manager を使用した Real Application Clusters のパフォーマンスの監視」

Real Application Clusters でのデータベース の配置方法

この章では、**Oracle Real Application Clusters** 環境に対するデータベースの配置方法について説明します。内容は次のとおりです。

- **Real Application Clusters** に対するデータベース配置の原理
- **Real Application Clusters** での表領域の使用
- **Real Application Clusters** でのオブジェクトの作成およびパフォーマンス
- **Real Application Clusters** での順序番号の使用
- **Real Application Clusters** のアプリケーションのチューニングに関する推奨事項
- アドバンスド・キューイングおよび **Real Application Clusters**
- 結論およびガイドラインの概要

Real Application Clusters に対するデータベース配置の原理

Real Application Clusters に対してデータベースを配置するときは、シングル・インスタンス・データベースに対して使用する метод論と同じ方法論を使用します。効率的なシングル・インスタンス設計の場合、アプリケーションは Real Application Clusters で効率的に実行されます。

Real Application Clusters での表領域の使用

シングル・インスタンスの Oracle データベースでの表領域の使用方法は、Real Application Clusters データベースでの表領域でも使用できます。シングル・インスタンスの Oracle データベースの表領域にあるオブジェクトを制御する方法と同じ方法で、Real Application Clusters の特定の表領域に常駐するオブジェクトを制御します。

表領域管理を簡単にするため、オラクル社は、Real Application Clusters 環境での自動セグメント領域管理の使用をお勧めします。自動セグメント領域管理によって、エクステンツ管理が大幅に向上し、新規データの挿入時に空き領域を検索して割り当てるためのオーバーヘッドが削減されます。

自動セグメント領域管理に必要なローカル管理表領域を使用できない場合は、[付録 B「Real Application Clusters での空きリストと空きリスト・グループの使用 \(オプション\)」](#)で説明する空きリスト・グループの使用方を参照してください。また、すべてのノードで順序番号を使用する場合は、順序番号の生成を構成できます。

関連項目： 表領域とパフォーマンスに関する方法論の詳細は、『Oracle9i データベース・パフォーマンス・プランニング』を参照してください。

Real Application Clusters でのオブジェクトの作成およびパフォーマンス

原則として、DDL 文の使用はメンテナンス・タスクのみに限定し、システム操作のピーク時には実行しないようにします。ほとんどのシステムでは、新しいオブジェクトの生成量とその他の DDL 文に対する制限が必要です。シングル・インスタンスの Oracle データベースの場合と同様に、オブジェクトの作成と削除が多くなるとパフォーマンスのオーバーヘッドが増加する可能性があります。

オブジェクトのパーティション化

キャッシュ・フュージョンによって、クラスタ間でデータを効率的に同期化できるため、グローバル共有データベースのパーティションに関連するほとんどのコストが削減されます。ただし、オブジェクトのパーティション化では、アプリケーションを変更せずに、表と索引のホット・ブロックについてパフォーマンスを改善できます。この改善は、オブジェクトをハッシュ・パーティション化オブジェクトまたはコンポジット・パーティション化オブジェクトとして再作成することで実現されます。

たとえば、挿入率が高く、索引の主キーとして順序番号を使用する表を考えてみます。全ノード上のすべてのセッションは、右端の索引リーフ・ブロックにアクセスします。したがって、回避できない索引ブロックの分割によって、ボトルネックの原因となるシリアライズ・ポイントが作成される可能性があります。これを解決するには、表とその索引を、たとえば 16 方向のハッシュ・パーティション化オブジェクトとして再作成します。これによって、16 の索引リーフ・ブロック間で負荷が均等に分散されます。

注意： これらの方法は、Real Application Clusters 固有の方法ではなく、シングル・インスタンス環境でも使用できます。

Real Application Clusters での順序番号の使用

Real Application Clusters のアプリケーションを配置する場合は、できるだけ Oracle 順序番号をキャッシュします。順序番号の使用を最適化するには、各インスタンスのキャッシュに、順序を保持するための十分なサイズが必要です。デフォルトのキャッシュ・サイズでは、20 個の順序番号が保持されます。これを、200 個の順序番号を保持できるようにするには、次の構文を使用します。

```
ALTER SEQUENCE sequence_name CACHE 200;
```

キャッシュを抑制するには、順序付け機能を使用します。通常、インスタンス障害の後または SHUTDOWN コマンドの実行後は、番号がいくつか失われます。この問題は、シングル・インスタンス構成においても発生します。順序付けの使用を回避できない場合、順序キャッシュを使用禁止にする必要がある場合があります。この場合、ある程度のパフォーマンスのオーバーヘッドが予想されます。

関連項目： 順序の詳細は、『Oracle9i データベース概要』を参照してください。

順序におけるグローバル競合の検出

順序のキャッシュが不十分な場合や、順序がまったくキャッシュされない場合は、待機時間の増加によって重大なパフォーマンスの問題が発生する場合があります。パフォーマンスの問題が発生した場合は、V\$SYSTEM_EVENT ビューで統計を調べ、問題が順序の使用によるものかどうかを判断します。

このような状況では、DC_SEQUENCES パラメータの、DLM_REQUESTS に対する DLM_CONFLICTS の割合が高くなります。この割合が 10 ~ 15% を超え、「row cache locks」待機時間が合計待機時間の割合の大部分を占める場合、順序のキャッシュが不十分なために、サービス時間が低下することがあります。

データベース表を使用した順序番号の生成

アプリケーションが順序番号の損失に対応できない場合は、順序をデータベース表に格納して実装します。ただし、この方法では、パフォーマンス・オーバーヘッドが発生する可能性があります。この問題は、シングル・インスタンス環境においても発生します。一般的には、順序番号を格納する行を短期間ロックすることをお勧めします。

Real Application Clusters では、順序番号の格納に必要なキャッシュの一貫性で、小規模なオーバーヘッドが発生する場合があります。単一のデータ・ブロックに順序番号をいくつか格納しており、2 つ以上のインスタンスでそれらの順序番号を必要とする場合、データ・ブロックは、インスタンス間で頻繁に転送されます。

ブロックの頻繁な転送による悪影響を最小化するには、PCTFREE の値を高く設定して、各データ・ブロックに、順序番号を含む表の行が 1 つのみ格納されるようにします。この場合、キャッシュ転送は、インスタンスが同じ順序番号を同時に要求したときのみ行われます。

Real Application Clusters のアプリケーションのチューニングに関する推奨事項

この項では、Real Application Clusters ベースのアプリケーションにおけるパフォーマンス問題の識別方法および解決方法について説明します。内容は次のとおりです。

- [問合せのチューニングのヒント](#)
- [トランザクション処理に関するヒント](#)
- [アドバンスド・キューイングおよび Real Application Clusters](#)

問合せのチューニングのヒント

問合せ集中型のアプリケーションでは、I/O 要求ごとのデータ量を最大にするチューニング方法によって効果が得られます。このチューニング方法を使用する前と後にパフォーマンスを監視して、その効果を評価します。その方法は次のとおりです。

- [大きいブロック・サイズの使用](#)
- [DB_FILE_MULTIBLOCK_READ_COUNT の値の増加](#)

大きいブロック・サイズの使用

ブロック・サイズを大きくして、各操作によって取り出される行数を増加させます。行数の増加によって、アプリケーションの索引ツリーの階層も浅くなります。データベースが主に問合せ処理に使用されている場合は、ブロック・サイズを 8KB 以上にする必要があります。

DB_FILE_MULTIBLOCK_READ_COUNT の値の増加

DB_FILE_MULTIBLOCK_READ_COUNT の値を、できるだけ大きい値に設定する必要もあります。これによって、表をスキャンするために必要な読み込み数が削減され、全表スキャンがより高速になります。システム I/O は、読み込まれたブロック数をブロック・サイズに掛けた値に制限されていることに注意してください。

オペレーティング・システムによるディスクの**ストライプ化**を行っている場合は、DB_FILE_MULTIBLOCK_READ_COUNT に DB_BLOCK_SIZE の 2 倍の値を掛けた値にストライプ・サイズを設定します。システムで索引ストライプと表データ・ストライプを区別できる場合は、DB_BLOCK_SIZE に 2 を掛けたストライプ・サイズを索引に使用します。

次のことも考慮してください。

- データ・ファイルと索引に対しては、読み込み専用表領域を使用すること
- 一時セグメントを保持する表領域を TEMPORARY 型として定義すること

関連項目： データ・ウェアハウス環境での問合せパフォーマンスの改善にパラレル化を使用する方法の詳細は、『Oracle9i データベース・パフォーマンス・プランニング』を参照してください。

トランザクション処理に関するヒント

一般的に、トランザクション・ベースのアプリケーションは、その他のタイプのアプリケーションより多くのデータをディスクに書き込みます。データベース・ライター・プロセス (DBWR n) の機能を改善して大量のデータを高速で書き込むには、非同期の I/O を使用します。アクセスがランダムな場合は、小さなブロック・サイズの使用を考慮します。この方法でチューニングを開始する前後に、アプリケーションのパフォーマンスを監視して、システムのパフォーマンスが適切であることを確認してください。

注意： これらの方法はすべてのタイプのシステムで使用できるわけではなく、すべてのプラットフォームが非同期 I/O をサポートしているわけではありません。

アドバンスト・キューイングおよび Real Application Clusters

Real Application Clusters 環境でアドバンスト・キューイングを使用すると、次の項で示すとおり、機能性およびパフォーマンス関連の問題が発生する可能性があります。

- [キュー・テーブルのインスタンス・アフィニティ](#)
- [グローバル・キャッシュ・サービス・リソースの取得](#)
- [アドバンスト・キューイングおよびキュー・テーブルのキャッシュ転送](#)

キュー・テーブルのインスタンス・アフィニティ

キュー・テーブルのインスタンス・アフィニティによって、プライマリおよびセカンダリ・インスタンスのプロパティをキュー・テーブルに割り当てることができます。これによって、インスタンスの停止時および再起動時にキュー・テーブルの所有権が自動的に割り当てられます。キュー・テーブルのインスタンス・アフィニティは、次のビューを問い合わせることによって評価できます。

- `DBA_QUEUE_TABLES`
- `USER_QUEUE_TABLES`

グローバル・キャッシュ・サービス・リソースの取得

グローバル・キャッシュ・サービス・リソースの取得は、ローカル・リソースまたはローカル・エンキューの取得より高コストです。アドバンスト・キューイングの配置が不適切な場合、リソース制御動作によって、Real Application Clusters 環境のパフォーマンスが低下する場合があります。この問題を回避するためには、次のことを検討します。

- キュー・テーブルに対してロックを使用禁止にする。
- COMMIT の数を削減する。

自動セグメント領域管理を使用できない場合は、最高水位標を繰り上げるとき、空きリストに追加されるブロック数を増加します。

アドバンスト・キューイングおよびキュー・テーブルのキャッシュ転送

一般的に、キュー・テーブルのデータ・ブロックおよびキュー・テーブルの索引ブロックのキャッシュ転送は、次の場合に発生します。

- 異なるインスタンスが同時にキューにアクセスした場合
- Oracle がキュー・テーブルの所有権を正しく割り当てなかったため、キュー・モニターが、エンキュー操作またはデキュー操作が実行されたインスタンスとは異なるインスタンスからのキューをスケジューリングした場合
- Oracle がキュー・テーブルに対する領域トランザクションを実行する場合

キュー・テーブルのブロックのキャッシュ転送頻度を低下させるには、特定のインスタンスに対するアフィニティがあるキュー・テーブルを作成し、このインスタンスからキューまたはキュー・テーブルにアクセスします。

関連項目： アドバンスト・キューイングの概要は、『Oracle9i アプリケーション開発者ガイド-アドバンスト・キューイング』を参照してください。

結論およびガイドラインの概要

Real Application Clusters およびキャッシュ・フュージョンが提供する改善されたディスクレス・アルゴリズムによって、キャッシュの一貫性を効率的に管理できます。また、複雑さが軽減されたデータベース設計を配置しながら、最適なパフォーマンスが得られます。Real Application Clusters でキャッシュ・フュージョンを使用すると、通常、複数インスタンスでのリソース競合時に発生する処理オーバーヘッドのほとんどを取り除くことができます。

一般的に、パフォーマンスの問題の 80% 以上は、20% 以下の特定のワークロードが原因となっています。いくつかの簡単なガイドラインに従ってこの 20% を処理すると、最小限の工数で具体的な効果を得ることができます。これらの問題は、次の方法で解決できます。

- 自動セグメント領域管理とローカル管理表領域を使用して、エクステンツ管理処理を削減します。
- データが固定の場合は、必ず、読取り専用表領域を使用します。
- Oracle 順序を使用して一意の番号を生成し、CACHE パラメータを高い値に設定します。

Real Application Clusters のパフォーマンスの監視

この章では、**Oracle Real Application Clusters** のパフォーマンスの監視方法について説明します。内容は次のとおりです。

- [Real Application Clusters データベースの監視の概要](#)
- [最適なパフォーマンスを得るための構成に関する推奨事項](#)
- [Real Application Clusters のパフォーマンス・ビュー](#)
- [Real Application Clusters のパフォーマンス統計](#)
- [Statspack と統計を使用した Real Application Clusters のパフォーマンスの監視](#)

関連項目： Oracle Enterprise Manager を使用した Real Application Clusters のパフォーマンス監視方法については、第 5 章「[Oracle Performance Manager を使用した Real Application Clusters のパフォーマンスの監視](#)」を参照してください。

Real Application Clusters データベースの監視の概要

シングル・インスタンスのすべてのチューニング方法は、Real Application Clusters データベースで実行されているアプリケーションにも使用できます。したがって、『Oracle9i データベース・パフォーマンス・プランニング』で説明しているシングル・インスタンスのチューニング方法論を検討および実装してください。

Real Application Clusters データベースでは、シングル・インスタンスの Oracle データベース以上のチューニングは必要ありません。これは、キャッシュ・フュージョンでは、キャッシュの一貫性のためにディスク書込みを使用しないためです。したがって、Real Application Clusters データベースに対するパフォーマンスの監視作業は、この章で説明する手順に限定されます。

最適なパフォーマンスを得るための構成に関する推奨事項

この項では、Real Application Clusters 固有の構成に関する次の推奨事項について説明します。

- ユーザー・モードの IPC プロトコルの使用
- バッファ・キャッシュと共有プールのサイズ指定
- Real Application Clusters のインターコネクト設定の検証

ユーザー・モードの IPC プロトコルの使用

キャッシュ・フュージョンでは高速 IPC が使用されるため、Real Application Clusters では、**クラスタ・データベース・インターコネクト**で使用されている待機時間の短い通信リンクに関する最新テクノロジーによるパフォーマンス上の利点を活用できます。**仮想インタフェース・アーキテクチャ** (VIA) やユーザー・モードの IPC などのより効率的なプロトコルによって、さらにパフォーマンスの向上が期待できます。

キャッシュ・フュージョンは、ユーザー・モードの IPC (メモリマップド IPC ともいう) を使用して、UNIX、Windows NT および Windows 2000 ベースのプラットフォームでの CPU 使用率を削減します。ハードウェアが適切にサポートされている場合は、**オペレーティング・システムのコンテキストのスイッチング**を最小化します。また、高コストのデータ・コピーおよびシステム・コールも排除されます。

ユーザー・モードの IPC は、ハードウェアで効率的に実装されている場合、CPU の使用率を削減します。これは、ユーザー・モードの IPC ではユーザー・プロセスがオペレーティング・システムのカーネルを使用しないで通信するためです。ユーザー実行モードからカーネル実行モードへ切り替える必要はありません。

バッファ・キャッシュと共有プールのサイズ指定

Real Application Clusters におけるバッファ・キャッシュと共有プール機能の要件は、シングル・インスタンスの Oracle データベースの要件と比較して、わずかに厳しくなります。リカバリを容易にするため、Real Application Clusters では、データ・ブロックのコピーであるパスト・イメージ (PI) を、複数インスタンスにキャッシュされたブロック用に維持するための追加メモリーが必要になる場合があります。このため、バッファ・キャッシュのサイズを約 10%、共有プールのサイズを約 15% 増やしてください。

シングル・インスタンスの Oracle 環境と同様に、頻繁に使用しないバッファを積極的にフラッシュすることで、バッファ・キャッシュのヒット率を改善できます。たとえば、増分チェックポイントを使用して、データベース・ライター (DBWR n) のアクティビティを増加できます。

増分チェックポイントの率は、FAST_START_MTTR_TARGET、LOG_CHECKPOINT_TIMEOUT、LOG_CHECKPOINT_INTERVAL などの各種パラメータの設定によって異なります。これらのパラメータを 0 (ゼロ) 以外の値に設定すると、使用済みバッファまたはコールド・バッファ (使用頻度の低いバッファ) の書込み頻度が増加します。その結果、インスタンス・リカバリ時間が短縮され、バッファ・キャッシュの使用率が向上します。これによって、キャッシュ・データ・ブロックで使用できる領域が増加します。

関連項目： チェックポイントとパフォーマンスの詳細は、『Oracle9i データベース・パフォーマンス・プランニング』を参照してください。

Real Application Clusters のインターコネクト設定の検証

インターコネクトとノード間通信のプロトコルは、キャッシュ・フュージョンのパフォーマンスに影響を与える場合があります。さらに、インターコネクトの帯域幅、その待機時間および IPC プロトコルの効率によって、キャッシュ・フュージョンが CR ブロック要求を処理する速度が決まります。

インターコネクト処理への影響

一度インターコネクトが動作可能になると、インターコネクトのパフォーマンスを大きく変更することはできません。ただし、IPC のバッファ・サイズを調整することで、インターコネクト・プロトコルの効率を変更できます。

関連項目： IPC バッファ・サイズの調整の詳細は、ベンダー固有のインターコネクトのドキュメントを参照してください。

グローバル・キャッシュ・サービス (GCS) とグローバル・エンキュー・サービス (GES) 処理のメッセージ通信では、適切なインターコネクトを使用する必要があります。Real Application Clusters 関連の通信で使用する IP アドレスまたは NIC が不明な場合は、プラットフォームに依存しない次の SQL*Plus 文を実行します。

```
SQL> oradebug setmypid  
SQL> oradebug ipc
```

このコマンド・シーケンスによって、Oracle が `user_dump_dest` ディレクトリのトレース・ファイルへのインターコネクト通信に使用する IP アドレスに関する詳細が書き込まれます。ほとんどの場合、このパラメータを設定する必要はありませんが、次の例に示すように、`CLUSTER_INTERCONNECTS` パラメータを使用すると、プライベート・ネットワーク IP アドレスまたは NIC を割り当てることができます。

```
CLUSTER_INTERCONNECTS=10.0.0.1
```

オペレーティング・システム固有のベンダー IPC プロトコルを使用している場合は、トレース情報から IP アドレスが判明しない場合があります。ただし、Oracle では、ベンダー固有の IPC ライブラリの使用に基づいて正しいネットワーク・インタフェースを使用します。

関連項目： `CLUSTER_INTERCONNECTS` パラメータの設定については、『Oracle9i Real Application Clusters 管理』を、このパラメータに対する設定の Oracle による解釈方法については、プラットフォーム固有のマニュアルを参照してください。

Real Application Clusters のパフォーマンス・ビュー

各インスタンスには、それぞれ一連のインスタンス固有のビューがあります。グローバル動的パフォーマンス・ビューに問い合せて、すべてのインスタンスからパフォーマンス情報を取り出すことができます。グローバル動的パフォーマンス・ビュー名には、GV\$ という接頭辞が付きます。

グローバル・ビューには、各インスタンス固有のビューのすべての列が含まれます。また、`INST_ID` 列も含まれます。この列には、Oracle が関連するインスタンス固有の情報を取得した **インスタンス番号** が表示されます。このインスタンス番号をフィルタとして使用すると、インスタンスのサブセットから情報を取り出すことができます。たとえば、次の問合せは、インスタンスの 2 と 5 から情報を取り出します。

```
SELECT INST_ID,NAME,VALUE FROM GV$SYSSTAT WHERE NAME LIKE 'global cache%' AND (INST_ID =2 OR INST_ID=5);
```

各グローバル・ビューには、GLOBAL ヒントが含まれています。GLOBAL ヒントは、各インスタンスのローカル・ビューの内容を取り出すパラレル問合せを作成します。インスタンス内のパラレル実行プロセスの数が `PARALLEL_MAX_SERVERS` で設定した制限数に達している状態で、GV\$ ビューに対して追加の問合せを発行すると、この目的のために追加の **パラレル実行** プロセスが起動されます。この追加プロセスは、GV\$ 問合せの実行以外のパラレル操作には使用できません。

