

Oracle9i Real Application Clusters

概要

リリース 2 (9.2)

2002 年 7 月

部品番号 : J06259-01

ORACLE®

Oracle9i Real Application Clusters 概要, リリース 2 (9.2)

部品番号 : J06259-01

原本名 : Oracle9i Real Application Clusters Concepts, Release 2 (9.2)

原本部品番号 : A96597-01

原本著者 : Mark Bauer

原本協力者 : David Austin, Wilson Chan, Sashikanth Chandraserkaran, Jonathan Creighton, Lisa Eldridge, Mitch Flatland, Rick Greenwald, Eugene Ho, Bill Kehoe, Raj Kumar, Kotaro Ono, Stefan Pommerenk, Rebecca Reitmeyer, Joao Rimoli, Daniel Semler, Vinay Srihari, Alok Srivastava, Tak Wang, Shari Yamaguchi, Tolga Yurek, Valarie Moore

Copyright © 1998, 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おります。

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に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に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	xi
------------	----

Real Application Clusters の新機能	xix
--------------------------------------	-----

第 I 部 クラスタ・データベース処理の基本

1 Real Application Clusters の概要

Real Application Clusters とは	1-2
Real Application Clusters の利点	1-3
全体的な所有コストの削減	1-3
拡張性の増大	1-3
高可用性	1-4
透過性	1-4
バッファ・キャッシュ管理	1-4
行ロック	1-5
マルチバージョン読込み一貫性	1-5
Recovery Manager、オンライン・バックアップおよびアーカイブ	1-5

2 Real Application Clusters システムのアーキテクチャ

クラスタ・データベース・システムのコンポーネントの概要	2-2
ノードとそのコンポーネント	2-3
クラスタ・インターコネクトおよびプロセス間通信（ノード間）	2-3
メモリー、インターコネクトおよびストレージ	2-4

高速な IPC インターコネクト	2-4
共有ディスク・ストレージとクラスタ・ファイル・システムのメリット	2-4

3 Real Application Clusters ソフトウェアのアーキテクチャ

オペレーティング・システム固有のクラスタウェア	3-2
Cluster Manager	3-2
Node Monitor	3-2
インターコネクト	3-3
Real Application Clusters の共有ディスク・コンポーネント	3-3
Real Application Clusters 固有のデーモンとインスタンス・プロセス	3-4
グローバル・サービス・デーモン	3-4
Real Application Clusters 固有のインスタンス・プロセス	3-4
グローバル・キャッシュとグローバル・エンキュー・サービス	3-6
アプリケーションに対する透過性	3-6
分散アーキテクチャにおけるグローバル・リソース・ディレクトリ	3-6
リソースのマスター化とアフィニティ	3-6
GCS と GES の Cluster Manager との相互作用	3-7

4 Real Application Clusters の拡張性

Real Application Clusters の拡張性機能	4-2
Real Application Clusters の利点を生かしたシステム・タイプ	4-2
トランザクション・システムと Real Application Clusters	4-2
データ・ウェアハウス・システムと Real Application Clusters	4-3
拡張性のレベル	4-3
ネットワークの拡張性	4-3
ネットワークの拡張性とクライアント / サーバー接続	4-4
オペレーティング・システムの拡張性	4-6

第 II 部 Real Application Clusters でのリソースの調整

5 Real Application Clusters リソースの調整

Real Application Clusters リソースの調整の概要	5-2
グローバル・リソース・ディレクトリの内容	5-2
Real Application Clusters の同期化プロセス	5-3

エンキュー	5-3
パスト・イメージ	5-3
リソースのモードとロール	5-4
リソースのモード	5-4
リソースのロール	5-4
グローバル・キャッシュ・サービスの動作	5-5
システム変更番号処理	5-6
Lamport SCN 生成	5-6

6 キャッシュ・フュージョンとグローバル・キャッシュ・サービス

キャッシュ・フュージョン処理の概要	6-2
複数ノードでの並行読み込み	6-2
異なるノードでの並行読み込みと書き込み	6-2
異なるノードでの並行書き込み	6-2
書き込みプロトコルとパスト・イメージの追跡	6-3
リソース制御、キャッシュ間転送およびキャッシュ一貫性	6-3
ブロック・アクセス・モードおよびバッファの状態	6-4
キャッシュ・フュージョンの使用例	6-5
変更されたブロックの変更操作要求	6-5
ディスクへのブロックの書き込み	6-6
Real Application Clusters でのリカバリとキャッシュ・フュージョン	6-8

7 グローバル・エンキュー・サービスによるリソースの調整

グローバル・エンキュー・サービスの処理	7-2
グローバル・エンキューの並行処理制御	7-2
グローバル・エンキュー・サービスで管理するリソース	7-3
ディクショナリ・キャッシュ・ロック	7-3
ライブラリ・キャッシュ・ロック	7-3

第 III 部 Real Application Clusters の実装

8 Real Application Clusters の記憶域上の考慮点

Real Application Clusters の記憶域の概要	8-2
Real Application Clusters のデータ・ファイル	8-2
Real Application Clusters のデータ・ファイルの検証	8-2
Real Application Clusters でのデータ・ファイルの追加	8-2
Real Application Clusters のパラメータ・ファイルの記憶域	8-3
サーバー・パラメータ・ファイルの位置	8-3
Real Application Clusters の REDO ログ・ファイルの記憶域	8-4
自動セグメント領域管理	8-5
Real Application Clusters での UNDO 領域の管理	8-5
プライベート・ロールバック・セグメントとパブリック・ロールバック・セグメント	8-5

9 Real Application Clusters 環境用の管理ツール

Real Application Clusters の管理の概要	9-2
Real Application Clusters のインストール、設定および構成の管理	9-2
Real Application Clusters の管理	9-3
Oracle Enterprise Manager	9-3
Database Configuration Assistant	9-4
サーバー制御ユーティリティ (SRVCTL)	9-5
グローバル・サービス・デーモンの管理コマンド	9-5
Real Application Clusters のパフォーマンス監視用の管理ツール	9-6
Oracle Enterprise Manager を使用したパフォーマンスの監視	9-6
Statspack を使用したパフォーマンスの監視	9-6
Real Application Clusters のバックアップとリカバリの管理	9-6

第 IV 部 高可用性および Real Application Clusters

10 Real Application Clusters における高可用性の概念と実践

高可用性について	10-2
高可用性のための Real Application Clusters の構成	10-2
クラスター・コンポーネントおよび高可用性	10-3
災害時計画	10-4
障害保護の検証	10-4

フェイルオーバーおよび Real Application Clusters	10-5
フェイルオーバーの基本	10-5
クライアント・フェイルオーバー	10-6
透過的アプリケーション・フェイルオーバーの使用	10-7
サーバー・フェイルオーバー	10-11
Real Application Clusters でのフェイルオーバー処理	10-13
障害の検出	10-13
クラスタ・メンバーシップの再編成	10-13
データベース・リカバリの実行	10-14
高可用性の構成	10-17
デフォルトの n ノード構成	10-17
高可用性の基本構成	10-17
高可用性の共有ノード構成	10-23
Real Application Clusters Guard II での完全なアクティブ構成	10-24
高可用性の配置	10-24

第 V 部 リファレンス

A 制限事項

互換性	A-2
制限付き SQL 文	A-2
データ・ファイルの最大数	A-2

B マルチブロック・ロック割当ての使用（オプション）

ロック設定を使用する場合	B-2
ロック設定を使用する方法	B-2
ロックの粒度	B-2
ロック管理	B-3

用語集

索引



2-1	クラスタ・データベース処理におけるクラスタ・コンポーネント	2-2
3-1	Real Application Clusters 固有のインスタンス・プロセス	3-5
6-1	変更されたブロックの変更操作要求	6-5
6-2	ディスクへのブロックの書込み	6-7
10-1	専用サーバー環境でのプライマリ / セカンダリ構成	10-19
10-2	専用サーバーのプライマリ / セカンダリ構成とノード障害	10-20
10-3	共有サーバー環境でのプライマリ / セカンダリ構成	10-21
10-4	共有サーバーのプライマリ / セカンダリ構成とノード障害	10-22
B-1	GC_FILES_TO_LOCKS 設定に対するブロックへのロックの割当て例	B-3

表

5-1 グローバル・キャッシュ・サービスのリソースのモード 5-4

はじめに

このマニュアルでは、すべてのオペレーティング・システム上で動作する Real Application Clusters の概要を説明します。このマニュアルは、『Oracle9i Real Application Clusters セットアップおよび構成』、『Oracle9i Real Application Clusters 管理』および『Oracle9i Real Application Clusters 配置およびパフォーマンス』を読む前に読んでください。

ただし、このマニュアルを読むには、シングル・インスタンスの Oracle データベース処理について理解しておく必要があります。詳細は、『Oracle9i データベース概要』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管理者およびアプリケーション開発者を対象としています。

このマニュアルの構成

このマニュアルは、5 部で構成されています。まず、Real Application Clusters に対するクラスタ・データベース処理の基本について説明します。次に、インスタンス間のリソース処理、Real Application Clusters の実装方法の基本、高可用性と Real Application Clusters について説明します。最後にリファレンス情報として、Real Application Clusters の実装上の制限、ロック設定の概要について説明します。また、用語集も記載しています。

第 I 部：「クラスタ・データベース処理の基本」

第 I 部では、Real Application Clusters の概要を説明し、そのアーキテクチャ、ソフトウェア・コンポーネントおよび拡張性機能について説明します。

第 1 章「Real Application Clusters の概要」

この章では、Real Application Clusters でのクラスタ・データベース処理の概要を説明します。

第 2 章「Real Application Clusters システムのアーキテクチャ」

この章では、クラスタ環境の特徴を表すシステム・コンポーネントおよび高レベルのアーキテクチャ・モデルについて説明します。

第 3 章「Real Application Clusters ソフトウェアのアーキテクチャ」

この章では、Real Application Clusters のソフトウェア・コンポーネントについて説明します。

第 4 章「Real Application Clusters の拡張性」

この章では、Real Application Clusters の拡張性機能について説明します。

第 II 部：「Real Application Clusters でのリソースの調整」

第 II 部では、インスタンス間のリソース調整およびグローバル・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) が実行する処理について説明します。

第 5 章「Real Application Clusters リソースの調整」

この章では、Real Application Clusters 環境で行われるインスタンス間の調整について説明します。

第 6 章「キャッシュ・フュージョンとグローバル・キャッシュ・サービス」

この章では、キャッシュ・フュージョンおよび GCS によって管理されるインスタンス間の調整アクティビティの詳細を説明します。

第 7 章「グローバル・エンキュー・サービスによるリソースの調整」

この章では、GES によって管理されるインスタンス間の調整アクティビティの詳細を説明します。

第 III 部：「Real Application Clusters の実装」

第 III 部では、Real Application Clusters の実装上の考慮点および管理コンポーネントについて説明します。

第 8 章「Real Application Clusters の記憶域上の考慮点」

この章では、Real Application Clusters アプリケーションの記憶域上の考慮点について説明します。

第 9 章「Real Application Clusters 環境用の管理ツール」

この章では、Real Application Clusters 環境用の管理コンポーネントについて説明します。

第 IV 部：「高可用性および Real Application Clusters」

第 IV 部では、Real Application Clusters を使用して高可用性を実装する方法を説明します。

第 10 章「Real Application Clusters における高可用性の概念と実践」

この章では、Real Application Clusters における高可用性の実装の概念と、その実践について説明します。

第 V 部：「リファレンス」

第 V 部では、Real Application Clusters に関するリファレンス情報を記載します。

付録 A「制限事項」

この付録では、Real Application Clusters の配置上の制限事項について説明します。

付録 B「マルチブロック・ロック割当ての使用（オプション）」

この付録では、オプションのロック設定を使用して、キャッシュ・フュージョンが提供するインスタンス間の調整を無効にする方法を説明します。

用語集

この用語集では、このマニュアルで使用する重要な用語を定義しています。

関連文書

- このマニュアルを読んだ後に、『Oracle9i Real Application Clusters セットアップおよび構成』、『Oracle9i Real Application Clusters 管理』および『Oracle9i Real Application Clusters 配置およびパフォーマンス』を読んでください。
- また、Oracle Real Application Clusters Guard I については、『Oracle9i Real Application Clusters Real Application Clusters Guard I - Concepts and Administration』を、Real Application Clusters Guard II については、『Oracle9i Real Application Clusters Guard II Concepts, Installation, and Administration』を参照してください。

詳細は、次の Oracle ドキュメントを参照してください。

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』
- Oracle Real Application Clusters Guard I のインストレーション・ガイド（プラットフォーム固有のインストレーション・ガイドが何種類かあります）
- 『Oracle9i Real Application Clusters Guard II Concepts, Installation, and Administration』

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 Guard II Concepts, Installation, and Administration』

クラスタ・データベース管理

- 『Oracle9i データベース概要』
- 『Oracle9i データベース管理者ガイド』
- 『Oracle Enterprise Manager 管理者ガイド』
- 『Oracle Enterprise Manager Oracle Diagnostics Pack スタート・ガイド』

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 を使用するには、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](#)

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、本文または用語集（あるいはその両方）で定義されている用語を示します。	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	固定幅フォントの大文字は、システムにより指定される要素を示します。この要素には、パラメータ、権限、データ型、 Recovery Manager キーワード、 SQL*Plus またはユーティリティ・コマンド、パッケージとメソッド、システム指定の列名、データベース・オブジェクト、データベース構造体、ユーザー名、およびロールがあります。	NUMBER 列に対してのみ、この句を指定できます。 BACKUP コマンドを使用すると、データベースのバックアップを作成できます。 USER_TABLES データ・ディクショナリ・ビューの TABLE_NAME 列を問い合わせてください。 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	固定幅フォントの小文字は、実行可能ファイル、ファイル名、ディレクトリ名およびサンプルのユーザー指定要素を示します。この要素には、コンピュータ名、データベース名、ネット・サービス名、接続識別子の他、ユーザー指定のデータベース・オブジェクトとデータベース構造体、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があります。 注意： 一部のプログラム要素には、大文字と小文字の両方が使用されます。この場合は、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。 このメソッドは JRepUtil クラスに実装されません。

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>Uold_release</i> .SQL を実行します。 <i>old_release</i> は、アップグレード前にインス トールしたリリースです。

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、そのうちの 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示します。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。	CREATE TABLE ... AS <i>subquery</i> ;
	<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。 	SELECT <i>col1</i> , <i>col2</i> , ... , <i>coln</i> FROM employees;
.	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。	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.

規則	意味	例
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記号は、示されているとおりに入力してください。	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
イタリック	イタリックの文字は、特定の値を指定する必要があるプレースホルダまたは変数を示します。	CONNECT SYSTEM/system_password DB_NAME = database_name
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。	SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;
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、列名またはファイル名などです。 注意： 一部のプログラム要素には、大文字と小文字の両方が使用されます。この場合は、記載されているとおりに入力してください。	SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjones IDENTIFIED BY ty3MU9;

Real Application Clusters の新機能

ここでは、Oracle9i Real Application Clusters リリース 2 (9.2) の新機能について説明し、追加情報の参照先を示します。次の項では、Oracle9i Real Application Clusters リリース 2 (9.2) の新機能について説明します。

- [Real Application Clusters リリース 2 \(9.2\) の新機能](#)

Real Application Clusters リリース 2 (9.2) の新機能

- Oracle Universal Installer および Database Configuration Assistant

Oracle Universal Installer および Database Configuration Assistant 用に、多数のクラスタ化の拡張機能が用意されています。また、XML DB および Oracle Data Mining 機能を使用するには、2 つの追加の表領域が必要です。

関連項目： Database Configuration Assistant の拡張機能の詳細は、『Oracle9i Real Application Clusters セットアップおよび構成』を参照してください。

- Database Upgrade Assistant (DBUA)

Database Migration Assistant は Database Upgrade Assistant (DBUA) に名称が変更され、Real Application Clusters をサポートします。

関連項目： DBUA の詳細は、『Oracle9i データベース移行ガイド』を参照してください。

- クラスタ・ファイル・システムのサポート

Real Application Clusters のクラスタ・ファイル・システムが、特定のプラットフォームで使えるようになりました。

関連項目： RAW パーティション表領域サイズ要件に関する補足の詳細は、『Oracle9i Real Application Clusters セットアップおよび構成』およびリリース 2 (9.2) の README ファイルを参照してください。また、クラスタ・ファイル・システムの詳細は、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。

- 動的なローカルとリモートのリスナー・パラメータ

SQL 文の ALTER SYSTEM SET を使用すると、LOCAL_LISTENER パラメータと REMOTE_LISTENER パラメータを動的に更新できます。さらに、Real Application Clusters データベースとの間でノードの追加または削除を行うと、Oracle はこれらのパラメータを動的に更新します。再構成時に、プロセス・モニター (PMON) プロセスは、インスタンス間でリスナー登録を行うために、クラスタ内のすべてのリスナーに関する再構成情報を記録します。

■ Real Application Clusters の Recovery Manager の拡張機能

Real Application Clusters の Recovery Manager (RMAN) の自動位置検索機能は、Real Application Clusters 構成のノードの中で、バックアップまたはリストアを行うファイルにアクセスできるノードを自動的に検出します。次の種類のファイルが自動検出されます。

- バックアップ・ピース（バックアップまたはリストア時）
- アーカイブ REDO ログ（バックアップ時）
- データ・ファイルまたは制御ファイルのコピー（バックアップまたはリストア時）

関連項目： Recovery Manager の詳細は、『Oracle9i Recovery Manager ユーザーズ・ガイド』および『Oracle9i Recovery Manager リファレンス』を参照してください。Real Application Clusters 環境での Recovery Manager の使用方法については、『Oracle9i Real Application Clusters 管理』を参照してください。

■ Real Application Clusters Guard II

Real Application Clusters Guard II は、包括的なワークロード管理をサポートし、Real Application Clusters データベースとそのアプリケーションの高可用性を維持します。Real Application Clusters Guard II は、サービス名の概念に基づいて、アプリケーションの負荷を転嫁します。このため、Real Application Clusters Guard II は、サービス・レベルとデータベース・サービスを使用するアプリケーションに基づいてワークロード管理をサポートします。

サービス名は高可用性のために採用されました。これによって、サービス名を実装するためにアプリケーションを変更する必要はありません。さらに、サービス名によって、サービスを提供するデータベース・インスタンスに位置の透過性が提供されます。サービス名によって、ワークロードの構成、操作およびリカバリを簡素化する単一システムのイメージが可能となります。

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

第 I 部

クラスタ・データベース処理の基本

第 I 部では、Real Application Clusters の処理の基本について説明します。第 I 部に含まれる章は、次のとおりです。

- 第 1 章「Real Application Clusters の概要」
- 第 2 章「Real Application Clusters システムのアーキテクチャ」
- 第 3 章「Real Application Clusters ソフトウェアのアーキテクチャ」
- 第 4 章「Real Application Clusters の拡張性」

Real Application Clusters の概要

この章では、クラスタ・データベース処理および Real Application Clusters の機能について説明します。この章の内容は、次のとおりです。

- [Real Application Clusters とは](#)
- [Real Application Clusters の利点](#)

Real Application Clusters とは

Real Application Clusters は、相互に接続された複数のコンピュータの処理能力を統合して利用します。Real Application Clusters ソフトウェアと**クラスタ**というハードウェアの集合によって、各コンポーネントの処理能力が統合され、強力な単一のコンピューティング環境が構築されます。

Real Application Clusters は、従来の Oracle クラスタ対応ソフトウェア製品の機能を超える拡張性と高可用性を備えた画期的なクラスタ・ソフトウェア・アーキテクチャです。Real Application Clusters は、すべての種類のシステムに対して大きなメリットを提供します。Real Application Clusters に追加された機能を使用すると、すべてのシステムとアプリケーションで、クラスタ環境を効果的に利用できます。

Real Application Clusters を使用すると、パフォーマンスとスループットが向上し、**高可用性**が得られます。ただし、Real Application Clusters を配置する前に、Real Application Clusters の処理を理解しておく必要があります。

Real Application Clusters 環境でのアクティブなすべてのインスタンスは、共有データベースに対して同時にトランザクションを実行できます。Real Application Clusters は、各インスタンスの共有データへのアクセスを調整し、データの一貫性と整合性を提供します。

複数のクラスタの能力を統合して利用することで、大きなメリットが生まれます。1つの大きな作業を小さな作業に分割して複数のノードに分散することで、1つのノードで作業全体を処理する場合よりも、高速にかつ効率的に作業を完了できます。クラスタ処理によるパフォーマンス向上によって、増大するワークロードと急速に増加するユーザーに対応することもできます。

Real Application Clusters では、増大するデータ処理要求に対応するために、アプリケーション・コードを変更することなくアプリケーションを拡張できます。ノードや記憶域などのリソースを追加すると、Real Application Clusters はそれらのリソースの処理能力を、個々のコンポーネントの上限を超えて拡張します。

読取り専用データにアクセスするデータ・ウェアハウス・アプリケーションは、Real Application Clusters に最適です。また、Real Application Clusters は、**オンライン・トランザクション処理** (OLTP) システムの他に、読み書き両用と読取り専用アプリケーションの両方の特性を組み合わせたハイブリッド・システムを適切に管理します。Real Application Clusters は、堅牢な高可用性ソリューションの重要なコンポーネントとしても機能し、わずかな停止時間または停止時間なしで障害に対処します。

Real Application Clusters の利点

ここでは、Real Application Clusters の次の利点について説明します。

- 全体的な所有コストの削減
- 拡張性の増大
- 高可用性
- 透過性
- バッファ・キャッシュ管理
- 行ロック
- [Recovery Manager](#)、[オンライン・バックアップ](#)および[アーカイブ](#)

全体的な所有コストの削減

Real Application Clusters は、他のクラスタ・データベース製品よりも効率的に全体的な所有コストを削減します。これには、Real Application Clusters アーキテクチャが提供する単一システムのイメージが大きく寄与しています。

Real Application Clusters によるシステム統合によって、使用するソフトウェアやコンピュータ機器を他のクラスタ・データベース製品に比較して大幅に削減できます。さらに、Real Application Clusters は、高可用性と拡張性が要求されるあらゆる構成に対して配置できます。また、Real Application Clusters と相互に作用する包括的で使いやすい管理ツールによって、プロジェクト期間中の管理オーバーヘッドを大幅に削減できるため、全体的なコストを低く抑えることができます。

拡張性の増大

スケーラブルな環境では、ノードの追加によってパフォーマンスが改善され、容量が増加します。一部のプラットフォームでは、クラスタの実行中にノードを動的に追加できます。

Real Application Clusters が実際にサポートできるノードの数は、従来の実装より大幅に増加しています。主に高可用性を目的とした小規模なシステムは、2 ノードのみで構成されます。一方、大規模な構成では、32 ～ 64 ノードになることもあります。

注意： クラスタにノードを動的に追加できるかどうかは、ご使用のクラスタウェアの機能によって異なります。Oracle クラスタウェアを使用している場合は、ほとんどのプラットフォームでノードとインスタンスの動的な追加が可能です。

関連項目： 拡張性の詳細は、[第 4 章「Real Application Clusters の拡張性」](#)を参照してください。

高可用性

高可用性システムとは、障害時でも、一貫性のある継続したサービスを提供する冗長コンポーネントを備えたシステムを指します。最も可用性の高い構成では、互いのノードが独立しているため、1つのノードで障害が発生しても、システム全体に影響することはありません。

透過性

透過性の概念では、Real Application Clusters 環境はシングル・インスタンスの Oracle データベース構成と機能的に同等です。つまり、アプリケーションがシングル・インスタンスの Oracle 構成で効率的に動作している場合は、コードを変更せずにそのアプリケーションを Real Application Clusters に配置できます。

キャッシュ・フュージョン機能は、1つのクラスタ内の複数のバッファ・キャッシュの融合化を意味し、Real Application Clusters 環境の管理を簡素化します。また、Real Application Clusters とキャッシュ・フュージョンを使用すると、容量計画の実行が不要になります。

透過性は、アプリケーションとシステムの両方のレベルでの効率的なリソース使用によっても実現できます。たとえば、Real Application Clusters が自動的にリソース構成を行うため、データ・アクセスのパターンを調べてリソース構成を実行するという時間を要する作業は必要ありません。

バッファ・キャッシュ管理

Oracle では、データ・ブロック情報などのリソースをメモリー内に常駐するバッファ・キャッシュに格納します。この情報をローカルに格納することで、データベース操作とディスク I/O が削減されます。各インスタンスには独自のメモリーがあるため、Real Application Clusters では、ディスク I/O を最小化すると同時に、複数ノードのバッファ・キャッシュを調整します。この調整によって、パフォーマンスが最適化され、有効なメモリーがクラスタ・データベース内のすべてのメモリー合計とほぼ同じになるまで拡張されます。

このために、Real Application Clusters は**グローバル・キャッシュ・サービス**（GCS）を使用して複数のバッファ・キャッシュ間で操作を調整し、Oracle の高パフォーマンス機能を最適化します。また、**グローバル・エンキュー・サービス**（GES）は、ノード間の通信を管理して同期化を支援します。GCS と GES の詳細は、このマニュアルの第 II 部で説明します。

関連項目： バッファ・キャッシュの詳細は、『Oracle9i データベース概要』を参照してください。

行ロック

Oracle の行ロックおよびマルチバージョン読込み一貫性によって、高度な並行性およびスループットが提供されます。行ロックによって、同じデータ・ブロック内の異なる行を変更する複数のトランザクションは、他のトランザクションのコミットを待機する必要がなくなります。

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

マルチバージョン読込み一貫性

マルチバージョン読込み一貫性によって、読込み操作と書き込み操作が相互にブロックしないことが保証されます。マルチバージョン読込み一貫性によって、まだコミットされていないトランザクションで変更されたブロックのスナップショットまたは読込み一貫性バージョンが作成されます。この方法には、次の主要な利点があります。

- データを変更するユーザーは読込み操作による読込みを妨げないため、読込み操作でのリソース待機は不要です。
- データのスナップショットは、特定の時点から表示できます。

Recovery Manager、オンライン・バックアップおよびアーカイブ

Real Application Clusters は、**Recovery Manager (RMAN)** および **Oracle Enterprise Manager** のすべての機能をサポートしています。Real Application Clusters は、シングル・インスタンスの Oracle データベースで使用可能な Oracle のバックアップ機能とアーカイブ機能もサポートします。このサポートには、データベース全体または個々の表領域に関するオンラインとオフラインの両方のバックアップが含まれます。

ARCHIVELOG モードで Oracle を操作している場合、ログ・ファイルが満杯になると、Oracle は、情報を上書きする前に、そのログ・ファイルをアーカイブ・ログ・ファイルに変換します。Real Application Clusters では、各インスタンスがその REDO ログ・ファイルを自動的にアーカイブするか、1 つ以上のインスタンスが、クラスター・データベースの一部またはすべてのインスタンスの REDO ログ・ファイルをアーカイブできます。

データベースを NOARCHIVELOG モードで操作している場合に可能なバックアップは、オフライン・バックアップのみです。データの消失を防ぐため、ARCHIVELOG モードの使用をお勧めします。

関連項目： Real Application Clusters での Recovery Manager については、『Oracle9i Real Application Clusters 管理』を参照してください。

