SYBASE®
An SAP® Company

Clusters ユーザーズ・ガイド

Adaptive Server® Enterprise

15.7

ドキュメント ID：DC00980-01-1570-01
改訂：2012 年 2 月

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

目次

第 1 部

Cluster Edition の設定

ここでは、クラスタ環境で実行する Adaptive Server® の設定手順を示します。

第 1 章

Cluster Edition の概要

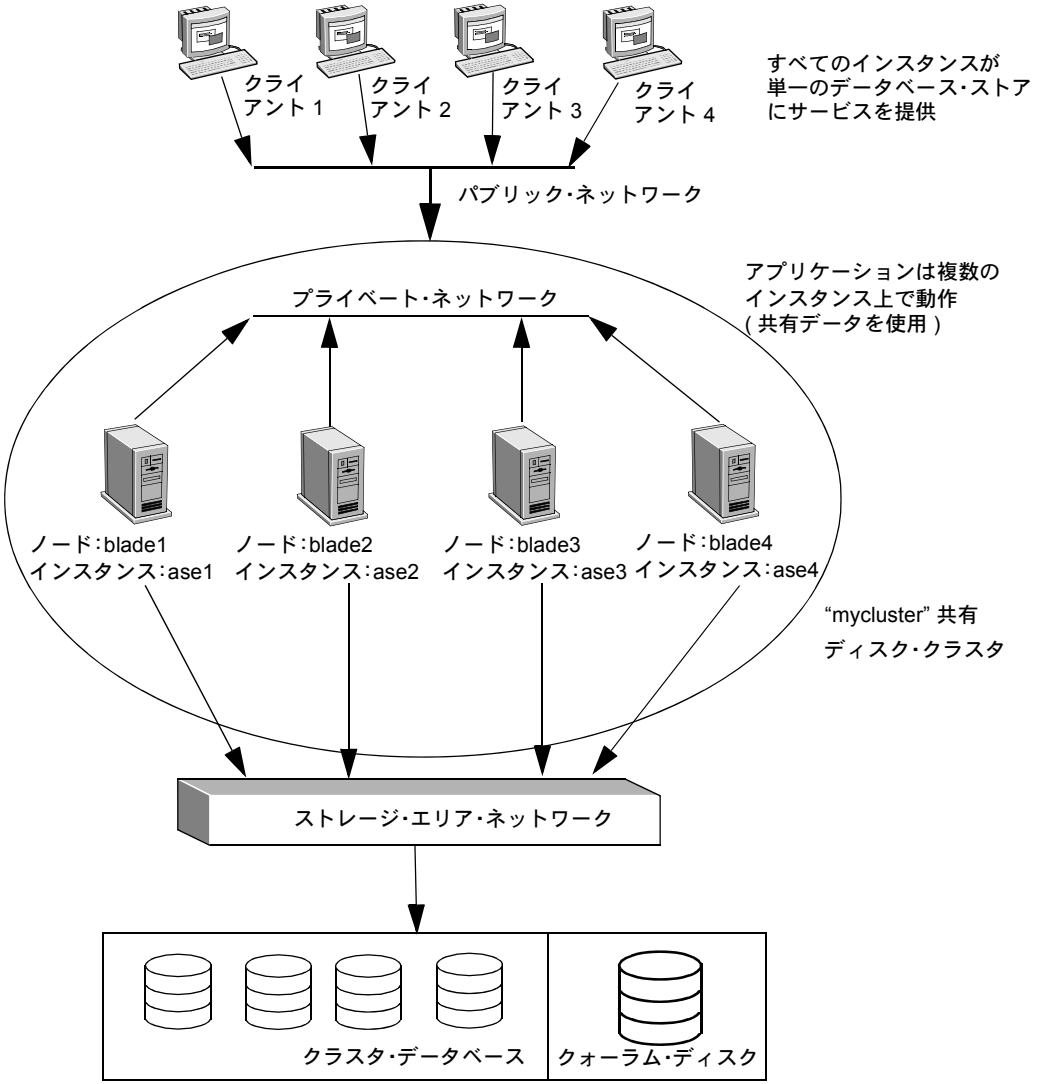
Cluster Edition とは

Cluster Edition を使用すると、複数の Adaptive Server を共有ディスク・クラスタとして実行できます。複数のマシンが共有のディスク・セットと高速のプライベート相互接続 (たとえば、ギガビット・イーサネットなど) に接続することで、Adaptive Server は複数の物理ホストや論理ホストを使用して、その規模を調整できます。