注意： インスタンスの `PARALLEL_MAX_SERVERS` パラメータを 0 (ゼロ) に設定した場合、GV\$ 問合せのための追加のパラレル実行サーバー・プロセスは起動しません。

インスタンスでのパラレル実行プロセス数が `PARALLEL_MAX_SERVERS` で設定した値に達している状態で、GV\$ ビューへの複数の問合せを発行すると、最初問合せ以外は正常に実行されません。ほとんどのパラレル問合せでは、プロセスを割り当てることができない場合、エラーになるか、または問合せコーディネータによって問合せが順次実行されます。

関連項目：

- 2-4 ページ [「Real Application Clusters 環境での柔軟なパラレル化」](#)
- GV\$ ビューの制限事項、および関連するパラメータとビューの詳細は、『Oracle9i データベース・リファレンス』を参照してください。

CATCLUST.SQL による Real Application Clusters のデータ・ディクショナリ・ビューの作成

Real Application Clusters データベースを Database Configuration Assistant (DBCA) を使用して作成しなかった場合は、CATCLUST.SQL スクリプトを実行して Real Application Clusters 関連のビューと表を作成します。このスクリプトの実行には、SYSDBA 権限が必要です。

関連項目： 動的パフォーマンス・ビューの詳細は、『Oracle9i データベース・リファレンス』を参照してください。

Real Application Clusters のパフォーマンス統計

この項では、V\$ ビューと GV\$ ビューの統計の概要を説明します。この統計は、クラスタ内のブロック通信量の評価に使用できます。これらの統計を使用して、インターコネクトのブロック転送率や Real Application Clusters データベース全体のパフォーマンスを分析します。この項の内容は次のとおりです。

- [Real Application Clusters 統計の内容](#)
- [統計の記録](#)

Real Application Clusters 統計の内容

Real Application Clusters 固有の統計は、メッセージ要求カウンタまたは定期的な統計として表示されます。メッセージ要求カウンタには、特定のタイプのブロック・モード変換の数を示す統計が含まれます。定期的な統計は、特定のタイプの操作での読み取りおよび書き込み I/O に対する、合計または平均の待機時間を示します。

多くの統計は、キャッシュ・レイヤー、トランザクション・レイヤー、I/O レイヤーなど、データベース・カーネルの様々なコンポーネントが行う作業を計測します。たとえば、定期的な統計は、ある要求を処理する時間や特定のイベントを待機する時間を示します。ほとんどの統計は、各インスタンスのシステム・グローバル領域 (SGA) に記録されます。

統計の記録

オラクル社では、あるイベントが発生する割合についての統計を記録することをお勧めします。システム・パフォーマンスの履歴を維持することで、そうした統計がどのような傾向で変化しているのか、識別することができます。パフォーマンスの傾向によって、応答時間の増大やスループット低下の原因となる問題も識別されます。また、それらの情報によって、容量のピーク使用時における処理要件の変化も識別しやすくなります。

Oracle は、バッファ・キャッシュおよび**グローバル・キャッシュ・サービス (GCS)** レイヤーから、キャッシュ・フュージョン関連のパフォーマンス統計を収集します。Oracle は、Real Application Clusters 全般のパフォーマンス統計の他に、ブロック要求とブロック・モード変換の待機に関する統計も収集します。

Oracle は、オブジェクト・レベルの統計をデフォルトで記録します。つまり、Oracle は、STATISTICS_LEVEL パラメータを `typical` に設定します。また、定期的に統計を記録するには、TIMED_STATISTICS パラメータを `true` に設定してください。Oracle は、統計を数百分の 1 秒ごとに記録します。

オブジェクト・レベルの統計と定期的な統計で示されたパフォーマンスの傾向に加えて、Real Application Clusters 環境内の、特定のトランザクションに関する情報を示す統計も調べます。この作業は、次項の説明に従って、Oracle9i Statspack などのユーティリティを使用して行います。

注意： 統計を収集しないで、統計収集に関連するパフォーマンス上のオーバーヘッドを回避するには、TIMED_STATISTICS を `false` に設定してください。

関連項目： DBA_OBJECT_STATS、V\$OBJ_STAT ビューと V\$OBJ_STAT_NAME ビューの使用の詳細は、『Oracle9i パフォーマンス・メソッド』を参照してください。

Statspack と統計を使用した Real Application Clusters のパフォーマンスの監視

この章の後半にある次の項では、Real Application Clusters での Statspack の使用方法、およびその統計を分析してパフォーマンスの傾向を評価する方法について説明します。

- [Real Application Clusters での Statspack の使用](#)
- [GCS 統計と GES 統計の分析によるパフォーマンスの監視](#)
- [待機イベントの分析](#)
- [CACHE_TRANSFER ビューを使用した Real Application Clusters 統計の分析](#)

Real Application Clusters での Statspack の使用

Statspack は、シングル・インスタンスの Oracle データベースで使用方法と同じ方法で Real Application Clusters で使用できます。Statspack は、長期間にわたるパフォーマンスの傾向を示す統計を表示します。次のページに、Real Application Clusters のパフォーマンス情報が表示されます。

- 「Cluster Statistics」 ページ
- 「GES Statistics」 ページ

これらのページでは、次の統計を監視します。

- ワークロードの特性を示すグローバル・キャッシュ・サービス (GCS) 統計
- グローバル・エンキュー・サービス (GES) 統計
- GCS と GES のメッセージ機能の統計

件数、経過時間、待機時間など、ほとんどの統計は Statspack によって計算されます。これらの統計によって、アプリケーションのワークロードとプロファイルの特性が定義されます。これらの統計は、次の方法で計算されます。

- トランザクションごとの統計 (トランザクションごとの global cache current blocks received など)
- 率の統計 (毎秒ごとの global cache current blocks received など)
- 比率の統計 (キャッシュ・ヒット率、SQL を処理する全体 CPU 時間に対する解析 CPU 時間の比率)

Statspack の最初のページには、負荷データおよびプロファイル・データのほとんどが表示されます。また、グローバル・キャッシュ・サービスの、1 要求あたりの平均時間 (カレント・ブロックの受信に要した時間など) も表示されます。Statspack には、上位の待機イベントとその合計待機時間に対する割合も表示されます。これらの統計は、特定の要求に関連するオーバーヘッドを示します。

関連項目： Statspack の詳細は、ORACLE_HOME/rdbms/admin ディレクトリにある spdoc.txt 文書ファイルを参照してください。

GCS 統計と GES 統計の分析によるパフォーマンスの監視

クラスタ・データベースのパフォーマンスを分析し、Real Application Clusters での競合を識別するには、次の各項で説明するように、ブロック転送率、待機イベントおよび統計を調べます。

- Real Application Clusters のブロック転送の分析
- GCS 統計と GES 統計を使用したパフォーマンス分析

Real Application Clusters のブロック転送の分析

Real Application Clusters のブロック転送を分析してグローバル処理のコストを判断し、インスタンス間での一貫性の保持に必要なリソースを明確にします。これを行うには、次の各項の説明に従って、統計を分析します。これらの手順を継続して行い、処理の傾向を識別して、パフォーマンスを最適化します。次の統計は、Real Application Clusters 関連のパフォーマンス特性を示します。

- バッファ・キャッシュ関連の統計 (consistent gets、db block gets など)
- グローバル・キャッシュ・サービス統計 (global cache current block receive time、global cache cr block receive time、global cache open、global cache convert request など)
- グローバル・キャッシュ・サービスの待機イベント (global cache s to x、global cache null to s など)

キャッシュ間転送の応答時間は、ログ書込み以外の I/O 関連の要素に含まれません。これは、ブロックがリモート・インスタンスに転送される前に、データ・ブロックの修正が REDO ログに書き込まれるためです。つまり、Real Application Clusters では、キャッシュの置換、チェックポイント、および新規に要求されたブロックの読み込みのために物理 I/O を実行する以外は、ディスク I/O に関するオーバーヘッドが発生しません。この動作は、シングル・インスタンスの Oracle 環境と同じです。

GCS 統計と GES 統計を使用したパフォーマンス分析

この項では、競合の識別によって、グローバル・キャッシュ・サービスとグローバル・エンキュー・サービスのパフォーマンスを監視する方法を説明します。内容は次のとおりです。

- グローバル・キャッシュ・サービスの待機イベント
- グローバル・キャッシュ・サービスのタイミング
- グローバル・エンキュー・サービスの統計
- 注意を要するグローバル・キャッシュの統計

グローバル・キャッシュ・サービスの待機イベント

次の統計値を調べて、各キャッシュのグローバルなホット・リソース（使用頻度の高いリソース）を識別します。

- global cache busy
- buffer busy global cache
- buffer busy global CR

V\$SESSION_WAIT ビューを使用して、パフォーマンスに問題があるオブジェクトを識別します。次の問合せの例に示すように、P1 列と P2 列でオブジェクトのファイル番号とブロック番号を識別します。

```
SELECT P1 FILE_NUMBER, P2 BLOCK_NUMBER FROM V$SESSION_WAIT WHERE EVENT = 'BUFFER
BUSY GLOBAL CR';
```

この最初の間合せによる出力は、次のようになります。

```
FILE_NUMBER BLOCK_NUMBER
-----
          12          3841
```

次の問合せを発行します。

```
SELECT OWNER,SEGMENT_NAME,SEGMENT_TYPE FROM DBA_EXTENTS WHERE FILE_ID = 12 AND 3841
BETWEEN BLOCK_ID AND BLOCK_ID+BLOCKS-1;
```

次のように出力されます。

```
OWNER          SEGMENT_NAME          SEGMENT_TYPE
-----
SCOTT          W_ID_I                INDEX PARTITION
```

競合がある場合、通常、最も競合が発生するブロックは索引リーフ・ブロックです。各ブロックでの競合を削減するには、より小さいサイズのブロックを索引の表領域に使用するか、ハッシュ・パーティション化を行います。この変更によって、より多くの個別のブロック間で負荷が分散されます。

グローバル・キャッシュ・サービスのタイミング

グローバル・キャッシュ・サービスのタイミングを調べて、インターコネク트에待機時間の問題があるかどうかを判断します。次の統計を調べます。

- global cache cr/current block receive time

待機時間が長くなる原因として、次の可能性があります。

- 多数のプロセスが CPU 待ちの実行キューにある場合、またはスケジュールの遅延。
- プラットフォーム固有のオペレーティング・システムのパラメータ設定によって、IPC バッファリングまたは処理スケジュールが影響を受けている場合。
- インターコネク트가低速、ビジーまたは障害が発生している場合。このような場合は、削除されたパケット、再送信または巡回冗長検査 (CRC) エラーを検索します。ネットワークがプライベートであること、およびインスタンス間の通信がパブリック・ネットワークを経由していないことを確認してください。

注意： CRC エラーは、通常、ファームウェアまたはハードウェアに関する問題を示します。このエラーは、通常システム・ログ・ファイルに表示されます。

Real Application Clusters でこのようなパフォーマンスの問題が発生しているかどうかを評価する手順は、プラットフォームによって異なります。たとえば、Solaris オペレーティング・システムでは、次のコマンドを実行します。

```
netstat -l
netstat -s
sar -c
sar -q
vmstat
```

グローバル・エンキュー・サービスの統計

待機時間に関連する主要な待機イベントの中に「enqueue」がある場合は、V\$ENQUEUE_STATS ビューの出力を分析して、待機時間が最も長いエンキューを識別します。

- SQ エンキュー順序に関する競合があることを示します。ほとんどの場合、ALTER SEQUENCE 文を実行して、アプリケーションで使用する順序のキャッシュ・サイズを増やすことができます。SQ エンキュー競合の再発を回避するため、順序では NOORDER キーワードを必ず使用してください。
- TX エンキュー—通常は、行ロックに関するアプリケーション関連の問題です。これは、シングル・インスタンスの Oracle データベースでも問題となる場合があります。ただし、Real Application Clusters プロセスでは、TX エンキュー待機の影響が拡大する場合

があります。また、昇順にデータが増加する索引のリーフ・ブロックにおけるパフォーマンス上のボトルネックが、索引ブロック分割中の TX エンキュー待機として表示される場合があります。

TX エンキューのパフォーマンスの問題を解決するには、INITRANS パラメータ値を、クラスタ内のノード数とノードごとの CPU 数の乗算結果に 0.75 を乗算した値に設定します。ただし、このパラメータは、100 を超える値に設定しないことをお勧めします。このため、MAXTRANS も 100 を超える値に設定しないでください。これらのパラメータ値のいずれかが大きすぎると、トランザクション・レイヤーのブロック・ヘッダー用に使用する領域が多くなり、データ・レイヤーの変数ヘッダー用に使用する領域が少なくなります。この結果、I/O が増加する可能性があります。

注意を要するグローバル・キャッシュの統計

注意を要する統計は次のとおりです。これらの統計では、常に 0（ゼロ）または 0 に近い値が示されている必要があります。

- **global cache blocks lost** — この統計は転送中のブロックの損失を示し、高い値はネットワークに問題があることを示している可能性があります。UDP など信頼性の低い IPC プロトコルを使用していると、global cache blocks lost の値が 0（ゼロ）以外になる場合があります。このような場合は、global cache blocks lost を、global cache current blocks served と global cache cr blocks served の加算結果で除算した率をできるだけ低くする必要があります。global cache blocks lost の値が 0（ゼロ）以外の場合でも必ずしも問題が発生しているわけではありません。これは、ブロック転送操作は正常終了するまで再試行されるためです。TCP/IP と異なり、UDP/IP は、提供されるエラー・リカバリ・サービスが少ないため、信頼性が低いとみなされています。
- **global cache blocks corrupt** — この統計は、転送中に破損したブロックがあるかどうかを示します。高い値は、IPC、ネットワークまたはハードウェアに問題があることを示します。

待機イベントの分析

動的パフォーマンス・ビューまたは Statspack で高い合計時間が示されたイベントのほとんどは、実際には問題ありません。イベントの応答時間が長くなり、クラスタ・アクセスで待機時間の比率が高いことが Statspack で示される場合は、その待機の原因を特定します。Statspack には、待機時間の割合が高い順に、5 つの待機イベントが表示されます。監視する必要がある統計イベントは、次のとおりです。

- `global cache open s` および `global cache open x`
- `global cache null to s` および `global cache null to x`
- `global cache cr request`
- 論理読取りに対するグローバル・キャッシュ・サービスの使用率

`global cache open s` および `global cache open x`

これらのイベントは、インスタンスによる特定のデータ・ブロックの初期アクセスに関連しています。待機時間は短く、多くの場合、待機後にディスクからの読み込みが行われます。これは、クラスタ・データベース内のインスタンスにブロックがキャッシュされていないためです。

これらのイベントの合計待機時間やトランザクションごとの待機時間が長い場合は、ローカル・インスタンスにデータ・ブロックがキャッシュされず、別のインスタンスからブロックを取得できないために、ディスク読取りが行われる可能性が高くなります。同時に、ローカルなバッファ・キャッシュ・ヒット率が観察されます。

`global cache null to s` および `global cache null to x`

これらは、ブロックがインスタンスで使用され、別のインスタンスに転送されて、最初のインスタンスで再度要求された場合の待機イベントです。これらのイベントを待つプロセスは、通常、最後にブロックを変更したインスタンスからそのブロックが転送されるのを待機します。あるインスタンスが、キャッシュされたデータ・ブロックを別の複数のインスタンスに要求すると、通常、これらのイベントは合計待機時間のかなりの割合を占めます。

`global cache cr request`

このイベントは、インスタンスが読み込み一貫性のためのデータ・ブロックを要求し、転送されたブロックが要求元のインスタンスにまだ到達していない場合の待機イベントです。

グローバル・キャッシュ・サービス時間に関する一般的な注意 `global cache waits` が待機時間に対して高い割合を占め、Statspack レポートの最初のページにリストされ、応答時間またはスループットがサービス・レベルの要件に適合しない場合は、Statspack レポートの「Cluster Statistics」ページで、グローバル・キャッシュ・サービスのワークロードの特性を確認してください。

要求ごとのグローバル・キャッシュ・サービスの平均時間が長い場合は、競合、システム負荷またはネットワークにおける問題が原因である可能性があります。システム・ログとオペ

レーティング・システム統計情報から、ネットワーク・リンクが混雑していること、パケットがプライベート・インターコネクトではなくパブリック・ネットワークを経由していること、実行キューのサイズが増加していることなども判断できます。

CPU 使用率が最大限に近く、プロセスが CPU に対してキューされている場合は、グローバル・キャッシュ・サービス・プロセス (LMSn) の優先順位を他のプロセスより上位にすると、グローバル・キャッシュ・サービスの時間を大幅に短縮できます。データベース・サーバー上のプロセス数の削減、CPU のサーバーへの追加、ノードのクラスタ・データベースへの追加なども考慮してください。

関連項目： Real Application Clusters データベースにノードとインスタンスを追加する手順は、『Oracle9i Real Application Clusters 管理』を参照してください。

論理読取りに対するグローバル・キャッシュ・サービスの使用率

バッファ・キャッシュ読取りまたは論理読取りの数に相対するグローバル・キャッシュ・サービスの使用率を見積もるには、グローバル・キャッシュ・サービス要求の合計を、指定の統計収集間隔の論理読取り数で除算します。

ユーザーがデータ・ブロックの読込みまたは変更のためにバッファ・キャッシュにアクセスし、そのブロックがローカル・キャッシュに存在しない場合は、グローバル・キャッシュ・サービス要求が行われます。この結果、リモート・キャッシュ読取り、ディスク読取りまたはアクセス権限の変更が行われます。つまり、論理読取りは、グローバル・キャッシュ・サービス操作のスーパーセットです。計算式は次のとおりです。

$$\frac{\text{global cache gets} + \text{global cache converts} + \text{global cache cr blocks received} + \text{global cache current blocks received}}{\text{consistent gets} + \text{db block gets}}$$

一部のブロックが非常にホットな（つまり、ローカル・ユーザーおよびリモート・ユーザーからの要求の頻度が高い）場合があります。このため、ローカル・ユーザーの作業完了まで、ブロック転送が数ミリ秒遅延する場合があります。遅延が発生する可能性の概算は、次の率で示すことができます。

$$\frac{\text{global cache defers}}{\text{global cache current blocks served}}$$

率が 0.3 を超える場合は、非常にホットなデータ・セットであることを示します。このような場合は、busy waits に含まれるブロックを詳しく分析します。分析を行うには、V\$CACHE_TRANSFER ビューの NAME、TYPE、FORCED_READS、FORCED_WRITES などの列

を問い合わせるか、または V\$OBJ_STATS ビューの CR_TRANSFERS 列および CURRENT_TRANSFERS 列を調べます。また、4-9 ページの「グローバル・キャッシュ・サービスの待機イベント」で説明したように、global cache busy、buffer busy global cache および buffer busy global CR の各統計で表示される値も調べます。

問題を検出した場合は、その問題が発生しているオブジェクト、そのオブジェクトにアクセスしているインスタンス、およびそのオブジェクトへのアクセス方法を識別します。必要に応じて、次の要因を削減して競合を解決します。

- ホット・スポット—シングル・インスタンスの Oracle データベースと同様に、索引ブロックまたはデータ・ブロックへのアクセスを分散させたり、Oracle のハッシュ・パーティション化またはレンジ・パーティション化を可能なかぎり使用して削減します。
- オブジェクトの並行性—ロード・バランシングまたはリソース管理を実装して削減します。たとえば、オブジェクトに対する変更の割合を削減（使用するデータベース・プロセスを少なくする）します。

前述したとおり、シングル・インスタンスの Oracle データベースおよび Real Application Clusters では、エージング、キャッシュの置換またはチェックポイントのためにのみ、ブロックがディスクに書き込まれます。エージングのためにデータ・ブロックがキャッシュから置換される場合、またはチェックポイントが発生し、ブロックが以前に別のインスタンスで変更されたがディスクに書き込まれていない場合、Oracle は、データ・ブロックをディスクに書き込むことを他のインスタンスに通知するメッセージを送信します。これは**フュージョン書込み**と呼ばれます。次の率は、Oracle がこの方法で管理する書込みの割合を示します。