第 I 部の残りの章では、Real Application Clusters のアーキテクチャとその拡張性機能について説明します。

Real Application Clusters システムの アーキテクチャ

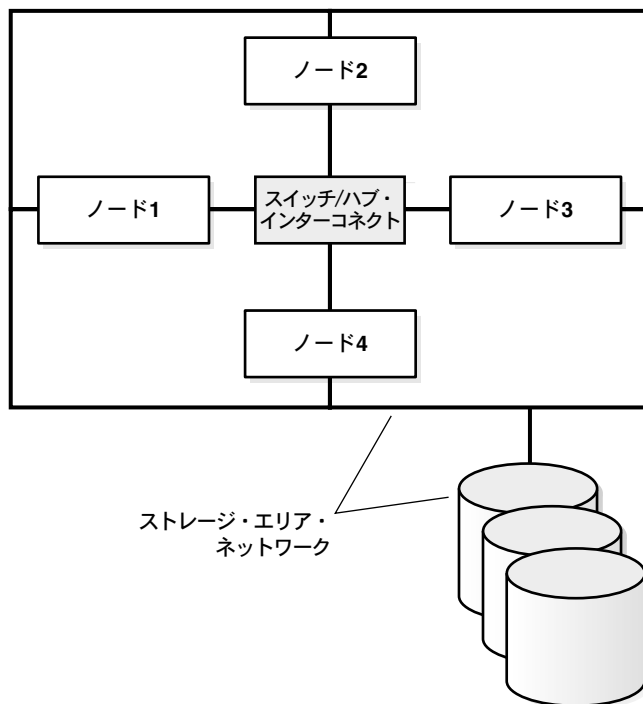
この章では、ほとんどのクラスタ・データベース環境の特徴を表すシステム・コンポーネントとアーキテクチャ・モデルについて説明します。また、ノード用のハードウェア、およびノードをクラスタ・データベースに結合するハードウェアとソフトウェアについても説明します。この章の内容は、次のとおりです。

- クラスタ・データベース・システムのコンポーネントの概要
- メモリー、インターコネクトおよびストレージ
- 高速な IPC インターコネクト
- 共有ディスク・ストレージとクラスタ・ファイル・システムのメリット

クラスタ・データベース・システムのコンポーネントの概要

クラスタ・データベースは、**インターコネクト**でリンクされた2つ以上のノードで構成されます。インターコネクトは、クラスタ・データベース内の各**ノード**間の通信経路として機能します。各 Oracle **インスタンス**では、各インスタンスでの共有リソースの使用を同期化するメッセージ機能に、このインターコネクトが使用されます。また、Oracle は、複数のインスタンスが共有するデータ・ブロックの転送でもインターコネクトを使用します。主要な共有リソースは、すべてのノードがアクセスするデータ・ファイルです。図 2-1 に、インターコネクトがクラスタ・データベース内のノードにリンクする方法、クラスタがストレージ・デバイス上の共有データ・ファイルにアクセスする方法の概要図を示します。

図 2-1 クラスタ・データベース処理におけるクラスタ・コンポーネント



クラスタとそのインターコネクトは、ストレージ・エリア・ネットワークによって、ストレージ・デバイス、つまり**共有ディスク・サブシステム**にリンクされます。次の各項では、ノードとインターコネクトの詳細を説明します。

- ノードとそのコンポーネント
- クラスタ・インターコネクトおよびプロセス間通信（ノード間）

ノードとそのコンポーネント

ノードには、次の主要コンポーネントがあります。

- CPU – コンピュータのメイン・メモリーからの読み込み、またはメイン・メモリーへの書き込みを行う、コンピュータのメイン処理コンポーネント。
- メモリー – プログラムの実行とデータのバッファに使用されるコンポーネント。
- インターコネクト – ノード間の通信リンク。
- ストレージ – データを格納するデバイス。通常は、その内容を変更するために読み込み / 書き込みトランザクションでアクセスする必要がある永続ストレージ。

これらのコンポーネントは、多様な構成で購入できます。これらのコンポーネントの構成によって、ノードによるメモリーとストレージへのアクセス方法が決定します。

注意： Real Application Clusters は、Real Application Clusters データベースでの使用が認定されている構成とともに配置することをお勧めします。

クラスタ・インターコネクトおよびプロセス間通信（ノード間）

Real Application Clusters では、ノード間通信に、高速な**プロセス間通信**（IPC）コンポーネントが使用されます。IPC は、インスタンス間でメッセージを転送するために、Real Application Clusters 環境に必要なプロトコルおよびインタフェースを定義します。メッセージは、このインタフェースにおける通信の基本単位です。核となる IPC 機能は、非同期のキュー・メッセージング・モデルで構築されています。

メモリー、インターコネクトおよびストレージ

一般的に、すべてのクラスタ・データベースは、同じ方法で CPU を使用します。ただし、メモリー、ストレージおよびインターコネクトは、様々な目的に対して様々な構成で配置できます。Real Application Clusters を配置するアーキテクチャは、処理の目的によって異なります。

クラスタ・データベース内の各ノードには、1 つ以上の CPU があります。通常、複数の CPU を持つノードがメイン・メモリーを共有するように構成されます。これによって、スケーラブルなシステムの配置が可能になります。

高速な IPC インターコネクト

高速**プロセス間通信** (IPC) インターコネクトは、クラスタ内の複数のノードをリンクする、高帯域幅で待機時間の短い通信機能です。インターコネクトは、メッセージおよび他のクラスタ通信を送信し、各ノードのリソースへのアクセスを調整します。

インターコネクトには、イーサネット、**光ファイバ分散データ・インタフェース** (FDDI) または他の独自のハードウェアを使用できます。また、プライマリ・インターコネクトに障害が発生した場合に備えて、バックアップ・インターコネクトをインストールすることも検討してください。インターコネクトのバックアップを準備すると、可用性が向上し、インターコネクトが単一の致命的な障害箇所になる可能性を減らすことができます。

Real Application Clusters は、ユーザー・モードの IPC およびメモリーマップド IPC をサポートします。これらのタイプの IPC によって、CPU の消費と IPC の待機時間が大幅に削減されます。

共有ディスク・ストレージとクラスタ・ファイル・システムのメリット

Real Application Clusters では、インスタンスがデータベースに同時アクセスするために、すべてのノードが共有ディスクに同時にアクセスする必要があります。共有ディスク・サブシステムは、使用するオペレーティング・システムに基づいて実装されます。クラスタ・ファイル・システムを使用するか、またはファイルを RAW デバイス上に配置できます。クラスタ・ファイル・システムによって、Real Application Clusters のインストールと管理が大幅に簡素化されます。

通常、Real Application Clusters のメモリー・アクセス構成は均一です。これは、クラスタ内の各ノードがメモリーにアクセスするためのオーバーヘッドは、同じであることを意味します。ただし、典型的なストレージ・アクセス構成には、均一と不均一の両方があります。使用するストレージ・アクセス構成は、メモリー構成とは関係がありません。

メモリー構成と同様に、ほとんどのシステムが均一ディスク・アクセスを Real Application Clusters データベースで使用します。クラスタ・データベースの均一ディスク・アクセス構成によって、ディスク・アクセス管理が簡素化されます。

Real Application Clusters ソフトウェアの アーキテクチャ

この章では、**Oracle Real Application Clusters** 固有のアーキテクチャ・コンポーネントについて説明します。これらのコンポーネントの一部は、**Oracle** データベース・ソフトウェアで提供されますが、他にベンダー固有のコンポーネントもあります。この章では、次の **Real Application Clusters** のコンポーネントについて説明します。

- オペレーティング・システム固有のクラスタウェア
- **Real Application Clusters** の共有ディスク・コンポーネント
- **Real Application Clusters** 固有のデーモンとインスタンス・プロセス
- グローバル・キャッシュとグローバル・エンキュー・サービス

オペレーティング・システム固有のクラスタウェア

Real Application Clusters の処理では、**オペレーティング・システム依存 (OSD) クラスタウェア**を、オペレーティング・システムへのアクセスとクラスタ関連のサービス処理（インスタンスの起動と停止に関する通信情報など）で使います。UNIX オペレーティング・システム用の OSD クラスタウェアはベンダーが提供し、Windows NT および Windows 2000 オペレーティング・システム用の OSD クラスタウェアは Oracle が提供します。OSD には、次のサブコンポーネントがあります。

- **Cluster Manager**
- **Node Monitor**
- **インターコネクト**

Cluster Manager

Cluster Manager (CM) は、**インターコネクト**を経由して転送されるノード間のメッセージ機能を監視し、ノード間の操作を調整します。**Cluster Manager** は、クラスタとノードのグローバル・ビュー、およびそのメンバーであるインスタンスも提供します。**Cluster Manager** は、クラスタ・メンバーシップも制御します。

Node Monitor

Cluster Manager には、**Node Monitor** という機能があります。**Node Monitor** は、ノード、インターコネクトのハードウェアとソフトウェア、共有ディスクも含めて、クラスタ内の各**リソース**の状態をポーリングします。Oracle が提供する Windows 用の **Cluster Manager** では、**Node Monitor** は Oracle インスタンスもポーリングします。

Cluster Manager は、クラスタ・リソースの状態の変化を、クライアントと Oracle サーバーに知らせます。これは、別のインスタンスが **Cluster Manager** に登録された場合、または既存のインスタンスが切断された場合に、**Real Application Clusters** がクラスタ・データベースを再構成してクラスタ・メンバーシップを管理しているためです。

また、**Node Monitor** は、**Cluster Manager** に対して次の作業も行います。

- ノード管理インタフェース・モジュールを提供します。
- クラスタ全体にノード・メンバーシップの共通ビューを提供することによって、ノードのメンバーシップ状態を検出および追跡します。
- アクティブ・ノードの状態の変化を検出および診断し、これらの変更イベントに関する情報を通知します。

インターコネクト

プロセス間通信（IPC）ソフトウェア、つまりインターコネクトは、第2章で説明するとおり、OSD の主要なコンポーネントの1つです。IPC は、ノード間のメッセージ機能を制御します。Real Application Clusters では、インスタンス間のデータ・ブロック転送にも IPC が使用されます。

Real Application Clusters の共有ディスク・コンポーネント

Real Application Clusters データベースには、シングル・インスタンスの Oracle データベースと同じコンポーネントがあります。これらのコンポーネントには、1つ以上の制御ファイル、一連のオンライン REDO ログ・ファイル、アーカイブ・ログ・ファイル（オプション）、データ・ファイルなどが含まれます。このため、制御ファイルとデータ・ファイルごと、およびすべての REDO ログ・グループのオンライン REDO ログ・メンバーごとに、共有ディスクへのアクセスを提供する必要があります。また、推奨する自動 UNDO 管理機能を使用するには、UNDO 表領域データ・ファイル用に共有ディスクを構成する必要があります。

Windows NT および Windows 2000 の場合のみ、Oracle がクラスタ構成情報を格納する投票ディスクまたはクォーラム・ディスクへのアクセスを用意する必要があります。このディスクは、クラスタ・ファイル・システム（CFS）上または RAW デバイス上に配置できます。Node Monitor は、この定数ディスクの構成情報を使用して、クラスタ構成を管理します。

Oracle の構成ツールと管理ツールでは、共有ディスクに格納されているクラスタ構成データへのアクセスも必要です。Database Configuration Assistant（DBCA）、Oracle Enterprise Manager（EM）およびサーバー制御ユーティリティ（SRVCTL）を使用するには、共有ディスク・リソースを構成する必要があります。Windows NT および Windows 2000 では、投票ディスクと構成データは同じディスク・リソースを共有します。

サーバー・パラメータ・ファイルを使用すると、Real Application Clusters でのパラメータ・ファイル管理が簡素化されます。グローバルとインスタンス固有のパラメータ設定値を1つのファイルで管理するには、このファイルを共有ディスクに格納します。

関連項目： 表領域の要件については、『Oracle9i Real Application Clusters セットアップおよび構成』を参照してください。Real Application Clusters でのサーバー・パラメータ・ファイルの使用の詳細は、『Oracle9i Real Application Clusters 管理』を参照してください。

Real Application Clusters 固有のデーモンとインスタンス・プロセス

ここでは、Real Application Clusters 固有のデーモンとインスタンス・プロセスについて、次の各項で説明します。

- グローバル・サービス・デーモン
- Real Application Clusters 固有のインスタンス・プロセス

グローバル・サービス・デーモン

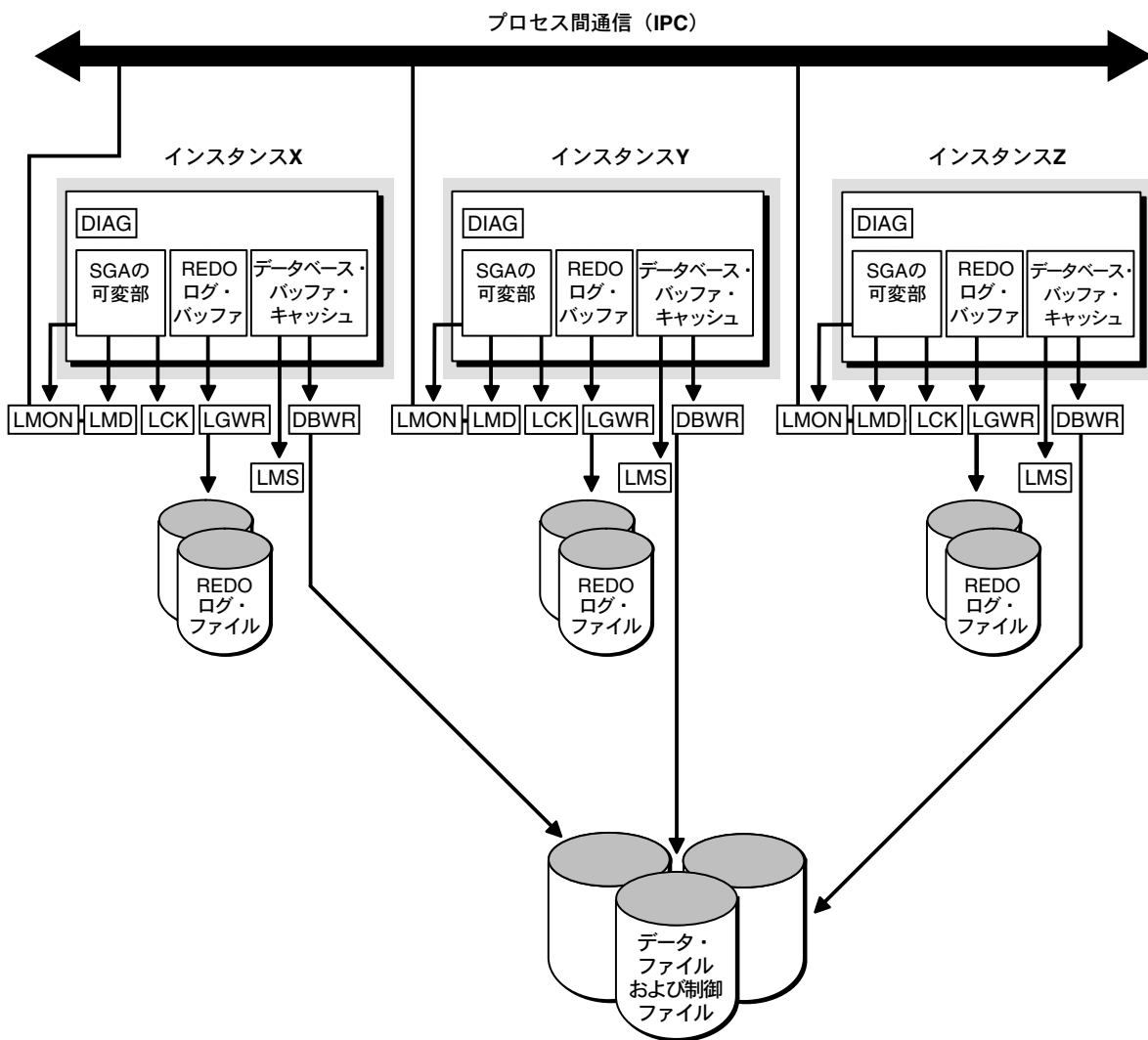
グローバル・サービス・デーモン（GSD）は各ノードで実行され、1つのノードで1つのGSDプロセスが実行されます。GSDはCluster Managerと協調して、DBCA、EM、SRVCTLユーティリティなどのクライアントから要求を受信し、インスタンスの起動や停止などの管理ジョブ・タスクを実行します。GSDはOracleインスタンスのバックグラウンド・プロセスではないため、Oracleインスタンスとともに起動されません。

Real Application Clusters 固有のインスタンス・プロセス

Real Application Clusters データベースには、プロセス・モニター（PMON）、データベース・ライター（DBWR n ）、ログ・ライター（LGWR）など、シングル・インスタンスのOracleデータベースと同じプロセスがあります。これに加えて、図 3-1 に示すとおり、Real Application Clusters 固有のプロセスがあります。これらのプロセスおよび作成されるトレース・ファイルの正確な名前は、プラットフォームによって異なります。

- グローバル・キャッシュ・サービス・プロセス（LMS n ）は、リモート・インスタンスへのメッセージ・フローを制御し、グローバル・データ・ブロック・アクセスを管理します。 n の範囲は0～9で、この数はメッセージの通信量によって異なります。また、LMS n プロセスは、異なるインスタンスのバッファ・キャッシュ間でブロック・イメージを転送します。このプロセスは、キャッシュ・フュージョン機能の一部です。
- グローバル・エンキュー・サービス・モニター（LMON）は、クラスタ全体のグローバル・エンキューとリソースを監視し、グローバル・エンキューのリカバリ操作を実行します。エンキューは、行の更新をシリアル化する複数のメモリー構造によって共有されます。
- グローバル・エンキュー・サービス・デーモン（LMD）は、グローバル・エンキューとグローバル・リソースへのアクセスを管理します。各インスタンス内で、LMDプロセスは、受信するリモート・リソース要求を管理します。
- ロック・プロセス（LCK）は、ライブラリや行キャッシュ要求など、キャッシュ・フュージョン以外のリソース要求を管理します。
- 診断用デーモン（DIAG）は、インスタンス内で発生したプロセス障害に関する診断データを獲得します。このデーモンは自動的に操作され、アラート・ログ・ファイルを更新して、実行するアクティビティを記録します。

図 3-1 Real Application Clusters 固有のインスタンス・プロセス



グローバル・キャッシュとグローバル・エンキュー・サービス

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)** は、共有データベースおよびデータベース内の共有リソースへの同時アクセスを調整する、**Real Application Clusters** の統合コンポーネントです。これらのサービスによって、一貫性とデータの整合性が維持されます。各インスタンス上の GCS と GES は、**Cluster Manager** と同様に、IPC を使用してインスタンス間およびクラスタ内の通信を行います。ここでは、GCS と GES に関する次の機能について説明します。

- アプリケーションに対する透過性
- 分散アーキテクチャにおけるグローバル・リソース・ディレクトリ
- リソースのマスター化とアフィニティ
- GCS と GES の **Cluster Manager** との相互作用

アプリケーションに対する透過性

GCS と GES によって実行されるリソースへのアクセス調整は、アプリケーションに対して透過的です。**Real Application Clusters** のアプリケーションでは、シングル・インスタンスの Oracle データベースと同様の並行処理メカニズムが使用されます。

分散アーキテクチャにおけるグローバル・リソース・ディレクトリ

GCS と GES は **グローバル・リソース・ディレクトリ** を管理して、リソースに関する情報を記録します。**グローバル・リソース・ディレクトリ** はメモリーに常駐し、クラスタ全体に分散されて、アクティブなすべてのインスタンスで使用できます。この分散アーキテクチャでは、各ノードがディレクトリ内の情報管理に関連します。この分散方式を採用することで、**フォルト・トレランス** が実現し、実行時のパフォーマンスが向上します。

GCS と GES によって、複数のノードに障害が発生した場合でも、**グローバル・リソース・ディレクトリ** の整合性が維持されます。共有データベースは、リカバリの完了後に 1 つ以上のインスタンスがアクティブであれば、常にアクセス可能です。また、**リソース・ディレクトリ** の **フォルト・トレランス** によって、**Real Application Clusters** インスタンスは、いつでも、どのような順序でも、起動および停止できます。

リソースのマスター化とアフィニティ

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** は、クラスタ内の各リソースに関する情報を管理します。GCS と GES は、1 つのインスタンスを指定して、特定リソースに関するすべての情報を管理します。このインスタンスを **リソース・マスター** といいます。GCS は、リソースのマスター化を定期的に評価し、データのアクセス・パターンに基づいて **リソース・マスター** を変更します。これによって、ネットワークの通信量が削減され、リソース取得時間も短縮されます。

GCS と GES の Cluster Manager との相互作用

GCS と GES は、Cluster Manager から独立して動作します。ただし、この 2 つのサービスは、クラスタ内のインスタンスの状態に関する正確な情報については Cluster Manager に依存しています。特定のインスタンスから必要な情報を入手できない場合、応答のないインスタンスは停止します。これによって、各インスタンスでは、共有ディスク・アクセスの調整で、他のすべてのアクティブなインスタンスが考慮されるため、Real Application Clusters データベースの整合性が保証されます。

Real Application Clusters の拡張性

この章では、Real Application Clusters の拡張性機能について説明します。この章の内容は、次のとおりです。

- [Real Application Clusters の拡張性機能](#)
- [Real Application Clusters の利点を生かしたシステム・タイプ](#)
- [拡張性のレベル](#)

Real Application Clusters の拡張性機能

Real Application Clusters の実装によって、アプリケーションのパフォーマンスを向上させ、高可用性を維持するいくつかの機能を使用できます。次のような機能があります。

- 使用率が高いときは、複数サーバーの接続を制御して、ノード間でワークロードを均衡化します。
- クライアントと Real Application Clusters データベース間で、フォルト・トレラント接続を持続します。

事前に構成されたデータベースの 1 つとともに Real Application Clusters をインストールした場合、**Database Configuration Assistant (DBCA)** は、これらのほとんどの機能を自動的に構成します。

関連項目： [拡張性機能を手動で構成する方法](#)については、『Oracle9i Net Services 管理者ガイド』を参照してください。

Real Application Clusters の利点を生かしたシステム・タイプ

ここでは、システム・タイプで Real Application Clusters の利点が生かされている背景を説明します。内容は次のとおりです。

- [トランザクション・システムと Real Application Clusters](#)
- [データ・ウェアハウス・システムと Real Application Clusters](#)

注意： ご使用のアプリケーションが適切に設計され、シングル・インスタンスの Oracle データベース上で効率的に稼動している場合、そのアプリケーションは Real Application Clusters 上でも適切に拡張できます。

トランザクション・システムと Real Application Clusters

トランザクション・システムの特徴は、データベースを更新する比較的短いトランザクションにあります。[トランザクション・システム](#)のアプリケーションには、E-Business や従来のオンライン・トランザクション処理 (OLTP) システムが含まれ、Real Application Clusters 上で適切に実行されます。これは、Real Application Clusters が優れたスケールアップを備えているためです。キャッシュ・フュージョンによって、オーバーヘッドを発生させずに複数のインスタンスで更新を行うことができます。

トランザクション・システムは、ユーザーがシステムに追加接続すると、さらに多くのトランザクションを管理するため、クラスタにノードを動的に追加することで、容量を増加できます。ただし、適切に設計されていないシステムの場合は、シングル・インスタンスの Oracle データベースまたは Real Application Clusters データベースのいずれでも拡張性が低くなります。つまり、不適切なアプリケーションの設計の場合は、要求の増加に従ってパフォーマンスが低下する可能性があります。

データ・ウェアハウス・システムと Real Application Clusters

更新アクティビティが少ないアプリケーションでは、追加のオーバーヘッドなしで異なるインスタンスを介してデータベースにアクセスできるため、データ・ウェアハウス・システムは Real Application Clusters 上で適切に実行されます。データ・ブロックを変更しない場合、複数のインスタンスが同じブロックをバッファ・キャッシュに読み込むことができ、追加の I/O なしで、そのブロックの問合せを実行できます。通常、データ・ウェアハウス・システムは、スケールアップの恩恵を受け、さらにスピードアップする場合があります。

Real Application Clusters での Oracle Parallel Execution

Real Application Clusters は、パラレル実行のためのフレームワークを提供します。パラレル実行によって大きな SQL 文を複数のインスタンス間に分散できるため、Real Application Clusters ではパラレル実行が効率的に動作します。トランザクションは、複数の CPU で実行されるため、処理速度が速くなります。

Real Application Clusters では、Oracle が、パラレル実行サーバー・プロセスを 1 つのインスタンス上でのみ実行するか、複数のインスタンス上で実行するかを実行時に決定します。通常、使用可能なリソースが十分ある場合、Oracle が使用するのは 1 つのインスタンスのみです。これによって、インスタンス間のメッセージ通信量が削減されます。

拡張性のレベル

クラスタ・データベースの実装には、この項で説明するように、次のレベルで最適化された拡張性が必要です。

- ネットワークの拡張性
- ネットワークの拡張性とクライアント / サーバー接続
- オペレーティング・システムの拡張性

注意： 適切に設計されていないアプリケーションは、システムで可能な拡張性を十分に使用できない場合があります。同様に、アプリケーションに拡張性があっても、拡張性のないハードウェアにアプリケーションを配置すると、最適なパフォーマンスは得られません。

ネットワークの拡張性

インターコネクトは、クラスタの拡張性における主要因です。それは、高速バスか低速なイーサネット接続かにかかわらず、すべてのシステムは CPU と接続する必要があるためです。そして、インターコネクトの帯域幅および待機時間が、ハードウェアの拡張性を決定します。

帯域幅および待機時間機能によるネットワークの拡張性

ハードウェアの拡張性で要求されるのは、非常に短い待機時間です。ほとんどのインターコネクには、Real Application Clusters のノード間処理に対する十分な帯域幅があります。パラレル実行など、ノード間の他の操作も帯域幅の大きさに依存しています。

ネットワークの拡張性とクライアント/サーバー接続

Real Application Clusters でのクライアント/サーバー接続は、複数のサブコンポーネントを使用して確立されます。クライアントは、ターゲット・ノードに常駐するリスナー・プロセスに接続要求を送信します。リスナーは、クライアントが特定のノードにアクセスできるように、いくつかの要因に基づいてクライアントに接続を許可します。リスナーは、次の各項で説明するように、接続オプションの構成方法に基づいて接続を許可します。

- [接続時ロード・バランシング機能](#)
- [向上したネットワークの拡張性と共有サーバー機能](#)

接続時ロード・バランシング機能

複数のインスタンスおよびディスパッチャのある Real Application Clusters 環境では、接続時ロード・バランシング機能によって、非常に優れた効果を得られます。接続時ロード・バランシングは、同じサービスに対する様々なインスタンスおよび共有サーバー・ディスパッチャ間の接続の数を均衡化することによって、パフォーマンスを改善します。ご使用のアプリケーションがパーティション化されている場合は、この機能を使用しない方がよい場合があります。

リモート・リスナーにクロス登録するサービス登録の機能によって、リスナーは常にすべてのインスタンスを把握しています。これによって、リスナーは、特定のサービスに対して受信したクライアント要求を、最も負荷の少ないインスタンスとディスパッチャに送信できます。また、サービス登録によって、共有サーバーの有無に関係なく使用できる、接続時フェイルオーバーとクライアント・ロード・バランシングが容易になります。さらに、使用している Real Application Clusters データベースに対してノードの追加や削除を行うと、接続時ロード・バランシングを制御する初期化パラメータが動的に更新されます。

複数リスナーに対する接続時フェイルオーバー サービス登録では、インスタンスが起動しているか停止しているかがリスナーに通知されます。このため、複数のリスナーがサービスをサポートしている場合に、接続時フェイルオーバーを使用できます。この機能を使用するには、最初のリスナーへの接続に失敗した場合は、クライアント要求を異なるリスナーにフェイルオーバーするようにクライアントを構成します。この場合、クライアントがリスナーへの接続に成功するまで再接続が試行されます。インスタンスで障害が発生した場合、リスナーはネットワーク・エラーを戻します。

複数リスナーに対するクライアント・ロード・バランシングと接続時ロード・バランシング クライアント・ロード・バランシングと接続時ロード・バランシングが、Oracle Net に用意されています。2つ以上のリスナーが1つのサービスをサポートする場合、クライアントは接続要求を様々なリスナーに対してランダムに送ることができます。ランダムな接続要求によって、負荷を分散し、単一のリスナーへの過剰な負荷を回避できます。

ロード・バランシングの他に、クライアントでは、ランダムに選択されたリスナーに接続できない場合には、クライアントの接続要求を特定のリスナーに自動的にフェイルオーバーするように指定することもできます。指定されたリスナーが起動していないか、またはインスタンスが起動していないために接続が失敗することもあり、この場合、リスナーは接続を受け入れることができません。