クラスタ環境では、各マシンは「ノード」、Adaptive Server は「インスタンス」と呼ばれます。接続された複数のインスタンスが「クラスタ」を構成し、共有ディスク上にあるデータベースの単一のセットを共同で管理します。いずれの場合も、複数のインスタンスが単一のシステムとして存在し、すべてのデータに任意の「インスタンス」からアクセスできます。Cluster Edition では、クラスタごとに固有の SPID が割り当てられるので、この SPID を使用してクラスタ内のすべてのインスタンスにわたって個々のプロセスが識別されます。

図 1-1 に示すクラスタ・システムでは、「mycluster」という名前の共有ディスク・クラスタに各クライアントが接続しています。このクラスタには、それぞれが「blade1」、「blade2」、「blade3」、「blade4」というマシン上で実行されている「ase1」、「ase2」、「ase3」、「ase4」というインスタンスが含まれています。この例では、単一のインスタンスが各ノード上に存在しています。

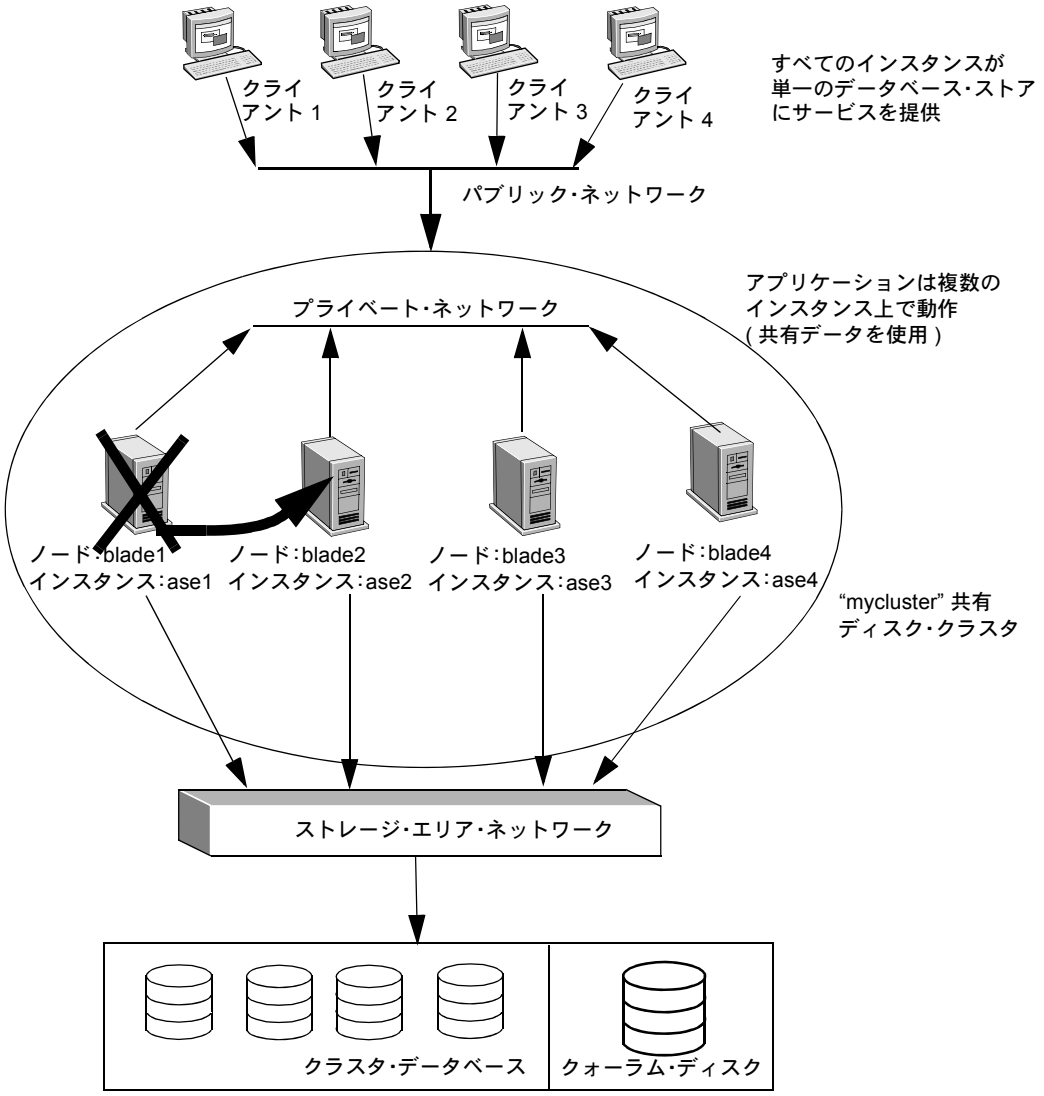
図 1-1: クラスタ対応の主要コンポーネント