DBWR fusion writes

physical writes

この率が高いほど、以前の変更時に他のインスタンスにコピーされた（書き込まれた）ブロックの数が多いことを示します。率が高い場合は、キャッシュ・サイズが不十分か、ある期間チェックポイントが発生しなかったことが原因である可能性があります。また、高い率は、キャッシュの置換またはチェックポイントが原因で書き込まれたバッファで構成されるグローバルな作業セットの割合を示します。たとえば、率が 0.1 の場合は、ディスクに書き込まれたバッファの 10% がグローバルに使用済みであることを示します。

フュージョン書込みは、ディスクへの追加の書込みではありません。ただし、フュージョン書込みには、転送処理と他のインスタンスを調整するためのメッセージ機能が必要です。したがって、フュージョン書込みは、インスタンスで発生するすべての **physical writes** のサブセットになります。

CACHE_TRANSFER ビューを使用した Real Application Clusters 統計の分析

V\$CACHE_TRANSFER ビューと V\$FILE_CACHE_TRANSFER ビューを使用して、Real Application Clusters 統計を調べます。V\$CACHE_TRANSFER ビューには、Oracle が **インターネット** 経由で転送するブロックのタイプとクラスがオブジェクトごとに示されます。Real Application Clusters のインスタンスが共有するオブジェクトのタイプを判断するには、FORCED_READS 列と FORCED_WRITES 列を使用します。FORCED_WRITES 列の値は、別のインスタンスがカレント・バージョンを要求したために、あるブロック・タイプにローカル・バッファ・キャッシュからの転送が発生した頻度を示します。

キャッシュ転送が発生したファイルを識別するには、V\$FILE_CACHE_TRANSFER を使用します。たとえば、V\$CACHE_TRANSFER の NAME 列には、オブジェクトの名前が表示されず。したがって、このビューを使用して、オブジェクトごとにブロック転送を評価します。

共有ディスク・アーキテクチャによってディスク書込みの強制実行が排除される場合でも、V\$CACHE_TRANSFER ビューと V\$FILE_CACHE_TRANSFER ビューには、ブロック・クラスまたはオブジェクトごとのブロック・モード変換数が示されます。ただし、FORCED_WRITES 列の値は 0 (ゼロ) になります。

Oracle Performance Manager を使用した Real Application Clusters のパフォーマンス の監視

この章では、**Oracle Real Application Clusters** 固有の **Oracle Performance Manager** チャートについて説明します。この章で説明するチャートを表示するには、Oracle Performance Manager をインストールして構成する必要があります。この章は、『Oracle Enterprise Manager Oracle Diagnostics Pack スタート・ガイド』に記載されている情報の補足としてお読みください。内容は次のとおりです。

- **Oracle Performance Manager** の概要
- **Real Application Clusters** での **Oracle Performance Manager** チャートの表示

関連項目：

- Oracle Performance Manager のインストールの詳細は、『Oracle9i Real Application Clusters セットアップおよび構成』を参照してください。
- Performance Manager チャートに表示される統計とその解釈方法の詳細は、『Oracle9i データベース・パフォーマンス・ガイドおよびリファレンス』を参照してください。
- Performance Manager チャートのフィールドとそれが導出される V\$ ビューの詳細は、『Oracle9i データベース・リファレンス』を参照してください。

Oracle Performance Manager の概要

Diagnostics Pack の一部である Oracle Performance Manager は、主要なパフォーマンス統計を表示します。この統計を監視して、Real Application Clusters データベースのパフォーマンスを最適化できます。Oracle Performance Manager で提示されるパフォーマンス・データは、Intelligent Agent によって取得および計算されたデータです。Oracle Performance Manager チャートには、インスタンス固有の V\$ ビューの情報およびグローバルな GV\$ ビューの情報が表示されます。

以前のリリースの Oracle クラスタ・ソフトウェアでの Performance Manager チャートの使用

現行および以前のバージョンの Oracle クラスタ・データベースのパフォーマンスを調べるには、Oracle Performance Manager を使用します。リリース 2 (9.2) より前のバージョンの Performance Manager を使用するには、使用するバージョンの Oracle クラスタ・ソフトウェアのマニュアルを参照してください。

注意： Oracle Real Application Clusters リリース 1 (9.0.1) のマニュアルで説明されていたチャートの一部は、リリース 1 (9.0.1) ではすでに使用できません。

Real Application Clusters での Oracle Performance Manager チャートの表示

Oracle Performance Manager には、[図 5-1](#) に示すように、「Cluster Database」アイコンの下にあるアイコンからアクセスできる主レベルのチャートがあります。一部の主チャートには、インスタンス別、ファイル別およびオブジェクト別のパフォーマンス情報を表示する下位レベルのチャートもあります。

図 5-1 Oracle Performance Manager の Real Application Clusters 固有のチャート

The screenshot displays the Oracle Performance Manager interface for the instance OPSMB1.US.ORACLE.COM - SYSTEM. The left pane shows a tree view of charts, with 'Cluster Database' expanded to show RAC-specific charts. The right pane shows a table of charts in the selected group.

Chart Name	Description
General Overview	Top Transfers, Global Cache Converts and
Top Consumers	User Sessions and Top Sessions charts
Resource Contention	Library Cache Lock and Row Cache Lock ch

Real Application Clusters 固有の Performance Manager チャートの階層

表 5-1 は、Performance Manager チャートの階層を示します。左側の列にある主チャートから、右側の列にある下位レベルのチャートにナビゲートできます。

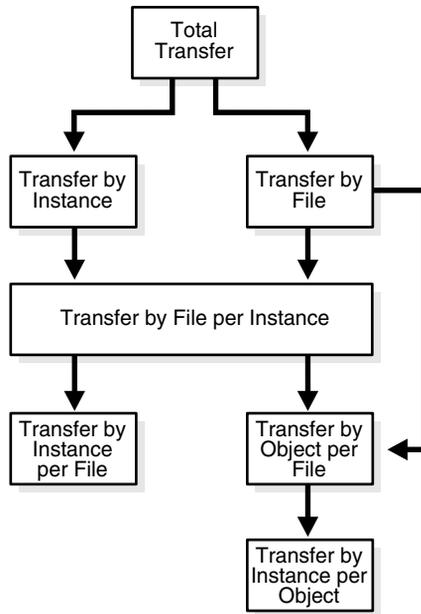
表 5-1 Real Application Clusters 固有のチャートの階層

主チャート	下位レベルのチャート
RAC Database Health Overview	-
Total Transfer	Transfer By Instance
Total Transfer チャートの階層の詳細は、 図 5-2 を参照してください。	Transfer By File この 2 つのチャートから、次のチャートにアクセスできます。 Transfer By File Per Instance Transfer By Instance Per File Transfer By Object Per File Transfer By Instance Per Object
Global Cache CR Request	Global Cache CR Request By Instance
Global Cache Convert	Global Cache Convert By Instance
Library Cache Lock	Library Cache Lock By Instance
Row Cache Lock	Row Cache Lock By Instance
Global Cache Current Block Request	Global Cache Current Block Request By Instance
File I/O Rate	File I/O Rate By Object File I/O Rate By Instance
Sessions	
Users	Users By Instance
Real Application Clusters Top Sessions	-

関連項目： これらのチャートの詳細は、Performance Manager のオンライン・ヘルプを参照してください。

図 5-2 は、Total Transfer チャートとその下位レベルのチャートの階層を示します。

図 5-2 Real Application Clusters 固有の Enterprise Manager チャートの階層



次の各項で、主レベルと下位レベルのチャートについて説明します。

- Total Transfer チャート
- Global Cache CR Request チャート
- Global Cache Convert チャート
- Library Cache Lock チャート
- Row Cache Lock チャート
- Global Cache Current Block Request チャート
- File I/O Rate チャート
- Session チャート
- Users チャート
- Real Application Clusters Top Sessions チャート

Performance Manager の Real Application Clusters 固有のチャート

この章で説明する Performance Manager チャートは、一般的に、一定の時間内に発生したアクティビティの比率を表示します。この項では、Real Application Clusters 固有の次のチャートについて説明します。

- Total Transfer チャート
- Global Cache CR Request チャート
- Global Cache Convert チャート
- Library Cache Lock チャート
- Row Cache Lock チャート
- Global Cache Current Block Request チャート

Total Transfer チャート

Total Transfer チャートは、クラスタ・データベース全体に対する次のアクティビティの比率を表示します。

- Logical Reads (論理読取り) : 読み一貫性およびカレント・ブロックが要求された回数 (consistent gets と db block gets の値の合計)
- Physical Reads (物理読取り) : バッファ・キャッシュをバイパスしないで、ディスクからブロックを読み込んだ回数 (physical reads と physical reads direct の差異)
- キャッシュ転送 : 読み一貫性およびカレント・ブロックがリモート・キャッシュから読み込まれた回数 (global cache cr blocks received と global cache current blocks received の値の合計)

インスタンスごとにこれらの比率を調べるには、Transfer By Instance チャートにドリルダウンします。

このチャートを使用すると、Real Application Clusters データベースの次のパフォーマンスを分析できます。

- physical reads の比率が低く、キャッシュ間転送の比率も低い場合は、理想的なワークロードであり、ノードの追加が必要な場合にスケーラブルなシステムであることを示します。
- physical reads の比率が高く、キャッシュ間転送の比率も高い場合は、アプリケーションの作業セットが大きいため、全体をクラスタにキャッシュできないことを示します。各インスタンスのバッファ・キャッシュのサイズを増やすか、別のノードを追加することで、この比率を低くすることができます。
- キャッシュ間転送の logical reads の比率が高く、physical reads の比率が低い場合は、Transfer By File チャートにドリルダウンして詳細に分析します。

Transfer By File チャート

Transfer By File チャートには、クラスタ・データベースの各データ・ファイルについて、CR ブロックおよびカレント・ブロックの転送の比率が表示されます。この統計は、GV\$FILE_CACHE_TRANSFER ビューの `cr_transfers` と `cur_transfers` の値の合計から導出されます。インスタンスごとの転送率は、Transfer By File Per Instance チャートにドリルダウンして取得します。

このチャートを使用して、転送率が最も高いファイルを識別します。ファイルを選択してから、Transfer By Object Per File チャートにドリルダウンして、最も高い転送率の対象となるオブジェクトを識別します。また、このチャートには、各ノードからそのオブジェクトにアクセスする方法も表示されます。これによって、表や索引などのホット・オブジェクトを識別できます。その後、Oracle のハッシュ・パーティション化やレンジ・パーティション化の使用などの対処措置を識別できます。

Global Cache CR Request チャート

Global Cache CR Request チャートには、クラスタ・データベース全体に関する次の統計が表示されます。

- 平均 global cache CR request time。global cache cr block receive time の 10 倍を、global cache cr blocks received の値で除算した比率です。

インスタンスごとの平均 CR request time を調べるには、Global Cache CR Request By Instance チャートにドリルダウンします。

チャートに表示された CR request time の値が高い場合は、システム負荷が高い、プライベート・ネットワークではなくパブリック・インターコネクトを使用している、ネットワーク・エラーの発生、LMS プロセスでの低い CPU 使用率など、考えられる原因を調べます。

Global Cache Convert チャート

Global Cache Convert チャートには、クラスタ・データベース全体に関する次の統計が表示されます。

- 平均 global cache convert time — global cache convert time の 10 倍を global cache converts で除算した比率です。
- 平均 global cache get time — global cache get time の 10 倍を global cache gets で除算した比率です。

インスタンスごとにこれらの統計を調べるには、Global Cache Convert By Instance チャートにドリルダウンして、次の点を考慮します。

- 平均 convert time の値が高い場合は、昇順にデータが増加する索引ツリーが原因である可能性があります。使用している環境でこの問題が発生した場合は、索引でハッシュ・パーティション化またはレンジ・パーティション化を使用して、索引リーフ・ブロックへのアクセスを分散させます。
- 平均 convert time と平均 get time の両方の値が高い場合は、Global Cache CR Request By Instance チャートで前述したように、システム、インターコネクト、ネットワーク、LMS プロセス、CPU 使用率のすべてに関連する問題が発生している可能性があります。

Library Cache Lock チャート

Library Cache Lock チャートには、クラスタ・データベース全体に関する次の統計が表示されます。

- dlm_lock_requests — データベース・オブジェクトに対してロックが要求された回数を表します。
- dlm_pin_requests — データベース・オブジェクトに対して確保が要求された回数を表します。
- dlm_invalidations — 他のインスタンスから受信した無効の数を表します。

インスタンスごとにこれらの統計を調べるには、Library Cache Lock By Instance チャートにドリルダウンします。

ロック要求と確保要求の回数が多い場合は、PL/SQL ユニットの頻繁な実行、高い解析量、バインド変数の非効率な使用などの原因を検索します。

無効の数が多く場合は、スキーマ・オブジェクトを分析、削除または変更する DDL 文または PL/SQL コンパイル・コマンドを検索します。

Row Cache Lock チャート

Row Cache Lock チャートには、クラスタ・データベース全体に関する次の統計が表示されます。

- `dml_requests` – CR ブロックに対して生成された DLM ロック要求の回数を表します。
- `dml_conflicts` – インスタンスが要求したモードと競合するモード内の CR ブロックに対するロックを、他のインスタンスがすでに所有していた場合の回数を表します。
- 結果として `dml_conflicts` となる `dml_requests` の割合。

インスタンスごとにこれらの統計を調べるには、**Row Cache Lock By Instance** チャートにドリルダウンします。このチャートで示す統計によって、ディクショナリの競合が識別されません。DLM 競合の割合が高い場合は、ディクショナリ管理表領域、キャッシュされていない順序、データベース・オブジェクトの頻繁な作成と削除など、ディクショナリ競合の原因を調べます。

Global Cache Current Block Request チャート

Global Cache Current Block Request チャートには、クラスタ・データベース全体に関する次の統計が表示されます。

- 平均 `global cache current block request time` – `global cache current block receive time` の 10 倍を、`global cache current blocks received` で除算した比率です。
- 平均 `global cache current block serve time` – `global cache current block serve time` の 10 倍を、`global cache current blocks served` で除算した比率です。

インスタンスごとにこれらの統計を調べるには、**Global Cache Current Block Request By Instance** チャートにドリルダウンします。`serve time` の値が高い場合は、一般的に、`LMSn` プロセスのパフォーマンスが低いことが原因です。平均 `global cache CR request time` の場合と同様に、平均 `current block request time` と平均 `current block serve time` の値が高い場合は、高いシステム負荷、パブリック・インターコネクトの使用、ネットワーク・エラーの発生、`LMSn` プロセスの低い優先順位、`LMSn` プロセスでの低い CPU 使用率などが原因である可能性があります。

Real Application Clusters 固有の Oracle チャート

次に、クラスタ化されたシングル・インスタンスの Oracle データベースのチャートについて説明します。この項で説明するチャートは次のとおりです。

- [File I/O Rate チャート](#)
- [Session チャート](#)
- [Users チャート](#)
- [Real Application Clusters Top Sessions チャート](#)

File I/O Rate チャート

File I/O Rate チャートは、クラスタ・データベース内のすべてのファイルに対する **physical reads** および **physical writes** の割合を表示します。データ・ファイルごとに **physical reads** および **physical writes** の割合を調べるには、**File I/O Rate By Object** チャートにドリルダウンします。インスタンスごとに **physical reads** および **physical writes** の割合を調べるには、**File I/O Rate By Instance** チャートにドリルダウンします。

Session チャート

Session チャートは、クラスタ・データベース全体と連結されたセッションとその関連情報（インスタンス名、セッション ID、プロセス ID、状態、ユーザー名など）を表示します。

Users チャート

Users チャートは、クラスタ・データベース全体について次の統計を表示します。

- アクティビティの生成に関係なく、ログインしたユーザー・セッションの合計数
- アクティブなユーザー・セッションの合計数

ログインしたユーザー数とアクティブなユーザー数をインスタンスごとに調べるには、**Users By Instance** チャートにドリルダウンします。

Real Application Clusters Top Sessions チャート

Real Application Clusters Top Sessions チャートは、クラスタ・データベースのインスタンスのアクティビティに最も影響を与えた一連のセッションを、選択したソート統計に基づいて表示します。

Real Application Clusters Database Health Overview チャート

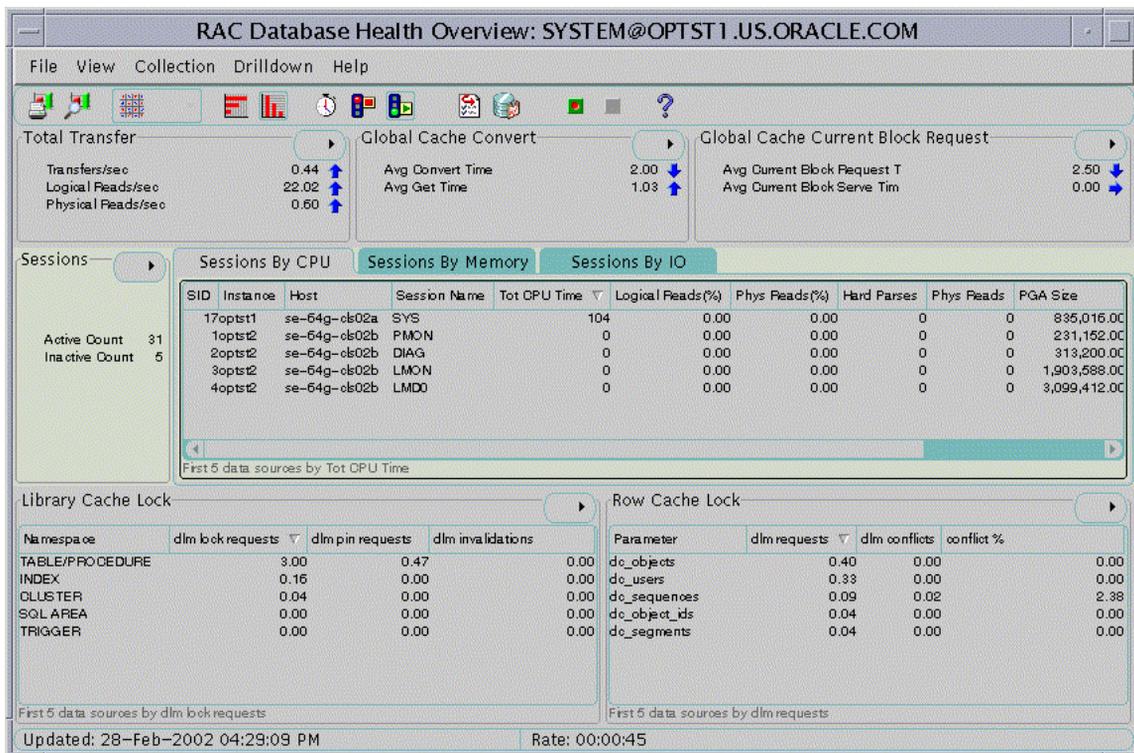
図 5-3 に、Real Application Clusters Database Health Overview チャートを示します。このチャートは、クラスタ・データベースに関する主要なパフォーマンス統計を表示するスタンドアロン・チャートの集合です。このチャートは、次のスタンドアロン・チャートで構成されています。

- Total Transfer
- Global Cache Convert
- Global Cache Current Block Request
- Library Cache Lock
- Row Cache Lock

これらのチャートは、スタンドアロン・チャートとして使用している場合と同様の方法で、下位レベルのチャートにドリルダウンできます。さらに、Database Health Overview チャートの中央には、スタンドアロンのチャートでは使用できない、クラスタ・データベースの場合のセッション情報が表示されます。セッション情報の表示は、CPU、メモリーまたは I/O に基づいてカスタマイズできます。

注意：この Database Health Overview チャートは、Enterprise Manager のコンソールからも表示できます。表示するには、クラスタ・データベースの右マウス・メニューからメニュー項目の「Performance View」を選択します。

図 5-3 Real Application Clusters Database Health Overview チャート



第 III 部

Real Application Clusters のリファレンス 情報

第 III 部では、**Oracle Real Application Clusters** のリファレンス情報を示します。第 III 部に含まれるリファレンス情報は、次のとおりです。

- 付録 A 「マルチブロック・ロックの構成 (オプション)」
- 付録 B 「Real Application Clusters での空きリストと空きリスト・グループの使用 (オプション)」
- 用語集