クライアント・ロード・バランシングと接続時ロード・バランシングの構成 クライアント・ロード・バランシングを使用可能にするには、初期化パラメータ・ファイルと `tnsnames.ora` ファイルにサービスを構成します。構成するには、`LOAD_BALANCE` パラメータを `ON`、`YES` または `TRUE` に設定し、接続記述子に同じサービス名を指定します。

注意： Oracle Net は PMON プロセスを使用して、サービス情報を自動的に登録します。このため、`listener.ora` ファイルを静的なインスタンス情報に変更する必要はありません。

接続時ロード・バランシングの場合、各ノードのディスパッチャへの接続数は、アクティブな接続の数に基づきます。Oracle Net は、処理ロードが最も低く接続数が最も少ないノードに新しい接続を割り当てます。共有サーバーを構成している場合、Oracle Net はロードが最も低いディスパッチャに接続します。

注意： クライアントは、サーバー・マシンにスケーラブルな方法で接続してください。つまり、ネットワークも、スケーラブルである必要があります。

関連項目： この項で説明した機能に関する概要と構成の詳細は、『Oracle9i Net Services 管理者ガイド』を参照してください。

向上したネットワークの拡張性と共有サーバー機能

共有サーバーを使用すると、一定のメモリー量で、専用サーバーを使用した場合よりも多くのユーザーをサポートできるという点で、拡張性が向上します。共有サーバーでは、ユーザーの追加ごとに必要なメモリーの追加コストは、専用サーバーを使用した場合より低くなります。

Oracle9i の機能増加によって、次のような共有サーバーが必要となります。

- Oracle9i JVM
- 接続プーリング
- 接続集中化機能
- アドバンスド・キューイング

サービス登録と共有サーバーの機能 共有サーバーの重要な機能の 1 つに、**サービス登録**があります。この機能は、リスナーに対して、データベースの 1 つ以上の**サービス名**、**インスタンス名**およびネットワーク・アドレスを提供します。すべてのインスタンスと共有サーバー・ディスパッチャに関する現在の状態とロードも提供します。この情報によって、リスナーは、クライアント接続要求を適切なディスパッチャと専用サーバーに送信します。

この情報はリスナーに登録されるため、データベース・サービスに関する静的情報を使用して **listener.ora** ファイルを構成する必要はありません。サービス登録の利点は、共有サーバーの機能である接続時ロード・バランシングにも生かされます。

注意： サービス登録は、専用サーバーでも使用できます。ただし、専用サーバーでは、共有サーバーの場合と同じ拡張性は提供されません。

オペレーティング・システムの拡張性

Real Application Clusters データベースの拡張性も、オペレーティング・システムの拡張性に依存しています。1 つのノードが共有メモリー・システム（対称型マルチプロセッサ（SMP）の単一メモリーに複数の CPU が接続されたシステム）である場合は、ソフトウェアの拡張性が重要な問題になります。オペレーティング・システムにおける同期化の方法によって、システムの拡張性を判断できます。たとえば、非対称型マルチプロセッシングでは、単一の CPU のみが I/O 割込みを処理できます。

次の第 II 部では、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・サービスによって **Real Application Clusters** 内のリソースを調整する方法の詳細を説明します。

第 II 部

Real Application Clusters でのリソースの調整

第 II 部では、Real Application Clusters のリソースの調整について説明します。この調整によって、データ・アクセスが同期化され、データの整合性が保証されます。第 II 部に含まれる章は、次のとおりです。

- 第 5 章「Real Application Clusters リソースの調整」
- 第 6 章「キャッシュ・フュージョンとグローバル・キャッシュ・サービス」
- 第 7 章「グローバル・エンキュー・サービスによるリソースの調整」

Real Application Clusters リソースの調整

この章では、**Real Application Clusters** リソースの調整と管理の概要を説明します。この章の内容は、次のとおりです。

- **Real Application Clusters** リソースの調整の概要
- リソースのモードとロール
- システム変更番号処理

Real Application Clusters リソースの調整の概要

クラスタ操作では、リソースへの共有アクセスを制御するため、すべてのインスタンス間での同期化が必要です。Real Application Clusters は、**グローバル・リソース・ディレクトリ**を使用して、クラスタ・データベース内でのリソースの使用に関する情報を記録します。**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) は、このディレクトリ内の情報を管理します。各インスタンスは、システム・グローバル領域 (SGA) 内のグローバル・リソース・ディレクトリの一部をメンテナンスします。

Real Application Clusters でのリソース調整は、インスタンス・レベルとクラスタ・データベース・レベルの両方で行われます。Real Application Clusters でのインスタンス・レベルのリソース調整は、**ローカル・リソース調整**といいます。クラスタ・レベルの調整は、**グローバル・リソース調整**といいます。

クラスタ・データベースでのローカル・リソース調整を管理するプロセスは、シングル・インスタンスの Oracle でローカル・リソースを調整するプロセスと同じです。これは、Real Application Clusters とシングル・インスタンスの Oracle では、行およびブロック・レベルのアクセス、領域管理、システム変更番号 (SCN) 作成、およびデータ・ディクショナリ・キャッシュとライブラリ・キャッシュの管理が同じであることを意味します。ただし、グローバル・リソース調整は、この章で後述するように、少し複雑です。

グローバル・リソース・ディレクトリの内容

各インスタンスのグローバル・リソース・ディレクトリ部分には、すべての共有リソースの現在の状態に関する情報が記述されています。グローバル・リソース・ディレクトリ内の情報は、インスタンス障害のリカバリ時やクラスタの再構成時にも使用されます。

グローバル・リソース・ディレクトリには、次のタイプの共有リソース情報が記述されています。

- データ・ブロック識別子 (データ・ブロック・アドレスなど)。
- 最新バージョンのデータ・ブロックの位置 (データ・ブロックがクラスタ内の複数ノードのバッファ・キャッシュに読み込まれている場合)。
- インスタンスがデータ・ブロックを保持している**モード**。(N) NULL、(S) 共有、(X) 排他のいずれかです。
- 各インスタンスがデータ・ブロックを保持している**ロール**。ローカルまたはグローバルです。

Real Application Clusters の同期化プロセス

インスタンスがリソース（データ・ブロックなど）を要求した場合、Real Application Clusters はインスタンスのブロック取得をローカルで管理します。後でこのブロックが 1 つ以上のインスタンスによって変更された場合、Oracle は、グローバル・レベルで同期化を実行し、クラスタ全体によるこのブロックへの共有アクセスを可能にします。この場合の同期化には、ノード間のメッセージ機能が必要です。また、ブロックの読み取り一貫性バージョンを準備し、クラスタ・データベース内のメモリー・キャッシュ間でブロックのコピーを転送する必要があります。

グローバル・キャッシュ・サービス・プロセス (LMSn) は、1 つ以上のインスタンスが、別のインスタンスによって更新されたデータ・ブロックを要求した場合に、最新のブロック・イメージの検索、準備および転送を行います。GCS 処理を調整するため、GES はノード間のメッセージをインターコネクトを経由して転送します。

注意： ノード内のクラスタ・データベース処理の場合、Oracle は共有メモリーを使用します。このため、メッセージ機能は必要ありません。

このため、Real Application Clusters では、リモート・インスタンスがローカルに保持されているリソースを要求すると、ローカル調整の一部の要素がグローバルになります。このグローバル化は、特に、次の各項で説明するように、データ・ブロックのエンキューとパスト・イメージの場合に顕著です。

エンキュー

エンキューは、特定の行に対する更新をシリアル化するメモリー構造です。エンキューの調整は、エンキューが割り当てられたリソースを別のインスタンスが要求するまで必要ありません。

ユーザーは、グローバル・エンキューを構成する必要はありません。グローバル・エンキューの数は、起動時に自動的に計算され、算出された値が `alert.log` ファイルに記録されます。

パスト・イメージ

パスト・イメージは、使用済ブロックのコピーです。このブロックは、使用済ブロック内の情報の内容に影響する書込みが完了するまで Oracle が維持します。GCS がパスト・イメージを管理し、GCS はパスト・イメージを障害時リカバリで使用します。

リソースのモードとロール

インスタンス間でのブロック転送の結果、同じデータ・ブロックが複数のキャッシュに存在する場合があります。複数のインスタンスが、異なるモードでブロックを保持できます。モードは保持するインスタンスにおける動作がデータの変更か、ブロックの内容の単なる読み込みかによって決まります。データ・ブロックの共有など、GCS がクラスタ内でリソース調整に使用するモードとロールについては、次の各項で説明します。

- [リソースのモード](#)
- [リソースのロール](#)

リソースのモード

[リソースのモード](#)によって、リソースの保持側インスタンスがリソースを変更できるかどうかが決まります。表 5-1 に、**NULL (N) モード**、**共有 (S) モード**および**排他 (X) モード**の比較を示します。

表 5-1 グローバル・キャッシュ・サービスのリソースのモード

リソースのモード	識別子	説明
NULL	N	このレベルでリソースを保持する場合、アクセス権限はありません。
共有	S	保護付きの読み取り。このレベルでリソースを保持する場合、インスタンスはリソースを変更できません。複数のインスタンスがリソースを読み込みます。
排他	X	このレベルでリソースを保持する場合、保持側インスタンスに排他的アクセス権限が付与されます。他のインスタンスは、リソースに書き込みません。古いリソースの読み込み一貫性は有効です。

リソースのロール

Oracle は、保持側インスタンスのリソースに対して、[リソース・ロール](#)を割り当てます。ロールは、ローカルまたはグローバルのいずれかであるため、相互に排他的です。ローカルとグローバルの間でのロールの移行は、次のように行われます。

- ブロックが 1 つのインスタンスのキャッシュに初めて読み込まれ、別のインスタンスでは読み込まれていない場合、そのブロックは**ローカルで管理**されています。したがって、GCS はそのブロックにローカル・ロールを割り当てます。
- ブロックがローカル・インスタンスによって変更され、別のインスタンスに転送されると、そのブロックは**グローバルで管理**されているとみなされます。GCS は、そのブロックにグローバル・ロールを割り当てます。

すべてのリソースにローカルのロールがあるのは、それらのリソースが1つのキャッシュに存在している場合のみです。データ・ブロックが、あるインスタンスで変更されてから、別のインスタンスに転送された場合、そのデータ・ブロックを含むバッファはグローバルで**使用済**であるとみなされるため、リソースにはグローバル・ロールがあります。

グローバル・キャッシュ・サービスの動作

GCS は、データ・ブロックの位置、モードおよびロールを追跡します。このため、GCS は、リソースに関連する様々なインスタンスのアクセス権限も管理します。現行のバージョンのデータ・ブロックが、1つのインスタンスのバッファ・キャッシュ内に存在し、別のインスタンスがそのブロックの更新を要求する場合、Oracle は、キャッシュ一貫性のために GCS を使用します。

インスタンスが排他モードでブロックを読み込むと、そのインスタンス内の複数トランザクションは、後続の操作で、一連のデータ・ブロックへのアクセスを GCS を使用せずに共有できます。ただし、そのブロックがローカル・キャッシュ外に転送されていない場合にかぎります。ブロックがローカル・キャッシュ外に転送されている場合、GCS は、リソースにグローバル・ロールがある **グローバル・リソース・ディレクトリ** を更新します。リソースのモードが排他モードから他のモードに変換するかどうかは、他のインスタンスによるリソースの使用方法によって決まります。

グローバル・キャッシュ・サービスの処理の例

クラスタ・データベース内のあるインスタンスで、データ・ブロックを更新する必要があるとします。同時に、別のインスタンスでも同じブロックを更新する必要があるとします。GCS が提供するキャッシュ一貫性がない場合は、両方のインスタンスが同じブロックを同時に更新する可能性があります。しかし、GCS の同期化制御があれば、1つのインスタンスのみがブロックを更新でき、もう一方のインスタンスは待機します。

GCS は、ブロックを更新できるのは1回に1インスタンスのみであることを保証することで、キャッシュ一貫性を維持します。これによって、インスタンスは、そのインスタンスのトランザクションが完了するまで、排他モードでブロックを保持する必要がありません。たとえば、あるインスタンスのトランザクションがブロックの行を変更し、その変更がコミットされていない時でも、別のインスタンスによる同じブロックの別の行の変更が可能です。

キャッシュ一貫性とグローバル・キャッシュ・サービス

GCS は、インスタンスに対して、データベース・ブロックの変更前に、クラスタ全体のレベルでリソースを取得するように要求することで、キャッシュ一貫性を保証します。これによって、GCS はグローバル・キャッシュ・アクセスを同期化し、1回に1インスタンスのみがブロックを変更できるようにします。

Oracle のマルチバージョン・アーキテクチャを使用すると、現行のデータ・ブロックと1つ以上の**読み取り一貫性** (CR) バージョンのブロックを区別できます。現行のブロックには、すべてのコミット済みのトランザクションと、まだコミットされていないトランザクションの変更が含まれます。

読み込み一貫性 (CR) バージョンのブロックは、ある時点でデータの一貫性があるスナップショットを示します。LMS n プロセスは、ロールバック・セグメント情報をパスト・イメージに適用することで、読み込み一貫性バージョンを作成します。現行のブロックと CR ブロックは、GCS によって管理されます。

あるインスタンスが変更したブロックを保持している場合に、別のインスタンスがそのブロックを要求すると、保持側インスタンスは、そのブロックのパスト・イメージ (PI) を維持します。障害が発生した場合、Oracle は、PI を読み込むことによって現行の読み込み一貫性バージョンのブロックを再構築できます。

システム変更番号処理

クラスタ内のリソース（特にデータ・ブロック）を同期化するため、Oracle ではデータ・ブロックの変更を継続して追跡する必要があります。つまり、Oracle は、ブロックのバージョンごとに数値識別子を割り当て、データ・ブロックに対する各変更を記録します。この記録によって、Oracle は、REDO ログを順序立てて生成し、後続のリカバリ処理に備えます。

Oracle は、各インスタンス内およびすべてのインスタンスにわたるデータ・ブロック変更イベントの順序付けに、システム変更番号 (SCN) という論理タイムスタンプを使用します。SCN の主な目的は、REDO ログ・エントリにマークを設定して、リカバリ処理を同期化することです。

Oracle は 1 つの SCN を各トランザクションに割り当てます。概念的には、SCN を生成するグローバル・シリアル・ポイントが存在します。ただし、実際には、SCN はパラレルで読み込みおよび生成できます。このような SCN 生成スキームの 1 つを **Lamport SCN 生成スキーム** といいます。

シングル・インスタンスの Oracle では、システム・グローバル領域 (SGA) で、データベースを排他モードでマウントしたインスタンスの SCN がメンテナンスおよび増分されます。Real Application Clusters では、SCN をグローバルにメンテナンスする必要があり、その実装はプラットフォームによって異なります。SCN は、GCS、Lamport SCN 生成スキームによって、またはハードウェア・クロックか専用 SCN サーバーを使用して管理されます。

Lamport SCN 生成

Lamport SCN 生成スキームは、SCN をすべてのインスタンス上にパラレルで生成するため、効率的かつスケーラブルです。このスキームでは、インスタンス間のすべてのメッセージが SCN を保持します。複数のインスタンスは、これらのインスタンス間で追加の通信を行わずに、SCN をパラレルで生成できます。

ほとんどのプラットフォームでは、MAX_COMMIT_PROPAGATION_DELAY の値がプラットフォーム固有のしきい値より大きい場合、Lamport SCN 生成スキームが使用されます。通常はこれがデフォルトで、値は通常、7 秒に設定されています。この値を変更すると、クラスター・データベースで使用する SCN ジェネレータが変更されます。

インスタンスの起動後に `alert.log` を検査して、Lamport SCN 生成スキームが使用されているかどうかを確認します。`MAX_COMMIT_PROPAGATION_DELAY` の値がしきい値より小さい場合は、ハードウェア・クロックの SCN 生成スキームが使用されます。

第 6 章では、キャッシュ・フュージョン処理の詳細を説明します。

キャッシュ・フュージョンとグローバル・キャッシュ・サービス

この章では、[キャッシュ・フュージョン](#)処理と[グローバル・キャッシュ・サービス](#)（GCS）操作について説明します。また、[Real Application Clusters](#)でのリソース制御メカニズムを説明し、一般的なキャッシュ・フュージョンの使用例を示し、[Real Application Clusters](#)でのリカバリ処理についても説明します。この章の内容は、次のとおりです。

- [キャッシュ・フュージョン処理の概要](#)
- [リソース制御、キャッシュ間転送およびキャッシュ一貫性](#)
- [ブロック・アクセス・モードおよびバッファの状態](#)
- [キャッシュ・フュージョンの使用例](#)
- [Real Application Clusters](#)でのリカバリとキャッシュ・フュージョン

キャッシュ・フュージョン処理の概要

デフォルトでは、インスタンスのキャッシュ内に常駐する各データ・ブロックにリソースが割り当てられています。別のインスタンスがブロック更新を要求した際に発生する**キャッシュ・フュージョン**処理とディスク書込みの排除で、インスタンス間で共有データを管理するためのパフォーマンス・オーバーヘッドが大幅に減少します。キャッシュ・フュージョンの並行処理制御は、パフォーマンスを大幅に改善する他に、Real Application Clusters 環境での管理の負荷を削減します。

キャッシュ・フュージョンは、次の各項で説明するように、様々な並行処理のタイプに対応しています。

- 複数ノードでの並行読み込み
- 異なるノードでの並行読み込みと書込み
- 異なるノードでの並行書込み

複数ノードでの並行読み込み

複数ノードでの並行読み込みは、2つのインスタンスが同じデータ・ブロックを読み込む必要がある場合に発生します。Real Application Clusters では、同期を行わずにこの問題を解決します。これは、複数のインスタンスが読み込みアクセスで、データ・ブロックをキャッシュ一貫性の競合を伴わずに共有できるためです。

異なるノードでの並行読み込みと書込み

あるインスタンスで変更されてディスクにまだ書き込まれていないブロックに対して、別のインスタンスから読み込みが要求された場合、その読み込み要求は、現行のバージョンまたは読み込み一貫性バージョンのいずれかのブロックに対する要求である可能性があります。いずれの場合も、**グローバル・キャッシュ・サービス・プロセス** (LMSn) は、そのブロックを保持側インスタンスのキャッシュから要求側インスタンスのキャッシュにインターコネクトを経由して転送します。

異なるノードでの並行書込み

異なるノードでの並行書込みは、同じデータ・ブロックが異なるインスタンスによって頻繁に変更される場合に発生します。このような場合、保持側インスタンスは、ブロックに対する要求を受け取った後に、データ・ブロックでの作業を完了します。次に、GCS はブロック上のリソースをグローバル管理に変換し、LMSn プロセスはそのブロックのコピーを要求側インスタンスのキャッシュに転送します。この処理に関する主な機能は、次のとおりです。

- **グローバル・キャッシュ・サービス** (GCS) は、各バージョンのデータ・ブロックを追跡します。このような各バージョンを**パスト・イメージ** (PI) といいます。障害が発生した場合は、パスト・イメージ (PI) の情報を使用して、現行バージョンのブロックを再構築できます。

- 高速な IPC インターコネクトを経由して、キャッシュからキャッシュへのデータ転送が行われるため、ディスク I/O が排除されます。
- キャッシュ・フュージョンでは、一連のラウンドトリップ・メッセージが減少するため、**コンテキストのスイッチング**数が制限されます。コンテキストのスイッチング数の減少によって、キャッシュ一貫性のプロトコルの効率が向上します。キャッシュ・フュージョンのブロック転送に、データベース・ライター（DBWn）プロセスは含まれません。

書込みプロトコルとパスト・イメージの追跡

インスタンスがブロックの変更を要求した場合、**グローバル・キャッシュ・サービス・プロセス**（LMSn）は、そのブロックを最後に変更したインスタンスから要求側インスタンスに送信します。さらに、LMSn プロセスは、元々そのブロックを保持していたインスタンスにブロックの PI を保持します。

ディスクへの書込みは、キャッシュの置換えおよびチェックポイント時にのみトリガーされます。たとえば、あるインスタンスがデータ・ブロックの書込みを開始し、そのブロックのリソースにはグローバル・ロールがあるとします。ただし、そのインスタンスにあるのはブロックの PI のみで、最新のバッファはないとします。この場合、インスタンスから通知を受けた GCS は、最新バージョンのブロックを保持するインスタンスに書込み要求を転送します。次に、保持側インスタンスは、完了メッセージを GCS に送信します。最後に、このブロックの PI を持つ他のすべてのインスタンスは、その PI を削除します。

リソース制御、キャッシュ間転送およびキャッシュ一貫性

GCS は、インスタンスのバッファ・キャッシュに読み込まれたデータ・ブロックごとにリソースを割り当て、オープンします。リソースがこれ以上バッファの管理を行わない場合、またはバッファされたブロックがキャッシュの置換えのためにディスクに書き込まれた場合は、リソースがクローズされます。Oracle は、クローズしたリソースをリストに戻し、そのリストから新しいリソースを割り当てることができます。

ブロック・アクセス・モードおよびバッファの状態

並行処理制御の概念の1つである**バッファの状態**は、インスタンスのローカル・キャッシュ内にあるバッファの状態を示します。ブロックのバッファの状態は、そのブロックのアクセス・モードに関連しています。たとえば、バッファの状態が**排他カレント**（XCUR）の場合、インスタンスは排他モードでリソースを所有しています。

バッファの状態を確認するには、V\$BH 動的パフォーマンス・ビューの STATUS 列を問い合わせます。この V\$BH ビューには、次のように、ブロック・アクセス・モードとそのバッファの状態名に関する情報が表示されます。

- ブロック・アクセス・モードが NULL で、バッファの状態名が CR の場合、インスタンスはブロックの一貫性のある読み込みを実行できます。これは、インスタンスに古いバージョンのデータが保持されている場合にかぎります。
- ブロック・アクセス・モードが S で、バッファの状態名が SCUR の場合、インスタンスはブロックに対して共有でアクセスし、ブロックの読み込みのみ実行できます。
- ブロック・アクセス・モードが X で、バッファの状態名が XCUR の場合、インスタンスはブロックに対して排他的にアクセスし、ブロックを変更できます。
- ブロック・アクセス・モードが NULL で、バッファの状態名が PI の場合、インスタンスによるブロック変更はあったが、そのコピーがバースト・イメージとして保持され、ブロックの変更前の状態が記録されます。

Real Application Clusters 固有のバッファの状態名は、SCUR と PI のみです。クラスタ・データベース内には、XCUR 状態でバッファされた1つのブロックのコピーが、常に1つのみ存在している可能性があります。ブロックを変更するには、プロセスが、データ・ブロックを含むバッファに XCUR のバッファ状態を割り当てる必要があります。

たとえば、別のインスタンスから、同じブロックの最新バージョンの読み込みアクセスが要求されると、アクセス・モードが排他から共有に変更され、ブロックの現行の読み込みバージョンが要求側インスタンスに送信されます。バッファに**使用済ブロック**がある場合は PI バッファが保持されます。

この時点では、現行のブロックは最初のインスタンスに存在し、要求側インスタンスが同じ現行のブロックを共有モードで所有します。このため、リソースのルールは、グローバルになります。このブロックには、常に、クラスタ・データベース全体のキャッシュされた複数の**共有カレント**（SCUR）バージョンが存在する可能性があります。

キャッシュ・フュージョンの使用例

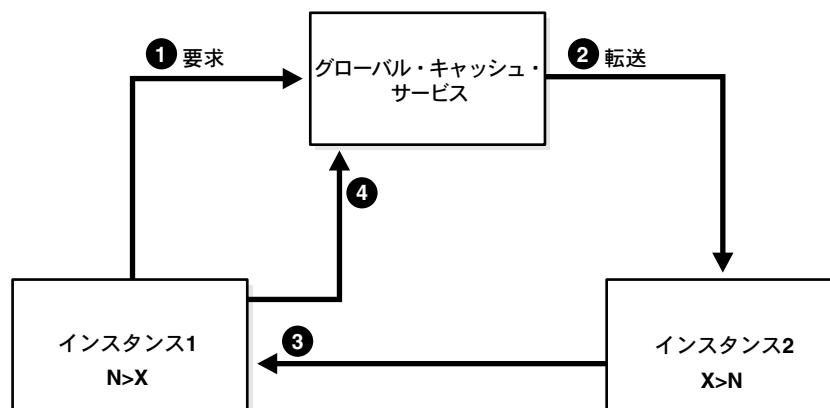
次の使用例では、キャッシュ・フュージョン処理の最も重要なポイントを示します。この使用例では、可能なすべての構成を示しているわけではありません。たとえば、ここでは読み込み操作についての説明はありません。内容は次のとおりです。

- 変更されたブロックの変更操作要求
- ディスクへのブロックの書込み

変更されたブロックの変更操作要求

図 6-1 に示す使用例では、1 つのインスタンスのみでデータ・ブロックが変更または使用され、排他モード (X) で保持されているとします。さらに、そのブロックにアクセスしているのは、これまでにそのブロックを変更したインスタンスのみであるとします。つまり、そのブロックのコピーは、クラスタ全体で 1 つのみ存在します。これは、このブロックにローカル・ロール (L) があることを示します。

図 6-1 変更されたブロックの変更操作要求



1. ブロックを変更しようとしているインスタンス（インスタンス 1）は、GCS に要求を送信します。
2. GCS は、保持側（インスタンス 2）に要求を転送します。

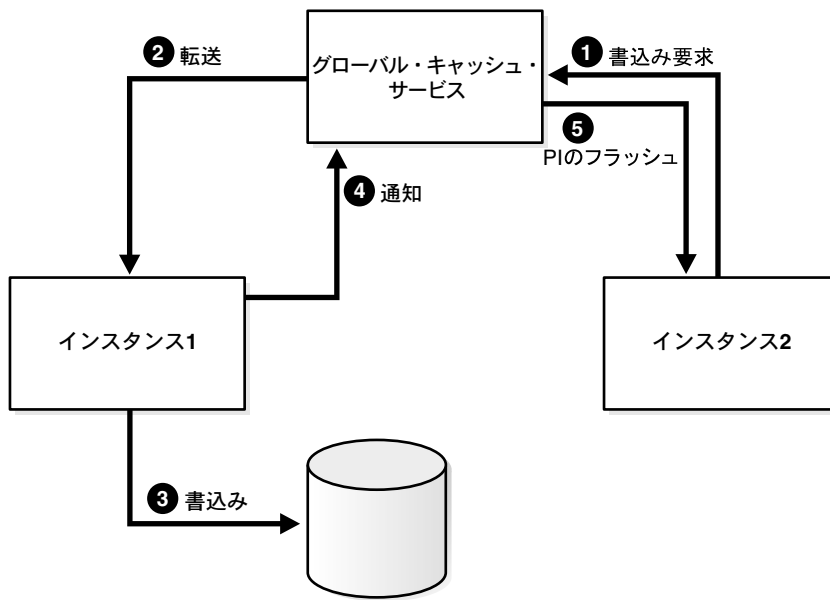
3. インスタンス 2 はメッセージを受信し、LMS プロセスはブロックをインスタンス 1 に送信します。ブロックを送信する前に、インスタンス 2 のリソースが排他モードから NULL モード (N) にダウングレードされ、インスタンス 2 は使用済みバッファを PI として保持します。複数のインスタンス内でブロックが使用済になるため、ロールはグローバル (G) に変わります。インスタンス 2 は、ブロックとともに、PI を NULL モード (N) で保持したことを要求側インスタンスに通知します。また、このとき、インスタンス 2 は、要求側がブロックを排他モード (X) とグローバル・ロール (G) で保持する必要があることを指定します。
4. ブロックを受信したインスタンス 1 は、ブロックを排他モードとグローバル・ロールで保持していることを GCS に通知します。データ・ブロックは、インスタンス 1 にリソースが付与されるまで、ディスクに書き込まれません。