クラスタ・メンバの 1 つに障害が発生すると、そのメンバが行っていた負荷は、残りの稼動中のクラスタ・メンバに転送されます。たとえば、「ase1」に障害が発生すると、そのインスタンスに接続していたクライアントは、残りのアクティブなインスタンスのいずれかにフェールオーバされます。図 1-2 (6 ページ) を参照してください。

注意　Cluster Edition では、複数の障害を処理し、後続の障害が発生する前に、初めの障害から完全に復旧できます (これらの障害が同時に発生しないことが前提)。

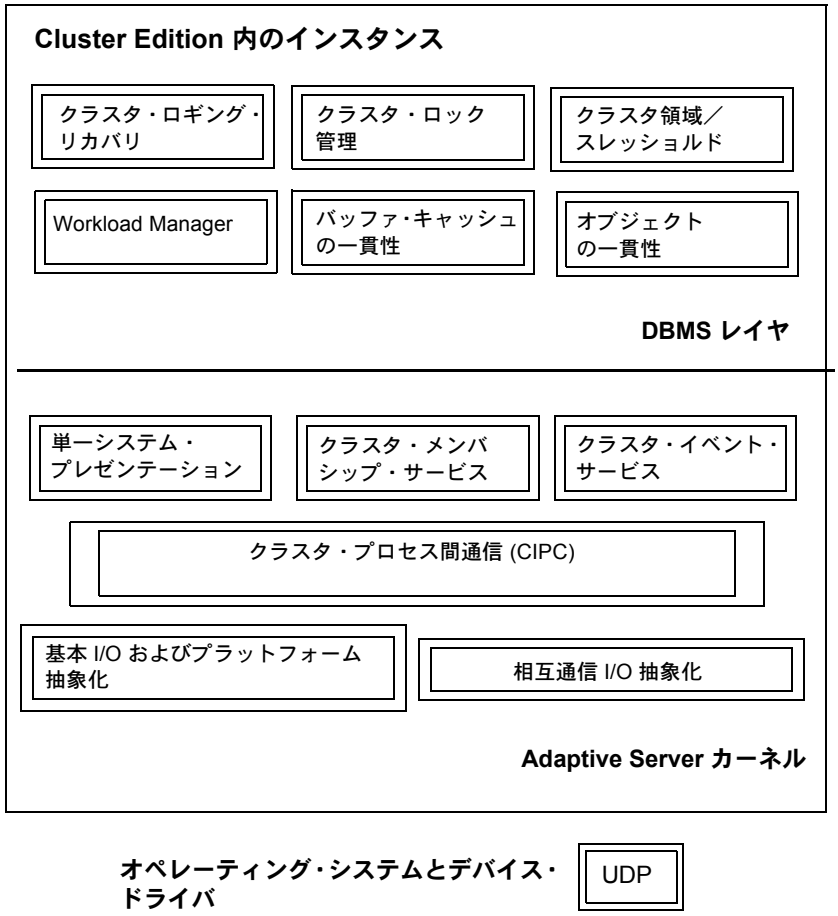
図 1-2: クラスタが障害を処理する方法



Adaptive Server 統合クラスタウェア

Cluster Edition クラスタウェアは、Adaptive Server に直接統合されています。Adaptive Server を実行するために外部のクラスタウェアを必要としません。Cluster Edition にはいくつかの新しいコンポーネントが含まれます。その他のコンポーネントは、既存の Adaptive Server インフラストラクチャのクラスタ対応の拡張になります。図 1-3 は、これらのコンポーネントを示しています。