マルチブロック・ロックの構成（オプション）

この付録では、複数ブロックへのアクセスを管理するためのロックの構成方法について説明します。この付録は、**グローバル・キャッシュ・サービス**（GCS）および**グローバル・エンキュー・サービス**（GES）で実行されるように、**Oracle Real Application Clusters** のデフォルト・リソース制御スキームを上書きするという例外的な状況にのみ参照してください。内容は次のとおりです。

- GCS および GES のリソース制御メカニズムの上書き前の注意
- GCS 処理および GES 処理の上書きの判断
- GC_FILES_TO_LOCKS の設定
- GC_FILES_TO_LOCKS の設定に対する追加の考慮点
- Real Application Clusters でのパラレル実行のチューニング
- Real Application Clusters の I/O 統計の分析
- 不適切な強制書込みの検出によるマルチブロック・ロック使用量の監視
- ロック名の書式

GCS および GES のリソース制御メカニズムの上書き前の注意

デフォルト・スキームは、すべての Real Application Clusters 環境で、すべての種類のシステムに非常に優れたパフォーマンスを提供します。さらに、ロックを割り当てるには、追加の管理工程が必要です。そのため、この付録の説明に従って、デフォルトのスキームを上書きするために必要なタスクを実行するよりも、デフォルトのスキームを使用することをお勧めします。

注意： この付録は、データ・アクセス・パターンがほとんど読み込みのみであるアプリケーションの場合など、例外的な場合にのみ参照してください。

GCS 処理および GES 処理の上書きの判断

キャッシュ・フュージョンは、ローカルにキャッシュされないデータのキャッシュ間転送を使用して、非常に優れた拡張性およびパフォーマンスを提供します。キャッシュ・フュージョンでは、**インスタンス**がディスクからデータ・ブロックを読み込む前に、Oracle は別のインスタンスのキャッシュから、要求されたデータの取得を試みます。要求されたブロックが別のキャッシュに存在する場合、そのデータ・ブロックは、**インターコネクト**を介して保持側インスタンスから要求側インスタンスに転送されます。

Real Application Clusters のリソース制御スキームは、複数インスタンスによって変更されたデータの整合性を保証します。デフォルトでは、インスタンスのバッファ・キャッシュ内の各データ・ブロックは、GCS によって保護されています。GCS は、アクセス・モード、ルール、権限およびこれらのリソースの状態を追跡します。

まれに、1つのファイル内の複数データ・ブロックを1つのロックで処理するマルチブロック・ロックを構成して、GCS および GES を上書きすることが必要となる場合があります。ブロックが同じインスタンスから頻繁にアクセスされる場合、または同時読み込みの共有モードなどの互換性のあるモードでブロックが複数**ノード**からアクセスされる場合、ロック構成によって、パフォーマンスが向上することがあります。

これを実行するには、GC_FILES_TO_LOCKS パラメータを設定し、特定のファイルで使用されるロックの数を指定します。また、パラメータの構文では、ファイル・グループへのロックの割当て、およびロックごとに処理される連続データ・ブロックの数が指定できます。GC_FILES_TO_LOCKS の値を無計画に使用すると、過剰な**ディスク書き込みの強制実行**などの、パフォーマンスを低下させる操作が行われる可能性があります。そのため、GC_FILES_TO_LOCKS は、次の場合のみに設定します。

- 読取り専用または読取りが主なファイルおよび表領域
- 一時表領域、READ ONLY とマークされた表領域、およびロールバック・セグメントを含む表領域に対応付けられていないデータ・ファイル

注意： 特定のファイルに対して GC_FILES_TO_LOCKS パラメータを設定できるため、その特定のファイルについてのみキャッシュ・フュージョン処理を使用禁止にすることができます。したがって、同じ Real Application Clusters 環境内で、マルチブロック・ロック割当ておよび GCS と GES (キャッシュ・フュージョン) 処理を使用できます。

ロックを使用する場合

各ファイルにロックを複数使用すると、表 A-1 に示すようなデータに効果的です。

表 A-1 ロックを使用する場合

状況	理由
データ構成が主に読取り専用の場合	少数のロックで、多くのブロックを処理できます。ロック・オペレーションもほとんど必要ありません。これらのロックは、別のインスタンスでデータを変更する必要がある場合のみ解放されません。ロックを割り当てると、読取り専用データの パラレル実行 処理のパフォーマンスが向上します。データが厳密に読取り専用の場合は、表領域を読取り専用として指定することを考慮する必要があります。
比較的少数のインスタンスの集合によって、多量のデータが変更される場合	ロック割当てによって、キャッシュされていないデータベース・ブロックへのアクセスが、 パラレル・キャッシュ管理 アクティビティなしで継続できます。ただし、これは、そのブロックが要求側インスタンスのキャッシュにすでにある場合にのみ有効です。

ロックを使用すると、異なるデータベース・ブロックを変更するインスタンス間で競合が発生することがあるため、より多くのインスタンス間キャッシュ管理アクティビティが発生する場合があります。不適切なディスク書込みの強制実行、または過剰なディスク書込みの強制実行を解決するには、現在ブロックにアクセスしているインスタンスのキャッシュから、いくつかのブロックを書き込む必要があります。

GC_FILES_TO_LOCKS の設定

データ・ファイルまたはデータ・ファイルの集合内のデータ・ブロックを処理するロックの数を指定するには、GC_FILES_TO_LOCKS 初期化パラメータを設定します。この項の内容は次のとおりです。

- GC_FILES_TO_LOCKS 構文
- ロックの割当て例

GC_FILES_TO_LOCKS 構文

GC_FILES_TO_LOCKS パラメータの構文では、ロックとファイルの関係を指定できます。表 A-2 に、このパラメータの構文と変数の意味を示します。

```
GC_FILES_TO_LOCKS="{file_list=#locks[!blocks] [EACH][:]} . . ."
```

表 A-2 GC_FILES_TO_LOCKS の変数とその意味

変数	意味
<i>file_list</i>	次のように、単一のファイル、ファイルの範囲、またはファイルおよび範囲のリストを指定します。 <i>fileidA[-fileidC][fileidE[-fileidG]] ...</i> データ・ディクショナリ・ビュー DBA_DATA_FILES を問い合せて、ファイル名とファイル ID 番号の間で対応するファイルを検索します。
<i>#locks</i>	<i>file_list</i> に割り当てるロックの数を設定します。
<i>!blocks</i>	各ロックによって処理される、連続するデータ・ブロックの数（ブロッキング要素）を指定します。
EACH	<i>#file_list</i> 内の各ファイルに割り当てるロックの数を、 <i>#locks</i> に設定します。

注意： すべてのインスタンスの GC_FILE_TO_LOCKS の値は同一である必要があります。また、GC_FILES_TO_LOCKS パラメータ構文の引用符内には、空白を使用しないでください。

!blocks のデフォルト値は 1 です。*blocks* を指定する場合、連続するデータ・ブロックは、*#locks* ロックの各自によって処理されます。*blocks* の値を指定するには、「!」セパレータを使用します。主に *blocks* を指定し、EACH キーワードは指定しないで、複数のデータ・ファイルを処理するロックの集合を割り当てます。

オプションの空きリスト・グループも使用する場合は、データ・パーティション化への干渉を回避するため、必ず、*!blocks* に値を設定します。通常、エクステンツを事前に割り当てる必要はありません。行が表に挿入され、新しいエクステンツが割り当てられる場合、

GC_FILES_TO_LOCKS の *!blocks* で指定された連続するブロックは、1つのインスタンスに対応付けられた空きリスト・グループに割り当てられます。

ロックの割当て例

初期化パラメータ・ファイルに次の行を追加して、300のロックをファイル1に、100のロックをファイル2に割り当てることができます。

```
GC_FILES_TO_LOCKS = "1=300:2=100"
```

次のエントリによって、合計1500のロックが、ファイル1、2および3に500ずつ割り当てられます。

```
GC_FILES_TO_LOCKS = "1-3=500EACH"
```

対照的に、次のエントリでは、3つのファイルに合計500のロックが割り当てられます。

```
GC_FILES_TO_LOCKS = "1-3=500"
```

次のエントリは、1000の個別のロックを使用して、ファイル1を保護する必要があることを示します。ファイル内のデータは、25の連続するロックごとに保護されます。

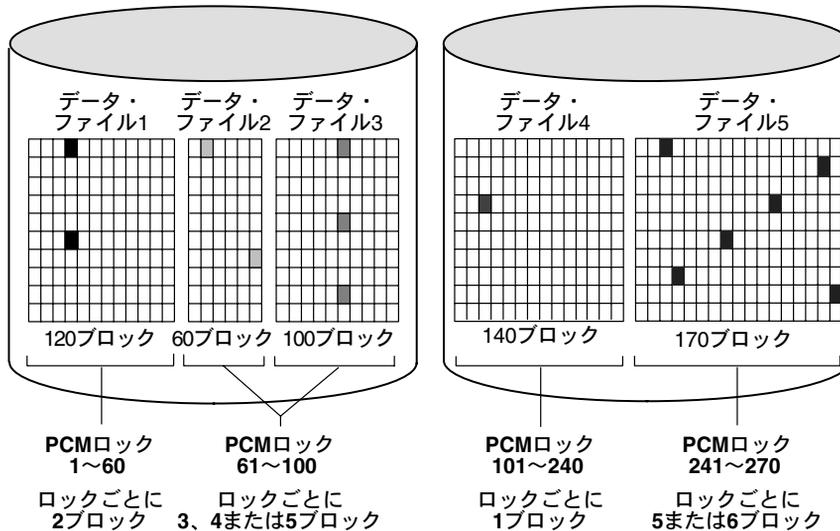
```
GC_FILES_TO_LOCKS = "1=1000!25"
```

データ・ファイルを AUTOEXTEND 句で定義するか、または ALTER DATABASE... DATAFILE... RESIZE 文を発行する場合は、ロック割当てを調整する必要がある場合があります。

新しいデータ・ファイルを追加する場合は、これらの新しいファイルを GCS のデフォルト制御の対象とするかどうか、または GC_FILES_TO_LOCKS 初期化パラメータを使用してロックを割り当てかどうかを判断します。

次の例では、異なる方法を使用したロックへのブロックのマッピング、および複数のデータ・ファイルでの同じロックの使用方法を示します。

図 A-1 データ・ブロックへのロックのマッピング



例 1 図 A-1 に、パラメータ値 `GC_FILES_TO_LOCKS = "1=60:2-3=40:4=140:5=30"` に対するロックへのブロックのマッピング例を示します。

図 A-1 に示すデータ・ファイル1では、60のロックが、60の倍数である120ブロックにマップされます。各ロックは2つのデータ・ブロックを処理します。

データ・ファイル2および3では、40のロックが合計160ブロックにマップされます。1つのロックは、データ・ファイル2では1つまたは2つのデータ・ブロックを、データ・ファイル3では2つまたは3つのデータ・ブロックを処理できます。したがって、1つのロックが、この2つのデータ・ファイル全体の、3つ、4つまたは5つのデータ・ブロックを処理できます。

データ・ファイル4では、ロックの数とデータ・ブロックの数と同じであるため、各ロックは単一のデータ・ブロックにマップされます。

データ・ファイル5では、30のロックが、30の倍数でない170ブロックにマップされます。したがって、各ロックは5つまたは6つのデータ・ブロックを処理します。

図 A-1 で示したロックはそれぞれ、共有読み込みモードまたは読み込み排他モードのどちらかに保持されます。

例 2 次のパラメータ設定では、データ・ファイル1に500のロックを、データ・ファイル2、3、4、10、11および12にそれぞれ400のロックを、データ・ファイル5に150のロックを、データ・ファイル6に250のロックを、ファイル7~9にまとめて300のロックを割り当てます。

```
GC_FILES_TO_LOCKS = "1=500:2-4,10-12=400EACH:5=150:6=250:7-9=300"
```

この例では、 $(500 + (6 \times 400) + 150 + 250 + 300) = 3600$ で、合計 3600 のロックを割り当てます。さらに多くのデータ・ファイルを追加する場合は、これより多くのロックを指定できます。

例 3 例 2 では、句「7-9=300」によって 300 のロックがまとめてデータ・ファイル 7、8 および 9 に割り当てられます。キーワード EACH は省略されています。これらのデータ・ファイルにそれぞれ 900 のデータ・ブロックがある場合、データ・ブロックの合計は 2700 となるため、各ロックは 9 つのデータ・ブロックを処理します。データ・ファイルが 300 の倍数であるため、9 つのロックは、各データ・ファイルで 3 つのデータ・ブロックを処理します。

例 4 次のパラメータ値は、ファイル 1～3 にそれぞれ 200 のロックを、データ・ファイル 4 に 50 のロックを、データ・ファイル 5、6、7 および 9 にまとめて 100 のロックを、データ・ファイル 8 と 10 の組合せに、連続する 50 のブロック・グループの 20 のロックを割り当てます。

```
GC_FILES_TO_LOCKS = "1-3=200EACH 4=50:5-7,9=100:8,10=20!50"
```

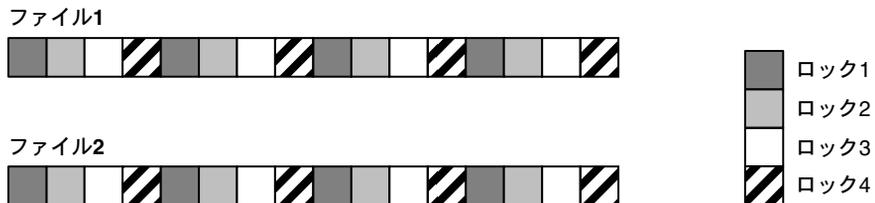
この例では、結合したデータ・ファイル 5、6、7 および 9 に割り当てられたロックは、各データ・ファイルにおいて 1 つ以上のデータ・ブロックを処理します。ただし、1 つのデータ・ファイルが、99 以下のデータ・ブロックを含む場合を除きます。データ・ファイル 5～7 がそれぞれ 500 のデータ・ブロックを含み、データ・ファイル 9 が 100 のデータ・ブロックを含む場合、各ロックは 16 のデータ・ブロックを処理します。この場合、データ・ファイル 9 で 1 つ、その他のデータ・ファイルでそれぞれ 5 つのデータ・ブロックが処理されます。または、データ・ファイル 9 が 50 のデータ・ブロックを含む場合は、ロックの半数が 16 のデータ・ブロック（そのうちの 1 つはデータ・ファイル 9）を処理し、残りの半分のロックは 15 のデータ・ブロックのみ（データ・ファイル 9 にはなし）を処理します。

データ・ファイル 8 および 10 にまとめて割り当てられた 20 のロックは、50 のデータ・ブロックの連続するグループを処理します。データ・ファイルが 50 の倍数のデータ・ブロックを含み、かつデータ・ブロックの合計数が 20×50 （つまり 1000）以下の場合、各ロックは、データ・ファイル 8 またはデータ・ファイル 10 のいずれかのデータ・ブロックを処理します。これは、ロックがそれぞれ 50 の連続するデータ・ブロックを処理するためです。データ・ファイル 8 のサイズが、50 のデータ・ブロックの倍数でない場合、1 つのロックは、両方のファイルのデータ・ブロックを処理する必要があります。データ・ファイル 8 および 10 のサイズが、1000 のデータ・ブロックを超える場合、いくつかのロックは、50 のデータ・ブロックのグループを 2 つ以上処理する必要があります。このグループは、1 つのデータ・ファイルにあるとは限りません。

例 5 GC_FILES_TO_LOCKS=「1-2=4」

この例では、ファイル 1 および 2 に対して 4 つのロックが指定されています。したがって、各ロックが処理するブロックの数は $8((16+16)/4)$ です。これらのブロックは連続していません。

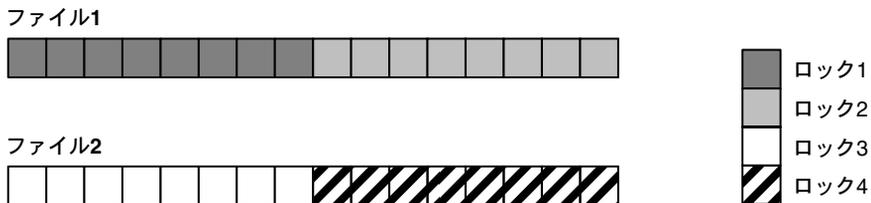
図 A-2 GC_FILES_TO_LOCKS の例 5



例 6 GC_FILES_TO_LOCKS=「1-2=4!8」

この例では、ファイル 1 および 2 に対して 4 つのロックが指定されています。ロックは、8 つの連続するブロックを処理する必要があります。

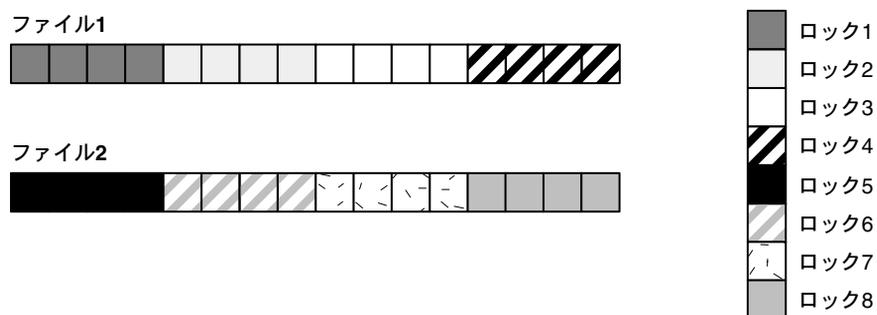
図 A-3 GC_FILES_TO_LOCKS の例 6



例 7 GC_FILES_TO_LOCKS=「1-2=4!4EACH」

この例では、ファイル 1 に対して 4 つのロック、およびファイル 2 に対しても 4 つのロックが指定されています。ロックは、4 つの連続するブロックを処理する必要があります。

図 A-4 GC_FILES_TO_LOCKS の例 7



例 8 GC_FILES_TO_LOCKS=「1=4:2=0」

この例では、ファイル1が、4つのロックでマルチブロック・ロック制御されています。ファイル2には、ロックは割り当てられていません。

図 A-5 GC_FILES_TO_LOCKS の例 8



GC_FILES_TO_LOCKS の設定に対する追加の考慮点

Real Application Clusters で GC_FILES_TO_LOCKS を設定する際には、さらに考慮が必要な項目があります。たとえば、GC_FILES_TO_LOCKS を設定すると、監視によるオーバーヘッドが増大し、データベースが増大する場合やファイルを追加する場合に、頻繁にパラメータを調整する必要があります。また、GC_FILES_TO_LOCKS の設定を動的に変更することはできません。この設定を変更するには、インスタンスを停止して設定を変更し、すべてのインスタンスを再起動する必要があります。さらに、この項で説明する次のことを考慮する必要があります。

- データ・ファイルの拡張または追加
- GC_FILES_TO_LOCKS 設定に含めないファイル

データ・ファイルの拡張または追加

継続的に実行しているサイトは、パラメータ値を調整するために停止することはできません。したがって、GC_FILES_TO_LOCKS パラメータを使用する場合、増大またはファイルの拡張に備えて領域を確保しておく必要があります。

また、読取り専用ファイルや主に読取り専用ファイルなどの、あまり増大しないファイルでのロックの使用方法を慎重に検討する必要があります。複数ブロックに割り当てるロック数を減らすことで、パフォーマンスが向上する場合があります。ただし、ロック数の減少によって予想される CPU およびメモリー使用の節約よりも、管理オーバーヘッドの方が大きい場合、GCS および GES のリソース制御スキームを使用してください。

GC_FILES_TO_LOCKS 設定に含めないファイル

次の種類のファイルは、GC_FILES_TO_LOCKS パラメータ・リストに含めないでください。

- ロールバック・セグメントを含むファイル
- 一時表領域の一部であるファイル
- 明示的に READ ONLY に設定される、表領域内の読取り専用データを格納したファイル。例外として、表領域が予備ロックに対して競合する必要がないことを保証するために、単一のロックを割り当てることができます（このロックの設定は必須ではありません）。

Real Application Clusters でのパラレル実行のチューニング

デフォルトのリソース制御スキームを使用せずに Real Application Clusters 環境でパラレル実行を最適化するには、GC_FILES_TO_LOCKS パラメータを正確に設定する必要があります。データ・ブロック・アドレス・ロックは、デフォルト動作では、各ブロックに1つのロックを割り当てます。たとえば、全表スキャンの実行中に、読み込まれた各ブロックに対してロックが取得される必要があります。全表スキャンを高速化するには、次のいずれかの操作を行います。