ディスクへのブロックの書込み

図 6-2 に示す使用例では、インスタンスが必要時にチェックポイントを実行する方法、または使用可能バッファの要求のために、キャッシュ内でバッファを置き換える方法について説明します。クラスタ内のインスタンスのキャッシュには、変更された複数バージョンのデータ・ブロックが存在している可能性があります。このため、GCS が管理する書込みプロトコルによって、ディスクに書き込まれるのは、最新バージョンのデータのみであることが保証されます。また、この書込みプロトコルによって、以前のバージョンすべてが他のキャッシュから削除されることも保証されます。データ・ブロックに対する書込み要求は、ブロックの現行バージョンまたはパスト・イメージが存在するどのインスタンスでも発生する可能性があります。

この使用例では、PI バッファを NULL モードで保持するインスタンスが、ディスクへのバッファの書込みを要求するとします。

図 6-2 ディスクへのブロックの書き込み



1. インスタンス 2 が、書き込み要求を GCS に送信します。
2. GCS は、インスタンス 1（現行のブロックの保持側）に要求を転送します。
3. インスタンス 1 は書き込み要求を受信し、ディスクにそのブロックを書き込みます。
4. インスタンス 1 は、GCS を使用した書き込み操作の完了を記録し、現行のブロックの書き込みを実行したため、リソースのロールがローカルになることを GCS に通知します。
5. 通知を受信した GCS は、すべての PI の保持側に PI を廃棄（フラッシュ）するように指示します。これらの PI は、すでにリカバリには不要なものです。バッファは解放され、以前の NULL モードで保持されたリソースはクローズされます。

次の項では、Real Application Clusters でのリカバリについて説明します。

Real Application Clusters でのリカバリとキャッシュ・フュージョン

Real Application Clusters リカバリで、ノード障害後のリカバリに必要な処理量は、障害が発生したノードの数に比例します。通常、データ・ブロックは、リカバリした後すぐに使用可能になります。

あるインスタンスで発生した障害が別の Oracle インスタンスによって検出された場合、Oracle によって次のリカバリ手順が実行されます。

1. リカバリの第 1 フェーズである GES の再構成時に、GES エンキューが再構成されます。次に、GCS リソースが再構成されます。この間、すべての GCS リソース要求と書き込み要求は、一時的に中断されます。ただし、すでに必要なエンキューを取得しているプロセスとトランザクションでは、データ・ブロックの変更を継続できます。
2. GES が制御するエンキューを再構成した後、ログの読み込みと GCS リソースの再マスタ化が平行で実行されます。この手順の最後に、リカバリする必要があるブロック・リソースが識別されます。
3. リカバリ用のバッファ領域が割り当てられ、前述のログ読み込みで識別されたリソースが、リカバリ・リソースとして要求されます。ここで、クラスタ・データベース内の別のキャッシュには、リカバリ対象のブロックの PI が存在すると仮定した場合は、他のインスタンスから、リソース・バッファが要求されます。そのリソース・バッファから、特定のブロックに対するリカバリを開始します。
4. 後続の処理に必要なすべてのリソースとエンキューを取得すると、グローバル・リソース・ディレクトリは、この時点で解放されます。また、リカバリ中でないすべてのデータ・ブロックに対してアクセス可能となります。システムは、部分的には、すでに使用可能であることに注意してください。
5. キャッシュ・レイヤーは、手順 2 で識別された各ブロックをリカバリして書き込み、ブロック・リカバリの直後にリカバリ・リソースを解放するため、キャッシュ・リカバリが進行するにつれて、より多くのブロックが使用可能になります。
6. すべてのブロックがリカバリされ、リカバリ・リソースが解放されると、システムは完全に使用可能な状態に戻ります。リカバリされたブロックは、リカバリ完了後に使用可能になります。

つまり、リカバリされたデータベースまたはデータベースの一部は、すべてのリカバリ手順が完了する前に、使用可能になります。このため、システムがより早く使用可能になり、スケラブルなリカバリが可能になります。

データ・ブロックの PI バッファや現行のバッファが、障害が発生していないインスタンスのいずれのキャッシュにも存在しない場合は、障害が発生したインスタンスのログのマージが実行されます。リカバリ処理の一般論として前述したように、ログのマージによるパフォーマンスのオーバーヘッドは、障害が発生したインスタンスの数と各インスタンスの REDO ログ・ファイルのサイズに比例します。ただし、ログ・ファイルのサイズは、Oracle のチェックポイント機能を使用して制御できます。高度に設計された Real Application Clusters のリカバリは、同時に発生した複数の障害および連続して発生した障害を管理する

ことができます。また、共有サーバー機能も、リカバリ中のインスタンス障害に対応できます。

関連項目： Real Application Clusters でのリカバリの詳細は、『Oracle9i Real Application Clusters 管理』を参照してください。

[第7章](#)では、グローバル・エンキュー・サービスで実行するリソースの調整について説明します。

グローバル・エンキュー・サービスによる リソースの調整

この章では、**グローバル・エンキュー・サービス**（GES）が Real Application Clusters で実行する、インスタンス間のリソースを調整するメッセージ機能について説明します。GES は、**リソース**、データ、およびインスタンス間のデータ要求を使用して、インスタンス間の調整を行います。この章の内容は、次のとおりです。

- **グローバル・エンキュー・サービスの処理**
- **グローバル・エンキューの並行処理制御**
- **グローバル・エンキュー・サービスで管理するリソース**

関連項目： グローバル・キャッシュ・サービス（GCS）については、第 6 章「**キャッシュ・フュージョンとグローバル・キャッシュ・サービス**」を参照してください。

グローバル・エンキュー・サービスの処理

GES は、キャッシュ・フュージョン以外のすべてのインスタンス間のリソース操作を管理し、すべての Oracle エンキュー・メカニズムの状態を追跡します。GES は、複数のインスタンスがアクセスするリソースに対してこの処理を実行します。GES が制御する主要なリソースは、ディクショナリ・キャッシュ・ロックとライブラリ・キャッシュ・ロックです。GES は、インスタンス間で発生するリソースに関する通信を管理します。ただし、これらのリソースは、データ・ファイル・ブロックを保護しません。また、GES は、デッドロックの影響を受けやすいすべてのエンキューとリソースに対して、デッドロック検出を実行します。

グローバル・エンキューの並行処理制御

Oracle データベースの他のレイヤーでは、データ構造をグローバルに保護します。該当するレイヤーは、次のとおりです。

- データ・ディクショナリ・アクセス・レイヤー
- ライブラリ・キャッシュ・レイヤー
- トランザクション管理コード・レイヤー

各レイヤーには、他のレイヤーがそれぞれのデータ構造へのアクセスで使用する特定のプロトコルがあります。Oracle のすべてのレイヤーでは、GES の API サービスを使用して、レイヤーに常駐するリソースを取得、変換および解放します。GES は、インスタンスの起動時に、グローバル・エンキュー・リソースに影響を与えるパラメータを使用して、エンキュー・リソース管理操作に必要な正確なメモリー要件を自動的に計算します。

トランザクション・レイヤーでは、グローバル・エンキュー・リソースは、非常に短い時間のみ保持されて、すぐに解放される場合があります。たとえば、通常、TX ロックはトランザクションの開始時に取得され、トランザクションのコミット時に解放されます。TX ロックは、ライブラリ・キャッシュ・ロックなど、DDL 文がライブラリ・キャッシュ・オブジェクトを無効にすることがあるイベントの通知に使用される場合があります。通常、Real Application Clusters でのトランザクションと表のロック処理は、シングル・インスタンスの Oracle データベースの場合と同じ方法で実行されます。次の項では、GES で管理するリソースについて詳細に説明します。

グローバル・エンキュー・サービスで管理するリソース

ここでは、Oracle が GES エンキューを使用して、Oracle Real Application Clusters 環境のトランザクション、表およびその他のエンティティで操作されるリソースの並行性を管理する方法について説明します。次のリソースは、シングル・インスタンスの Oracle データベースではローカルですが、GES の制御下ではグローバルです。

- [ディクショナリ・キャッシュ・ロック](#)
- [ライブラリ・キャッシュ・ロック](#)

ディクショナリ・キャッシュ・ロック

Real Application Clusters データベースの各インスタンスには、データ・ディクショナリ情報が記述されたディクショナリ・キャッシュまたは[行キャッシュ](#)があります。データ・ディクショナリ構造は、クラスタ・データベース内の Oracle インスタンスの場合もシングル・インスタンスの Oracle データベース内のインスタンスの場合も同じです。ただし、Real Application Clusters では、クラスタ全体のすべてのディクショナリ・キャッシュが同期化されます。Real Application Clusters では、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、ラッチを使用して同期化を行います。

5 つのノードで構成される Real Application Clusters 環境があり、そのうちの 1 つのノードで、ユーザーが表を削除したとします。5 つのディクショナリ・キャッシュには、それぞれ削除した表の定義のコピーがあるため、表をキャッシュから削除するノードは、他の 4 つのディクショナリ・キャッシュに対して、削除した表の定義のコピーを削除するように通知する必要があります。Real Application Clusters は、GES で管理するメッセージを使用して、これを自動的に処理します。

ライブラリ・キャッシュ・ロック

SQL 文、DML 文、DDL 文、PL/SQL 文または Java 文の解析中に、データベース・オブジェクト（表、ビュー、プロシージャ、パッケージ、索引など）が参照された場合、文を解析するプロセスでは、ライブラリ・キャッシュ・ロックが取得されます。Oracle9i では、解析またはコンパイルの完了まで（つまり、解析コールの存続期間）、ロックが保持されます。

第 III 部では、Real Application Clusters の実装について説明します。

第 III 部

Real Application Clusters の実装

第 III 部では、Real Application Clusters の実装に固有の項目について説明します。第 III 部に含まれる章は、次のとおりです。

- 第 8 章「Real Application Clusters の記憶域上の考慮点」
- 第 9 章「Real Application Clusters 環境用の管理ツール」

Real Application Clusters の記憶域上の考慮点

この章では、Real Application Clusters の記憶域上の考慮点について説明します。この章の内容は、次のとおりです。

- Real Application Clusters の記憶域の概要
- Real Application Clusters のデータ・ファイル
- Real Application Clusters のパラメータ・ファイルの記憶域
- Real Application Clusters の REDO ログ・ファイルの記憶域
- Real Application Clusters での UNDO 領域の管理

Real Application Clusters の記憶域の概要

Real Application Clusters とシングル・インスタンスの Oracle データベースの記憶域の主な相違は、Real Application Clusters 内のすべてのデータ・ファイルと構成ファイルは、クラスタ・ファイル・システムまたは共有 RAW デバイス上のいずれかに存在している必要があります。また、インスタンスごとに 2 つ以上の追加 REDO ログを作成する必要があります。さらに、自動 UNDO 管理を使用する場合は、各インスタンスに独自の表領域用のデバイスを 1 つ作成する必要があります、自動 UNDO 管理を使用しない場合は、データベース用のロールバック・セグメント表領域を 1 つ作成する必要があります。

ご使用のプラットフォームがクラスタ・ファイル・システムをサポートしていない場合、または Real Application Clusters のデータ・ファイルの格納にクラスタ・ファイル・システムを使用しない場合は、『Oracle9i Real Application Clusters セットアップおよび構成』に説明するように、追加の RAW デバイスを作成する必要があります。次の各項では、Real Application Clusters 固有の記憶域について説明します。

Real Application Clusters のデータ・ファイル

すべての Real Application Clusters インスタンスは同じデータ・ファイルにアクセス可能であることが必要です。データベース・ファイルの構成は、Real Application Clusters とシングル・インスタンスの Oracle で同じです。したがって、Real Application Clusters で使用するためにデータ・ファイルを変更する必要はありません。ただし、データ・ファイルの物理的な配置は変わります。これは、前述のように、すべてのインスタンスがファイルにアクセスする必要があります。

Real Application Clusters のデータ・ファイルの検証

最初に起動する Real Application Clusters インスタンスは、メディア・リカバリを実行するかどうかを判断するために、すべてのオンライン・ファイルへのアクセスを検証します。その他のインスタンスは、すべてのオンライン・データ・ファイルにアクセスしなくても操作できますが、検証されていないファイルを使用しようとすると失敗し、エラー・メッセージが生成されます。

Real Application Clusters でのデータ・ファイルの追加

データ・ファイルを表領域に追加したり、データ・ファイルをオンライン化すると、すべてのインスタンスはそのファイルへのアクセスを検証します。他のインスタンスがアクセスできないディスクに新しいデータ・ファイルを追加する場合、検証は失敗しますが、それらのインスタンスは実行し続けます。検証は、インスタンスが同じデータ・ファイルの別のコピーにアクセスした場合も失敗します。

どのインスタンスで検証が失敗しても、問題を診断して解決してください。ALTER SYSTEM CHECK DATAFILES 文を使用して、アクセスを検証してください。リカバリの実行では、ファイル全体をリカバリする必要はありません。ファイル全体ではなく、個々のブロックをリカバリできます。

関連項目：

- ブロック・メディア・リカバリ（BMR）の詳細は、『Oracle9i Recovery Manager ユーザーズ・ガイド』を参照してください。
- ALTER SYSTEM CHECK DATAFILES 文の使用の詳細は、『Oracle9i SQL リファレンス』を参照してください。
- Oracle Managed File（OMF）機能と関連ファイルのネーミング規則の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

Real Application Clusters のパラメータ・ファイルの記憶域

パラメータの管理には、サーバー側のパラメータ・ファイルまたはクライアント側のパラメータ・ファイルの 2 種類のファイルを使用できます。パラメータ管理を簡素化し、Oracle の高度なセルフ・チューニング機能を利用するために、サーバー・パラメータ・ファイルを使用してパラメータを管理することをお薦めします。Oracle のセルフ・チューニングによって、サーバー・パラメータ・ファイルのパラメータ設定が自動的に変更されます。

Database Configuration Assistant (DBCA) は、サーバー・パラメータ・ファイルをバイナリ・ファイルとして作成します。デフォルトでは、Oracle は 1 つの **パラメータ・ファイル** (PFILE) を基にサーバー・パラメータ・ファイルを作成します。サーバー・パラメータ・ファイルは、アスタリスク (*) を使用してグローバル・パラメータ設定を識別し、Oracle **システム識別子** (SID) を使用してインスタンス固有の設定を指定します。

サーバー・パラメータ・ファイルのパラメータ設定は、**Oracle Enterprise Manager** または SQL 文の ALTER SYSTEM SET を使用して変更できます。従来のクライアント側のパラメータ・ファイルを使用する場合、セルフ・チューニングの結果として Oracle が行ったパラメータの変更は、Oracle の停止後は保持されません。

注意： サーバー・パラメータ・ファイル内のセルフ・チューニング・パラメータの値は、変更しないでください。変更すると、パフォーマンスが著しく低下する場合があります。

サーバー・パラメータ・ファイルの位置

DBCA を使用して Real Application Clusters データベースを作成している場合、DBCA がサーバー・パラメータ・ファイル用に提供する、シングル・インスタンスの Oracle データベースのデフォルトの位置は、Real Application Clusters では適切ではありません。すべてのインスタンスが同じサーバー・パラメータ・ファイルを使用する必要があります。このため、DBCA が提供するデフォルトの位置を変更する必要があります。

ただし、デフォルトの位置の変更先は、ご使用のファイル・システム構成によって異なります。ご使用のプラットフォームがクラスタ・ファイル・システムをサポートしている場合、または単一ノードの Real Application Clusters データベースを使用している場合は、サー

パー・パラメータ・ファイルをファイル・システム上に配置できます。それ以外の場合は、サーバー・パラメータ・ファイルの位置として RAW デバイス名を入力する必要があります。

関連項目： サーバー・パラメータ・ファイルの構成方法および使用方法の詳細は、『Oracle9i Real Application Clusters セットアップおよび構成』および『Oracle9i Real Application Clusters 管理』を参照してください。

Real Application Clusters の REDO ログ・ファイルの記憶域

Real Application Clusters では、それぞれのインスタンスはその一連のオンライン REDO ログ・ファイルに書き込みます。インスタンスが書き込んだ REDO を REDO **スレッド**といいます。それぞれのオンライン REDO ログ・ファイルは、特定のスレッド番号に対応付けられます。オンライン REDO ログがアーカイブされると、Oracle はスレッド番号を記録し、Oracle がリカバリを実行するときにそのスレッド番号を識別します。

プライベート・スレッドは、THREAD 句を指定した ALTER DATABASE ADD LOGFILE 文を使用して作成された REDO ログです。**パブリック・スレッド**は、THREAD 句を指定しない ALTER DATABASE ADD LOGFILE 文を使用して作成された REDO ログです。

THREAD 初期化パラメータが指定されている場合、起動するインスタンスは、その値によってプライベート・スレッドとして識別されるスレッドを取得します。THREAD 初期化パラメータがデフォルトの 0（ゼロ）である場合、そのインスタンスはパブリック・スレッドを取得します。一度取得されると、その取得インスタンスは排他的に REDO スレッドを使用します。

注意： オンライン REDO ログ・ファイルは、多重化またはミラー化できません。

関連項目： 多重 REDO ログ・ファイルの詳細は、『Oracle9i データベース概要』および『Oracle9i データベース管理者ガイド』を参照してください。

自動セグメント領域管理

Real Application Clusters では、別々のインスタンス上のトランザクションが同じ表に同時にデータを挿入できます。この場合、新しいレコード用の空き領域を検索するための競合は発生しません。これを実現するため、Oracle では、行が使用可能な領域を超える場合があるトランザクションのために、領域のあるブロックを厳密に管理します。Oracle は、これを表、クラスタ、索引などのデータベース・オブジェクトごとに実行します。

領域管理の制御には、自動セグメント領域管理の使用をお薦めします。この機能は、ビットマップを使用して、セグメント内の空き領域を管理します。空きリストを使用する場合は、使用する Real Application Clusters データベース内のインスタンスと同じ数の空きリスト・グループを作成してください。

関連項目： 空きリストおよび空きリスト・グループの詳細は、『Oracle9i Real Application Clusters 配置およびパフォーマンス』を参照してください。自動セグメント領域管理の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

Real Application Clusters での UNDO 領域の管理

UNDO ブロックは、トランザクション（コミットされていない場合も含む）で変更された以前のデータ値の記録です。UNDO ブロックは、読み取り一貫性を維持し、ロールバックまたは異常終了したトランザクションが行った変更を取り消したり、データベースをリカバリする際に必要です。

Real Application Clusters での UNDO 領域の制御には、自動 UNDO 管理の使用をお薦めします。この機能は、ロールバック・セグメント UNDO より簡単に管理できます。また、自動 UNDO 管理による領域管理のパフォーマンスは、ロールバック UNDO モードでの領域管理パフォーマンスより優れています。

注意： 自動 UNDO 管理を使用する場合は、各インスタンスに独自の UNDO 表領域が必要です。

プライベート・ロールバック・セグメントとパブリック・ロールバック・セグメント

ロールバック・セグメント UNDO モードを使用する場合、Real Application Clusters の各インスタンスは、SYSTEM ロールバック・セグメントを共有して使用し、1 つ以上の専用ロールバック・セグメントを必要とします。インスタンスは、起動時にプライベート・ロールバック・セグメントまたはパブリック・ロールバック・セグメントのいずれかを取得して、排他的に使用できます。SYSTEM ロールバック・セグメントは例外です。ただし、他のイン

スタンスでは、ロールバック・セグメントを読み込んで、読み込み一貫性のあるスナップショットを作成したり、インスタンス・リカバリを実行することができます。

Real Application Clusters の各インスタンスでプライベート・ロールバック・セグメントを使用し、特定のインスタンスを起動するときは同じロールバック・セグメント識別子を使用することをお勧めします。ROLLBACK_SEGMENTS パラメータを使用する場合、プライベート・ロールバック・セグメントには、SID を指定する必要があります。インスタンスは、ロールバック・セグメントがオフラインにされるまで、またはロールバック・セグメント・パラメータで指定した取得インスタンスが停止されるまで、ロールバック・セグメントを使用します。

パブリック・ロールバック・セグメントは、ロールバック・セグメントが必要なインスタンスが起動して、それを取得およびオンラインにするまでオフラインであり、インスタンスでも使用されません。一度オンラインにすると、それを取得したインスタンスは排他的にパブリック・ロールバック・セグメントを使用します。

関連項目： ロールバック・セグメントのパフォーマンス、およびロールバック・セグメントの追加による影響については、『Oracle9i データベース管理者ガイド』を参照してください。このマニュアルでは、自動 UNDO 管理、UNDO 表領域およびロールバック・セグメントの管理についても説明されています。

Real Application Clusters 環境用の 管理ツール

Oracle には、Real Application Clusters データベースの構成と管理を簡素化する多数の管理ツールがあります。この章の内容は、次のとおりです。

- [Real Application Clusters の管理の概要](#)
- [Real Application Clusters のインストール、設定および構成の管理](#)
- [Real Application Clusters の管理](#)
- [Real Application Clusters のパフォーマンス監視用の管理ツール](#)
- [Real Application Clusters のバックアップとリカバリの管理](#)

関連項目： これらの機能用の管理ツールについては、『Oracle9i Real Application Clusters Real Application Clusters Guard I - Concepts and Administration』および『Oracle9i Real Application Clusters Guard II Concepts, Installation, and Administration』を参照してください。

Real Application Clusters の管理の概要

Oracle には、Real Application Clusters データベース配置のすべてのフェーズに対して、管理ツールが用意されています。Oracle の管理フレームワークによって、Real Application Clusters データベースのインストール、構成、管理および配置を簡素化するユーザー・フレンドリなツールが提供され、優れた運用性と拡張性が提供されます。

Real Application Clusters は、完全に自動化されたプロセス、またはカスタマイズされたインストール・プロセスのいずれかを使用してインストールできます。システムを構成すると、Real Application Clusters データベースは、データのロードと本番操作に対する準備が整います。

Real Application Clusters データベースの継続的な管理は、Oracle Enterprise Manager (EM) の機能を使用して簡素化できます。Enterprise Manager は、Graphical User Interface (GUI) ベースのツールで、Real Application Clusters データベース環境内のすべてのアクティビティに関するクラスタ全体のビューを提供します。また、サーバー制御ユーティリティ (SRVCTL) とグローバル・サービス・デーモン (GSD) のコマンドによって、管理操作を迅速に実行できます。

Real Application Clusters では、Oracle9i セルフ・チューニング機能も利用します。さらに、バックアップ操作とリカバリ操作の自動化に、Oracle Recovery Manager (RMAN) の一連の機能を使用できます。次の各項では、これらの機能について詳細に説明します。

Real Application Clusters のインストール、設定および構成の管理

Oracle Universal Installer は、Database Configuration Assistant および Network Configuration Assistant とともに機能し、Oracle9i および Real Application Clusters ソフトウェアをインストールし、Real Application Clusters データベースを構成します。Oracle Universal Installer には、ユーザー入力をほとんど必要としない、事前構成済インストールがいくつか用意されています。構成は、Transaction Processing、Data Warehousing、General Purpose の各構成タイプから選択できます。これらの構成タイプによって、各タイプの処理環境に適した事前構成済オプションがインストールされます。

Oracle Universal Installer と構成ツールを個別に実行して、Oracle Real Application Clusters ソフトウェアを手動でインストールし、クラスタ・データベース環境を構成することもできます。また、一部のプラットフォームでは、Real Application Clusters のデータ・ファイルをクラスタ・ファイル・システムに格納できます。これによって、インストールと管理の手順が大幅に簡素化されます。それ以外のプラットフォームの場合は、Real Application Clusters のインストール前に、RAW デバイスを構成する必要があります。

関連項目： Oracle9i ソフトウェア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。クラスタ・ファイル・システムについては、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。

Real Application Clusters の管理

Oracle には、次の各項で説明するように、Real Application Clusters データベースの管理を簡素化するためのツールがいくつか用意されています。

- [Oracle Enterprise Manager](#)
- [Database Configuration Assistant](#)
- [サーバー制御ユーティリティ \(SRVCTL\)](#)
- [グローバル・サービス・デーモンの管理コマンド](#)

Oracle Enterprise Manager

Oracle Enterprise Manager は、インスタンスとデータベースの両レベルの操作を制御します。EM を使用すると、クラスタ全体を単一エンティティとして表示したり、個々のインスタンスに関する詳細情報を取得できます。EM によって使用可能になる情報は、クラスタ・データベースの場合とシングル・インスタンスの Oracle データベースの場合とで同じです。つまり、EM を使用すると、シングル・インスタンスの Oracle データベースの場合と同じマスター・ビュー、詳細ビュー、メニューを操作して、クラスタ・データベースとその関連要素を管理できます。

Oracle Enterprise Manager の要件

Enterprise Manager を使用するには、Oracle Intelligent Agent が、Real Application Clusters データベースの一部である各ノード上に常駐する必要があります。Oracle Intelligent Agent は、Oracle Universal Installer による Oracle データベース・ソフトウェアのインストール時にインストールされます。

Enterprise Manager を使用した管理タスク

Enterprise Manager を使用すると、Oracle Intelligent Agent が検出したインスタンス、リスナーおよびクラスタ・データベースを起動および停止できます。さらに、インスタンスの特定グループとそのサービスも起動および停止できます。Oracle Enterprise Manager は、Real Application Clusters データベース内のすべてのインスタンスの状態也表示します。また、Enterprise Manager を使用すると、次の管理タスクを実行できます。

- サーバー・パラメータ・ファイル内の初期化パラメータ値を表示および設定します。
- UNDO 領域管理をさらに簡素化するために、自動 UNDO 管理を制御します。
- すべてのインスタンス上のセッションとその詳細を調べて、任意のインスタンスのセッションを終了します。
- 操作の効率を向上させるために、各インスタンスに異なるリソース・プランを割り当てて、リソース管理を制御します。

EM を使用すると、クラスタ・ファイル・システムのファイルと RAW デバイス上のデータ・ファイルの両方の表領域を表示することもできます。EM は、必要に応じてこれらのオブジェクトをオンラインまたはオフラインにするためにも使用します。また、EM を使用して、REDO ログ・グループをクラスタ・データベース・インスタンスの特定スレッドに割り当て、スキーマ・オブジェクトとユーザーを管理できます。

Enterprise Manager を使用すると、Real Application Clusters 環境内で実行するジョブを作成して管理できるため、反復して実行するタスクを自動化できます。ジョブは、Intelligent Agent が実行されているノードまたはノード・グループ上にスケジュールできます。また、ジョブは、クラスタ・データベースまたはクラスタ・データベース・インスタンスに対してもスケジュールできます。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、Enterprise Manager は複数のクラスタ・データベース・インスタンスのレポートやクラスタ・データベース環境全体のレポートを作成し、Web に公開します。

関連項目： Oracle Enterprise Manager については、『Oracle Enterprise Manager 概説』を参照してください。

Database Configuration Assistant

Database Configuration Assistant は、Real Application Clusters 環境を管理するための機能を備えています。DBCA は、クラスタ内の任意のノード（そのノード上のインスタンスが起動しているかどうかに関係なく）から操作できます。

DBCA の Instance Management 機能を使用して、インスタンス、ノードおよびデータベースをクラスタに追加したりクラスタから削除します。Instance Management は、データベース作成スクリプトの作成と管理にも使用できます。この機能を使用すると、既存のデータベース用のスクリプトを作成して、スキーマのリバース・エンジニアリングを行うことができます。また、Instance Management によって、インスタンス固有のオブジェクトを割り当てることができます。つまり、ロールバック・セグメントと REDO ログ・グループを特定のインスタンスに割り当てることができます。

DBCA は、クラスタ・ファイル・システムをサポートするオペレーティング・システム用に、Oracle Managed Files (OMF) もサポートしています。OMF を使用すると、ファイル管理のオーバーヘッドが大幅に削減されます。Oracle で追加のファイルが必要な場合、またはファイルが不要になった場合に、OMF は自動的にファイルの追加または削除を行います。

サーバー制御ユーティリティ (SRVCTL)

サーバー制御ユーティリティ (SRVCTL) を使用すると、Oracle Enterprise Manager とともに実行するデータベース・レベルの多数の管理タスクを実行できます。たとえば、SRVCTL を使用すると、クラスタ・データベース・タスクとデータベース構成タスクを実行できます。