図 1-3: インスタンスのクラスタ対応の主要コンポーネント



Adaptive Server カーネル

次に示すのは、新しいネイティブのクラスタ・インフラストラクチャ・コンポーネントです。

- クラスタ・メンバシップ・サービス ― クラスタのメンバシップを管理し、インスタンスの障害の検出と処理を行います。
- クラスタ・プロセス間通信 (CIPC) ― インスタンスが冗長な通信経路を通じてお互いに通信できるようにするメッセージ・サービスと相互通信抽象化レイヤを提供します。
- クラスタ・イベント・サービス ― クラスタ全体のイベントに対する、イベントの汎用のパブリッシングとサブスクリプションのメカニズムをサポートします。

DBMS レイヤ

Adaptive Server DBMS レイヤに含まれる次の主要コンポーネントは、Cluster Edition 環境で動作するように拡張されています。

- バッファ・キャッシュの一貫性 ― 共有バッファ・キャッシュに関する一貫性の問題を処理し、アロケーション・ページ、インデックス・ページ、データ・ページ、オブジェクト・アロケーション・マップ・ページ (OAM ページ)、グローバル・アロケーション・マップ・ページ (GAM ページ) 向けのキャッシュからキャッシュへの転送をサポートします。
- クラスタ・ロック・マネージャ ― クラスタ全体にわたって、一貫性を制御するための分散ロックをサポートします。
- クラスタ・ロギングとリカバリ ― すべてのインスタンスのロギングとフェールオーバ・データベースのリカバリを処理します。
- クラスタ領域とスレッショルド ― 分散環境における領域とスレッショルド管理を処理します。
- オブジェクトの一貫性 ― メタデータとグローバル変数の共有および転送に関する一貫性の問題を処理します。オブジェクトの一貫性では、共有オブジェクトに対する更新が連続的に行われ、クラスタ内のすべてのインスタンスで利用可能な最新の変更が行われる必要があります。
- Workload Manager ― Adaptive Server モジュールの 1 つで、リソース割り当て、可用性、負荷分散をアプリケーションレベルで管理できるようにします。

クラスタ・コーディネータ

クラスタ・コーディネータはメンバシップの管理とリカバリに関連する特定のタスクを処理します。既存のクラスタにジョインしようとするインスタンスは、まずこのクラスタ・コーディネータとやりとりを行います。

特定のインスタンスをクラスタ・コーディネータとして指定する開始パラメータはありません。クラスタ・コーディネータは、クラスタ内の他のインスタンスと同様に設定します。最初に開始したインスタンスがクラスタ・コーディネータになり、クラスタ・コーディネータが抜け出た場合、別のインスタンスが、動的にコーディネータの役割を担うようになります。

クォーラム・デバイス

クォーラム・デバイスには、クラスタの構成情報が含まれており、すべてのクラスタ・メンバによって共有されます。クォーラム・デバイスは、ロー・パーティションに配置し、クラスタ・インスタンスをホストしているすべてのノードからアクセスできるようにします。

Adaptive Server Cluster Edition では、次の目的でクォーラム・ディスクを使用します。

- メンバの参加に関する投票やアービトレーションなどの、クラスタ・メンバシップ管理を実行する場所
- インスタンスと UAF によって使用される構成データを保管する永続的な場所
- 通信メディアと同期ポイント

クォーラム・デバイスには、次に関する情報が含まれます。

- クラスタ名、クラスタ内のインスタンス数、interfaces ファイルを含むディレクトリへのパス、ログ・ファイル、マスタ・デバイス、その他必要な構成情報
- クラスタ内の各インスタンスの状態 (起動または停止) を示すクラスタ・ビュー・レコード
- Adaptive Server のインスタンス障害が検出された場合に、正確なクラスタ・メンバシップを決定するために使用する領域

クラスタの設定時にクォーラム・デバイスを作成します (『インストール・ガイド』を参照)。初期設定の後、sybcluster または qrmutil ユーティリティを使用して、クォーラム・デバイスのバックアップ、リストア、再設定を行います。詳細については、『ASE ユーティリティ・ガイド』を参照してください。