- データ・ファイルに読取り専用データのみが含まれている場合は、表領域を読取り専用を設定します。すると、ロック・オペレーションが発生しなくなります。
- または、データ・ファイルに含まれているデータのほとんどが読取り専用である場合は、ごく少数のハッシュ・ロック (2つの共有ロックなど) を割り当てます。すると、データを読み込むときに取得する必要があるのは、これらのロックのみになります。
- データ・ブロック・アドレス・ロックまたはファイングレイン・ロックが必要な場合、!オプションを使用して、ロックごとに制御されるブロックをグループ化します。これは、デフォルトのデータ・ブロック・アドレス・ロックよりメリットがあります。デフォルトでは、100万個のブロックを読み込むためには100万個のロックを取得する必要があります。ブロックをグループ化すると、グループ係数によって、割り当てられたロックの数を削減できます。したがって、!10のグループ化で取得する必要があるロック数は、デフォルトで取得する必要があるロック数の10分の1で済みます。ロック割当て数が大幅に削減されるため、パフォーマンスが向上します。これまでの実績では、!10のグループ化でのパフォーマンスは、ハッシュ・ロックでの速度と同等です。

次のガイドラインは、メモリー使用量に影響し、間接的にパフォーマンスにも影響します。

- 一時表領域のデータ・ファイルには、ロックを割り当てないでください。
- SYSTEM 表領域には、固有のロックを割り当てます。これによって、領域管理などのデータ・ディクショナリ・アクティビティが、キャッシュ管理レベルでデータ表領域と干渉 (エラー 01575) しないことが保証されます。

Real Application Clusters の I/O 統計の分析

GC_FILES_TO_LOCKS を設定した場合、キャッシュ・フュージョンは使用禁止になります。この場合は、V\$SYSSTAT ビューの 3 つの統計を使用して、グローバル・キャッシュの同期化に関連する次の I/O のパフォーマンスを計測できます。

- DBWR cross-instance writes
- リモート・インスタンスの UNDO ヘッダー書込み
- リモート・インスタンスの UNDO ブロック書込み

DBWR cross-instance writes は、要求側ノードが要求したブロックを使用する前に、Oracle がブロックをディスクに書き込んで、インスタンス間のデータ・ブロックの使用を解決したときに発生します。

キャッシュ・フュージョンは、ブロックの現行のバージョンおよび読み一貫性バージョンに対するディスク I/O を排除します。これによって、各インスタンスが実行する **physical writes** および物理読み込みが大幅に削減されます。

V\$SYSSTAT を使用した Real Application Clusters の I/O 統計の分析

次の統計を取得して、グローバル・キャッシュの同期化に必要な書込み I/O の量を明確にできます。

1. 次の構文を使用して、V\$SYSSTAT ビューを問い合わせます。

```
SELECT NAME, VALUE FROM V$SYSSTAT
WHERE NAME IN ('DBWR cross-instance writes',
              'remote instance undo block writes',
              'remote instance undo header writes',
              'physical writes');
```

次のよう出力されます。

NAME	VALUE
physical writes	41802
DBWR cross-instance writes	5403
remote instance undo block writes	0
remote instance undo header writes	2
4 rows selected.	

physical writes 統計は、特定のインスタンスから発生した DBWR が実行するすべての物理書込みを示します。また、**DBWR cross-instance writes** の値は、別のインスタンスが変更を行うために要求したデータ・ブロックを含む使用済みバッファを書き込むことによって発生する、すべての書込みを示します。DBWR プロセスは、クロス・インスタンス書込みも処理するため、DBWR cross-instance writes はすべての **physical writes** のサブセットになります。

2. 次の式で、物理 I/O 全体に対する Real Application Clusters 関連 I/O の割合を計算します。

$$\frac{DBWR \text{ cross-instance writes}}{\text{physical writes}}$$

3. 次の式で、リモート・インスタンスが、ローカル・インスタンスによって使用されているロールバック・セグメントから読み込む必要がある場合に、ロールバック・セグメントへの書込みが発生した回数を計算します。

$$\frac{(\text{remote instance undo header writes} + \text{remote instance undo block writes})}{DBWR \text{ cross-instance writes}}$$

この割合は、ロールバック・セグメントへの書込みに関連するディスク I/O の量を示します。

4. グローバル・キャッシュ同期化による読み込みの数または割合を見積もるには、NULL (N) から共有モード (S) への変換に対するロック要求数を使用します。この要求数は、V\$LOCK_ACTIVITY、または V\$SYSSTAT の **physical reads** 統計にカウントされません。

次の式では、**lock buffers for read** が N-to-S ブロック・アクセス・モード変換を示すローカル作業のみに関する読み込みの割合が計算されます。

$$\frac{(\text{physical reads} - (\text{lock buffers for read})) \times 100}{\text{physical reads}}$$

これらは強制読み込みと呼ばれ、ローカル・インスタンスによって変更されたキャッシュ・データ・ブロックが、別のインスタンスからの要求によってディスクに書き込まれる必要があり、その後ローカル・インスタンスが、読み込みのためにブロックを再取得する場合に発生します。

不適切な強制書込みの検出によるマルチブロック・ロック使用量の監視

不適切な強制書込みは、異なるノードからブロックが同時に更新される場合に、2つ以上のブロックを保護するロックが下位変換されるときに発生します。たとえば、2つのノードがそれぞれ異なるブロックを更新するとき、それらのブロックが同じロックによって保護されているとします。この場合、各ノードは1つのブロックのみを更新するために、2つのブロックをディスクに書き込む必要があります。これは、同じロックが両方のブロックを処理するためです。

不適切な強制書込みアクティビティを表示できる統計はありません。不適切な強制書込みの検出は、この項で説明する周囲の状況から判断するしかありません。

次の SQL 文は、書込みの原因となるロック・オペレーションの数と、実際に書き込まれたブロックの数を表示します。

```
SELECT VALUE/ (A.COUNTER + B.COUNTER + C.COUNTER) "PING RATE"
FROM V$SYSSTAT,
     V$LOCK_ACTIVITY A,
     V$LOCK_ACTIVITY B,
     V$LOCK_ACTIVITY C
WHERE A.FROM_VAL = 'X'
      AND A.TO_VAL = 'NULL'
      AND B.FROM_VAL = 'X'
      AND B.TO_VAL = 'S'
      AND C.FROM_VAL = 'X'
      AND C.TO_VAL = 'SSX'
      AND NAME = 'DBWR cross-instance writes';
```

表 A-3 に、強制書込み率の解析方法を示します。

表 A-3 強制書込み率の解析

ディスク書込みの強制実行の割合	意味
< 1	不適切な強制書込みが発生している場合がありますが、ディスク書込みの強制実行を上回るロック・オペレーションがあります。DBWR が十分な速さでブロックを書き込んでいるため、ロック・アクティビティに対する書込みは発生していません。これは、soft ping ともいいます。I/O アクティビティがディスク書込みの強制実行に対してではなく、ロック・アクティビティにのみ要求されていることを表します。
=1	書込みが発生する可能性がある各ロック・アクティビティで、実際に書込みが発生します。不適切な強制書込みが発生している場合があります。
> 1	不適切な強制書込みが確実に発生しています。

次の式を使用して、不適切な強制書込みの確率を計算します。

$$\frac{(\text{ping_rate} - 1)}{\text{ping_rate}} \times 100$$

書込みの合計数をチェックし、不適切な強制書込みによる書込み数を計算します。

```
SELECT Y.VALUE "ALL WRITES",
       Z.VALUE "PING WRITES",
       Z.VALUE * pingrate "FALSE PINGS",
FROM V$SYSSTAT Z,
     V$SYSSTAT Y,
WHERE Z.NAME = 'DBWR cross-instance writes'
AND Y.NAME = 'physical writes';
```

次の SQL 文によって、`ping_rate` が導出されます。

```
CREATE OR REPLACE VIEW PING_RATE AS
SELECT ((VALUE/(A.COUNTER+B.COUNTER+C.COUNTER))-1)/
       (VALUE/(A.COUNTER+B.COUNTER+C.COUNTER)) RATE
FROM V$SYSSTAT,
     V$LOCK_ACTIVITY A,
     V$LOCK_ACTIVITY B,
     V$LOCK_ACTIVITY C
WHERE A.FROM_VAL = 'X'
      AND A.TO_VAL = 'NULL'
      AND B.FROM_VAL = 'X'
      AND B.TO_VAL = 'S'
      AND C.FROM_VAL = 'X'
      AND C.TO_VAL = 'SSX'
      AND NAME = 'DBWR cross-instance writes';
```

目標は、全体のディスク書込みの強制実行のみでなく、不適切な強制書込みを削減することです。このためには、`GC_FILES_TO_LOCKS` 内のインスタンス・ロックの配分を確認して、ファイル内のデータをチェックします。

ロック名の書式

次の項では、ロックの名前と書式について説明します。この項の内容は次のとおりです。

- [ロック名の書式](#)
- [ロックの名前](#)
- [ロックのタイプと名前](#)

ロック名の書式

内部的に使用されるグローバル・ロック名の書式は、次のいずれかです。[表 A-4](#) で各パラメータを説明します。

- type ID1 ID2
- type, ID1, ID2
- type (ID1, ID2)

表 A-4 ロック名の書式で使用する変数

変数	意味
type	ロック・タイプを表す 2 文字のタイプ名です。BL、TX、TM などがあります。
ID1	1 番目のロック識別子です。この識別子の意味および書式は、ロック・タイプごとに異なります。
ID2	2 番目のロック識別子です。この識別子の意味および書式は、ロック・タイプごとに異なります。

たとえば、領域管理ロックの名前は ST00 のようになります。ロックの名前は、BL 1900 のようになります。

ロック・マネージャのクライアントは、ロック・タイプを定義します。たとえば、ロックに対して、BL および 2 つのパラメータ (ID1 と ID2) を定義し、これら 2 つのパラメータを GCS API に渡してロックをオープンします。ロック・マネージャは、異なるタイプのロックを識別しません。Oracle の各コンポーネントは、必要に応じてタイプおよび 2 つのパラメータを定義するので、ID1 および ID2 は、各コンポーネントの要件と一貫性があるといえます。

ロックの名前

すべてのロックは、バッファ・キャッシュ管理ロックです。バッファ・キャッシュ管理ロックのタイプは BL になります。ロックの名前の構文は、type ID1 ID2 です。各項目の意味は次のとおりです。

- type –ロックはバッファ・ロックであるため、常に BL です。
- ID1 –ブロックのデータベース・アドレスです。
- ID2 –ブロック・クラスです。

ロックの名前の例は、次のとおりです。

- BL (100, 1) –ロック要素 100 を持つデータ・ブロックです。
- BL (1000, 4) –ロック要素 1000 を持つセグメント・ヘッダー・ブロックです。
- BL (27, 1) –ロールバック・セグメント 10 を持つロールバック・セグメント・ヘッダーです。ロールバック・セグメントの計算式は、 $7+(10 \times 2)$ です。

ロックのタイプと名前

ロックには、タイプと名前が異なるものがあります。表 A-5 に、タイプと名前のリストを示します。

表 A-5 ロックのタイプと名前

タイプ	ロック名	タイプ	ロック名
CF	制御ファイル・トランザクション	PS	パラレル実行プロセスの同期化
CI	インスタンス間コール起動	RT	REDO スレッド
DF	データ・ファイル	SC	システム変更番号
DL	ダイレクト・ローダー索引作成	SM	SMON
DM	データベース・マウント	SN	順序番号
DX	分散リカバリ	SQ	順序番号エンキュー
FS	ファイル・セット	SV	順序番号値
KK	REDO ログ「Kick」	ST	領域管理トランザクション
IN	インスタンス番号	TA	トランザクション・リカバリ
IR	インスタンス・リカバリ	TM	DML エンキュー
IS	インスタンス状態	TS	一時セグメント (および表領域)
MM	マウント定義	TT	一時表

表 A-5 ロックのタイプと名前 (続き)

タイプ	ロック名	タイプ	ロック名
MR	メディア・リカバリ	TX	トランザクション
IV	ライブラリ・キャッシュ無効化	UL	ユーザー定義ロック
L[A-P]	ライブラリ・キャッシュ・ロック	UN	ユーザー名
N[A-Z]	ライブラリ・キャッシュ・ピン	WL	REDO ログ書込み開始
Q[A-Z]	行キャッシュ	XA	インスタンス登録属性ロック
PF	パスワード・ファイル	XI	インスタンス登録ロック
PR	プロセス起動		

Real Application Clusters での空きリストと 空きリスト・グループの使用（オプション）

オラクル社がお薦めする自動セグメント領域管理を使用できない場合は、この付録で説明する、**Oracle Real Application Clusters** 環境での空きリストと空きリスト・グループの使用手順を参照してください。内容は次のとおりです。

- [空きリスト・グループを使用した複数ノードからの同時挿入](#)
- [エクステンツ管理](#)

空きリスト・グループを使用した複数ノードからの同時挿入

ローカル管理表領域および自動セグメント領域管理を使用できない場合は、空きリストと空きリスト・グループを使用して手動で空き領域を管理することを検討してください。ただし、オラクル社は、自動セグメント領域管理の使用をお勧めします。

空きリストと空きリスト・グループの目的

自動セグメント領域管理がない状態で、複数ノードから表にデータが頻繁に挿入され、その表がパーティション化されていない場合は、空きリスト・グループを使用してパフォーマンスの問題を回避できます。この場合、パフォーマンスの問題の原因は、データ・ブロック、表セグメント・ヘッダーおよびその他のグローバル・リソース要求への並行アクセスであると考えられます。

空きリスト・グループは、表の空き領域管理に関連するデータ構造を、個々のインスタンスが使用できるように分割された集合に分離します。空きリスト・グループを使用すると、異なるインスタンスで作業するプロセス間の、パフォーマンスの問題を軽減できます。これは、挿入を行うために十分な空き領域があるデータ・ブロックが、インスタンスごとに管理されるためです。

関連項目： 空きリスト・グループの概要については、『Oracle9i Real Application Clusters 概要』を参照してください。

空きリスト・グループを使用したデータベース・オブジェクトの作成

複数の**インスタンス**が同じ表に頻繁にランダム挿入を行う場合は、通常、空きリストと空きリスト・グループが必要です。プロセスがデータ・ブロックの領域を検索するため、同じブロックおよび表ヘッダーに対して競合が発生する場合があります。また、並行度、およびあるインスタンスから別のインスタンスにデータとヘッダー・ブロックを送信するオーバーヘッドによって、パフォーマンスが低下する可能性があります。このような場合は、空きリスト・グループを使用してください。

重要な表の識別

挿入率が高い表は、次の例に示すように、V\$SQL ビューを問い合わせして INSERT コマンドを検索することによって識別できます。

```
SELECT SUBSTR(SQL_TEXT,80), DECODE(COMMAND_TYPE,2,'INSERT'), EXECUTIONS
FROM V$SQL
WHERE COMMAND_TYPE = 2
ORDER BY EXECUTIONS;
```

最も実行頻度の高い文の文字列にある表の名前を検索します。実行頻度の高い文、および検索された表に対して作成されている索引が、空きリスト・グループの候補になります。

FREELIST GROUPS 再編成の必要性の判断

空きリスト・グループのパフォーマンスを監視するには、V\$CLASS_CACHE_TRANSFER ビューを使用してキャッシュ転送および**ディスク書込みの強制実行**の割合を調査します。V\$CLASS_CACHE_TRANSFER ビューには、ブロックの各クラスに対してインスタンスが起動した後のキャッシュ転送の回数が示されます。次の SELECT 文で、セグメント・ヘッダーおよび空きリストのディスク書込みの強制実行に対して、比較的高い値（全体の 5% を超える値など）が結果として出力される場合は、パフォーマンスを改善するために、いくつかの表の FREELIST GROUPS パラメータを変更することを検討してください。

```
SELECT CLASS, (X_2_NULL_FORCED_STALE + X_2_S_FORCED_STALE) CACHE_TRANSFER
FROM V$CLASS_CACHE_TRANSFER;
```

V\$CLASS_CACHE_TRANSFER ビューはキャッシュ転送をオブジェクト名ごとに識別しないため、キャッシュ転送の数に大きく影響するオブジェクトを識別するには、他のビューを使用します。たとえば、V\$CACHE_TRANSFER ビューには、転送されたバッファ・キャッシュの各ブロックに関する情報が示されます。ブロック・クラス 4 はセグメント・ヘッダーを識別し、ブロック・クラス 6 は空きリスト・ブロックを識別します。次の SELECT 文を実行すると、空きリスト・グループの値を増やすメリットがあるオブジェクトが示されます。

```
SELECT NAME, CLASS#, SUM(XNC) CACHE_TRANSFER
FROM V$CACHE_TRANSFER
WHERE CLASS# IN (4,6)
GROUP BY NAME, CLASS#
ORDER BY CACHE_TRANSFER DESC;
```

注意： Database Configuration Assistant を使用してデータベースを作成していない場合、V\$CLASS_CACHE_TRANSFER などの Real Application Clusters 固有のビューを使用できるのは、CATCLUST.SQL スクリプトを実行した後のみです。

FREELISTS および FREELIST GROUPS を使用した表、クラスタおよび索引の作成

空きリストおよび空きリスト・グループを作成するには、CREATE TABLE 文、CREATE CLUSTER 文または CREATE INDEX 文に FREELISTS 記憶域パラメータおよび FREELIST GROUPS 記憶域パラメータを指定します。データベースは排他モードまたは共有モードのいずれでもオープンできます。空きリスト・グループを使用する必要がある場合、原則として、各 Real Application Clusters インスタンスに、空きリスト・グループを1つ以上作成します。

注意： ALTER TABLE 文、ALTER CLUSTER 文または ALTER INDEX 文で FREELIST GROUPS の値を変更するには、表をエクスポート、削除、再構築および再ロードする必要があります。ただし、FREELISTS の設定は、ALTER TABLE 文、ALTER INDEX 文または ALTER CLUSTER 文で動的に変更できます。

FREELISTS パラメータ

FREELISTS パラメータには、空きリスト・グループごとの空きリスト数を指定します。FREELISTS のデフォルトおよび最小値は1です。最大値は、データ・ブロックのサイズによって異なります。指定した値が大きすぎると、エラー・メッセージが表示され、最大値が示されます。FREELISTS の最適値は、特定の表の空きリスト・グループ1つ당りに予想される同時挿入の数によって異なります。

注意： オブジェクトが常駐する表領域が自動セグメント領域管理モードの場合、FREELISTS の設定は無視されます。

FREELIST GROUPS パラメータ

各空きリスト・グループは、起動時に1つ以上のインスタンスに対応付けられます。FREELIST GROUPS のデフォルト値は1です。これは、セグメントの既存の空きリストすべてをすべてのインスタンスが使用できることを意味します。前述のとおり、通常、FREELIST GROUPS には、使用する Real Application Clusters 環境内のインスタンスの数を設定します。

挿入および更新のために十分な空き領域がある空きリスト・グループ・ブロックは、一度特定の空きリスト・グループに割り当てられると効率的に分割されます。ただし、1つのインスタンスに割り当てられたデータ・ブロックは、別のインスタンスによって割り当てを解除されると、元のインスタンスでは使用できなくなります。これによって、領域の一部が使用不可能になり、偏りが発生する場合があります。

注意： 複数の空きリスト・グループを使用すると、空きリスト構造はセグメント・ヘッダーから切り離され、空きリスト・ブロック（個別のブロック）に置かれます。これによってセグメント・ヘッダーのパフォーマンスの問題が軽減され、インスタンスに個別の空きブロック・リストが提供されます。

例： 次の文では、それぞれ4つの空きリストを含む7つの空きリスト・グループを持つ、department という名前の表が作成されます。

```
CREATE TABLE department
  (deptno  NUMBER(2),
   dname   VARCHAR2(14),
   loc     VARCHAR2(13) )
STORAGE ( INITIAL 100K          NEXT 50K
          MAXEXTENTS 10        PCTINCREASE 5
          FREELIST GROUPS 7    FREELISTS 4 );
```

クラスタ化表の FREELISTS および FREELIST GROUPS の作成

レコードが、グループとして1つ以上の SELECT 文によって頻繁にアクセスされる場合、クラスタ化表を使用して、別の表からレコードを格納します。クラスタ化表を使用すると、読み込み処理のオーバーヘッドを減少させてパフォーマンスを改善できます。ただし、クラスタ化表は DML 文に対してあまり有効でない場合もあります。