クラスタ・データベース・レベルで SRVCTL を使用すると、クラスタ・データベースの起動と停止、クラスタ・データベース・インスタンスの起動と停止、およびクラスタ・データベースとそのインスタンスの状態の取得を行うことができます。また、SRVCTL を使用して、次のクラスタ・データベース構成タスクを実行できます。

- クラスタ・データベース構成情報の追加と削除
- クラスタ・データベース構成でのインスタンスの追加と削除
- クラスタ・データベース構成内のノード間でのインスタンスの移動
- クラスタ・データベース構成内のインスタンス環境の設定と設定解除
- クラスタ・データベース構成内のクラスタ・データベース全体の環境設定と設定解除

SRVCTL は、グローバル・サービス・デーモン、仮想 IP アドレス、リスナーおよび EM エージェントが含まれるプログラム・グループの起動と停止を行う場合にも使用します。

グローバル・サービス・デーモンの管理コマンド

`gsdctl` コマンドを使用して、グローバル・サービス・デーモン (GSD) の起動、停止および状態の取得を行うことができます。`gsdctl` コマンドは、すべてのプラットフォームで機能し、Real Application Clusters データベース内のすべてのノード上での GSD 操作を制御します。

関連項目： Oracle Real Application Clusters を管理するための管理コンポーネントの詳細は、『Oracle9i Real Application Clusters 管理』を参照してください。

Real Application Clusters のパフォーマンス監視用の管理ツール

次の 2 つの Oracle 管理ツールを使用して、Real Application Clusters のパフォーマンスを監視できます。詳細は、後の各項を参照してください。

- [Oracle Enterprise Manager](#) を使用したパフォーマンスの監視
- [Statspack](#) を使用したパフォーマンスの監視

Oracle Enterprise Manager を使用したパフォーマンスの監視

Oracle Enterprise Manager には、Real Application Clusters データベースで実行されるアプリケーションを監視するためのパフォーマンス・チャートが用意されています。パフォーマンス・チャートは、アクティブなすべてのインスタンスから情報を収集して集計し、単一の包括的なビューを示します。このチャートを使用して、クラスタ・データベースとクラスタ・データベース・インスタンスの両方を監視します。

Statspack を使用したパフォーマンスの監視

Statspack は、Real Application Clusters の統計を表示して、長期間にわたるパフォーマンスの傾向を示します。Statspack で表示するのは、パフォーマンス特性を示すグローバル・キャッシュ・サービス (GCS) の統計、およびグローバル・エンキュー・サービス (GES) の統計です。Statspack は、GCS と GES のメッセージ機能の統計も表示します。

関連項目： Statspack および Real Application Clusters のパフォーマンスの監視については、『Oracle9i Real Application Clusters 配置およびパフォーマンス』を参照してください。

Real Application Clusters のバックアップとリカバリの管理

Real Application Clusters は、Oracle Recovery Manager (RMAN) のすべての機能をサポートしています。アーカイブ・ログ・スキームを『Oracle9i Real Application Clusters 管理』で説明するとおり構成し、クラスタ・データベース内のすべてのインスタンスですべてのアーカイブ・ログが読取り可能になると、Recovery Manager を使用して、バックアップ操作とリカバリ操作を自動化できます。

関連項目： Recovery Manager の使用方法については、『Oracle9i Recovery Manager ユーザーズ・ガイド』を参照してください。

第 IV 部

高可用性および Real Application Clusters

第 IV 部では、高可用性を提供する Oracle 製品および機能について説明します。第 IV 部に含まれる章は、次のとおりです。

- [第 10 章「Real Application Clusters における高可用性の概念と実践」](#)

Real Application Clusters における高可用性 の概念と実践

この章では、Real Application Clusters における高可用性の実装の概念と、その実践について説明します。この章の内容は、次のとおりです。

- 高可用性について
- 高可用性のための Real Application Clusters の構成
- 災害時計画
- 障害保護の検証
- フェイルオーバーおよび Real Application Clusters
- Real Application Clusters でのフェイルオーバー処理
- 高可用性の構成
- 高可用性の配置

高可用性について

常に可用性を提供するように構成されたコンピュータ環境を、**高可用性**システムといいます。そのようなシステムは、通常、障害が発生してもシステムを使用できるようにするための冗長ハードウェアおよび冗長ソフトウェアを備えています。適切に設計された高可用性システムには、致命的な障害箇所は1箇所ありません。障害が発生する可能性のあるハードウェアまたはソフトウェアのコンポーネントはどれでも、同じタイプの冗長コンポーネントを備えています。

障害が発生すると、障害コンポーネントが実行していた処理が、**フェイルオーバー**・プロセスによって、バックアップ・コンポーネントに移されます。このプロセスは、システム全体のリソースを再マスター化し、部分トランザクションまたは障害トランザクションをリカバリして、システムを通常の状態に戻します。可能な場合、これらは100万分の数秒のうちに実行されます。フェイルオーバーがユーザーに対して透過的であるほど、システムの可用性は高いといえます。

Oracle には、高可用性を提供する製品と機能が用意されています。これらの製品と機能には、Real Application Clusters、**Oracle Real Application Clusters Guard I**、**Oracle Real Application Clusters Guard II**、Oracle Replication および **Oracle9i Data Guard** が含まれます。これらの機能を組み合わせて使用することで、特定の高可用性ニーズを満たすことができます。Real Application Clusters システムは高可用性の環境であり、計画的な停止および計画外の停止の際に連続的にサービスを提供できます。Real Application Clusters Guard II は、計画外の障害発生時や、オンライン・メンテナンスを実行中も連続的なサービスを提供します。

関連項目： これらの機能の詳細は、『Oracle9i Real Application Clusters Real Application Clusters Guard I - Concepts and Administration』および『Oracle9i Real Application Clusters Guard II Concepts, Installation, and Administration』を参照してください。

高可用性のための Real Application Clusters の構成

Real Application Clusters は、Oracle の標準機能の上に高レベルの可用性を構築します。ファスト・スタート・リカバリやオンライン再編成など、すべてのシングル・インスタンスの高可用性機能は、Real Application Clusters にも適用されます。ファスト・スタート・リカバリを使用すると、オンライン・アプリケーションのパフォーマンスへの影響を最小限に抑えながら、平均リカバリ時間 (MTTR) を大幅に削減できます。オンライン再編成によって計画的な停止時間の長さが短縮され、ユーザーが基礎となるオブジェクトを更新しているときも、多くの操作をオンラインで実行できます。

これらの機能に加えて、Real Application Clusters では、クラスタ化によって提供される冗長性を利用して、 n ノードのクラスタ内の $n-1$ ノード障害に対しても可用性を提供します。つまり、すべてのユーザーは、クラスタ内に1つでも使用可能なノードがあれば、すべてのデータにアクセスできます。Real Application Clusters を高可用性システムとして構成するには、次の項で説明するように、クラスタのハードウェア・コンポーネントとソフトウェア・コンポーネントについて考慮してください。

クラスタ・コンポーネントおよび高可用性

ここでは、次の高可用性およびクラスタ・コンポーネントの項目について説明します。

- クラスタ・ノード
- クラスタ・インターコネクト
- データベース・ソフトウェア

関連項目： これらのコンポーネントの詳細は、[第2章](#)を参照してください。

クラスタ・ノード

Real Application Clusters 環境は完全に冗長であり、すべてのノードがすべてのデータベースにアクセスします。1つのノードでの障害は、別のノードがトランザクションを処理する能力に影響しません。障害のないノードがクラスタに1つでもあれば、すべてのデータベース・クライアントは、すべてのトランザクションを処理できます。ただし、クライアントでは、障害が発生していないノードの容量制限によって応答時間が増加します。

クラスタ・インターコネクト

インターコネクトの冗長性は、クラスタの中でよく見落とされます。これは、平均障害時間 (MTTF) が、通常、数年に及ぶためです。そのため、クラスタ・インターコネクトの冗長性の優先順位が低い場合があります。また、システムと複雑さのレベルに従って、冗長なクラスタ・インターコネクトのコストが高くなる場合があります。

ただし、冗長なクラスタ・インターコネクトは、完全冗長クラスタにとって重要です。これがないと、システムに単一の致命的な障害箇所がないとはいいきれません。クラスタ・インターコネクトの障害は、様々な理由で発生する可能性があり、それらすべてを防ぐことはできません。

データベース・ソフトウェア

Real Application Clusters では、Oracle 実行可能プログラムが、クラスタ・ファイル・システムまたは各ノードのローカル・ディスクのいずれかにインストールされ、クラスタの各ノードで1つ以上のインスタンスが実行されます。ご使用のプラットフォームでクラスタ・ファイル・システム (CFS) がサポートされ、その CFS を使用している場合、インストールされる Oracle Real Application Clusters ソフトウェアのコピーは1つのみです。すべてのインスタンスはすべてのデータに同様にアクセスでき、どのようなトランザクションでも処理できます。このように、Real Application Clusters では、完全なデータベース・ソフトウェア冗長性が保証されています。

災害時計画

Real Application Clusters は、基本的に単一サイトの、高可用性ソリューションです。つまり、クラスタ内のノードは、一般的に、同じ部屋ではないとしても、同じビルの中に存在します。そのため、災害時計画は重要です。システムがどれほどミッション・クリティカルであるか、また、システムのある場所がそれらの災害に対してどれほど耐えられるかによって、災害時計画は、高可用性の重要なコンポーネントとなる場合があります。

Oracle では、Oracle9i Data Guard などの他のソリューションを提供し、包括的な災害時リカバリ計画を促進しています。これらのソリューションは、1つのクラスタがプライマリ・データベースのホストとして機能し、別のリモート・システムまたはクラスタが災害時リカバリ・データベースのホストとして機能する Real Application Clusters で使用できます。ただし、災害時リカバリ用の Real Application Clusters は、いずれのサイトにも必要ありません。

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。

障害保護の検証

システム・レベルの問題を慎重に考慮した後、ご使用の Real Application Clusters 環境が、障害から最も効果的に保護されていることを検証してください。障害保護システムの計画とトラブルシューティングには、次に示す障害のポイントを使用してください。

- クラスタ・コンポーネント
- CPU
- メモリー
- インターコネクト・ソフトウェア
- オペレーティング・システム
- Cluster Manager
- Oracle データベース・インスタンス・メディア
- 破損または消失した制御ファイル、ログ・ファイルまたはデータ・ファイル
- 削除されたデータベース・オブジェクト
- 人為的エラー

Real Application Clusters 環境を使用すると、クラスタ・コンポーネント障害およびソフトウェア障害から保護できます。ただし、メディア障害および人為的エラーは、システム停止時間の原因となる場合があります。Real Application Clusters は、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、一連のファイルを操作します。このため、メディア障害による悪影響を回避するには、効果的な対策を採用する必要があります。

RAID ベースの冗長性を使用することによって、ファイルの消失は回避できますが、ファイルの破損は回避できない場合があります。Real Application Clusters 環境で誤ってデータベース・オブジェクトを削除した場合、シングル・インスタンスのデータベースの場合と同じ方法でそのオブジェクトをリカバリできます。これらは、Real Application Clusters システムの主な制限です。これらの制限を除けば、このシステムは非常に強力な高可用性システムです。

システムを配置した後の重要な問題は、フェイルオーバーの透過性とその期間です。次の項で詳細を説明します。

フェイルオーバーおよび Real Application Clusters

ここでは、フェイルオーバーの基本、およびフェイルオーバーを高可用性システムで実装するために Real Application Clusters が提供する機能について説明します。内容は次のとおりです。

- フェイルオーバーの基本
- クライアント・フェイルオーバー
- 透過的アプリケーション・フェイルオーバーの使用
- サーバー・フェイルオーバー

フェイルオーバーの基本

フェイルオーバーには、正確なインスタンス監視機能または**ハートビート**・メカニズムを備えた高可用性システムが必要です。通常操作に対する機能に加えて、システムはフェイルオーバーの間、すばやく正確にリソースを同期化する必要があります。

同期化のプロセス、つまり**再マスター化**では、そのシステムでマスター化されたリソースの制御を的確に想定すると同時に、障害システムを正常に停止する必要があります。Real Application Clusters では、システムは、ローカルと同様に、リモートのノードにもリソース情報を記録します。これによって、フェイルオーバーおよびリカバリに必要な情報が、リカバリするインスタンスに使用可能になります。

関連項目：

- システムに Oracle Real Application Clusters Guard I を設定する方法については、『Oracle9i Real Application Clusters Real Application Clusters Guard I - Concepts and Administration』を参照してください。
- Oracle Real Application Clusters Guard II の使用方法については、『Oracle9i Real Application Clusters Guard II Concepts, Installation, and Administration』を参照してください。
- Real Application Clusters でのリカバリの詳細は、『Oracle9i Real Application Clusters 管理』を参照してください。
- リカバリの詳細は、『Oracle9i Recovery Manager ユーザーズ・ガイド』を参照してください。

フェイルオーバーの所要時間

フェイルオーバーの所要時間には、システムがシステム全体のリソースを再マスター化するための時間、および障害からリカバリするための時間が含まれます。フェイルオーバー・プロセスにかかる時間は、認定されたプラットフォーム上では比較的短い時間で済む場合があります。

- すでに処理を行っているユーザーの場合、フェイルオーバーは、サーバー・フェイルオーバー処理およびクライアント・フェイルオーバー処理の両方を意味します。
- 新しく処理を開始するユーザーの場合、フェイルオーバーはサーバー・フェイルオーバーの処理時間のみを意味します。

クライアント・フェイルオーバー

データベース・クライアント接続からシステム障害を隠すことは重要です。このような接続には、クライアント・サーバー環境のアプリケーション・ユーザー、またはマルチティア・アプリケーション環境のミドルティア・データベース・クライアントが関係します。適切に構成されたフェイルオーバー・メカニズムは、クライアント・セッションの経路をクラスター内の使用可能なノードに透過的に変更します。Oracle データベースのこの機能を、透過的アプリケーション・フェイルオーバーといいます。

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）では、接続が失敗した場合、アプリケーション・ユーザーは自動的にデータベースへ再接続できます。アクティブ・トランザクションはロールバックしますが、別のノードを使用して確立された新しいデータベース接続は、元のものと同じです。これは、接続が失敗した原因にかかわらず同じです。

透過的アプリケーション・フェイルオーバーによって影響を受ける要素

次に、アクティブなデータベース接続に対応付けられた要素を示します。

- クライアント / サーバー・データベース接続
- コマンドを実行するユーザーのデータベース・セッション
- フェッチに使用されるオープン・カーソル
- アクティブ・トランザクション
- サーバー側プログラム変数

透過的アプリケーション・フェイルオーバーでは、これらの要素の一部が自動的にリストアされます。たとえば、通常のクライアント / サーバー・データベース操作の間、クライアントはクライアントとサーバーが通信できるようにデータベースへの接続を維持します。サーバーに障害が発生すると、接続にも障害が発生します。クライアントが次にその接続を使用しようとする、エラーが発生します。この場合、ユーザーはデータベースにログインしなおす必要があります。

ただし、透過的アプリケーション・フェイルオーバーがあれば、Oracle はデータベースへの新しい接続を自動的に取得します。これによって、ユーザーは、元の接続に障害がなかったかのように作業を継続できます。したがって、透過的アプリケーション・フェイルオーバーでは、アプリケーションに使用できるアクティブなインスタンスが 1 つでも残っていれば、クライアントは接続の切断に気付きません。

関連項目： 透過的アプリケーション・フェイルオーバーの背景と構成に関する情報は、『Oracle9i Net Services 管理者ガイド』を参照してください。

透過的アプリケーション・フェイルオーバーの使用

クライアント・セッションをフェイルオーバーする機能は、透過的アプリケーション・フェイルオーバーの重要なメリットですが、透過的アプリケーション・フェイルオーバーには、システムの可用性も向上させるというメリットがあります。次の内容を説明します。

- [トランザクションの停止](#)
- [データベースの静止](#)
- [ロード・バランシング](#)
- [フェイルオーバー中のデータベース・クライアント処理](#)
- [透過的アプリケーション・フェイルオーバーの制限事項](#)

トランザクションの停止

メンテナンスまたは修正のために、ノードをサービスから外す必要がある場合もあります。たとえば、アプリケーション・クライアントへのサービスに割り込まずに、パッチ・リリースを適用する必要がある場合があります。トランザクションの停止によって、データベース全体ではなく選択したノードを簡単に停止できます。トランザクションの停止には、次の2つのオプションを使用できます。

- すべての既存のトランザクションが完了するまで停止イベントを遅延するように、SHUTDOWN 文の TRANSACTIONAL 句を使用して、ノードをサービスから削除します。このように、クライアント・セッションは、トランザクションの境界で、クラスタの別のノードに移行することができます。
- SHUTDOWN 文の TRANSACTIONAL LOCAL 句を使用して、特定のローカル・インスタンスでトランザクションの停止を実行します。この文を使用すると、新しいトランザクションがローカルで開始されないようにし、すべてのローカル・トランザクションの完了直後に停止を実行することができます。このオプションを使用すると、選択したインスタンスをトランザクション終了とともに停止して、すべてのセッションをインスタンス間で適切に移動できます。

トランザクションの停止が実行された後、新しく発行されたトランザクションは別のノードに転送されます。すべての既存のトランザクションが完了したとき、ノードで即時に停止が実行されます。

関連項目： 10-10 ページ「[停止中の透過的アプリケーション・フェイルオーバー処理](#)」

データベースの静止

管理タスクを実行する場合、同時ユーザーのトランザクションまたは問合せからこのタスクを切り離す必要があります。このためには、データベース静止機能を使用できます。たとえば、この機能を使用すると、管理タスクを実行するために、データベースを停止して、制限モードで再オープンする必要がなくなります。

このためには、ALTER SYSTEM 文で、QUIESCE RESTRICTED 句を使用します。QUIESCE RESTRICTED 句を使用すると、同時ユーザーのトランザクションまたは問合せから切り離して管理タスクを実行できます。

注意： あるノードで静止しているデータベースは、別のインスタンスでオープンできません。ALTER SYSTEM QUIESCE RESTRICTED 文を発行しても、処理が完了していない場合、そのデータベースをオープンできません。データベースがすでに静止状態の場合も、そのデータベースをオープンできません。

関連項目： データベース静止機能の詳細は、『Oracle9i Real Application Clusters 管理』および『Oracle9i データベース管理者ガイド』を参照してください。ALTER SYSTEM QUIESCE RESTRICTED 構文の詳細は、『Oracle9i SQL リファレンス』を参照してください。

ロード・バランシング

データベースは、トランザクションを適時に処理した場合に使用可能といえます。負荷がノードの容量を超えた場合、クライアント・トランザクション応答時間が影響を受け、データベースの可用性が低下します。そこで、応答時間とアプリケーションの可用性を維持するために、クライアント・セッションを負荷が少ないノードに手動で移行することが必要になります。

Real Application Clusters では、Transport Network Services (TNS) リスナー・ファイルが、ノード全体にわたる自動ロード・バランシングを共有サーバーと専用サーバーの両方の構成で提供します。インスタンス間での登録を制御するパラメータは動的であるため、Real Application Clusters のロード・バランシング機能によって、クラスタ構成の変更に対する調整が自動的に行われます。たとえば、クラスタ・データベースにノードを追加すると、クラスタ内のすべてのリスナー・ファイルは、新しいノードのリスナー情報で更新されます。

フェイルオーバー中のデータベース・クライアント処理

問合せクライアントに対するフェイルオーバー処理は、データ操作言語クライアントの場合と異なります。いずれの場合も、フェイルオーバー処理中の重要な問題は、障害が既存のクライアント接続から、できるだけ見えないようにすることです。次の項では、両方の種類のフェイルオーバー処理について説明します。

問合せクライアント フェイルオーバーの際は、処理中の問合せは再発行され、最初から処理されます。このため、元の問合せの完了に時間がかかった場合、次の問合せの所要時間が延長される場合があります。透過的アプリケーション・フェイルオーバー (TAF) では、障害は、問合せクライアントに通知されません。クライアントに対する影響は、応答時間の増加のみです。クライアントの問合せが、そのクライアントが再接続した障害のないノードのバッファ・キャッシュ内のデータで対応できる場合、応答時間の増加は最小限に抑えられます。TAF の PRECONNECT メソッドを使用することで、障害のないインスタンスへの再接続が不要となり、応答時間をさらに短縮できます。ただし、PRECONNECT は、フェイルオーバー・イベント待ちのリソースを割り当てます。

フェイルオーバー後、データ・ファイルへのアクセスは、サーバー側のリカバリが完了してから許可されます。サーバー側のリカバリがまだ完了していなければ、クライアント・トランザクションでは、サーバー側のリカバリが完了するまでシステムが休止されます。

OCI コールを介してコールバック関数を使用し、クライアントが遅延を障害と誤解しないように、クライアントにフェイルオーバーを通知することもできます。これによって、クライアントが接続を手動で再開させないようにできます。

データ操作言語クライアント データ操作言語（DML）のデータベース・クライアントは、INSERT、UPDATE および DELETE の各操作を実行します。Oracle は特定のエラーを処理し、これらのエラーが戻されたときに再接続を実行します。

このアプリケーション・コードを使用しない場合、障害インスタンスでの INSERT、UPDATE および DELETE 操作は、未処理の Oracle エラー・コードを戻します。トランザクションを再発行すると、Oracle は、クライアント接続を障害のないインスタンスに転送します。クライアント・トランザクションは、サーバー側のリカバリが完了するまで一時的に停止します。

停止中の透過的アプリケーション・フェイルオーバー処理

停止処理の完了後、ネットワーク上の問合せはフェイルオーバーされます。ただし、Oracle は、停止中の問合せに関するエラーを戻します。したがって、TAF は、オペレーティング・システムがネットワーク・エラーを戻し、インスタンスが完全に停止している場合のみ機能します。

トランザクションの停止で TAF を使用するアプリケーションには、エラー「ORA-01033 Oracle の初期化またはシャットダウン中です。」を処理するコードを記述する必要があります。障害が発生した場合、停止処理が開始されると、インスタンスはエラー ORA-01033 を戻します。このようなアプリケーションは、複数の ORA-01033 エラーがレポートされても、失敗した操作を定期的に再試行する必要があります。停止処理が完了すると、TAF は、インスタンスへのネットワーク接続の障害を認識し、使用可能なインスタンスへの接続をリストアします。

接続時ロード・バランシングは、複数ディスパッチャ間のアクティブな接続の数を均衡化することによって、接続のパフォーマンスを向上します。シングル・インスタンスの Oracle 環境では、TNS リスナーが最も負荷が少ないディスパッチャを選択し、受信したクライアント要求を処理します。Real Application Clusters 環境の接続時ロード・バランシングには、複数インスタンス間でアクティブな接続の数を均衡化する機能もあります。

動的なサービス登録によって、リスナーは常にすべてのインスタンスとディスパッチャを、その位置にかかわらず把握しています。このロード情報によって、リスナーは、受信したクライアント要求の送信先インスタンスまたは（共有サーバー構成を使用している場合は）ディスパッチャを決定します。

共有サーバー構成でのリスナーは、次の条件を順に適用してディスパッチャを選択します。

1. 負荷が最も少ないノード
2. 負荷が最も少ないインスタンス
3. そのインスタンスに対して負荷が最も少ないディスパッチャ

専用サーバー構成でのリスナーは、次の順序でインスタンスを選択します。

1. 負荷が最も少ないノード
2. 負荷が最も少ないインスタンス

データベース・サービスが、複数のノード上に複数のインスタンスを持つ場合、リスナーは最も負荷が少ないノード上の最も負荷が少ないインスタンスを選択します。共有サーバーを構成している場合は、選択されたインスタンスの最も負荷が少ないディスパッチャが選択されます。

関連項目： ロード・バランシングの詳細は、『Oracle9i Net Services 管理者ガイド』を参照してください。

透過的アプリケーション・フェイルオーバーの制限事項

接続に失敗すると、次のことが発生します。

- サーバー上のすべての PL/SQL パッケージの状態は、フェイルオーバー時に消失します。
- ALTER SESSION 文が消失します。
- トランザクション処理中にフェイルオーバーが発生した場合、ユーザーが OCITransRollback コールを発行するまで、後続の各コールはエラー・メッセージの原因となります。その後、Oracle は、Oracle Call Interface (OCI) の成功メッセージを発行します。このメッセージを確認して、追加の操作が必要かどうかを判断してください。
- Oracle がデータベース接続をフェイルオーバーし、サービス名表記の FAILOVER_MODE セクションが TYPE=SELECT の場合、Oracle は問合せのフェイルオーバーも試行します。
- フェイルオーバー状態のカーソルで作業を継続すると、エラー・メッセージが表示される場合があります。

フェイルオーバー後の最初のコマンドが SQL SELECT 文または OCISmtFetch 文ではない場合、エラー・メッセージが表示されます。フェイルオーバーは、OCI リリース 8.0 以上を使用してプログラムされたアプリケーションにのみ効力があります。

サーバー・フェイルオーバー

Real Application Clusters のサーバー側のフェイルオーバー処理は、多くのサーバー・プラットフォームで使用可能な、ホストベースのフェイルオーバー・ソリューションとは異なります。次の項では、両方の種類のフェイルオーバー処理について説明します。

Real Application Clusters のフェイルオーバー

Real Application Clusters は、迅速なサーバー側フェイルオーバーを提供します。これは、Real Application Clusters の並行性（アクティブ・アクティブ・アーキテクチャ）によって実現されます。つまり、複数の Oracle インスタンスが複数ノードで並行にアクティブで、これらのインスタンスが同じデータベースへのアクセスを同期化します。

また、すべてのノードが並行性所有権を持ち、すべてのディスクにアクセスします。1つのノードで障害が発生した場合、クラスタ内の他のすべてのノードは、すべてのディスクへのアクセスを維持します。転送するためのディスク所有権はなく、データベース・アプリケーション・バイナリはすでにメモリーにロードされています。

データベースのサイズによって、フェイルオーバーの所要時間は異なります。データベースが大きいほど、または、そのデータ・ファイルのサイズが大きいほど、**Real Application Clusters** を使用して得られるフェイルオーバーの効果も大きくなります。

ホストベース・フェイルオーバー

多くのオペレーティング・システム・ベンダー、およびその他のクラスタ・ソフトウェア・ベンダーは、高可用性のアプリケーション・フェイルオーバー製品を提供しています。これらのフェイルオーバー・ソリューションは、特定のプライマリ・クラスタ・ノード上のアプリケーション・サービスを監視します。次に、必要に応じて、サービスをセカンダリ・クラスタ・ノードにフェイルオーバーします。ホストベースのフェイルオーバー・ソリューションには、一般的に、任意のデータベース・アプリケーションのために有用な作業を実行する1つのアクティブ・インスタンスがあります。セカンダリ・ノードがプライマリ・ノードのアプリケーション・サービスを監視し、このプライマリ・ノードのサービスが使用できない場合にフェイルオーバーを開始します。

ホストベース・システムのフェイルオーバーには、通常、次の手順があります。

1. ハートビートを監視することによって障害を検出します。
2. **Cluster Manager** でクラスタ・メンバーシップを再編成します。
3. ディスクの所有権をプライマリ・ノードからセカンダリ・ノードに転送します。
4. アプリケーションおよびデータベース・バイナリ（**Oracle** 実行可能プログラム）を再起動します。
5. アプリケーションおよびデータベースのリカバリを実行します。
6. フェイルオーバー・ノードへのクライアント接続を再確立します。

Real Application Clusters でのフェイルオーバー処理

ここでは、Real Application Clusters のサーバー・フェイルオーバーのリカバリ処理について説明します。

- 障害の検出
- クラスタ・メンバーシップの再編成
- データベース・リカバリの実行

障害の検出

Real Application Clusters は、障害検出に Cluster Manager ソフトウェアを利用します。これは、Cluster Manager がハートビート機能をメンテナンスするためです。Cluster Manager が、動作していないノードの検出に要する時間は、構成可能なハートビート・タイムアウト・パラメータに関係します。