クラスタ化表の場合は、CREATE TABLE 文に FREELISTS 記憶域パラメータおよび FREELIST GROUPS 記憶域パラメータを指定できません。空きリスト・パラメータは、個々の表についてではなく、クラスタ全体について指定する必要があります。これは、クラスタ化表では、CREATE CLUSTER 文の記憶域パラメータが使用されるためです。

自動セグメント領域管理を使用しない場合、Real Application Clusters では、インスタンスで複数の空きリストと空きリスト・グループを使用できます。ハッシュ・クラスタの中には、ハッシュ関数についてユーザー定義のキーを使用して作成され、そのキーがインスタンス別にパーティション化されている場合に、複数の空きリストおよび空きリスト・グループを使用できるものもあります。

注意： TRUNCATE TABLE *table_name* REUSE STORAGE 構文を使用すると、空きリスト・グループに対するエクステントのマッピングが削除され、最高水位標が最初のエクステントの開始点に再設定されます。

索引の FREELISTS の作成

CREATE INDEX 文に FREELISTS パラメータおよび FREELIST GROUPS パラメータを使用することもできます。ただし、索引への挿入と表への挿入は異なることに注意してください。これは、使用されるブロックが、索引キー値によって判断されるためです。

たとえば、複数の空きリスト・グループのある表があり、その索引にも複数の空きリスト・グループがあると想定します。2つのセッションが異なるインスタンスに接続し、その表に行を挿入する場合、Oracle は異なるブロックを使用して表データを格納します。これによって、影響を受けるデータ・セグメントに対するキャッシュ・ブロック転送が最少になります。ただし、これらのセッションが同様の索引キー値を挿入する場合、索引セグメントのキャッシュ・ブロック転送が行われる可能性があります。この場合、Oracle は、索引の空きリストを格納するためにさらに多くの索引ブロックを使用する必要があるため、索引セグメント・ヘッダーのキャッシュ転送はほとんど削減されません。

関連項目： SQL の詳細は、『Oracle9i SQL リファレンス』を参照してください。

インスタンスおよびユーザー・セッションと空きリスト・グループの対応付け

Oracle が複数の空きリスト・グループを使用してオブジェクトを作成する場合、空きリスト・グループ・ブロック数は、オブジェクトのデータ・ディクショナリの定義の一部になります。インスタンスおよびユーザーは、空きリスト・グループ・ブロックと対応付ける必要があります。初期化パラメータを使用して固定**インスタンス番号**をインスタンスに割り当てるか、DDL 文にインスタンス番号を指定することによって、静的にこの対応付けを行うことができます。

インスタンスと空きリスト・グループの対応付け

次のようにして、インスタンスと空きリスト・グループを対応付けます。

- INSTANCE_NUMBER パラメーター INSTANCE_NUMBER 初期化パラメータと同時に様々な SQL 句を使用して、データ・ブロックのエクステンツとインスタンスを対応付けることができます。
- SET INSTANCE 句—ALTER SESSION 文の SET INSTANCE 句を使用すると、セッションが接続しているインスタンスに関係なく、そのセッションが特定のインスタンスに対応付けられている空きリスト・グループを使用できます。次に例を示します。

```
ALTER SESSION SET INSTANCE = inst_no
```

SET INSTANCE 句は、1つのインスタンスに障害が発生し、他のインスタンスに接続する場合に有効です。たとえば、表の空きリスト・グループに領域が事前に割り当てられているデータベースを考えてみます。1つのインスタンスに障害が発生し、すべてのユーザーが他のインスタンスにフェイルオーバーされた場合、それらのセッションが、障害が発生したインスタンスに対応付けられた空きリスト・グループを使用するように設定できます。

SET INSTANCE 句を省略すると、フェイルオーバーされたセッションではブロックへのデータ挿入が開始され、エクステントはフェイルオーバー先のインスタンスに割り当てられます。その後、障害インスタンスがリストアされ、ユーザーがそのインスタンスに再接続すると、挿入されたデータが、他のインスタンスの空きリスト・グループと対応付けられた一連のブロックに組み込まれます。このため、インスタンス間の通信が増加する場合があります。

エクステント管理

内容は次のとおりです。

- [エクステントの事前割当て](#)
- [エクステント管理およびローカル管理表領域](#)

エクステントの事前割当て

Oracle が表に行を挿入する前は、表には、その表に割り当てられた多数の空きブロックを持つ初期エクステントのみがあります。エクステントがない場合は、表は空です。したがって、空きリスト・グループに表用の領域を事前に割り当てることを考慮する必要があります。これによって、空きブロックを持つエクステントを空きリスト・グループへ最適に割り当てることができます。この結果、インスタンスへの割当ても最適化されます。また、事前割当てによって、エクステント割当てのオーバーヘッドも回避できます。

事前割当てのメリットは、事前に物理記憶域のレイアウトを判断できることです。さらに、エクステント割当て方法によって、新規エクステントの割当て元の物理ファイルまたはボリュームを選択できます。ただし、次に示すような事前割当てを行う場合は、ALLOCATE EXTENT 句の実装とその方法、およびいくつかの Oracle 初期化パラメータの使用方法について考慮する必要があります。

- [ALLOCATE EXTENT 句を使用したエクステントの事前割当て](#)
- [MAXEXTENTS、MINEXTENTS および INITIAL を使用したエクステントの事前割当て](#)

ALLOCATE EXTENT 句を使用したエクステントの事前割当て

ALTER TABLE 文または ALTER CLUSTER 文の ALLOCATE EXTENT 句では、エクステントのサイズ、データ・ファイルおよびオブジェクトに対応付ける空きリスト・グループを指定するパラメータを使用して、エクステントを表、索引またはクラスタに事前に割り当てることができます。

排他モードと共有モードおよび ALLOCATE EXTENT 句 データベースを排他モードで実行している間は、共有モードの場合と同様に、ALTER TABLE (または CLUSTER) ALLOCATE EXTENT 文を使用できます。インスタンスは、排他モードで実行している間でも、領域の検索については同じ規則に従います。トランザクションは、このインスタンスに対して、マスター空きリストまたは特定の空きリスト・グループを使用できます。

SIZE パラメータおよび ALLOCATE EXTENT 句 ALLOCATE EXTENT 句の SIZE パラメータには、エクステント・サイズをブロック・サイズの倍数に切り上げて、バイト単位で指定します。SIZE を指定しない場合、エクステント・サイズは、NEXT 記憶域パラメータおよび PCTINCREASE 記憶域パラメータの値に従って計算されます。

SIZE 値は、後続のエクステント割当ての計算には使用されません。後続のエクステント割当ては、NEXT パラメータおよび PCTINCREASE パラメータに設定された値に従って計算されます。

DATAFILE パラメータおよび ALLOCATE EXTENT 句 このパラメータは、エクステント用の領域を確保するためのデータ・ファイルを指定します。このパラメータを省略すると、該当する表を含む表領域にあるすべてのアクセス可能なデータ・ファイルから、領域が割り当てられます。

ファイル名は、大 / 小文字の区別も含めて、**制御ファイル**に格納されている文字列と正確に一致している必要があります。この文字列は、DBA_DATA_FILES データ・ディクショナリ・ビューの FILE_NAME 列を問い合せて確認できます。

INSTANCE パラメータおよび ALLOCATE EXTENT 句 このパラメータは、**インスタンス番号**の整数に対応付けられた空きリスト・グループに新しい領域を割り当てます。各インスタンスは、起動時に、そのインスタンスを空きリスト・グループにマップする一意のインスタンス番号を取得します。最小インスタンス番号は、0 (ゼロ) ではなく 1 です。最大値は、オペレーティング・システム固有です。構文は次のとおりです。

```
ALTER TABLE tablename ALLOCATE EXTENT (... INSTANCE n )
```

ここで、*n* は同じ番号の空きリスト・グループにマップされます。インスタンス番号が空きリスト・グループの数より大きい場合は、次のようにハッシュされ、割り当てられる空きリスト・グループが判断されます。

$$\text{modulo}(n, \#_freelistgroups) + 1$$

INSTANCE パラメータを指定しないと、新しい領域は表に割り当てられますが、空きリスト・グループには割り当てられません。このような領域は、他の領域が使用できない場合に、必要に応じて空きブロックのマスター空きリストに含められます。

注意： INSTANCE の値には、実際のインスタンス番号ではなく、使用する空きリスト・グループの番号に対応した値を使用してください。

関連項目： INSTANCE パラメータの詳細は、『Oracle9i Real Application Clusters 管理』を参照してください。

MAXEXTENTS、MINEXTENTS および INITIAL を使用したエクステントの事前割当て

エクステントを特定のインスタンスに対応付けられている空きリスト・グループに事前に割り当て、MAXEXTENTS をエクステントの現在の数（事前割当てされたエクステントに MINEXTENTS を加えた数）に設定することによって、自動割当てを防止できます。MINEXTENTS を 1（デフォルト）に設定し、INITIAL をその最小値（2つのデータ・ブロック、またはブロック・サイズが 2048 バイトの場合は 10KB）に設定して表またはクラスタを作成すると、初期割当てを最小化できます。また、データ・ブロックに対するインスタンス間のパフォーマンスの問題を最小限に抑えるには、各表に複数のデータ・ファイルを作成し、各インスタンスを異なるファイルに対応付けます。

システムのノード数を増やす場合は、追加インスタンスを考慮して、現行のインスタンス数より多くの空きリスト・グループを持つ表またはクラスタを作成します。これらの空きリスト・グループには、必要になるまで領域を割り当てる必要はありません。この場合、自動的に割り当てられる領域を持っているのは、空きブロックのマスター空きリストのみです。

データ・ブロックを空きリスト・グループに対応付けるには、データ・ブロックの使用量を、PCTUSED に設定した値より、その空きリスト・グループを使用するインスタンスで実行中のプロセス分だけ低く設定するか、またはデータ・ブロックをその空きリスト・グループに特別に割り当てる必要があります。したがって、使用されていない空きリスト・グループは、未使用の空きデータ・ブロックから切り離されることはありません。

エクステント管理およびローカル管理表領域

エクステントの割当ておよび割当て解除は、高コストの操作であるため最小限に抑える必要があります。Real Application Clusters では、これらの操作のほとんどでインスタンス間の調整が必要です。さらに、頻繁にエクステント管理操作を行うと、シングル・インスタンス環境より Real Application Clusters 環境の方が、パフォーマンスが低下する場合があります。この問題は、特にディクショナリ管理の表領域に発生します。

エクステント管理問題の識別

「row cache lock」イベントが V\$SYSTEM_EVENT の非アイドル待機時間の主な原因である場合は、データ・ディクショナリ・キャッシュにパフォーマンスの問題が発生しています。エクステント割当て操作および割当て解除操作がこの原因である場合があります。

V\$ROWCACHE は、DC_USED_EXTENTS および DC_FREE_EXTENTS のデータ・ディクショナリ・キャッシュ情報を提供します。この問題は、特に、これらのパラメータに対する DLM_CONFLICTS の値が時間とともに大幅に増加する場合に発生します。これは、エクステント管理アクティビティが過剰に発生していることを示します。

エクステント管理操作の最小化

表、索引、一時セグメントおよびロールバック・セグメントの記憶域パラメータを適切に構成することによって、エクステント割当ておよび割当て解除の頻度を減らすことができま

す。これを行うには、INITIAL パラメータ、NEXT パラメータ、PCTINCREASE パラメータ、MINEXTENTS パラメータおよび OPTIMAL パラメータを使用します。

ローカル管理表領域の使用

ローカル管理表領域を使用すると、エクステントの割当ておよび割当て解除によるオーバーヘッドを大幅に減らすことができます。パフォーマンスを最適化し、領域を最適に使用するには、ローカル管理表領域にあるセグメントの領域割当て特性が、すべて同じであることが理想的です。この場合、セグメント用に計算された理想的なエクステント・サイズの増分に対応する適切な均一エクステント・サイズを使用して表領域を作成できます。

たとえば、均一エクステント・サイズが 10MB である表領域には、かなり頻繁に挿入が行われる表を格納できます。一方、均一エクステント・サイズが 100KB である表領域には、DML アクティビティが限られている小規模な表を置くことができます。既存のシステムの表領域がセグメント・サイズに基づいて構成されていない場合、このタイプの構成を行うにはかなりの再構成の工数が必要になり、効果も限られています。そのため、UNIFORM エクステント割当てのかわりに AUTOALLOCATE を使用して、大部分の表をローカル管理するように作成します。

関連項目： CREATE TABLESPACE 文の AUTOALLOCATE 句と UNIFORM 句の詳細は、『Oracle9i SQL リファレンス』を参照してください。

用語集

buffer busy global cache

別のプロセスがブロックのリソースを取得しているため、そのブロックが使用可能になるまでプロセスが待機する必要がある場合に通知される待機イベント。

buffer busy waits

その時点で別のプロセスがバッファを使用しているため、プロセスがそのバッファを取得できない場合に通知される待機イベント。

cache convert waits

[global cache null to S](#)、[global cache null to X](#)、[global cache S to X](#) などの、すべての上位変換操作に対する合計待機数。

cache open waits

[global cache open S](#) および [global cache open X](#) に対する合計待機数。

Cluster Manager (CM)

オペレーティング・システム固有のコンポーネント。クラスタ間にメンバーシップの共通ビューを提供して、各ノードのメンバーシップ状態を検出および追跡する。CM はプロセスの状態（特にデータベース・インスタンスの状態）も監視する。[グローバル・キャッシュ・サービス](#)の状態を監視するバックグラウンド・プロセスである[グローバル・エンキュー・サービス・モニター](#)は、CM への登録と CM からの登録解除を行う。

consistent gets

読みみ一貫性（CR）モードで取得されるバッファの数。

cr request retry

Oracle が、保持側インスタンスを使用できなくなったことを検出したときに、読みみ一貫性要求を再発行するたびに発生する待機。

db block changes

DML 用に排他モードで取得されたカレント・バッファの数。

db block gets

読み込み用に取得されたカレント・バッファの数。

DBWR cross-instance writes

強制書込みともいう。以前は排他的に保持されていたブロックを、他のインスタンスがバッファ・キャッシュに読み込むことができるようにするために、インスタンスがディスクに対して実行する必要がある書込みの数。実際には、GC_FILES_TO_LOCKS パラメータを 1 以上の値に指定しないかぎり、キャッシュ・フュージョンによって DBWR cross-instance writes は発生しない。

DFS lock handle

グローバル・リソースへのポインタ。GES リソースを操作するには、プロセスは、まず DFS handle を取得する必要がある。

flow control messages received

LMD プロセスが受信するフロー制御 (nullreq および nullack) メッセージの数。

flow control messages sent

LMS プロセスが送信するフロー制御 (nullreq および nullack) メッセージの数。

global cache bg acks

LMS プロセスがその操作を完了したときの、インスタンスの起動または停止中のみ発生する可能性がある、待機イベント。

global cache busy

セッションが、リリース上で進行している処理の完了を待機する必要がある場合に必ず発生する、待機イベント。

global cache convert timeouts

リソース操作のタイムアウト数。

global cache convert time

すべてのセッションが、GCS リソースに対してグローバル変換を実行するために必要な、累計時間。

global cache converts

バッファ・キャッシュ・ブロックのリソース変換。この統計は、GCS リソースが NULL から排他、共有から排他、または NULL から共有に変換されるたびに増加する。

global cache cr block build time

LMS プロセスが保持側インスタンス上で CR ブロックを作成するために必要な時間。

global cache cr block flush time

CR 要求が行われたときのログ・フラッシュの待機時間。一度 LGWR がログ・フラッシュ・キューにあるバッファへの変更のフラッシュを完了すると、LMS は、これを送信できる。サービス時間の一部。

global cache cr block receive time

読み込み一貫性要求の完了にかかった合計時間（CR ブロックに対するすべての要求のラウンドトリップ時間の累計）。

global cache cr block send time

LMS が CR ブロックを送信するために必要な時間。各要求に対して、タイミングはブロックが送信されたときに開始し、送信が完了したときに停止する。この統計では、送信するために必要な時間のみが測定される。ブロックが要求側インスタンスに到着するまでの経過時間は測定されない。

global cache cr blocks received

あるプロセスが、ローカル・キャッシュにはないデータ・ブロックの読み込み一貫性を要求する場合に、プロセスが要求を別のインスタンスへ送信する回数。一度要求が完了する（バッファが受信される）と、Oracle はこの統計を増加させる。

global cache cr blocks served

LMS が行った CR ブロックに対する要求の数。Oracle は、ブロックが送信されたときにこの統計を増加させる。

global cache cr cancel wait

取り消された CR 要求に対する獲得割り込みが完了するまで、セッションが待機する場合に発生する待機イベント。要求の取消しは、キャッシュ・フュージョン書込みプロトコルの一部である。

global cache cr request

ペンディング CR 要求が完了するまで、プロセスが待機する必要がある場合に、必ず発生する待機イベント。プロセスは、ブロックをディスクからキャッシュに読み込む前に、ブロックへの共有アクセス権が付与されるまで待機するか、または保持側インスタンスの LMS によってブロックが送信されるまで待機する。

global cache current block flush time

ブロックが要求側インスタンスに送信される前に、ブロックへの変更をディスクにフラッシュ（強制ログ・フラッシュともいう）する時間。

global cache current block pin time

カレント・ブロックを要求側インスタンスに送信する前に、ブロックを確保する時間。ブロックの確保は、別のインスタンスへのブロックの送信準備中に、ブロックへのさらなる変更を許可しないために必要である。

global cache current block receive time

カレント・ブロックに対するすべての要求の累計ラウンドトリップ時間。

global cache current block send time

カレント・ブロックを **インターコネクト** 経由で要求側インスタンスに送信する時間。

global cache current blocks received

保持側インスタンスから **インターコネクト** 経由で受信したカレント・ブロックの数。

global cache current blocks served

インターコネクト 経由で要求側インスタンスに送信されたカレント・ブロックの数。

global cache freelist waits

Oracle がリソース要素の空きリストが空であることを検出した回数。

global cache freelist wait

Oracle がローカル要素の空きリストが空であることを検出した後で待機する必要がある場合に発生する待機イベント。

global cache get time

ローカル・バッファ用の GCS リソースをオープンするために必要なすべてのセッションの累計時間。

global cache gets

GCS によって新しいリソースがオープンされたバッファ取得の数。

global cache null to S

任意のリソース変換が完了するまで、セッションが待機する必要がある場合に、必ず発生する待機イベント。

global cache null to X

このリソース変換が完了するまで、セッションが待機する必要がある場合に、必ず発生する待機イベント。

global cache open S

要求されたリソースへの共有アクセス権を受信するまで、セッションが待機する必要がある場合に発生する待機イベント。

global cache open X

要求されたリソースへの排他アクセス権を受信するまで、セッションが待機する必要がある場合に発生する待機イベント。

global cache pending ast

Oracle がリソース要素をクローズする前に、プロセスが獲得割込みを待機する場合に発生する可能性がある待機イベント。

global cache pred cancel wait

取り消された前回の読み込み要求に対する獲得割込みが完了するまで、セッションが待機する必要がある場合に発生する待機イベント。前回の読み込み要求の取消しは、キャッシュ・フュージョン書き込みプロトコルの一部である。

global cache retry prepare

Oracle が読み込み一貫性またはキャッシュ・フュージョン要求用のバッファの準備に失敗するたびに、また Oracle がこの失敗を無視またはスキップできない場合に発生する待機イベント。

global cache S to X

任意のリソース変換が完了するまで、セッションが待機する必要がある場合に、必ず発生する待機イベント。

global lock async converts

Oracle が非互換モードから変換したリソースの数。

global lock async gets

Oracle が非同期でオープンする必要がある GES リソースの数。async gets は、GES リソースに対してのみ使用され、global cache gets が含まれる。

global lock convert time

すべての global lock sync converts および global lock async converts に必要な累計時間。

global lock get time

Oracle がすべての GES リソースをオープンするために必要な累計時間。

global lock sync converts

Oracle が非互換モードから変換した GES リソースの数。sync converts は、主に GES リソースに対して発生する。

global lock sync gets

Oracle が同期でオープンする必要がある GCS リソースの数。sync gets は主に GES リソース (ライブラリ・キャッシュ・リソースなど) に対するものである。

lock buffers for read

NULL から共有への上位変換の回数。

messages flow controlled

直接送信される予定だったが、実際には LMD/LMS によってキューされ、後で配信されたメッセージの数。

messages received

LMD プロセスが受信したメッセージの数。

messages sent directly

Oracle プロセスによって直接送信されたメッセージの数。

messages sent indirectly

Oracle プロセスによって明示的にキューされたメッセージの数。

Oracle Data Gatherer

Oracle Performance Manager 用のパフォーマンス統計を収集するコンポーネント。Oracle Data Gatherer は、ネットワーク上の **ノード** にインストールする必要がある。