このパラメータの使用方法は、ご使用のプラットフォームによって異なります。デフォルト値は、使用するクラスタウェア（Sun Cluster や HP 社の ServiceGuard OPS Edition など）によって大きく異なります。タイムアウトの間隔の設定が低すぎると、クラスタが、一時的な障害のためにノードに障害が発生したと誤って判断する場合がありますため、このパラメータ値は、不適切な障害検出の数に悪影響を与えます。障害が検出されると、クラスタの再編成が行われます。

クラスタ・メンバーシップの再編成

ノードに障害が発生した場合、Oracle はそのノードのクラスタ・メンバーシップ状態を変更する必要があります。これを**クラスタ再編成**といい、通常、迅速に行われます。クラスタ再編成の所要時間は、そのクラスタ内の障害のないノードの数に比例します。

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）は、ソフトウェアに Cluster Manager インタフェースを提供し、ノードがクラスタに追加または削除されたときに、Oracle インスタンスにクラスタ・メンバーシップ・マップを公開します。クラスタ・ノード上の LMON プロセスは、各ノードの Cluster Manager と通信し、その情報をそれぞれのインスタンスに公開します。また、LMON は、次の 2 つの有効な機能も提供します。

- LMON を実行中のノードから、メッセージを継続的に送信する機能
- 共有ディスクに頻繁に書き込む機能

あるノードでこれらの機能の実行に失敗すると、他のノードでは、そのノードがクラスタのメンバーから外されたとみなします。この障害によって、クラスタ内のノードのメンバーシップ状態が変更されます。その後、LMON 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）のリソースの再マスター化やインスタンス・リカバリなども含めて、リカバリ処理を開始します。

この段階で、Real Application Clusters 環境はシステムの停止状態にあり、必要なリソースがないために未完了のクライアント・トランザクションは、Oracle がリカバリ処理を完了するまで中断されます。ただし、それ以外の処理中のトランザクションは、継続して処理されます。

インスタンス・メンバーシップ・リカバリ

インスタンス・メンバーシップ・リカバリ (IMR) のプロセスは、次の処理を実行して、クラスタのすべてのメンバーが機能していることを保証します。

- すべてのメンバー間で通信が実行可能であること、およびすべてのメンバーが応答可能であることを確認します。
- アクティブ状態ではないメンバーを削除するためのメカニズムを提供します。IMR は、メンバーシップを判定し、クラスタに属していないと判断したメンバーを削除します。
- 制御ファイルを使用して、メンバーシップを投票します。各メンバーは、制御ファイルにビットマップを書き込みます。これは、チェックポイント進行記録の一部です。
- 通信障害に基づいて、メンバーを削除します。IMR は、ハートビート・メッセージを定期的に制御ファイルに送信しないメンバー、または状態についての問合せに応答しないメンバーを、障害が発生しているメンバーであるとみなします。
- メンバーシップ投票を処理し、判定機能を使用して**制御ファイル投票結果レコード** (CFVRR) をロックします。

この CFVRR は、すべてのインスタンスで読み込まれます。メンバーがメンバーシップ・マップに存在しない場合、IMR は、ノードが異常終了しているとみなし、適切な診断情報が提供されます。IMR が現在構成されているため、すべてのメンバーは、ノードの停止が通知されるまで無期限で待機します。インスタンスの強制削除は行われません。Real Application Clusters のフォルト・トレランスの一部は、IMR 判定機能自体に障害が発生した場合のために用意されています。

データベース・リカバリの実行

インスタンスに障害が発生した場合、Oracle では、次の各項で説明するように、障害インスタンスの GCS リソースを障害が発生していないクラスタ・ノード上で再マスター化し、インスタンス・リカバリを実行する必要があります。

- **障害インスタンスのグローバル・キャッシュ・サービス・リソースの再マスター化**
- **インスタンス・リカバリ**

障害インスタンスのグローバル・キャッシュ・サービス・リソースの再マスター化

リソースの再マスター化に必要な時間は、障害インスタンス内の GCS リソースの数に比例します。この数は、バッファ・キャッシュのサイズに依存します。

この段階では、以前障害インスタンスでマスター化されたすべてのリソースが、残りのインスタンス全体に再分散されます。これらのリソースは、新しいマスター・インスタンスで再構築されます。以前障害のないインスタンスでマスター化されたその他すべてのリソースには、影響はありません。どのようなリソース要求の場合でも、要求がローカルで満たされる可能性は $1/n$ で、要求がリモート操作を含む可能性は $(n-1)/n$ です。

クラスタ・データベース内に障害のないインスタンスが 1 つのみ存在する場合は、すべてのリソース操作はローカルで実行されます。障害インスタンスの GCS リソースの再マスター化が完了すると、Oracle は、その障害インスタンスの実行中のトランザクションをリカバリします。これを **インスタンス・リカバリ** といいます。

インスタンス・リカバリ

インスタンス・リカバリには、キャッシュ・リカバリおよびトランザクション・リカバリが含まれます。インスタンス・リカバリでは、アクティブな Real Application Clusters インスタンスが障害を検出し、障害インスタンスのリカバリ処理を実行する必要があります。

LMON プロセスを使用して障害を検出した最初の Real Application Clusters インスタンスは、REDO ログ・ファイルを引き継ぐことによって障害インスタンスのリカバリを制御し、インスタンス・リカバリを実行します。このため、クラスタ・ファイル・システムのファイルまたは共有 RAW デバイスのいずれかには、REDO ログ・ファイルが必要です。

Oracle が障害インスタンスのオンライン REDO ログ・ファイルを再実行すると、インスタンス・リカバリが完了します。Oracle は、トランザクション・リカバリを遅延という方法で実行できるため、中断されたクライアント・トランザクションは、キャッシュ・リカバリの完了時に処理を開始できます。

関連項目： ブロック・メディア・リカバリ (BMR) の詳細は、『Oracle9i Recovery Manager ユーザーズ・ガイド』および『Oracle9i Recovery Manager リファレンス』を参照してください。

キャッシュ・リカバリ

キャッシュ・リカバリでは、Oracle は、障害インスタンスのオンライン REDO ログを再実行します。また、パラレル実行を使用したキャッシュ・リカバリを Oracle が行うように指定し、障害のある Oracle インスタンスの REDO ログをパラレル処理（スレッド）で再実行するようにできます。REDO ログを再実行する時間間隔を、予測可能な時間にしておくことも重要です。この時間間隔は、Oracle9i のファスト・スタート・リカバリ機能を使用して制御できます。

Oracle では、非ブロック化ロールバック機能も提供しています。これは、Oracle がオンライン・ログ・ファイルを再実行するとすぐに、完全なデータベース・アクセスを開始できるこ

とを意味します。キャッシュ・リカバリが完了すると、Oracle はトランザクション・リカバリを開始します。

関連項目： ファスト・スタート・リカバリの使用方法の詳細は、『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』を参照してください。

トランザクション・リカバリ

トランザクション・リカバリは、障害インスタンスのコミットされていないすべてのトランザクションのロールバックで構成されています。コミットされていないトランザクションとは、コミットされていない**進行中**のトランザクションです。

Oracle9i のファスト・スタート・ロールバック機能は、バックグラウンドで実行する遅延プロセスとしてロールバックを実行します。Oracle は、マルチバージョン読込み一貫性テクノロジーを使用して、異常終了したトランザクションによってブロックされていた行のみのオンデマンド・ロールバックを提供します。これによって、新しいトランザクションは、最小の遅延で実行できます。新しいトランザクションは、異常終了した長時間実行トランザクションがロールバックされるまで待機する必要はありません。したがって、通常、大規模なトランザクションでも、データベースのリカバリ時間には影響しません。

キャッシュ・リカバリと同様に、Oracle9i のファスト・スタート・ロールバックは、異常終了したトランザクションをパラレルでロールバックします。ただし、シングル・インスタンスの Oracle データベースでは、1 つのノードの CPU を使用して異常終了したトランザクションをロールバックします。

Real Application Clusters には、クラスタを意識したファスト・スタート・ロールバック機能があります。これは、クラスタのすべての CPU ノードを使用してパラレル・ロールバック操作を実行する機能です。各クラスタ・ノードは、リカバリ・コーディネータおよびリカバリ・プロセスを起動して、パラレル・ロールバック操作を支援します。データベースが、パラレル・ロールバック操作のために、すべてのクラスタ・リソースを認識して使用するため、ファスト・スタート・ロールバック機能は、このようにクラスタを意識したものとなっています。

デフォルトの動作はトランザクション・リカバリを遅延させますが、クライアント・トランザクションの処理を許可する前に、トランザクション・リカバリを完了するようにシステムを構成することもできます。この場合、複数ノード間でトランザクション・リカバリをパラレル化する Real Application Clusters の機能は、その有効性がユーザーにとってさらに明確になります。

高可用性の構成

ここでは、Real Application Clusters の次の高可用性構成について説明します。

- デフォルトの n ノード構成
- 高可用性の基本構成
- 高可用性の共有ノード構成
- Real Application Clusters Guard II での完全なアクティブ構成

デフォルトの n ノード構成

Real Application Clusters の n ノード構成は、デフォルトの環境です。クラスタのすべてのノードは、クライアント・トランザクション処理に関連し、クライアント・セッションは、接続時にロード・バランシングが可能です。高可用性環境を作成するためにクラスタ・ノード間に負荷を分散させることによって、応答時間は、CPU やメモリーなどの使用可能なクラスタ・リソースに最大限に利用されます。

n ノード構成の利点

ノード障害が発生した場合、別のノードのインスタンスが必要なリカバリ処理を実行します。障害インスタンス上のデータベース・クライアントは、クラスタの障害のない ($n-1$) インスタンス間でバランシングが可能です。障害のない各インスタンスに対して増加する負荷を最小限に抑えて、応答時間を許容範囲内に保つことによって可用性を高めることができます。この構成では、データベース・アプリケーションのワークロードをすべてのノード間に分散し、クラスタ・マシン・リソースの最適な使用方法を提供できます。

高可用性の基本構成

2 ノード環境の Real Application Clusters で高可用性の基本システムを簡単に構成できます。1 つのノード上のプライマリ・インスタンスは、ユーザー接続を受け入れますが、もう 1 つのノード上の**セカンダリ・インスタンス**は、プライマリ・ノードに障害が発生した場合、または INSTANCE_ROLE パラメータを介して特別に選択された場合に、接続を受け入れます。この構成は、特定のインスタンスへのトランザクションのルーティングを手動で制御することによって可能です。ただし、Real Application Clusters では、これを自動的行う**プライマリ / セカンダリ・インスタンス構成**機能を提供しています。

プライマリ / セカンダリ・インスタンス構成

プライマリ / セカンダリ・インスタンス機能は、**initsid.ora** の ACTIVE_INSTANCE_COUNT パラメータに 1 を設定することによって構成します。2 ノード環境では、データベースを最初にマウントするインスタンスが**プライマリ・インスタンス・ロール**とみなされます。もう 1 つのインスタンスは**セカンダリ・インスタンス・ロール**とみなされます。**プライマリ・インスタンス**に障害が発生した場合、セカンダリ・インスタンスがプライマリのロールを担い

ます。障害インスタンスがアクティブ状態に戻ったとき、それはセカンダリ・インスタンスになります。

リモート・クライアントとプライマリ/セカンダリ構成 セカンダリ・インスタンスがプライマリ・インスタンスになるのは、Cluster Manager からプライマリ・インスタンスの障害を通知された場合のみです。この移行は、GCS と GES の再構成およびキャッシュ・リカバリ処理とトランザクション・リカバリ処理が開始される前に発生します。障害のないインスタンスへのリダイレクションは透過的に実行され、アプリケーションのプログラミングは必要ありません。クライアント接続文字列へのわずかな構成変更のみが必要です。

プライマリ / セカンダリ・インスタンス構成では、他の n ノードの Real Application Clusters 環境と同様に、両インスタンスが並行して実行されます。ただし、データベース・アプリケーション・ユーザーは、指定されたプライマリ・インスタンスにのみ接続できます。プライマリ・ノードは、GCS と GES のすべてのリソースをマスター化します。このマスター化によって、ノード間の通信が最小限になり、従来のシングル・インスタンス・データベースと同等のパフォーマンス・レベルが提供されます。

セカンダリ・インスタンスは、管理クライアントという特別に構成されたクライアントによって、バッチ問合せレポート操作またはデータベース管理作業に使用される場合があります。これによって、セカンダリ・ノードがある程度使用可能になります。また、プライマリ・インスタンスの CPU の負荷を軽減するため、冗長ノードを導入するメリットも証明されます。

プライマリ / セカンダリ・インスタンス構成は、専用サーバー環境と共有サーバー環境の両方で動作します。ただし、次に説明するとおり、それぞれ異なる働きをします。

- 専用サーバー環境でのプライマリ / セカンダリ・インスタンス構成
- プライマリ / セカンダリ・インスタンス構成と共有サーバー

専用サーバー環境でのプライマリ/セカンダリ・インスタンス構成

現行の高可用性構成の場合、専用サーバー環境では、インスタンス間のリスナー登録は使用しません。特定のインスタンスのリスナーへの接続要求は、単に、そのインスタンスのサービスに接続されます。この動作は、専用サーバー環境のデフォルトの n ノード構成と同様です。

図 10-1 に、ノード障害が発生する前のクラスタ構成を示します。

1. SALES1 は、リスナーに接続中です。
2. クライアントは、リスナーに接続中です。
3. SALES1 がプライマリ・インスタンスになります。

図 10-1 専用サーバー環境でのプライマリ/セカンダリ構成

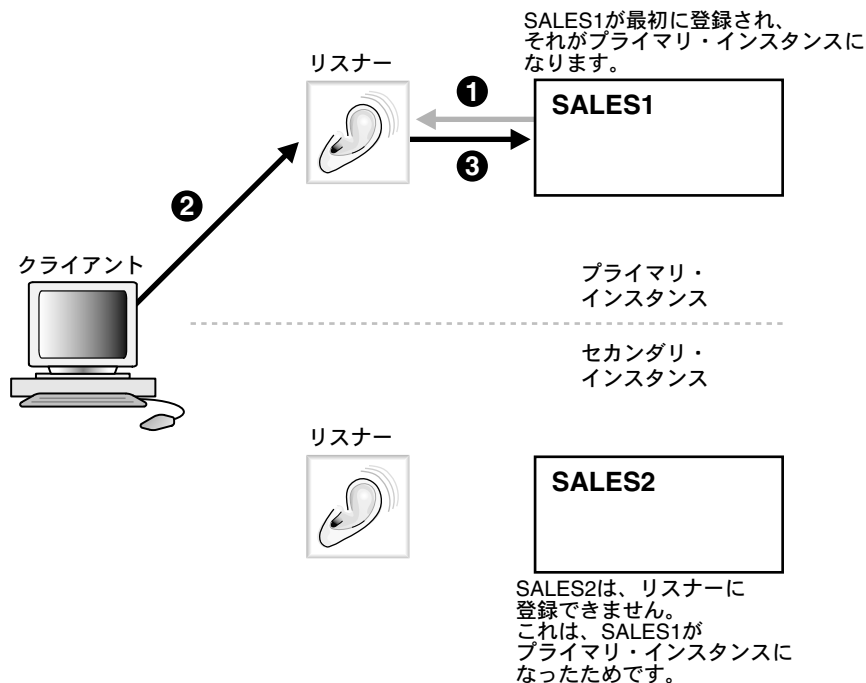
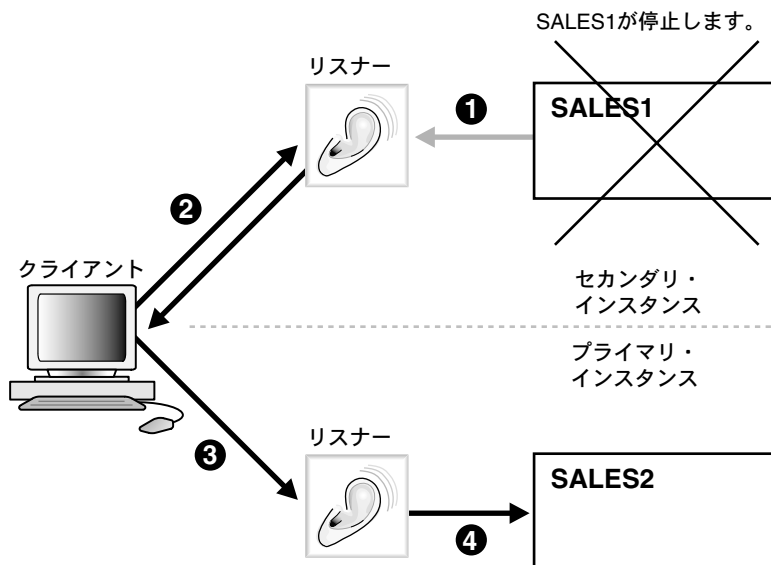


図 10-2 に示すとおり、プライマリ・インスタンスで障害が発生した場合は、次の手順が実行されます。

1. SALES1 での障害がクラスタ全体に通知されます。
2. クライアントからの再接続要求は障害インスタンスのリスナーによって拒否されます。
3. セカンダリ・インスタンスはリカバリを実行し、プライマリ・インスタンスになります。
4. クライアント要求を再発行すると、クライアントは新しいプライマリ・インスタンスのリスナーを介して接続を再確立します。新しいプライマリ・インスタンスのリスナーは、クライアントを新しいプライマリ・インスタンスに接続します。アドレス・リストを使用するか、またはクライアントが接続フェイルオーバーを使用するように構成されていると、接続は自動的に再確立されます。

図 10-2 専用サーバーのプライマリ/セカンダリ構成とノード障害



プライマリ/セカンダリ・インスタンス構成と共有サーバー

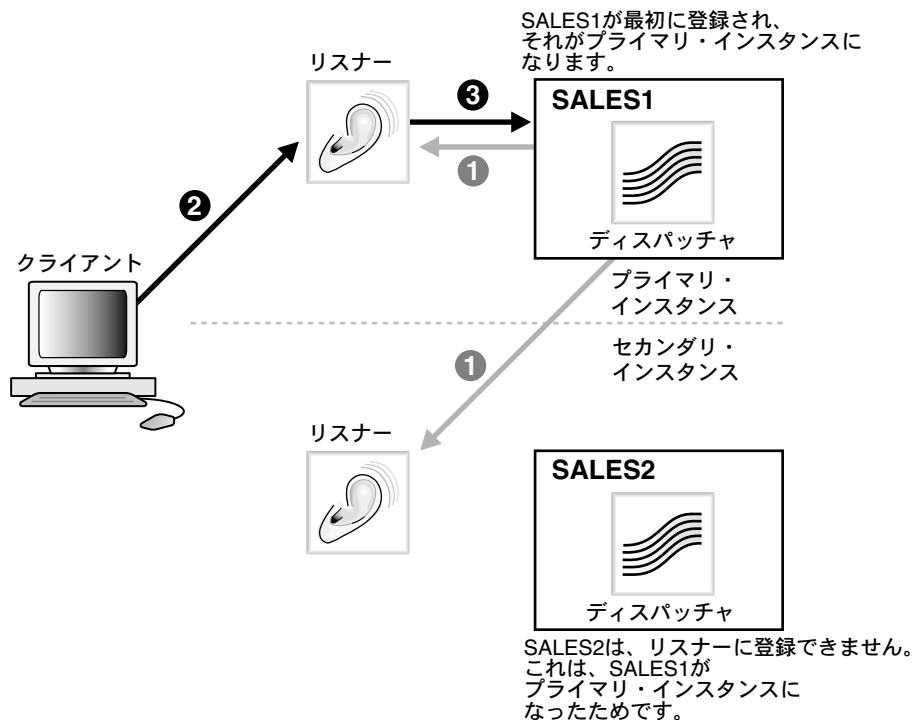
Real Application Clusters では、共有サーバー・モードで実行した場合、再接続パフォーマンスの効果が得られます。これは、クラスタ内のすべてのディスパッチャおよびリスナーのクロス登録によって行われます。

プライマリ / セカンダリ構成では、[図 10-3](#) に示すとおり、プライマリ・インスタンスのディスパッチャが、両方のリスナーにプライマリ・インスタンスとして登録します。

- クライアントはいずれのリスナーにも接続できます（図にはプライマリ・ノードのリスナーへの接続のみが示されています）。
- クライアントがリスナーに接続します。
- その後、リスナーは、クライアントをディスパッチャに接続します（図にはプライマリ・ノードのリスナー / ディスパッチャ接続のみが示されています）。

関連項目： クライアント接続文字列の構成については、『Oracle9i Real Application Clusters セットアップおよび構成』を参照してください。

図 10-3 共有サーバー環境でのプライマリ/セカンダリ構成



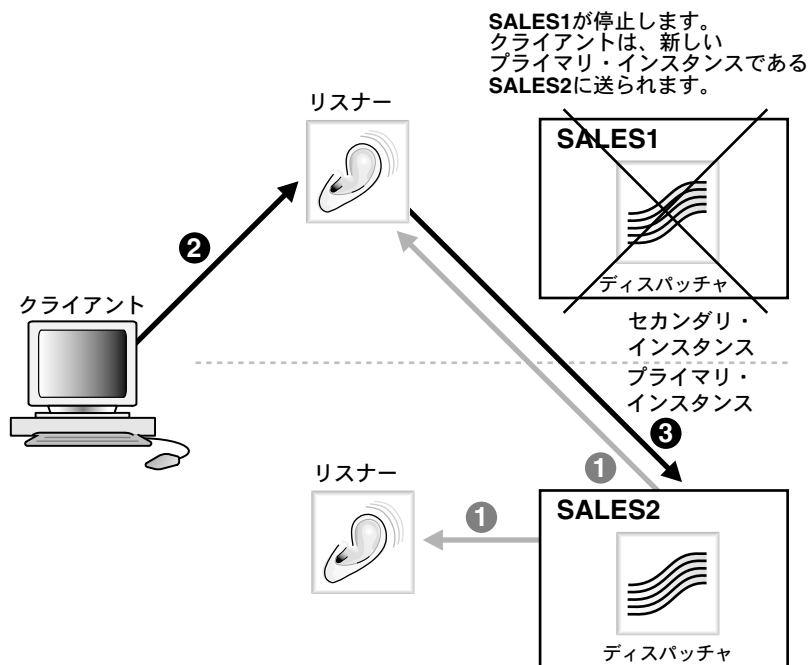
特別に構成されたクライアントは、セカンダリ・インスタンスを使用してバッチ操作ができます。たとえば、バッチ・レポート作業または索引作成操作は、セカンダリ・インスタンス上で実行できます。

関連項目： セカンダリ・インスタンスへの接続方法は、『Oracle9i Real Application Clusters 管理』を参照してください。

図 10-4 に、障害が発生したプライマリ・インスタンスが、新しいプライマリ・インスタンスにどのように置き換えられるかを示します。

1. プライマリ・ノードに障害が発生した場合、セカンダリ・インスタンスのディスパッチャが新しいプライマリ・インスタンスとしてリスナーに登録されます。
2. クライアントは、いずれかのリスナーを介してデータベースへの再接続を要求します。
3. リスナーは、新しいプライマリ・インスタンスのディスパッチャにその要求を送ります。

図 10-4 共有サーバーのプライマリ/セカンダリ構成とノード障害



セカンダリ・インスタンスのライブラリ・キャッシュのウォーミング

ライブラリ・キャッシュ内で頻繁に実行される SQL および PL/SQL 文に関する情報をメンテナンスすると、Oracle データベース・サーバーのパフォーマンスが改善されます。Real Application Clusters のプライマリおよびセカンダリ・インスタンス構成では、プライマリ・インスタンスに対応付けられたライブラリ・キャッシュに最新の情報が入れられます。セカンダリ・インスタンスのライブラリ・キャッシュを事前に移入しておかないと、フェイルオーバーが発生した場合にこの情報が無効になります。

DBMS_LIBCACHE パッケージを使用して、プライマリ・インスタンスのライブラリ・キャッシュにある情報を、セカンダリ・インスタンスのライブラリ・キャッシュに転送します。この処理を、**ライブラリ・キャッシュのウォーミング**といいます。新しいプライマリ・ライブラリ・キャッシュには、解析済 SQL 文およびコンパイル済 PL/SQL ユニットをロードする必要がないため、フェイルオーバー発生直後のパフォーマンスが向上します。

関連項目： ライブラリ・キャッシュのウォーミング機能のインストールと構成の詳細は、『Oracle9i Real Application Clusters Real Application Clusters Guard I - Concepts and Administration』を参照してください。また、DBMS_LIBCACHE の使用方法の詳細は、『Oracle9i PL/SQL パッケージ・プロシージャおよびタイプ・リファレンス』を参照してください。

高可用性の基本構成の利点

この例で、デフォルトの2ノード構成ではなく、プライマリ / セカンダリ・インスタンス機能を使用する理由はいくつかあります。プライマリ / セカンダリ・インスタンス機能は、次のことを提供します。

- n ノード構成にアップグレードするための実行可能な移行パス
- 1つのノードを超えて拡張する必要のないアプリケーションに対する、高可用性のソリューション
- シングル・インスタンス・データベースと同等のパフォーマンス
- シングル・インスタンスのアプリケーション環境から Real Application Clusters 環境への段階的な移行

高可用性の共有ノード構成

n ノード構成で Real Application Clusters を実行すると、クラスタ・リソースを最も効果的に利用できます。ただし、前述のとおり、これは常に可能または推奨できるとはかぎりません。一方、フェイルオーバーに備えてアイドル・ノードを持つための投資には、高い費用がかかる場合があります。このような状況の場合、高可用性の共有ノード構成が適しています。

このタイプの構成には、通常、複数のノードがあり、それぞれのノードが個別のアプリケーション・モジュールまたはサービスを実行しますが、すべてのアプリケーション・サービスで1つの Real Application Clusters データベースが共有されます。さらに、個別の指定ノードをフェイルオーバー・ノードとして設定できます。インスタンスがそのノードで実行している間は、通常の操作中にユーザーがそのノードに接続されることはありません。いずれかのノードに障害が発生した場合、Oracle は、そのワークロードをフェイルオーバー・ノードにリダイレクトできます。

この構成は別々のノードで実行する必要があるアプリケーションに有効ですが、中間層アプリケーションまたはトランザクション処理モニターが適切なアプリケーション・ユーザーを適切なノードに送る場合に、最も有効です。プライマリ / セカンダリ・インスタンス構成とは異なり、フェイルオーバー・ノードへのワークロードの移行を自動化するデータベース設定はありません。かわりに、アプリケーション、つまり中間層ソフトウェアは、ユーザーを障害アプリケーション・ノードからフェイルオーバー・ノードに送ります。また、このアプリケーションは、障害ノードが操作可能になった場合に、ユーザーのフェイルバックを制御する必要もあります。フェイルバックは、処理中のユーザー作業のあるフェイルオーバー・ノードを、引き続いて発生し得るノード障害を回避します。

高可用性の共有ノード構成の利点

この構成では、アプリケーション・パフォーマンスはフェイルオーバー時でも維持されます。それに対し n ノード構成では、同じワークロードがより小さいクラスター・ノード集合に再分散されるため、 $1/n$ の割合でアプリケーション・パフォーマンスが低下する場合があります。

Real Application Clusters Guard II での完全なアクティブ構成

高可用性と改善された管理機能は、Real Application Clusters Guard II でも利用できます。Real Application Clusters Guard II は完全なインスタンス環境です。この環境によって、サービスが実行されるすべてのインスタンスとそのフェイルオーバー・プロパティを制御できます。障害時に、Real Application Clusters Guard II は、サービスを中断することなく、アプリケーション・サービスの負荷を使用可能な別のノードに転送します。

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。

高可用性の配置

Real Application Clusters は、完全に冗長な耐障害環境を提供します。すべてのクラスター・ノードには、すべてのデータとリソースに同等にアクセス可能なアクティブ・インスタンスがあります。ノードに障害が発生した場合、ユーザーは、他のノードの障害のないインスタンスを使用して、データにアクセスできます。障害ノードの実行中のトランザクションは、障害を検出した最初のノードによってリカバリされます。このように、Real Application Clusters を使用すると、エンド・ユーザー・アプリケーションの可用性に対する中断を最小限に抑えることができます。