Oracle Enterprise Manager

異機種間環境を集中管理するための統合されたソリューションを提供する、システム管理ツール。Oracle Enterprise Manager によって、グラフィカルなコンソール、Management Server、Oracle Intelligent Agent、リポジトリ・データベースおよびツール製品が組み合わせられ、Oracle 製品を管理するための包括的な統合システム管理プラットフォームが提供される。

Oracle Enterprise Manager のコンソール (Oracle Enterprise Manager Console)

Oracle Enterprise Manager 製品を構成する一連の GUI ツール。

Oracle Intelligent Agent

各ノードで実行するプロセス。Management Server 経由でコンソールから送信されるジョブおよびイベントの実行者として機能する。エージェントは**コンソール**またはネットワーク接続の状態に関係なく機能できるため、Oracle Intelligent Agent によって高可用性が保証される。

Oracle Net

接続を可能にするソフトウェア・コンポーネント。Oracle Net Foundation レイヤーと呼ばれるコア通信レイヤー、およびネットワーク・プロトコル・サポートが含まれる。Oracle Net によって、サービスとそのアプリケーションを異なるコンピュータに常駐でき、ピア・アプリケーションとして通信できる。Oracle Net の主な機能には、ネットワーク・セッションの確立、およびクライアント・マシンとサーバー間または2つのサーバー間のデータ転送がある。ネットワーク・セッションの確立後は、Oracle Net はクライアントとサーバーのためのデータ伝達手段として機能する。

Oracle Performance Manager

Real Application Clusters のパフォーマンス統計を様々な表形式やグラフ形式で提供する **Oracle Enterprise Manager** のアドオン・アプリケーション。統計は、すべてのインスタンスの集計パフォーマンスを表す。

Oracle Real Application Clusters

複数クラスタによる共有データベースへのアクセスを可能にする画期的なアーキテクチャ。Real Application Clusters には、Oracle9i Enterprise Edition データベースを Oracle9i Real Application Clusters データベースにするために必要な、Real Application Clusters スクリプト、初期化ファイルおよびデータ・ファイルを提供するソフトウェア・コンポーネントが含まれる。

Oracle9i Enterprise Edition

オブジェクト・リレーショナル・データベース管理システム (ORDBMS)。データベース管理用のアプリケーションおよびファイルを提供する。その他のすべての Real Application Clusters コンポーネントは、Oracle9i Enterprise Edition の最上部のレイヤーに位置している。

Oracle システム識別子 (Oracle System Identifier: SID)

リリース 8.1 より前の Oracle データベースが動作しているときに、特定のインスタンスを識別する名前。Real Application Clusters データベースの場合は、クラスタ内の各ノードに、データベースを参照するインスタンスがある。各ノードの SID は、initdb_name.ora ファイルの DB_NAME パラメータで指定されたデータベース名と、一意の**スレッド ID** で構成される。スレッド ID は、クラスタ内の最初のインスタンスが 1 で始まり、2 番目以降のインスタンスの ID は 1 ずつ大きくなる。

physical reads

データ・ブロックに対する要求が、ローカル・キャッシュからでは満たされない場合に実行する必要がある、ディスク読取りの数。

physical writes

DBWN_n プロセスによって実行される書込み I/O の数。GC_FILES_TO_LOCKS が設定されている場合、この数値には、Oracle9i での DBWR cross-instance writes (強制書込み) の数が含まれる。特定のデータ・ファイルに対して GC_FILES_TO_LOCKS を設定すると、以前の ping プロトコルが使用可能になり、キャッシュ・フュージョン・アーキテクチャは活用されない。

RAW デバイス (raw device)

ファイル・システムが設定されていないディスクまたはディスク・ドライブ上のパーティション。RAW デバイスは、ディスクの共有を使用可能にするため、Real Application Clusters に使用される。

RAW ボリューム (raw volume)

「[RAW デバイス \(raw device\)](#)」を参照。

Real Application Clusters

「[Oracle Real Application Clusters](#)」を参照。

Recovery Manager (RMAN)

Oracle のツール製品の 1 つ。データ・ファイル、制御ファイルおよびアーカイブ REDO ログの、バックアップ、コピー、リストアおよびリカバリを行う。Recovery Manager は Oracle サーバーに含まれているため、個別にインストールする必要はない。Recovery Manager は、オペレーティング・システム (OS) のプロンプトからコマンドライン・ユーティリティとして起動するか、または GUI ベースの Enterprise Manager の一部である Backup Manager を使用できる。

remote instance undo block writes

DBWN が強制書込みの一部として UNDO ブロックをディスクに書き込んだ回数。

remote instance undo header writes

DBWN が強制書込みの一部として、ロールバック・セグメント・ヘッダー・ブロックをディスクに書き込んだ回数。

Server Management

Real Application Clusters 環境を管理するための包括的な統合システム管理ソリューション。Server Management を使用すると、異機種間環境でクラスター・データベースを管理できる。Server Management は、**Oracle Enterprise Manager** のオープンなクライアント / サーバー・アーキテクチャの一部である。クラスター・データベースの管理の他に、Server Management では、ジョブのスケジューリング、イベント管理の実行、パフォーマンスの監視および統計の取得を実行して、Real Application Clusters のパフォーマンスを監視できる。

Transmission Control Protocol/Interconnect Protocol (TCP/IP)

連携して動作するコンピュータがネットワーク上でリソースを共有できるようにする、プロトコルのセット。

意思決定支援システム (decision support system: DSS)

意思決定支援システムまたはデータ・ウェアハウス・システムを備えた、データベース環境およびアプリケーション環境。

インスタンス (instance)

システム・グローバル領域 (SGA) と Oracle データベースの各プロセスの組合せ。インスタンスのメモリーおよびプロセスは、対応付けられたデータベースのデータを管理し、データベースのユーザーのために機能する。各インスタンスには、一意の **Oracle システム識別子**、**インスタンス名**、**インスタンス番号**、**ロールバック・セグメント**および**スレッド ID**がある。

インスタンス・グループ (instance group)

パラレル操作に使用するインスタンスの数を制限するために使用する。インスタンス・グループは、それぞれに1つ以上のインスタンスを含めて任意の数だけ作成できる。次に、任意またはすべてのパラレル操作で使用するインスタンス・グループを指定できる。パラレル実行サーバーは、指定されたインスタンス・グループのメンバーであるインスタンス上でのみ使用される。

インスタンス番号 (instance number)

特定のインスタンスとデータ・ブロックのエクステントを対応付ける番号。インスタンス番号を使用すると、インスタンスを起動でき、挿入および更新に対して、そのインスタンスに割り当てられたエクステントを使用することが保証される。これによって、インスタンスが他のインスタンスに割り当てられた領域を使用しないことが保証される。インスタンス番号を使用して再起動されないかぎり、インスタンスは、他の空きリストのデータ・ブロックを使用できない。

INSTANCE_NUMBER 初期化パラメータでデータ・ブロックのエクステントをインスタンスに対応付けると、様々な SQL オプションを使用できる。

インスタンス番号は、init`sid`.ora インスタンス初期化ファイルの INSTANCE_NUMBER パラメータによって指定される。

インスタンス名 (instance name)

インスタンスの名前。複数インスタンスで共通のサービス名が共有される場合は、特定のインスタンスを一意に識別するために使用される。インスタンス名は、初期化パラメータ・ファイルの `INSTANCE_NAME` パラメータによって識別される。インスタンス名は、**Oracle システム識別子**と同じ。

インターコネクト (interconnect)

ノード間の通信リンク。

オペレーティング・システム依存 (OSD) のクラスタウェア (operating system-dependent (OSD) clusterware)

Oracle または他のベンダーが開発した複数のソフトウェア・コンポーネントで構成されたソフトウェア。OSD レイヤーによって、**Real Application Clusters** の操作に必要な、主要なオペレーティング・システム / クラスタウェア・サービスがマッピングされる。

オペレーティング・システムのコンテキストのスイッチング (operating system context switch)

スレッドの割当て時間が経過した場合、優先順位の高いスレッドが実行できる場合、または実行中のスレッドが I/O の完了などを待機する必要がある場合に発生する。

仮想インタフェース・アーキテクチャ (Virtual Interface Architecture: VIA)

ユーザー・モードの IPC の実装。

キャッシュ・フュージョン (Cache Fusion)

Real Application Clusters におけるディスクレスのキャッシュー貫性メカニズム。保持側インスタンスのメモリー・キャッシュから要求側インスタンスのメモリー・キャッシュに、ブロックのコピーを直接送信する。

共有サーバー (shared server)

非常に少数のサーバー・プロセスを、多数のユーザー・プロセスで共有できるように構成したサーバー。これによって、サポートできるユーザーの数が増える。共有サーバーでは、多数のユーザー・プロセスが**ディスパッチャ**に接続する。「**専用サーバー (dedicated server)**」と対比。

クラスタ (cluster)

協調して同じタスクを実行するインスタンスの集合。

クラスタ化データベース (clustered database)

「**クラスタ・データベース (cluster database)**」を参照。

クラスタ・データベース (cluster database)

Oracle Real Application Clusters データベースに対する一般的な用語。

クラスタリング (clustering)

「[クラスタ・データベース \(cluster database\)](#)」を参照。

グローバル・エンキュー・サービス (Global Enqueue Service: GES)

グローバルに共有されるエンキューを調整するサービス。

グローバル・エンキュー・サービス・デーモン (Global Enqueue Service Daemon: LMD)

GES リソース要求を管理するリソース・エージェント・プロセス。LMD プロセスは、デッドロック検出および GES 要求も処理する。リモート・リソース要求とは、別のインスタンスから発生する要求のことである。

グローバル・エンキュー・サービス・モニター (Global Enqueue Service Monitor: LMON)

クラスタ全体を監視して、グローバル・リソースを管理するバックグラウンド・プロセス。LMON は、インスタンスやプロセスの期限切れ、および GCS や GES に関連するリカバリを管理する。特に、グローバル・リソースに関連するリカバリの部分を処理する。LMON が提供するサービスは、クラスタ・グループ・サービス (CGS) とも呼ばれる。

グローバル・キャッシュ・サービス (Global Cache Service: GCS)

キャッシュ・フュージョンを実装するプロセス。ブロックのブロック・モードをグローバル・ロールで保持し、インスタンス間のブロック転送を行う。GCS は、LMS や GES などのバックグラウンド・プロセスを使用してこれらのタスクを実行する。

グローバル・キャッシュ・サービス・プロセス (Global Cache Service Process: LMSn)

リモートの[グローバル・キャッシュ・サービス](#)・メッセージを処理するプロセス。[Real Application Clusters](#) では、最大 10 のグローバル・キャッシュ・サービス・プロセスが提供される。LMS_n の数は、クラスタのノード間のメッセージ通信量によって異なる。LMS_n は、GCS リソースに対するリモート・インスタンスからの獲得割込み要求およびブロッキング割込み要求を処理する。クロス・インスタンス読込み一貫性要求の場合、LMS_n はブロックの読込み一貫性バージョンを作成して、要求側インスタンスに送信する。また、LMS_n は、リモート・インスタンスとの間のメッセージ・フローも制御する。

グローバル・キャッシュ初期化パラメータ (global cache initialization parameters)

すべてのインスタンス上のデータベース・バッファを保護する、グローバル・キャッシュの集合のサイズを設定する初期化パラメータ。

グローバル・データベース名 (global database name)

各データベースを一意に識別するための完全な名前。グローバル・データベース名の書式は、`database_name.database_domain` (sales.us.acme.com など) である。

コンソール (Console)

Oracle Enterprise Manager のコンソールでは、強力なシステム管理を実現する直観的なグラフィカル・ユーザー・インタフェース (GUI) を介して、Oracle 環境の主要部分を管理できる。

サーバー・クラスタリング (server clustering)

「**Oracle Real Application Clusters**」を参照。

サービス登録 (service registration)

PMON プロセス (共有サーバーを使用する場合は、共有サーバー・ディスパッチャ・プロセス) がリスナーに情報を自動的に登録する機能。この情報はリスナーに登録されるため、listener.ora ファイルはこの静的情報で構成する必要はない。

サービス名 (service name)

データベースの論理表現。データベースをクライアントに提示する方法。データベースは複数のサービスとして提供できる。また、サービスは複数のデータベース・インスタンスとして実装できる。サービス名は、**グローバル・データベース名**を表す文字列である。グローバル・データベース名は、インストール時またはデータベースの作成時に入力したデータベース名 (DB_NAME) およびドメイン名 (DB_DOMAIN) で構成される。

シード・データベース (seed database)

データベース作成時のユーザー入力を最小限に抑えた、すぐに使用できる構成済みのデータベース。

接続時フェイルオーバー (connect-time failover)

最初のリスナーが応答しない場合、クライアントの接続要求は他のリスナーに転送される。接続時フェイルオーバーは、リスナーが接続の試行前にインスタンスが起動されているかどうかを認識できるため、**サービス登録**によって使用可能になる。

接続時ロード・バランシング (connection load balancing)

同一サービスの様々なインスタンスおよび**共有サーバー**・ディスパッチャ間で、アクティブな接続数を均衡化する機能。

初期化パラメータ・ファイル (initialization parameter file)

データベース (initdb_name.ora) を初期化するパラメータ設定を含むファイル。Real Application Clusters の場合、このファイルはクラスタ内のインスタンス (initsid.ora) を初期化する。デフォルトの単一初期化パラメータ・ファイルは、SPFILE.ORA と呼ばれる。

スター・スキーマ (star schema)

問合せ中心のスキーマ。図で示すと、ファクト表がその中心にある。通常、ファクト表には、スキーマに対して操作される問合せの中心となるデータ要素が含まれる。多くの場合、ファクト表は非常に大規模であり、ファクト表内のデータの属性であるデータを含む複数のディメンション表に囲まれている。スター・スキーマでは、ディメンション表内の属性をファクト表のデータと結合する方法が直観的であるため、問合せ作成が簡単になる。スター・スキーマは、データ・ウェアハウス環境に最適であるが、OTLP 環境には有効ではない。

ストライプ化 (striping)

ディスク間での、データの関連ブロックのインターリーブ。ストライプ化を正しく実装すると、I/O が削減され、パフォーマンスが向上する。ストライプ化ソフトウェアは、オペレーティング・システム固有であるため、ベンダーのマニュアルを参照して、正しくインストールおよび構成する必要がある。

スレッド ID (thread ID)

インスタンス用の REDO スレッドの番号。使用可能状態の REDO スレッド番号が使用できるが、他のインスタンスと同じスレッド番号は使用できない。

制御ファイル (control file)

データベースの物理構造を記録し、データベース名、関連データベースおよびオンライン REDO ログ・ファイルの名前と位置、データベース作成のタイムスタンプ、現在のログの順序番号、チェックポイント情報、およびデータベースの構造と状態に関する他の様々な記録を含むファイル。

専用サーバー (dedicated server)

各ユーザー・プロセスに専用のサーバー・プロセスが必要なサーバー。サーバー・プロセスは、クライアントごとに1つある。Oracle Net によって、既存のサーバー・プロセスのアドレスがクライアントに返信される。その後、クライアントは指定されたサーバー・アドレスに接続要求を再送する。「共有サーバー (shared server)」と対比。

ディスク書き込みの強制実行 (forced disk writes)

インスタンスが DML 操作にデータ・ブロックを要求した場合に、他のインスタンスによって実行される、ディスクへのデータ・ブロックの強制書き込み。実際には、Oracle9i では、キャッシュ・フュージョンによって強制書き込みが発生しないが、GC_FILES_TO_LOCKS パラメータに 1:1 または 1:n の解放可能リソースまたは固定リソースを指定した場合は、引き続き強制書き込みが使用可能になる。この場合、その表領域ではキャッシュ・フュージョンが使用禁止になる。

ディスパッチャ (dispatcher)

多くのクライアントが、クライアントごとの専用サーバー・プロセスなしで同じサーバーに接続できるようにするプロセス。ディスパッチャは、複数の受信ネットワーク・セッション要求を処理して、共有サーバー・プロセスに送る。「**共有サーバー (shared server)**」を参照。

透過的アプリケーション・フェイルオーバー (Transparent Application Failover: TAF)

Real Application Clusters や Oracle Real Application Clusters Guard などの、高可用性の環境を目的としたランタイム・フェイルオーバー・メカニズム。これは、アプリケーションとサービス間の接続のフェイルオーバーおよび再確立を意味する。これによって、接続に失敗した場合、クライアント・アプリケーションは自動的にデータベースに再接続され、処理中の SELECT 文を再開することもできる。この再接続は、Oracle Call Interface (OCI) ライブラリ内から自動的に実行される。

トランザクションごとの CR blocks received (CR blocks received per transaction)

ブロックを排他アクセス・モードで保持するインスタンスから、そのブロックの CR バージョンを要求するインスタンスに転送される CR ブロックの数。

トランザクションごとの lock converts (lock converts per transaction)

トランザクションごとの global local sync converts および global lock async converts の数。

トランザクションごとの lock gets (lock gets per transaction)

トランザクションごとの global lock sync gets および global lock async gets の数。

トランザクション・モニター (transaction monitor)

オペレーティング・システムより上位のトランザクション実行レイヤーを提供するソフトウェア製品のクラス。トランザクション・モニターはデータベースの更新を組み合わせ、それらをデータベースに送る。このとき、トランザクション・モニターはデータベースの一貫性および正確性の一部を管理する。モニターによって、トランザクションのアトム性（更新が完全に実行されるか、またはまったく実行されないこと）の規則が守られることが保証される。トランザクション・モニターを使用するメリットには、スループットの向上がある。

ノード (node)

1つ以上のインスタンスが常駐するマシン。

ハイブリッド・データベース (hybrid database)

OLTP とデータ・ウェアハウス処理の両方の特長を備えたデータベース。

パラレル実行 (parallel execution)

単一のデータベース・トランザクションを完了するために同時に操作される、マルチ・プロセス。パラレル実行は、シングル・インスタンスとクラスタ・データベースの両方の Oracle 上で機能する。パラレル問合せともいう。

パラレル実行自動チューニング機能 (parallel automatic tuning)

パラレル実行に関連するすべてのパラメータの値を自動的に制御する機能。これらのパラメータは、サーバー処理の一部である**並列度 (DOP)**、マルチユーザー問合せ調整機能およびメモリーのサイズ指定に影響する。パラレル実行を初期化し、自動的にチューニングするには、`PARALLEL_AUTOMATIC_TUNING` 初期化パラメータを `true` に設定する。

プロセス間通信 (Inter-Process Communication: IPC)

高速オペレーティング・システム固有の転送コンポーネント。IPC は、様々なノード上のインスタンス間でメッセージを転送する。**インターコネクト**とも呼ばれる。

並列度 (degree of parallelism: DOP)

パラレル実行で使用されるプロセスまたはスレッドの数を指定する。各パラレル処理またはスレッドは、SQL 文の複雑さに応じて、1 つまたは2 つのパラレル実行プロセスを使用できる。

ユーザー・データグラム・プロトコル (User Datagram Protocol: UDP)

TCP/IP に似たプロトコル。しかし管理はより簡単である。UDP は、メッセージの順序付けなどを保証しないため、TCP/IP より信頼性に劣ると考えられる。

ユーザー・モードの IPC (user-mode IPC)

ネットワーク・ハードウェアに直接アクセスする IPC (プロセス間通信) ベースのプロトコル。カーネル・モードの IPC とは異なり、ユーザー・モードの IPC を使用すると、プロトコルによって、カーネル領域へのデータのコピー、システム・コールの作成およびコンテキストのスイッチングの発生によるオーバーヘッドが回避される。

読み込み一貫性 (consistent read: CR)