第 V 部

リファレンス

第 V 部に含まれる付録およびリファレンス情報は、次のとおりです。

- [付録 A「制限事項」](#)
- [付録 B「マルチブロック・ロック割当ての使用（オプション）」](#)
- [用語集](#)

制限事項

この付録では、Real Application Clusters の互換性の問題および制限事項について説明します。この付録の内容は、次のとおりです。

- [互換性](#)
- [制限付き SQL 文](#)
- [データ・ファイルの最大数](#)

互換性

Real Application Clusters は、**排他 (X) モード**で作成されたすべての Oracle データベースで動作します。排他モードの Oracle は、Real Application Clusters ソフトウェアが作成または変更したデータベースにアクセスできます。さらに、各インスタンスには、それ自体の REDO ログの集合が必要です。

制限付き SQL 文

クラスタ・データベース・モードでは、次の操作がサポートされません。

- データベースの作成 (CREATE DATABASE)
- 制御ファイルの作成 (CREATE CONTROLFILE)
- データベースのアーカイブ・モードの切替え (ALTER DATABASE の ARCHIVELOG および NOARCHIVELOG オプション)

これらの操作を実行するには、すべてのインスタンスを停止し、排他モードで 1 つのインスタンスを起動します。

データ・ファイルの最大数

Oracle がサポートするデータ・ファイルの数は、オペレーティング・システムによって異なります。この制限内で許可される最大値は、CREATE DATABASE コマンドで使用する値によって異なりますが、制御ファイルの物理サイズによって制限されます。この制限は排他モードとクラスタ・データベース・モードで同じですが、Real Application Clusters の追加インスタンスは、シングル・インスタンスのシステムよりも、ファイルの最大数を制限します。

関連項目：『Oracle9i SQL リファレンス』およびオペレーティング・システム固有の Oracle マニュアルを参照してください。

マルチブロック・ロック割当ての使用 (オプション)

この付録では、ロックを構成して複数ブロックを処理する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) とグローバル・エンキュー・サービス (GES) で実行される、Oracle Real Application Clusters のデフォルトのリソース制御方式を無効にするという非常にまれな環境の場合にのみ、参照してください。キャッシュ・フュージョンを無効にしてロックを使用することが適しているのは、まれなケースのみです。この付録の内容は、次のとおりです。

- [ロック設定を使用する場合](#)
- [ロック設定を使用する方法](#)
- [ロックの粒度](#)
- [ロック管理](#)

注意： GC_FILES_TO_LOCKS を設定する必要はありません。Real Application Clusters とキャッシュ・フュージョンによるリソース制御メカニズムによって、ロックを使用せずに最適なパフォーマンスが得られます。

ロック設定を使用する場合

この付録で説明する構成方法の大部分は、Real Application Clusters の**キャッシュ・フュージョン**に置き換えられています。キャッシュ・フュージョンを無効にしてロック設定を実行することが適しているのは、大量の読取り専用ブロックが存在する Real Application Clusters インストールの場合のみです。

ロック設定を使用する方法

ロックは、任意のクラスの 1 つ以上のブロックを管理できます。このブロックには、データ・ブロック、UNDO ブロック、セグメント・ヘッダーなどがあります。インスタンス間アクティビティの量（読込み / 書込み操作、書込み / 書込み操作など）、および対応する Real Application Clusters のパフォーマンスは、強制書込みの件数という観点から評価されます。キャッシュ・フュージョンを無効にすると、1 つのインスタンスが別のインスタンスに保持されているブロックを要求した場合に、強制書込みが発生する可能性があります。Oracle は、ブロックをディスクに書き込み、要求側インスタンスがブロックを最新の状態で読み込めるようにします。

関連項目： ロックの割当ての詳細は、『Oracle9i Real Application Clusters 配置およびパフォーマンス』を参照してください。

ロックの粒度

ロックの粒度には、1:1 ロック（デフォルト）と 1: n ロックの 2 つのレベルがあります。1: n ロックは、1 つのロックが、 n の値で定義されている複数のデータ・ブロックを管理することを示します。1: n ロックでは、少数のロックで多数のブロックを管理でき、ロック操作が減少します。読取り専用データの場合、パラレル実行などの特定の操作の間、1: n ロックは 1:1 ロックよりも実行速度が速くなります。GC_FILES_TO_LOCKS パラメータが有効なのは、1: n ロックの粒度を取得する場合のみです。

ロック管理

データ・ファイルに割り当てられたロック数、およびそれらのデータ・ファイル内のデータ・ブロック数によって、1つのロックによって管理されるデータ・ブロック数が決定します。

GC_FILES_TO_LOCKS パラメータをファイルに設定する場合、ロックごとのブロック数は、[図 B-1](#) に示すとおり、ファイル・レベルごとに計算できます。この例では、GC_FILES_TO_LOCKS=1:300,2:200,3-5:100 の値を想定しています。

図 B-1 GC_FILES_TO_LOCKS 設定に対するブロックへのロックの割当て例

ファイル1: $\frac{\text{ファイル1のブロック数}}{300\text{ロック}}$

ファイル2: $\frac{\text{ファイル2のブロック数}}{200\text{ロック}}$

ファイル3: $\frac{\text{合計 (ファイル3、ファイル4、ファイル5 のブロック数)}}{100\text{ロック}}$

ブロック数によるファイルのサイズが、それに割り当てられたロック数の倍数の場合、各 1:n ロックは、等式で求められるデータ・ブロック数と同数のデータ・ブロックを管理します。

ファイル・サイズが、ロック数の倍数でない場合、1:n ロックごとのデータ・ブロック数は、個々のデータ・ファイルに対して 1 つずつ異なります。たとえば、2,500 個のデータ・ブロックを含むデータ・ファイルに、400 個のロックを割り当てた場合、100 個のロックは 7 データ・ブロックずつ管理し、300 個のロックは 6 ブロックずつ管理します。GC_FILES_TO_LOCKS 初期化パラメータに指定されていないすべてのデータ・ファイルは、残りのロックを使用します。

n 個のファイルが同じ 1:n ロックを共有する場合、ロックごとのブロック数は最大で n 個異なります。GC_FILES_TO_LOCKS の分割句またはキーワード EACH を使用して、個々のファイルにロックを割り当てた場合、ロックごとのブロック数の違いは 1 以下になります。

1:n ロックをデータ・ファイルの集合にまとめて割り当てた場合、各ロックは通常、各ファイルの 1 つ以上のブロックを管理します。連続するブロックを指定した場合 (!blocks オプションを使用して)、またはファイルに含まれるブロック数が、ファイルの集合に割り当てたロック数よりも少ない場合には、例外が発生する場合があります。

関連項目： GC_FILES_TO_LOCKS パラメータの使用方法的詳細は、『Oracle9i Real Application Clusters 配置およびパフォーマンス』を参照してください。

用語集

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](#)上のクラスタ・メンバーシップに関する共通ビューを提供して、[ノード](#)のメンバーシップ状態を検出して追跡する。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](#)・プロセスは、CM への登録と登録解除を行う。

CR

「[読みみ一貫性 \(consistent read: CR\)](#)」を参照。

Data Guard

「[Oracle9i Data Guard](#)」を参照。

Database Configuration Assistant (DBCA)

データベースを作成および削除し、データベース・テンプレートを管理するための Oracle のツール製品の 1 つ。

DBCA

「[Database Configuration Assistant \(DBCA\)](#)」を参照。

DSS

「[意思決定支援システム \(Decision Support System: DSS\)](#)」を参照。

EMCA

「[Enterprise Manager Configuration Assistant \(EMCA\)](#)」を参照。

Enterprise Manager Configuration Assistant (EMCA)

Enterprise Manager の構成および設定を作成、削除および変更するためのツール。

Enterprise Manager

「[Oracle Enterprise Manager](#)」を参照。

GSD

「[グローバル・サービス・デーモン \(Global Services Daemon: GSD\)](#)」を参照。

HA

「[高可用性 \(high availability\)](#)」を参照。

IMR

「[インスタンス・メンバーシップ・リカバリ \(instance membership recovery: IMR\)](#)」を参照。

initdbname.ora

クラスタ全体に適用するグローバル・パラメータを含む初期化パラメータ・ファイル。

initSID.ora

[インスタンス](#)固有のパラメータを含み、データベース・パラメータ用の [initdbname.ora](#) を指す初期化パラメータ・ファイル。

IPC

「[プロセス間通信 \(interprocess communication: IPC\)](#)」を参照。

LCK プロセス (LCK process)

[インスタンス](#)のグローバル・エンキュー要求とインスタンス間のコール操作を管理するプロセス。複数の[グローバル・キャッシュ・サービス・プロセス](#)が存在する場合、ワークロードが自動的に共有され、負荷が平衡化される。

listener.ora

プロトコル・アドレスを識別するリスナー構成ファイル。リスナーは、リスニングの対象となる接続要求およびサービスを受け入れる。

LMD

「[グローバル・エンキュー・サービス・デーモン \(Global Enqueue Service Daemon: LMD\)](#)」を参照。

LMON

「[グローバル・エンキュー・サービス・モニター \(Global Enqueue Service Monitor: LMON\)](#)」を参照。

LMSn

「[グローバル・キャッシュ・サービス・プロセス \(Global Cache Service Processes: LMSn\)](#)」を参照。

N

「[NULL \(N\) モード \(null \(N\) mode\)](#)」を参照。

NULL (N) モード (null (N) mode)

保持プロセスがリソースを必要としていることを示す。ただし、アクセス権は、**排他 (X) モード**または**共有 (S) モード**の場合のみ付与される。

Oracle Database Configuration Assistant (DBCA)

「[Database Configuration Assistant \(DBCA\)](#)」を参照。

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され、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 Net

接続性を有効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核となる通信レイヤーおよびネットワーク・プロトコルのサポートを含む。サービスおよびそのアプリケーションを異なるコンピュータに常駐させ、ピア・アプリケーションとして通信できるようにする。

Oracle Net Configuration Assistant

インストール後に、基本ネットワーク・コンポーネントを構成するインストール後処理ツール。

Oracle Net Services

すべての Oracle ネットワーク・コンポーネントを含む用語。このコンポーネントには、[Oracle Net](#)、[リスナー](#)、Oracle Connection Manager、Oracle Names、[Oracle Net Configuration Assistant](#) および Oracle Net Manager が含まれる。

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、Real Application Clusters データベースで実行されている全インスタンスの集計パフォーマンスを表す。

Oracle Real Application Clusters

クラスタで共有データベースにアクセスできるようにする画期的なアーキテクチャ。Real Application Clusters には、Oracle9i Enterprise Edition 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。

Oracle Real Application Clusters Guard I

ファイルオーバー保護機能。Oracle Real Application Clusters Guard は Real Application Clusters の統合コンポーネントの 1 つである。Oracle Real Application Clusters Guard は、次の機能を提供する。

- Oracle **インスタンス**に障害が発生する問題からの高速リカバリおよびbounded リカバリ時間の自動設定。
- 特定の障害発生時の診断データの自動獲得。
- **プライマリ/セカンダリ・インスタンス構成**の強制。Oracle Net Services を通して接続するクライアントは、クラスタ内の別のノードに接続した場合でも、プライマリ・ノードに正しく転送される。
- 障害が発生した後の接続再確立時にクライアントで発生する遅延の回避。

Oracle Real Application Clusters Guard II

Real Application Clusters Guard II によって、可用性を維持するために、Real Application Clusters を使用するクラスタを監視できる。障害時に、Real Application Clusters Guard II は、アプリケーション・サービスの負荷を別のアクティブなノードに転送する。Real Application Clusters Guard II では、すべてのインスタンスはアクティブで、サービスをサポートできる。インスタンスは、サービスをサポートするために自動的に起動し、その間も別のサービスをサポートできる。

Oracle9i Data Guard

Oracle の**高可用性**製品。スタンバイ状態のバックアップ・データベースを提供する。この製品は、以前は**スタンバイ・データベース**と呼ばれていた。

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

「**システム識別子 (system identifier: SID)**」を参照。

OSD

「**オペレーティング・システム依存 (OSD) クラスタウェア (operating system-dependent (OSD) clusterware)**」を参照。

Performance Manager

「**Oracle Performance Manager**」を参照。

PFILE

「[パラメータ・ファイル \(parameter file: PFILE\)](#)」を参照。

PI

「[パスト・イメージ \(past image: PI\)](#)」を参照。

ping

「[ディスク書込みの強制実行 \(forced disk write\)](#)」と同義。

RAID

「[Redundant Array of Independent Disks \(RAID\)](#)」を参照。

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・ドライブ。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 パーティション \(raw partition\)](#)」も参照。

RAW パーティション (raw partition)

最低レベルでアクセスされる物理ディスクの一部。RAW パーティションは、拡張パーティションが作成され、それに対して物理パーティションがフォーマットされずに割り当てられた場合に作成される。フォーマット後は、[クックド・パーティション](#)と呼ばれる。「[RAW デバイス \(raw device\)](#)」も参照。

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。

Redundant Array of Independent Disks (RAID)

複数のハード・ディスク・ドライブを結合して、格納されている大量のデータへの高速アクセスを可能にするハードウェア・アーキテクチャ。

RMAN

「[Recovery Manager \(RMAN\)](#)」を参照。

S

「**共有 (S) モード (shared (S) mode)**」を参照。

SCUR

「**共有カレント (shared current: SCUR)**」を参照。

SID

「**システム識別子 (system identifier: SID)**」を参照。

SMP

「**対称型マルチ・プロセッサ (Symmetric Multi-Processor: SMP)**」を参照。

SRVCTL

管理者がインスタンスおよび **Real Application Clusters** データベースを管理するために使用するサービス制御 (SRVCTL) ユーティリティ。SRVCTL は、各ノード上にインストールされる。SRVCTL ユーティリティによって、**Oracle Enterprise Manager** のすべてのインスタンス情報が収集される。SRVCTL は、Oracle Intelligent Agent とノードの間における単一の制御ポイントとして機能する。SRVCTL との通信には、1 つのノードの Oracle Intelligent Agent のみが使用される。そのノード上の SRVCTL が、Java Remote Method Invocation (RMI) を通じて他のノードと通信する。

SV

「**順序番号値エンキュー (Sequence Number Value Enqueue: SV)**」を参照。

TAF

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

tnsnames.ora

ネット・サービス名が記述されているファイル。このファイルは、クライアント、ノード、コンソールおよび Oracle Performance Manager マシン上で必要である。

X

「**排他 (X) モード (exclusive (X) mode)**」を参照。

XCUR

「**排他カレント (exclusive current: XCUR)**」を参照。

空きリスト・グループ (free list group)

1 つ以上のインスタンスが使用する空きリストの集合。

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・システムに役立つデータベース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とプロセスでは、関連データベースのデータを管理し、データベース・ユーザーのために機能する。各インスタンスには、一意の**システム識別子**、インスタンス名、ロールバック・セグメントおよび**スレッド ID** が設定される。

インスタンス名 (instance name)

インスタンスの名前を表し、クラスタが共通のサービス名を共有する場合は、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。インスタンス初期化ファイル `init.ora` の `INSTANCE_NAME` パラメータによって識別される。このインスタンス名は、Oracle システム識別子 (SID) と同じである。「**システム識別子 (system identifier: SID)**」も参照。

インスタンス・メンバーシップ・リカバリ (instance membership recovery: IMR)

すべての**クラスタ**・メンバーが機能していることを保証するために Real Application Clusters で使用される方法。IMR はメンバーシップをポーリングし、判定を行う。制御ファイル・レコードでハートビートを示さないメンバー、または定期的な状態問合せに応答しないメンバーは異常終了しているとみなされる。インスタンス・メンバーシップ・リカバリ判定機能によって、クラスタ内のノードのメンバーシップの投票が処理され、**制御ファイル投票結果レコード**が作成される。

インターコネクト (interconnect)

ノード間の通信リンク。

エージェント (agent)

クライアント / サーバー・モデルにおけるシステムの一部。クライアントまたはサーバーのかわりに、情報の準備と交換を実行する。Intelligent Agent は、1 人以上のユーザーのかわりに、他のエージェントと通信して一部の共通タスクを実行できる自動プロセスを指す。

エンキュー (enqueue)

データベース・リソースへのアクセスをシリアル化する共有メモリ構造。Real Application Clusters が使用できない場合は、1 つの**インスタンス**に対してローカルである。Real Application Clusters を使用可能にした場合、データベースに対してローカルまたはグローバルになる。「**ラッチ (latch)**」、「**ロック (lock)**」および「**リソース (resource)**」も参照。

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

様々なオペレーティング・システムに合せて作成されるクラスタウェア。OSD クラスタウェアは、オペレーティング・システムと **Real Application Clusters** ソフトウェアとの間の通信リンクを提供する。

オンライン・トランザクション処理 (Online Transaction Processing: OLTP)

コンピュータによるリアルタイムでのトランザクション処理（「**トランザクション・システム (transaction systems)**」も参照）。

カーネル (kernel)

データベースに関して使用される場合は、データベースを実装するフォアグラウンド・プロセスとバックグラウンド・プロセスの組合せを指す。

拡張性 (scalability)

Real Application Clusters 環境にさらにノードを追加し、パフォーマンスを管理する機能。多くの場合、パフォーマンスを大幅に向上させる。

獲得割り込み (acquisition interrupt)

通知に使用されるソフトウェア同期化機能。Real Application Clusters では**ブロッキング割り込み**も使用される。

キャッシュ一貫性 (cache coherency)

複数キャッシュにおけるデータの同期化。どのキャッシュを通してメモリー位置を読み込んでも、その位置に書き込まれた最新のデータが他のキャッシュを通して戻される。「キャッシュの整合性」と呼ばれることもある。

キャッシュ・フュージョン (Cache Fusion)

Real Application Clusters でのディスクレスのキャッシュ一貫性メカニズム。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提供する。

行キャッシュ (row cache)

個々のレコード (行) に対する最新アクセス・データが格納されているメモリー。行キャッシュを使用すると、同じ行に格納されているデータに対する以降の要求の処理速度が速くなる。

共有 (S) モード (shared (S) mode)

リソースの保護付き読取りモード。書込みは許可されない。共有モードでは、どのユーザーでも、数に関係なく、リソースに並行読込みアクセスを持つことができる（「**排他 (X) モード (exclusive (X) mode)**」および「**NULL (N) モード (null (N) mode)**」も参照）。

共有カレント (shared current: SCUR)

共有リソースに対するバッファの状態名。

共有キャッシュ (shared cache)

クラスタ・データベース内のすべてのインスタンスのバッファ・キャッシュの集合。

共有サーバー (shared server)

多数のユーザー・プロセスが、非常に少数のサーバー・プロセスを共有できるように構成されたサーバー。共有サーバー構成では、多数のユーザー・プロセスがディスパッチャに接続する。

クックド・パーティション (cooked partition)

フォーマット済の物理ディスクの一部。このパーティションには、拡張パーティションが作成され、論理パーティションが割り当てられる。これに対し、フォーマットされていないパーティションを、**RAW パーティション**と呼ぶ。

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)

Real Application Clusters データベースを指す総称。

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)

ローカルおよびグローバルの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)

グローバル・エンキュー・サービス・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**グローバル・エンキュー・サービス**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（「**デーモン (daemon)**」も参照）。

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プロセス。**インスタンス**やプロセスの異常終了、および**グローバル・キャッシュ・サービス**や**グローバル・エンキュー・サービス**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。

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。グローバル・ロールのブロックに対して、ブロック・モードを維持する。インスタンス間のブロック転送を行う。グローバル・キャッシュ・サービスは、**グローバル・キャッシュ・サービス・プロセス**、**グローバル・エンキュー・サービス・デーモン**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 Processes: LMSn)

リモート・**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。LMSn 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。LMSn は、グローバル・キャッシュ・サービス・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。インスタンス間の読み込み一貫性要求の場合、LMSn はブロックの読み込み一貫性バージョンを作成して、要求側のインスタンスに送信する。リモート・インスタンスへのメッセージ・フローも制御する。

グローバル・キャッシュ要素 (global cache element)

キャッシュ・フュージョンのリソースを表す Oracle 固有のデータ構造体。**グローバル・キャッシュ・サービス**におけるグローバル・キャッシュ要素とキャッシュ・フュージョン・リソースの関係は 1 対 1 である。

グローバル・サービス・デーモン (Global Services Daemon: GSD)

SRVCTL から要求を受信し、起動や停止などの管理ジョブ・タスクを実行するコンポーネント。コマンドは各ノードでローカルに実行され、結果が SRVCTL に戻される。**デーモン**はデフォルトでノードにインストールされる。削除しないこと。

グローバル・リソース・ディレクトリ (Global Resource Directory)

グローバル・リソースに関連するデータ構造。**クラスタ**内のすべてのインスタンスに分散される。

計画外停止時間 (unplanned downtime)

システムを使用できなくなるシステム障害、データ (メディア)・エラーおよびサイト停止を含むシステム停止時間 (「**計画停止時間 (planned downtime)**」も参照)。

計画停止時間 (planned downtime)

ユーザーがシステムを使用できなくなるルーチン操作、メンテナンスおよびアップグレードのための時間を含む (「**計画外停止時間 (unplanned downtime)**」も参照)。

高可用性 (high availability)

ハードウェアまたはソフトウェア障害が発生しても、一貫性のある連続的なサービスを提供する、冗長なコンポーネントを持つシステム・タイプ。

コンテキストのスイッチング (context switches)

プログラムを実行するオペレーティング・システムは、ユーザー・モードまたはオペレーティング・システム・モードのいずれかで実行される。ユーザー・モードとオペレーティング・システム・モードとの間の切替えをコンテキストのスイッチングという。たとえば、ユーザー・モードでシステム・コールを行うプログラムは、コンテキストのスイッチングを行う。この例のように、ユーザー・プログラムのコンテキストを格納する一方で、オペレーティング・システム・カーネルのコンテキストをメモリーに転送する必要があるため、コンテキストのスイッチングによってパフォーマンスが低下する場合がある。複数のシステム・コールでオペレーティング・システムのリソースが競合する場合は、パフォーマンスがさらに低下する場合がある。

サーバー・クラスタリング (server clustering)

「[Real Application Clusters](#)」を参照。

サーバー制御ユーティリティ (Server Control utility)

「[SRVCTL](#)」を参照。

サービス登録 (service registration)

プロセス・モニター (PMON) が [リスナー](#) に自動的に情報を登録する機能。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、次の名前で構成される文字列である。

- グローバル・データベース名
- データベース名 (DB_NAME)
- ドメイン名 (DB_DOMAIN)

サービス名は、インストール時またはデータベース作成時に入力される。

グローバル・データベース名が不明な場合は、共通データベース初期化ファイル [initdbname.ora](#) の SERVICE_NAMES パラメータの結合値から取得できる。

サービス名は、接続記述子の CONNECT_DATA 部分に含まれている。

システム識別子 (system identifier: SID)

Oracle ソフトウェアが動作しているときに、その特定の [インスタンス](#) を識別する。Real Application Clusters データベースの場合、[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。

順序番号値エンキュー (Sequence Number Value Enqueue: SV)

順序の値を必要とするときにセッションが使用する**エンキュー**。

使用済ブロック (dirty block)

インスタンスによって変更されたデータ・ブロック。

診断用デーモン (diagnosability daemon)

インスタンス・プロセスの障害に関する診断データを獲得する Real Application Clusters のバックグラウンド・プロセス。このデーモンには、ユーザー制御は必要ない。

スケールアップ (scale up)

より大規模なシステムを使用することによって、同じ時間内にどれだけの作業を実行できるかということ（「**スピードアップ (speed up)**」も参照）。

スタンバイ・データベース (Standby Database)

「**Oracle9i Data Guard**」を参照。

ストレージ・エリア・ネットワーク (Storage Area Network: SAN)

共有ストレージ・デバイス的高速ネットワーク。通常、SANアーキテクチャを使用すると、Local Area Network (LAN) または Wide Area Network (WAN) 上のすべてのサーバーがすべてのストレージ・デバイスにアクセスできる。ほとんどのSANは、**ファイバ・チャネル**規格に準拠している。

スピードアップ (speed up)

より多くのハードウェアを使用することによって、元のシステムより少ない時間で同じ作業をどの程度実行できるかということ。ハードウェアを追加することによって、作業を一定に保持し、節約時間を測定する（「**スケールアップ (scale up)**」も参照）。

スリー・ナインの可用性 (three nines availability)

99.9%のシステム可用性を意味する口語表現。

スレッド (thread)

各Oracle **インスタンス**には、独自のオンライン REDO ログ・グループのセットがある。このようなグループを、オンライン REDO のスレッドと呼ぶ。Real Application Clusters 以外の環境では、データベースごとに1つのみスレッドが存在し、このスレッドは、そのデータベースにアクセスしているインスタンスに属している。Real Application Clusters 環境では、インスタンスごとに別々のスレッドがある。つまり、各インスタンスは、独自のオンライン REDO ログを持っている。各スレッドには、独自の現行のログ・メンバーがある。

制御ファイル投票結果レコード (control file voting results record: CFVRR)

リカバリ処理の一部。**クラスタ**にあるノードのメンバーシップ「投票」を処理した後、**インスタンス・メンバーシップ・リカバリ**判定機能によって作成される制御ファイル・レコード。

セカンダリ・インスタンス (secondary instance)

プライマリ/セカンダリ・インスタンス構成においては、プライマリ・インスタンスに障害が発生した場合、**プライマリ・インスタンス**にバックアップ・サービスを提供する**インスタンス**を指す。

セカンダリ・インスタンス・ロール (secondary instance role)

プライマリ/セカンダリ・インスタンス構成の場合、データベースを2番目にマウントする**インスタンス**が、セカンダリ・ロールを担うことになる。アプリケーション・セッションが要求する作業は、プライマリ・ロールを持つインスタンスによって実行されるが、選択された作業（レポート作業および予定された操作など）は、セカンダリ・インスタンス・ロールを持つインスタンスによって実行される場合がある（「**プライマリ・インスタンス (primary instance)**」、「**プライマリ・インスタンス・ロール (primary instance role)**」および「**セカンダリ・インスタンス (secondary instance)**」も参照）。

接続時フェイルオーバー (connect-time failover)

「**フェイルオーバー (failover)**」を参照。

対称型マルチ・プロセッサ (Symmetric Multi-Processor: SMP)

広帯域リンクで接続され、1つのオペレーティング・システムによって管理される2つ以上の同種のプロセッサ。各プロセッサは、I/O デバイスに同等にアクセスできる。

デーモン (daemon)

Disk and Execution Monitor の略。明示的には起動されず、一定の条件が満たされるのを待機している休止状態のプログラム。

ディスク書き込みの強制実行 (forced disk write)

Real Application Clusters の特定のデータ・ブロックは、一度に1つの**インスタンス**によってのみ変更できる。あるインスタンスが別のインスタンスで必要なデータ・ブロックを変更する場合、ディスク書き込みの強制実行が必要かどうかは、そのブロックに対して発行された要求のタイプによって異なる。

透過性 (transparency)

アクションがユーザーに表示されないまま実行される場合、そのアクションは透過性がある。

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 I** など、**高可用性**環境を目的としたランタイム・**フェイルオーバー**。TAF 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示す。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** 文を再開することができる。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。

トランザクション空きリスト (transaction free list)

コミットされなかったトランザクションによって解放されたブロックのリスト。

トランザクション・システム (transaction systems)

データベースへの更新内容に特徴があるアプリケーション・システム。トランザクション・システムの例には、E-Business や ERP アプリケーションなどがある。さらに具体的な例としては、通話料金請求システム、クレジット・カード取引および航空券予約システムなどがある。トランザクション・システムは、**オンライン・トランザクション処理**システムとも呼ばれる。

ノード (node)

インスタンスが常駐するマシン。

ハードウェア・フェイルオーバー (hardware failover)

プラットフォーム固有の **Cluster Manager (CM)** によって実行される**フェイルオーバー**のタイプ。ノードまたはノードで実行中の**インスタンス**に障害が発生した場合、Cluster Manager は、**クラスタ**内の別のノードでインスタンスを再起動する。インスタンスを再起動するには、IP アドレス、ボリュームおよび Oracle データ・ファイルを含むファイル・システムを移動する必要がある。Oracle サーバーを起動して、新しいノードでデータ・ファイルをオープンする必要もある。

ハートビート (heartbeat)

インスタンスがアクティブであることを示す定期的なメッセージ。

排他 (X) モード (exclusive (X) mode)

書込み専用のグローバル・リソース・モード。このモードでは、他のアクセスは許可されない。「**NULL (N) モード (null (N) mode)**」および「**共有 (S) モード (shared (S) mode)**」も参照。

排他カレント (exclusive current: XCUR)

排他リソースのバッファの状態名。

パスト・イメージ (past image: PI)

グローバル・キャッシュ・サービスで使用する**使用済ブロック**のコピー。ブロックのパスト・イメージは、対象のバージョンを含めた書込みが記録されるまで保持される。パスト・イメージは、障害時リカバリで使用する。

バッファの状態 (buffer state)

インスタンスのローカル・キャッシュ内にあるバッファの状態。

パブリック・ロールバック・セグメント (public rollback segment)

ロールバック・セグメントを必要とするすべての**インスタンス**が使用できるロールバック・セグメント。

パラメータ・ファイル (parameter file: PFILE)

Oracle サーバーが使用するファイル。データベースの起動時に使用する特定の値および構成の設定を指定する。キーワード PFILE は、起動コマンドで使用する。

パラレル実行 (parallel execution)

特定のタイプの SQL 文の処理作業を複数のパラレル実行サーバー・プロセス間に分散する。通常、**意思決定支援システム**・アプリケーションで使用する。

光ファイバ分散データ・インタフェース (Fiber Distributed Data Interface: FDDI)

伝送速度 100 Mbps (メガビット / 秒) の Local Area Network (LAN) アーキテクチャに関する ANSI (米国規格協会) の規格。基礎となるメディアは主に光ファイバで、トポロジは二重化された対循環とトークン・リングである。

ファイバ・チャネル (Fibre Channel)

ANSI (米国規格協会) によって規格化された高速シリアル・データ転送アーキテクチャの総称。ファイバ・チャネル・アーキテクチャは、コンピュータおよび大容量ストレージ・メーカーのコンソーシアムであるファイバ・チャネル産業協会 (Fibre Channel Industry Association: FCIA) によって開発された。最も有名なファイバ・チャネル規格は、ファイバ・チャネル判定ループ (FC-AL) である。「**ストレージ・エリア・ネットワーク (Storage Area Network: SAN)**」も参照。

ファイブ・ナインの可用性 (five nines availability)

99.999% のシステム可用性を意味する口語表現。

フェイルオーバー (failover)

Real Application Clusters で使用される障害の認識とリカバリの方法。

フォルト・トレランス (fault tolerance)

ハードウェアやソフトウェアの障害発生時に、正常な操作を継続するためのシステムまたはコンポーネントの機能。通常は、ある程度の冗長性を伴う。

プライベート・ロールバック・セグメント (private rollback segment)

インスタンスによるデータベースのオープン時に、その**インスタンス**が排他的に取得するロールバック・セグメント。

プライマリ/セカンダリ・インスタンス構成 (Primary/Secondary Instance Configuration)

すべてのクライアントがデータベースにアクセスする際のインスタンスがプライマリ・**インスタンス**である場合の構成。プライマリ・インスタンスに障害が発生した場合、セカンダリ・インスタンスは、プライマリ・インスタンスにバックアップ・サービスを提供する (「**プライマリ・インスタンス (primary instance)**」, 「**プライマリ・インスタンス・ロール (primary instance role)**」, 「**セカンダリ・インスタンス (secondary instance)**」 および 「**セカンダリ・インスタンス・ロール (secondary instance role)**」 も参照)。

プライマリ・インスタンス (primary instance)

プライマリ / セカンダリ構成の場合は、すべてのクライアントがデータベースへのアクセス時に使用する**インスタンス**を指す (「**セカンダリ・インスタンス (secondary instance)**」 も参照)。

プライマリ・インスタンス・ロール (primary instance role)

プライマリ / セカンダリ構成の場合、データベースを最初にマウントする**インスタンス**が、プライマリ・ロールを担うことになる。このインスタンスによって、アプリケーション・セッションが要求する作業が実行される。プライマリ・インスタンスに障害が発生した場合 (またはプライマリ・インスタンスが停止した場合)、**フェイルオーバー**が発生し、別のインスタンスがプライマリ・インスタンス・ロールを担うことになる (「**プライマリ/セカンダリ・インスタンス構成 (Primary/Secondary Instance Configuration)**」 および 「**セカンダリ・インスタンス・ロール (secondary instance role)**」 も参照)。

プロセス間通信 (interprocess communication: IPC)

高速のオペレーティング・システム固有の通信コンポーネント。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 (interconnect)**」 も参照。

ブロッキング割込み (blocking interrupt)

あるリソースへのアクセス権を保持するプロセスに対して、別のプロセスがそのリソースとは非互換モードでアクセスする必要があることを通知するソフトウェア同期化機能 (たとえば、**共有 (S) モード**と**排他 (X) モード**は非互換である)。「**獲得割込み (acquisition interrupt)**」 も参照。

平均障害間隔時間 (mean time between failures: MTBF)

コンポーネントが障害なしに動作する平均時間 (通常は時間単位で表される)。計測した合計動作時間数を合計障害数で除算して算出する。この用語を使用してユーザーは、障害が発生するまでのデバイスまたはシステムの動作期間を予想できる。

ボリューム・マネージャ (volume manager)

ストレージ・エリア・ネットワークにストレージ管理を提供するソフトウェア。通常、ホストベースの **Redundant Array of Independent Disks (RAID)** のソフトウェアを指す。ほとんどのボリューム・マネージャは、Graphical User Interface (GUI) コンソールを使用して、記憶ボリュームのパーティション化および状態を表示する。ボリューム・マネージャは、様々なディスクのボリュームを作成できる。このボリュームは、複数のディスクを含む場合がある (**RAW パーティション (raw partition)**) も参照)。

マスター空きリスト (master free list)

表内のすべてのエクステンツからの使用可能領域が格納されているブロックのリスト。

マスター化 (mastering)

「**リソースのマスター化 (resource mastering)**」を参照。

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、ブロックに加えられた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 つ以上のシステム・グローバル領域 (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。

ライブラリ・キャッシュのウォーミング (warming the library cache)

解析済 SQL 文およびコンパイル済 PL/SQL ユニットの情報を、プライマリ・**インスタンス**のライブラリ・キャッシュからセカンダリ・インスタンスのライブラリ・キャッシュに転送するプロセス。ライブラリ・キャッシュがすでに移入されているため、キャッシュのウォーミングによって、**フェイルオーバー**後のパフォーマンスが改善される。

ライブラリ・キャッシュ無効化 (library cache invalidation)

ライブラリ・キャッシュ内のオブジェクトは、オブジェクト依存性が原因で無効になる場合がある。このオブジェクトは、次回使用時に再コンパイルされる。

ラッチ (latch)

単純で低レベルのシリアル化メカニズム。システム・グローバル領域 (SGA) のメモリー内データ構造を保護する。ラッチは、データ・ファイルを保護せず、自動で、非常に短い時間のみ排他モードで保持される。ラッチは 1 つのノード内で同期化されるため、ノード間の同期化には利用できない (**エンキュー (enqueue)**)、「ロック (lock)」および「**リソース (resource)**」も参照)。

リスナー (listener)

サーバーで実行されるプロセス。クライアントの着信接続要求をリスニングし、サーバーへの通信量を管理する。クライアントがサーバーとのネットワーク・セッションを要求すると、リスナーがその要求を受け取る。クライアント情報がリスナー情報と一致すると、リスナーはサーバーへの接続を許可する。

リソース (resource)

[グローバル・キャッシュ・サービス](#)に関して使用される場合は、データ・ブロックに対する並行性制御を指す（「[グローバル・キャッシュ要素 \(global cache element\)](#)」、「[エンキュー \(enqueue\)](#)」、「[ロック \(lock\)](#)」、「[リソース \(resource\)](#)」および「[リソースのマスター化 \(resource mastering\)](#)」も参照）。

リソースのマスター化 (resource mastering)

[グローバル・キャッシュ・サービス](#)および[グローバル・エンキュー・サービス](#)によるリソース制御方法（「[グローバル・リソース・ディレクトリ \(Global Resource Directory\)](#)」も参照）。

リソースのモード (resource mode)

クラスタ内のインスタンスのグローバル・アクセス権を定義する並行性制御。

リソース・ロール (resource role)

データ・ブロックが、1つのインスタンス（ローカル）にのみキャッシュされているか、複数インスタンス（グローバル）にキャッシュされているかを定義する並行性制御。

リポジトリ・データベース (repository database)

[Oracle Enterprise Manager](#)に必要なデータが格納されている Oracle データベース内の表の集合。これは、ノード上のデータベースとは別のデータベースである。

ローカル・エンキュー (local enqueues)

[クラスタ](#)・データベース内の共有データ構造への同時アクセスを調整する同期化メカニズム。

割り込み (interrupt)

「[獲得割り込み \(acquisition interrupt\)](#)」および「[ブロッキング割り込み \(blocking interrupt\)](#)」を参照。

数字

1:1 ロック、設定、B-3

A

ACTIVE_INSTANCE_COUNT 初期化パラメータ、
10-18
ADD LOGFILE 句
 THREAD 句、8-4
ALERT ファイル、8-2
ALTER DATABASE 文
 ADD LOGFILE、8-4
 ログ・モードの設定、A-2
ALTER SYSTEM SET
 サーバー・パラメータ・ファイルの管理、8-3
ARCHIVELOG モード
 オンライン・バックアップおよびオフライン・バックアップ、1-5
 自動アーカイブ、1-5
 モードの変更、A-2

C

Cluster Manager、3-2
 障害検出、10-13
Cluster Manager (CM)
 Node Monitor、3-2
 グローバル・キャッシュ・サービス (GCS) との相互作用、3-7
 目的、3-2
CPU、2-3
CREATE CONTROLFILE 文
 排他モード、A-2

CREATE DATABASE 文
 排他モード、A-2

D

Database Configuration Assistant (DBCA)、4-2、8-3
 Real Application Clusters での管理、9-4
 グローバル・サービス・デーモン (GSD)、3-4
 サーバー・パラメータ・ファイルおよびデフォルトの位置、8-3
DBMS_LIBCACHE
 ライブラリ・キャッシュのウォーミング、10-22
DIAG
 診断用デーモン、3-4

F

FDDI
 Real Application Clusters で使用する場合、2-4

I

initids.ora のパラメータ、10-18
I/O、1-4
 割込み、4-6

L

Lamport SCN 生成、5-6
LCK
 ロック・プロセス、3-4
listener.ora ファイル、4-6
LMD
 グローバル・エンキュー・サービス・デーモン (LMD)、3-4

LMON

クラスタ再編成, 10-13
グローバル・エンキュー・サービス・モニター,
3-4

LMSn

グローバル・キャッシュ・サービス・プロセス,
3-4

M

MAX_COMMIT_PROPAGATION_DELAY パラメータ
Lamport SCN 生成, 5-6
MTTR, 10-2

N

NOARCHIVELOG モード
オフライン・バックアップ, 1-5
モードの変更, A-2
Node Monitor, 3-2
NULL (N) モード, 5-4
N ノード
Real Application Clusters 構成, 10-17

O

OMF 機能, 8-3
Oracle
データ・ファイルの互換性, 8-2
バックアップ, 1-5
Oracle Call Interface (OCI), 10-11
Oracle Enterprise Manager
Real Application Clusters での管理, 9-3
Real Application Clusters でのパフォーマンス, 9-6
サーバー・パラメータ・ファイルの管理, 8-3
Oracle Enterprise Manager (EM)
グローバル・サービス・デーモン (GSD), 3-4
Oracle Managed File 機能, 8-3
Oracle Parallel Execution, 4-3
説明, 4-3
Oracle Real Application Clusters Guard I, 10-2
Oracle Real Application Clusters Guard II, 10-2
Oracle9i Data Guard, 10-2

P

PI、廃棄, 6-7
PL/SQL, 7-3

R

RAID
災害時保護, 10-5
Real Application Clusters
Oracle Parallel Execution, 4-3
アーキテクチャの概要, 3-2
インスタンス・プロセス, 3-4
共有ディスク・コンポーネント, 3-3
所有コスト, 1-3
定義, 1-2
ディクショナリ・キャッシュ・ロック, 7-3
同期化プロセス, 5-3
リカバリ, 6-8
リソースの調整の概要, 5-2
利点を生かしたシステム・タイプ, 4-2
Real Application Clusters Guard II, 10-24
Real Application Clusters 接続時フェイルオーバー,
4-4
Real Application Clusters でのリカバリ, 6-8
Recovery Manager (RMAN)
Real Application Clusters の管理コンポーネント,
9-6
REDO ログ・ファイル
REDO スレッド, 8-4
アーカイブ, 1-5
上書き, 1-5
REDO ログ・ファイルのアーカイブ
オンライン・アーカイブ, 1-5

S

SCN, 5-6
SHUTDOWN TRANSACTIONAL LOCAL コマンド,
10-8
SHUTDOWN TRANSACTIONAL コマンド, 10-8
SID, 8-3
サーバー・パラメータ・ファイル, 8-3
SQL 文
制限事項, A-2

Statspack

Real Application Clusters で使用可能な統計, 9-6

Real Application Clusters での管理とパフォーマンス, 9-6

Structured Query Language (SQL), 7-3

T

THREAD オプション

パブリック・スレッド作成, 8-4

プライベート・スレッド作成, 8-4

U

UNDO 領域管理, 8-5

V

V\$BH ビュー, 6-4

W

Windows NT/Windows 2000

定数ディスク, 3-3

投票ディスク, 3-3

あ

アーカイブ

バックアップ, 1-5

アーカイブ・モードの切替え, A-2

アーキテクチャ

クラスタ・データベース処理, 2-1

コンポーネント, 3-1

アーキテクチャの概要

Real Application Clusters, 3-2

空き領域管理

自動セグメント領域管理, 8-5

アクセス, 2-4

アクティブ / アクティブ構成

Real Application Clusters Guard II, 10-24

アプリケーション

拡張性, 4-3, 4-6

い

イーサネット

Real Application Clusters で使用する場合, 2-4

意思決定支援システム (DSS), 4-3

一貫性

マルチバージョン読取り, 1-5

インスタンス, xi

スレッド番号, 8-4

リカバリ, 10-15

インスタンス障害

リカバリ, 6-8

インスタンス名, 4-6

インスタンス・メンバーシップ・リカバリ (IMR),
10-14

インスタンス・リカバリ

ロールバック・セグメント, 8-6

インストール

Real Application Clusters の管理コンポーネント,
9-2

インターコネクト, 2-3, 3-2

拡張性, 4-3

クラスタ・コンポーネント, 2-2

冗長性, 10-3

定義, 2-4

え

エンキュー

定義, 5-3

リカバリ処理, 6-8

お

オフライン・バックアップ, 1-5

オペレーティング・システム固有 (OSD) レイヤー,
3-2

オペレーティング・システムの拡張性, 4-6

オンライン REDO ログ・ファイル

REDO スレッド, 8-4

オンライン・アーカイブ, 1-5

オンライン・トランザクション処理 (OLTP), 1-2

オンライン・バックアップ, 1-5

か

概要, 2-2

Real Application Clusters, 1-1

書き込みプロトコルとパスト・イメージの追跡, 6-3

拡張

アプリケーション, 1-2

拡張性, 1-3

4つのレベル, 4-3

アプリケーション, 4-3, 4-6

共有メモリー・システム, 4-6

ネットワーク, 4-5

ハードウェア, 4-3

レベル, 4-3

拡張性のレベル, 4-3

可用性

インターコネクト, 2-4

クラスタ・データベースのメリット, 1-4

管理

Real Application Clusters での概要, 9-2

Real Application Clusters の管理コンポーネント,
9-3

Real Application Clusters のコンポーネント, 9-1

インストール, 9-2

管理, 9-3

設定と構成, 9-2

バックアップとリカバリ, 9-6

パフォーマンスの監視, 9-6

関連文書, xiv

き

記憶域

クラスタ・ファイル・システム, 8-2

起動, 3-4

ファイルへのアクセスの検証, 8-2

キャッシュ

ディクショナリのフラッシュ, 7-3

キャッシュ一貫性

グローバル・キャッシュ・サービス (GCS), 5-5

キャッシュ・フュージョン

使用例, 6-5

処理, 6-1

定義, 6-2

キャッシュ・リカバリ, 10-15

行

更新およびエンキュー, 5-3

共有 (S) モード, 5-4

共有カレント (SCUR), 6-4

共有サーバー

プライマリ / セカンダリ・インスタンス, 10-18

共有メモリー・システム

拡張性, 4-6

共有モード

データ・ファイル, 8-2

行レベル・ロック

システム共有リソース, 1-5

行ロック, 1-5

く

クォーラム・ディスク, 3-3

クライアント

フェイルオーバー, 10-6

ランダム化, 4-5

ロード・バランシング, 4-5

クラスタ

コンポーネント, 2-2

ストレージ・アクセス, 2-4

定義, 1-2

クラスタ再編成, 10-13

クラスタ・データベース

可用性, 1-4

クラスタ・データベース処理

定義, 1-2

ハードウェア, 2-1

有効な場合, 4-2

クラスタ・ハードウェア・アーキテクチャ, 2-1

クラスタ・ファイル・システム, 3-3

Real Application Clusters の記憶域, 8-2

メリット, 2-4

グローバル・エンキュー

自動的に計算, 5-3

グローバル・エンキュー・サービス (GES), 3-6, 5-2

リソースの調整, 7-1

グローバル・エンキュー・サービス (GES) の再構成

リカバリ処理のフェーズ, 6-8

グローバル・エンキュー・サービス・デーモン

(LMD), 3-4

グローバル・エンキュー・サービス・モニター

(LMON), 3-4

- グローバル・キャッシュ・サービス, 6-1
 - LMON、使用, 10-13
 - 説明, 3-6
 - 分散アーキテクチャ, 3-6
- グローバル・キャッシュ・サービス (GCS), 3-6, 5-2
 - CM との相互作用, 3-7
 - 機能, 3-6
 - 処理の例, 5-5
 - データ・ブロックの追跡動作, 5-5
 - フォルト・トレラント, 3-6
- グローバル・キャッシュ・サービス・プロセス (LMSn), 3-4
 - 同期化のためのプロセス, 5-3
- グローバル・サービス・デーモン (GSD)
 - Real Application Clusters 固有のバックグラウンド・プロセス, 3-4
 - Real Application Clusters での管理, 9-5
- グローバル・リソース・ディレクトリ, 3-6, 5-2
 - 内容, 5-2
- クロス登録, 10-20

こ

- 高可用性, 1-2
 - クラスタ・インターコネクト, 10-3
 - クラスタ・データベースのメリット, 1-4
 - 構成, 10-17
 - 構成、利点, 10-23
 - 定義, 1-4, 10-2
- 高可用性の共有ノード構成, 10-23
- 高可用性のノード構成
 - 利点, 10-24
- 更新
 - 並行, 1-5
- 構成
 - 高可用性, 10-17
- 互換性
 - 共有および排他モード, 8-2
- 異なるノードでの並行書込み, 6-2
- 異なるノードでの並行読込みと書込み, 6-2
- コンポーネント
 - 高可用性, 10-3

さ

- サーバー側フェイルオーバー, 10-11
- サーバー制御 (SRVCTL)
 - Real Application Clusters での管理, 9-5
- サーバー制御ユーティリティ (SRVCTL)
 - グローバル・サービス・デーモン (GSD), 3-4
- サーバー・パラメータ・ファイル, 8-3
 - Real Application Clusters での位置, 8-3
- サービス登録, 4-6
- サービス名, 4-6
- 災害時計画
 - 高可用性, 10-4
- 再マスター化
 - フェイルオーバー中, 10-5
 - リソース, 10-15

し

- システム
 - Real Application Clusters の利点を生かしたタイプ, 4-2
- システム固有の Oracle マニュアル
 - データ・ファイル、最大数, A-2
- システム変更番号 (SCN)
 - Lamport, 5-6
 - 増加, 5-6
- 実践, 10-1
- 自動 UNDO 管理, 8-5
 - Real Application Clusters で推奨, 8-5
- 自動セグメント領域管理
 - Real Application Clusters で推奨, 8-5
- 障害
 - ALERT ファイル, 8-2
 - Cluster Manager による検出, 10-13
 - インスタンスとリカバリ, 6-8
 - 保護の検証, 10-4
- 使用済ブロック, 6-4
- 初期化パラメータ・ファイル, 8-3
- 新機能, xx
- 診断用デーモン (DIAG), 3-4

す

ストレージ, 2-2, 2-3
スレッド, 8-4
スレッド・パラメータ
 インスタンス獲得スレッド, 8-4

せ

制御ファイル, 10-14
制御ファイル投票結果レコード, 10-14
制限事項
 ファイル操作, A-2
セカンダリ・インスタンス, 10-17, 10-18
接続時フェイルオーバー, 4-4
設定, B-2
設定と構成
 Real Application Clusters の管理コンポーネント,
 9-2
専用サーバー
 プライマリ / セカンダリ・インスタンス, 10-18

た

帯域幅, 2-4
 ネットワーク, 4-3
待機時間, 2-4
 ネットワーク, 4-3
対称型マルチプロセッサ (SMP), 4-6
単一ノードのクラスタ・データベース, 7-3

ち

チェックポイント, 6-6

て

ディクショナリ・キャッシュ, 7-3
 ロック, 7-3
停止
 トランザクション, 10-8
ディスク・サブシステム, 2-4
データ・ウェアハウス, 4-3
データ・ファイル
 共有, 8-2
 検証, 8-2
 最大数, A-2

 ブロックへのロックの割当て, B-3
 ロックに未指定, B-3
データ・ブロック, 1-4
データベース
 バックアップ, 1-5
データベース・インスタンス登録
 クライアント・ロード・バランシング, 4-5
 接続時フェイルオーバー, 4-4
データベース・ライター (DBWR), 6-3
デッドロック検出, 7-2

と

透過性
 定義, 1-4
透過的アプリケーション・フェイルオーバー
 使用, 10-7
 定義, 10-6
同期化プロセス
 Real Application Clusters, 5-3
投票ディスク, 3-3
トランザクション
 行ロック, 1-5
 更新, 1-5
 コミットされたデータ, 1-5
 並行, 1-5, 5-5
トランザクション・システム, 4-2
トランザクション・リカバリ, 10-16

ね

ネットワーク
 拡張性, 4-4
 帯域幅, 4-3
 待機時間, 4-3
 パフォーマンス、クライアント要求のランダム化に
 よる改善, 4-5

の

ノード
 高可用性, 10-3
 障害, 1-4
 定義, 2-3
 ハードウェア, 2-1
ノード間のメッセージ機能
 同期化プロセス, 5-3

は

ハードウェア

拡張性, 4-3

クラスタ・データベース処理, 2-1

ハードウェアおよびネットワークの拡張性, 4-3

排他 (X) モード, 5-4

排他カレント (XCUR), 6-4

排他モード

互換性, A-2

はじめに, xi, xix

パスト・イメージ (PI)

定義, 6-3

バックアップ

Real Application Clusters の管理コンポーネント,
9-6

オフライン, 1-5

オンライン, 1-5

バッファ・キャッシュ管理, 1-4

バッファの状態, 6-4

パフォーマンス

Real Application Clusters の管理コンポーネント,
9-6

パラメータ

設定, 8-3

ファイル (PFILE), 8-3

ファイル、サーバー, 8-3

パラレル・モード

ファイル操作の制限事項, A-2

ひ

非対称型マルチプロセッシング, 4-6

ビットマップ

自動セグメント領域管理, 8-5

表領域

バックアップ, 1-5

ふ

ファイル

ALERT, 8-2

REDO ログ, 8-4, 1-5

REDO ログのアーカイブ, 1-5

サイズ, B-3

最大数, A-2

ネーミング規則, 8-3

ファスト・スタート・リカバリ, 10-2

ファスト・スタート・ロールバック, 10-16

フェイルオーバー

Real Application Clusters でのリカバリ処理, 10-13

期間, 10-6

基本, 10-5

サーバー側, 10-11

接続時間, 4-4

定義, 10-2

透過的アプリケーション・フェイルオーバー, 4-4

ホストベース, 10-12

フォルト・トレランス, 3-6

複数ノードでの並行読み込み, 6-2

プライベート・ロールバック・セグメント

取得, 8-6

プライマリ・インスタンス, 10-18

プライマリ・インスタンス・ロール, 10-18

プライマリ / セカンダリ・インスタンス構成, 10-18

共有サーバー環境, 10-20

専用サーバー環境, 10-18

定義, 10-17

ライブラリ・キャッシュのウォーミング, 10-22

プロセス間通信 (IPC), 2-3, 2-4

ブロック

イメージ, 5-3

ディスクへの書込み, 6-6

複数バージョンに対する変更, 5-3

ブロック・メディア・リカバリ, 10-15

へ

平均障害時間 (MTTF), 10-3

ほ

ホストベース・フェイルオーバー, 10-12

ま

マルチティア・アプリケーション環境, 10-6

マルチバージョン読み込み一貫性, 1-5

め

メッセージ

ALERT ファイル, 8-2

ファイルへのアクセス, 8-2

メディア障害
 ファイルへのアクセス, 8-2
メモリー, 1-4, 2-3, 2-4
メモリー・キャッシュ
 同期化のための転送, 5-3
メモリマップド IPC
 使用方法, 2-4

も

モード
 アーカイブ, 1-5
 リソース, 5-4

ゆ

ユーザー・モードのプロセス間通信
 Real Application Clusters での使用方法, 2-4

よ

読込み一貫性
 マルチバージョン, 1-5
読込み一貫性 (CR), 5-6
読込み一貫性バージョン, 5-3
読取り専用アクセス, 1-5

ら

ライブラリ・キャッシュ
 ウォーミング, 10-22
ライブラリ・キャッシュのウォーミング
 DBMS_LIBCACHE, 10-22
ライブラリ・キャッシュ・ロック, 7-3
 文の解析, 7-3

り

リカバリ
 Real Application Clusters とキャッシュ・フュー
 ジョン, 6-8
 Real Application Clusters の管理コンポーネント,
 9-6
 インスタンス, 10-15
 キャッシュ, 10-15
 ファイルへのアクセス, 8-2

メディア障害, 8-2
 ロールバック, 8-6
リカバリ処理
 Real Application Clusters, 10-13
リスナー
 接続時フェイルオーバー, 4-4
 透過的アプリケーション・フェイルオーバー, 4-4
リスナー間での要求のランダム化, 4-5
リソース, 10-15
 グローバル・リソース・ディレクトリ内の情報,
 5-2
 再マスター化, 10-15
 システム変更番号 (SCN), 5-6
 調整、概要, 5-2
 調整、グローバル, 5-2
 調整、ローカル, 5-2
 モード, 5-4
 モードおよびバッファの状態, 6-4
 リカバリ処理, 6-8
 ルール, 5-4
領域管理
 自動セグメント領域管理, 8-5

ろ

ローカル調整
 グローバルになる, 5-3
ローカル・リソース調整, 5-2
ロード・バランシング
 クライアント・ロード・バランシング, 4-5
ロール
 リソース, 5-4
ロールバック・セグメント UNDO モード
 Real Application Clusters, 8-6
 プライベート・ロールバック・セグメント, 8-6
ロック, B-2
 エンキュー, 5-3
 ファイル集合, B-4
 ブロックのマッピング, B-3
 ライブラリ・キャッシュ, 7-3
ロック・プロセス (LCK), 3-4

わ

割当て
 ロック, B-3