グローバル・キャッシュ・サービスによって、CR ブロック (マスター・コピー・データ・ブロックともいう) の保持が保証される。CR ブロックは、ブロックに対するすべての変更に関する情報を記録するマスター・ブロック・バージョンである。これは、ブロックが変更される場合、クラスタ内の1 つ以上の SGA に保持される。インスタンスがブロックを読み込む必要がある場合、現行のバージョンのブロックは、共有リソースとして多くのバッファ・キャッシュに常駐できる。そのため、すべてのインスタンス上のすべてのトランザクションがコミットしたかどうかにかかわらず、すべての SGA 内のブロックの最新コピーには、それらのインスタンスによってブロックに加えられた変更が含まれる。

リスナー (listener)

リスナー構成ファイル。リスナーが接続要求およびサービスをリスニングして受け取るプロトコル・アドレスを識別する。

リポジトリ・データベース (repository database)

Real Application Clusters 環境を管理するために **Oracle Enterprise Manager** が使用するデータが格納されている、Oracle データベース内の表の集合。このデータベースは、**ノード**上のすべての共有 Real Application Clusters データベースとも異なる。

ロード・バランシング (load balancing)

インスタンス間でのアクティブ・データベース接続の均等分散。パラレル実行で使用される場合は、CPU およびメモリー・リソース間で作業を分散させるためのパラレル実行サーバー・プロセスの分散を示す。

索引

A

ALLOCATE EXTENT
 DATAFILE 句, B-8
 INSTANCE 句, B-8
 SIZE 句, B-8
 排他モード, B-7
ALTER CLUSTER 文
 ALLOCATE EXTENT 句, B-7
ALTER DATABASE 文
 DATAFILE RESIZE, A-5
ALTER SEQUENCE 文
 キャッシュ・サイズの増加, 4-10
ALTER SESSION 文
 SET INSTANCE 句, B-6
ALTER TABLE 文
 ALLOCATE EXTENT, B-7
AUTOEXTEND 句, A-5

C

CATCLUST.SQL スクリプト
 Real Application Clusters のビューを作成するための
 使用, 4-5
CLUSTER_INTERCONNECTS
 パラメータ, 4-4
CREATE CLUSTER 文, B-5
 FREELIST GROUPS 句, B-4
 FREELISTS 句, B-4
CREATE TABLE 文
 FREELISTS 句, B-4
 クラスタ化表, B-5
 初期記憶域, B-9
CREATE 文
 FREELISTS および FREELIST GROUPS の設定, B-4

D

Data Guard, 1-4
Database Configuration Assistant (DBCA)
 Real Application Clusters でのビューの作成, 4-5
DB_BLOCK_SIZE
 パラメータ, 3-5
DB_FILE_MULTIBLOCK_READ_COUNT
 全表スキャン用の増加, 3-5
 パラメータ, 3-5
DBA_QUEUE_TABLES
 アドバンスト・キューイングの表およびインスタ
 ンス・アフィニティの分析, 3-6
DBWRn プロセス
 チェックポイント, 4-3
 パフォーマンスの改善, 3-6
DC_SEQUENCES
 パラメータ, 3-4

E

E-Commerce
 Real Application Clusters のアプリケーション, 2-2
enqueue
 グローバル・エンキュー・サービス (GES) 統計,
 4-10

F

FAST_START_MTTR_TARGET, 4-3
 パラメータ, 4-3
File I/O Rate By Instance チャート, 5-4
File I/O Rate By Object チャート, 5-4
File I/O Rate チャート, 5-4, 5-10

FREELIST GROUPS 句, B-4
再編成の必要性の判断, B-3
パラメータ、索引の使用, B-6
パラメータ、使用, B-4

FREELISTS 句, B-4
STORAGE 句, B-4
クラスタ化表用の作成, B-5
最大値, B-4
索引用の作成, B-6
使用例, B-5
パラメータ、索引の使用, B-6
パラメータ、使用, B-4

G

GC_FILES_TO_LOCKS パラメータ, A-5, A-11
構文, A-4
設定, A-4
不適切な ping の削減, A-15
例, A-5

global cache blocks corrupt, 4-11
global cache blocks lost, 4-11
Global Cache Convert By Instance チャート, 5-4
Global Cache Convert チャート, 5-4, 5-8, 5-11
global cache cr request, 4-12
Global Cache CR Request By Instance チャート, 5-4
Global Cache CR Request チャート, 5-4, 5-7
Global Cache Current Block Request By Instance
チャート, 5-4
Global Cache Current Block Request チャート, 5-4,
5-9, 5-11
global cache null to s, 4-12
global cache null to x, 4-12
global cache open s, 4-12
global cache open x, 4-12
global cache waits, 4-12
GLOBAL ヒント, 4-4

I

INITIAL 記憶域パラメータ
最小値, B-9

INITRANS
パラメータ, 4-11

INSERTS
使用不可能な空き領域, B-7
同時, B-4

INST_ID 列, 4-4

INSTANCE_GROUPS
初期化パラメータ, 2-6

INSTANCE_NUMBER パラメータ
初期化パラメータ
INSTANCE_NUMBER, B-6

INSTANCE_ROLE
セカンダリ・インスタンス接続での使用, 1-5

INSTANCE 句
SET INSTANCE 文, B-6

IPC
キャッシュ・フュージョン, 4-2
バッファ・サイズ、調整, 4-3

J

JDBC OCI, 1-5

L

Library Cache Lock By Instance チャート, 5-4
Library Cache Lock チャート, 5-4, 5-8, 5-11

LOG_CHECKPOINT_INTERVAL, 4-3
パラメータ, 4-3

LOG_CHECKPOINT_TIMEOUT, 4-3
パラメータ, 4-3

M

MAXEXTENTS 記憶域パラメータ
自動割当て, B-9

MAXTRANS
パラメータ, 4-11

MINEXTENTS 記憶域パラメータ
自動割当て, B-9
デフォルト, B-9

O

OCI ライブラリ, 1-5

Oracle
互換性, B-7

Oracle Net
OCI ライブラリ, 1-5

Oracle Performance Manager
概要, 5-2
チャートの表示, 5-2

oradebug
IPC, 4-4
setmypid, 4-4

P

PARALLEL_INSTANCE_GROUP
パラメータ, 2-7
PARALLEL_MAX_SERVERS
パラメータ, 4-4, 4-5
PCTINCREASE パラメータ
表エクステンツ, B-8
Performance Manager
チャート、階層, 5-4
PL/SQL
Real Application Clusters, 1-6
パッケージおよび TAF, 1-5

R

RAW デバイス, 1-2
Real Application Clusters
E-Commerce, 2-2
ディスク・アフィニティ, 2-7
パラレル実行, A-11
表領域の使用, 3-2
Real Application Clusters Database Health Overview
チャート, 5-4, 5-11, 5-12
Real Application Clusters Guard I, 1-3
Real Application Clusters Guard II, 1-3
Real Application Clusters Top Sessions チャート, 5-4,
5-10
Recovery Manager
Real Application Clusters, 1-7
Row Cache Lock By Instance チャート, 5-4
Row Cache Lock チャート, 5-4, 5-9, 5-11

S

Session チャート, 5-4, 5-10
SQ エンキュー
V\$ENQUEUE_STATS, 4-10
STATISTICS_LEVEL
パラメータ, 4-6
Statspack
Real Application Clusters で使用, 4-7

T

TIMED_STATISTICS
パラメータ, 4-6
Total Transfer チャート, 5-4, 5-6, 5-11
下位レベルのチャートの階層, 5-5
Transfer By File Per Instance チャート, 5-4
Transfer By File チャート, 5-4, 5-7
Transfer By Instance Per File チャート, 5-4
Transfer By Instance Per Object チャート, 5-4
Transfer By Instance チャート, 5-4
Transfer By Object Per File チャート, 5-4
TX エンキュー
V\$ENQUEUE_STATS, 4-11

U

USER_QUEUE_TABLES
アドバンスド・キューイングの表およびインスタ
ンス・アフィニティの分析, 3-6
Users By Instance チャート, 5-4
Users チャート, 5-4, 5-10

V

V\$BH
競合オブジェクトの識別, 4-15
V\$CACHE
競合オブジェクトの識別, 4-15
V\$CACHE_TRANSFER
競合オブジェクトの識別, 4-15
VIA
インターコネクト・プロトコル, 4-2

あ

空きリスト
クラスタ, B-5
クラスタ化表用の作成, B-5
索引用の作成, B-6
排他モード, B-4, B-7
ハッシュ・クラスタ, B-5
リストの数, B-4
例, B-5
空きリスト・グループ
!blocks の設定, A-5

- セッションへの割当て, B-6
- 同時挿入, B-2
- アクティブ / アクティブ構成
 - Real Application Clusters, 1-3
- アドバンスト・キューイング
 - Real Application Clusters 内、パフォーマンス, 3-6
 - キュー・テーブルのインスタンス・アフィニティ, 3-6
 - キュー・テーブルのキャッシュ転送, 3-7
- アフィニティ
 - Real Application Clusters のデータ, 2-5
 - 表およびアドバンスト・キューイング, 3-6
- アプリケーションのワークロード
 - Statspack の統計, 4-7

い

- 移行
 - 排他モードへの戻り, B-7
- インスタンス
 - 空きリスト, B-7
 - 空きリスト・グループとの対応付け, B-6
 - インスタンスの追加, B-9
 - エクステンントへの対応付け, B-6
 - データ・ファイルへの対応付け, B-9
 - 番号, B-6
- インターコネクト
 - Real Application Clusters のプロトコル, 4-3
 - SQL*Plus を使用して識別, 4-3
 - 設定の検証, 4-3
 - 帯域幅, 4-3
 - 待機時間、グローバル・キャッシュ・サービス (GCS) のタイミングの分析, 4-10
 - パフォーマンス, 4-3

え

- エクステンント
 - 空きリスト・グループへのエクステンントの割当て, B-9
 - インスタンスへの未割当て, B-8
 - インスタンスへの割当て, B-6
 - サイズ, B-8
 - 事前割当て, B-7
 - 初期割当て, B-9
 - ファイルの指定, B-8
- エクステンント管理, 3-2

- エラー・メッセージ
 - 記憶域オプション, B-4

お

- 応答時間
 - 分析, 4-12
- オブジェクト
 - パフォーマンスの問題の識別, 4-9
 - ホット・ブロック, 4-14
 - ホット・ブロックのためのパーティション化, 3-3
- オペレーティング・システム
 - パフォーマンス用のストライプ化, 3-5
- オンライン・トランザクション処理
 - Real Application Clusters, 2-3

か

- 拡張性のテスト, 2-8
- 監視
 - Real Application Clusters の統計, 4-5
 - 手順, 4-8
- 完全なアクティブ構成, 1-3
- 管理
 - Real Application Clusters での拡張性, 2-9

き

- 記憶域, 1-2
 - オプション、エクステンント・サイズ, B-8, B-9
 - オプション、クラスタ化表, B-4
 - オプション、表, B-4
- 機能
 - 活用, 1-2
- キャッシュ
 - AQ でのキュー・テーブルのブロックの転送, 3-7
 - サイズおよび順序番号の使用, 3-4
 - サイズ指定, 4-14
 - サイズ、デフォルトおよび順序番号, 3-4
 - 順序番号, 3-4
- キャッシュ・フュージョン
 - E-Commerce アプリケーション, 2-2
 - パフォーマンス統計のソース, 4-6
 - リソース、共有, A-6
 - リソース、指定, A-5
 - リソース、排他, A-6

キャッシュ・フュージョン・リソースへのブロックの
マッピング, A-5

共有サーバー

Real Application Clusters, 1-4

接続時ロード・バランシング, 1-4

共有ディスク・アーキテクチャ

データ・アフィニティ, 2-5

共有プール

Real Application Clusters でのサイズ調整, 4-3

く

クラスタ

空きリスト, B-5

空きリスト・グループ, B-7

ハッシュ・クラスタ, B-5

パラレル実行のチューニング, A-11

クラスタ化表

空きリストおよび空きリスト・グループ, B-5

クラスタ・ファイル・システム

Real Application Clusters, 1-2

グローバル V\$ 表, 5-2

グローバル・エンキュー・サービス (GES)

統計, 4-8, 4-10

メッセージ通信とインターコネクト, 4-3

グローバル・キャッシュ・サービス (GCS)

インターコネクトの分析, 4-10

統計, 4-8

統計のソース, 4-6

メッセージ通信とインターコネクト, 4-3

リソース, A-11

リソース、取得, 3-6

論理読取り, 4-13

グローバル・キャッシュの待機イベント, 4-9

こ

高可用性

Real Application Clusters, 1-3

完全なアクティブ構成, 1-3

構成

完全なアクティブ, 1-3

互換性

共有モードおよび排他モード, B-7

コストベース・オブティマイザ, 2-5

コンポジット・パーティション化オブジェクト

ホット・ブロックのパフォーマンスの問題, 3-3

か

サーバー・パラメータ・ファイル, 1-2

サーバー・プロセス

パラレル実行, 2-5

再送信, 4-10

索引

空きリストおよび空きリスト・グループ, B-6

ブロックの分割, 3-3

索引リーフ・ブロック

パフォーマンス, 4-9

し

シェアード・ナッシング・システム

データ・アフィニティ, 2-5

識別子

リソース, A-16

システム・グローバル領域 (SGA)

統計の保持, 4-5

事前割当て

空きリスト・グループへのエクステンツの割当て,

B-9

エクステンツ, B-7

自動 UNDO 管理, 1-2

自動セグメント領域管理, 1-2, 3-2

巡回冗長検査 (CRC) エラー, 4-10

順序番号

キャッシュ、順序付けを使用した抑制, 3-4

グローバル競合, 3-4

使用, 3-4

生成および表領域の使用, 3-2

データベース表を使用した生成, 3-4

初期化パラメータ

CLUSTER_INTERCONNECTS, 4-4

DB_BLOCK_SIZE, 3-5

DB_FILE_MULTIBLOCK_READ_COUNT, 3-5

DC_SEQUENCES, 3-4

FAST_START_MTTT_TARGET, 4-3

INITRANS, 4-11

INSTANCE_GROUPS, 2-6

INSTANCE_ROLL, 1-5

LOG_CHECKPOINT_INTERVAL, 4-3

LOG_CHECKPOINT_TIMEOUT, 4-3

MAXTRANS, 4-11

PARALLEL_INSTANCE_GROUP, 2-7

PARALLEL_MAX_SERVERS, 4-4, 4-5

PCTINCREASE, B-8
STATISTICS_LEVEL, 4-6
TIMED_STATISTICS, 4-6

す

スタンバイ・データベース, 1-4
ストライプ化
ディスク・アフィニティ, 2-7

せ

制御ファイル
データ・ファイル, B-8
セグメント
ヘッダー, A-17
接続時ロード・バランシング, 1-5
専用サーバー
接続時ロード・バランシング, 1-5

そ

挿入
領域管理, 3-2
増分的な拡張, B-9

た

帯域幅
インターコネクト, 4-3
待機イベント
分析, 4-12

ち

チェックポイント, 4-3
頻度の低い, 4-14
チャート
Performance Manager, 5-2

つ

通信ジェネレータ, 2-8
通信プロトコル
設定の検証, 4-3

て

定期的な統計, 4-5
ディスク・アフィニティ
パラレル問合せ, 2-7
ディスク書込み
理由, 4-14
ディスク書込みの強制実行, A-15
不適切, A-14
理由, 4-14
データ・アフィニティ, 2-5
データ位置
Real Application Clusters, 2-5
データ・ウェアハウス
Real Application Clusters でのアプリケーションの
配置, 2-4
データ・ディクショナリ
ビューの問合せ, 4-5
データ・ファイル
エクステンツの割当て, B-8
各表の複数ファイル, B-9
データ・ブロック
アドバンスト・キューイングのキャッシュ転送,
3-7
データベース・ライター・プロセス
チェックポイント, 4-3
パフォーマンスの改善, 3-6
データベース・リソース
動的な移行, 2-3

と

問合せ
チューニングのヒント, 3-5
透過的アプリケーション・フェイルオーバー (TAF),
1-5
統計
Oracle による収集, 4-6
Real Application Clusters の監視, 4-5
記録の保持, 4-6
内容, 4-5
保持, 4-5
動的パフォーマンス・ビュー
作成, 4-5
パフォーマンスの監視用, 4-4
トランザクション・モニター, 1-4

の

- ノード
 - 追加, 2-9, B-9
- ノード内でのパラレル化, 2-5
- ノードの追加, 2-9

は

- バージョン、Oracle
 - 互換性, B-7
- 排他モード
 - 空きリスト, B-4, B-7
- パケット
 - 削除, 4-10
- パスト・イメージ (PI)
 - リカバリで使用, 4-3
- ハッシュ・クラスタ, B-5
- ハッシュ・パーティション化オブジェクト
 - ホット・ブロックのパフォーマンスの問題, 3-3
- バッファ・キャッシュ
 - Real Application Clusters でのサイズ調整, 4-3
 - 統計のソース, 4-6
- バッファ・サイズ
 - IPC、Real Application Clusters 用に調整, 4-3
- パフォーマンス
 - Real Application Clusters のアプリケーションにおける問題, 3-5
 - Statspack の使用, 4-7
 - 影響する主な要素, 4-3
 - 待機イベント、分析, 4-12
 - テスト, 2-8
 - 履歴のメンテナンス, 4-6
- パラレル・インスタンス・グループ, 2-6
- パラレル化
 - Real Application Clusters, 2-4
 - パラレル問合せの最適化, 2-5
- パラレル実行
 - クラスタ, A-11
 - サーバー・プロセスおよびデータ・アフィニティ, 2-5
 - プロセス, 4-5
 - ロード・バランシング, 2-6

ひ

- 非同期 I/O
 - 大量のデータの書込み, 3-6
- ビュー
 - Real Application Clusters での作成, 4-5
 - パフォーマンスの監視用, 4-4
- 表
 - アフィニティおよびアドバンスド・キューイング, 3-6
 - クラスタ, B-5
 - 使用不可能な空き領域, B-7
 - 初期記憶域, B-9
 - 高い挿入率, 3-3
 - 複数のファイル, B-9
- 表領域
 - Real Application Clusters で使用, 3-2
 - 自動セグメント領域管理のためのローカル管理, 3-2

ふ

- ファイル
 - エクステンツの割当て, B-8
- ファンクション・ SHIPPING, 2-4
- フェイルオーバー
 - Real Application Clusters, 1-3
- 負荷処理能力, 2-8
- 不適切な ping, A-3
- 不適切な強制書込み, A-14
- フュージョン書込み, 4-14
- プライマリ / セカンダリ・インスタンス構成
 - Real Application Clusters, 1-3
- TAF, 1-5
- プロセス
 - パラレル実行, 4-5
- ブロック・サイズ
 - 問合せパフォーマンス用の増加, 3-5
- ブロック転送
 - 分析, 4-8
- ブロック・モード変換
 - 統計, 4-5
- プロトコル
 - インターコネクツ, 4-2

へ

- 並行性
 - 挿入および更新, B-4
- 並列度 (DOP), 2-4

ほ

- ホット・ブロック
 - オブジェクト, 4-14
 - パフォーマンスの問題の解決, 3-3

め

- メッセージ要求カウンタ, 4-5
- メモリマップドIPC
 - キャッシュ・フュージョン, 4-2

ゆ

- ユーザー・セッション
 - 空きリスト・グループとの対応付け, B-6
- ユーザー・モードのIPC
 - キャッシュ・フュージョン, 4-2

よ

- 容量上の制限
 - ストレス・テストによる識別, 2-8

り

- リーフ・ブロック
 - 索引、パフォーマンスの問題, 4-9
- リソース
 - 識別子, A-16
 - 取得およびGCS, 3-6
 - 動的な移行, 2-3
 - 名前のフォーマット, A-16
- リソース共有システム, B-9
- 領域
 - インスタンスへの未割当て, B-8
 - エクステンツの割当て, B-9
 - 排他モードで使用不可能, B-7

ろ

- ローカル管理表領域, 3-2
- ロード・バランシング, 2-6
- ロック
 - GC_FILES_TO_LOCKS の設定による使用の判断, A-2
 - GC_FILES_TO_LOCKS を使用して設定, A-2
 - ロックを使用する場合, A-3
- ロックの設定, A-2
- 論理読取り
 - グローバル・キャッシュ・サービス (GCS), 4-13

わ

- ワークロード
 - Statspack の統計, 4-7
 - データ・ウェアハウス・アプリケーションでの分散, 2-6
- 割当て
 - エクステンツ、動的, A-11
 - キャッシュ・フュージョン・リソース, A-6, A-7
 - 自動, B-9