Cluster Edition におけるデータベース・デバイス

Cluster Edition では、ローカル・ユーザのテンポラリ・データベースによって使用されるプライベート・デバイスを除き、データベース・デバイスはロー・デバイス (キャラクター・デバイス) である必要があります。ブロック・デバイスは、個別のハードウェア・ノードでディスク書き込みがバッファされ、クラスタ・インスタンス間で矛盾が発生するため、データベース・デバイスとしては使用できません。

ブロック・デバイス上にプライベート・デバイスを作成できます。プライベート・デバイスは、ローカル・ユーザのテンポラリ・データベースでのみ使用されます。ロー・デバイスを設定する方法の詳細については、オペレーティング・システムのマニュアルを参照してください。

プライベート・デバイス上にローカル・ユーザのテンポラリ・データベースを作成できますが、ローカル・システムのテンポラリ・データベースは、共有デバイス上に作成してください。Cluster Edition では、より安価なローカル・ファイル・システム・デバイス (ブロック・デバイス) を使用して、クラスタ内のテンポラリ・データを保存するニーズを管理できます。これらのデバイスは、プライベート・デバイスとして追加され、ローカル・ユーザのテンポラリ・データベースのみが使用できます。「第 8 章 テンポラリ・データベースの使用」を参照してください。

たとえば、Linux システムでは、*/dev/sda* というパスはブロック・デバイスであるため、使用できません。ただし、このブロック・デバイスを */dev/raw/raw1* のようなロー・デバイスにバインドすることはできます。

Linux システムでは、コマンドを使用して表示されるファイル・タイプでキャラクター (ロー) デバイスを区別できます。次のように、ブロック・デバイスではファイル・タイプとして `b` が含まれ、キャラクター (ロー) デバイスではファイル・タイプとして `c` が含まれています。

```
[joeadminitrator@isles ~]$ ls -l /dev/sda
brw-rw----  1 root disk 8, 0 Nov 29 06:15 /dev/sda

[joeadministrator@isles ~]$ ls -l /dev/raw/raw1
crw-------  1 sybase sybase 162, 1 Nov 29 12:17 /dev/raw/raw1
```

Solaris システムでは、*/dev/dsk/c0t0d0s1* というパスはブロック・デバイスであるため、使用できません。ただし、これと同一の記憶領域には、*/dev/rdsk/c0t0d0s1* というパスを使用して、キャラクター・デバイスとしてアクセスできます。ls -l コマンドをキャラクター・デバイス上で使用すると、シンボリック・リンクの末尾に次のように `raw` と表示されます。

```
janeadministator% ls -l /dev/dsk/c0t0d0s1
lrwxrwxrwx   1 root     root          49 Apr 23  2007 /dev/dsk/c0t0d0s1 ->
../../devices/pci@780/pci@0/pci@9/scsi@0/sd@0,0:b
janeadministrator% ls -l /dev/rdsk/c0t0d0s1
lrwxrwxrwx   1 root     root          53 Apr 23  2007 /dev/rdsk/c0t0d0s1 ->
../../devices/pci@780/pci@0/pci@9/scsi@0/sd@0,0:b,raw
```

HP Itanium システムでは、*/dev/disk/disk4* というパスはブロック・デバイスであるため、使用できません。ただし、この記憶領域には、*/dev/rdisk/disk4* というパスを使用して、キャラクター・デバイスとしてアクセスできます。ls -l コマンドを使用して表示されるファイル・タイプでキャラクター (ロー) デバイスを区別できます。次のように、ブロック・デバイスではファイル・タイプとして `b` が含まれ、キャラクター (ロー) デバイスではファイル・タイプとして `c` が含まれています。

```
[jphui@hpiastr1-HP-UX]:/> ls -l /dev/disk/disk4
brw-r-----   1 bin        sys          3 0x000000 Feb 23 11:40 /dev/disk/disk4
[jphui@hpiastr1-HP-UX]:/> ls -l /dev/rdisk/disk4
crw-rw-rw-   1 bin        sys         23 0x000000 Feb 23 11:40 /dev/rdisk/disk4
```

IBM AIX システムでは、ls -l コマンドを使用して表示されるファイル・タイプでキャラクター (ロー) デバイスを区別できます。次のように、ブロック・デバイスではファイル・タイプとして b が含まれ、キャラクター (ロー) デバイスではファイル・タイプとして c が含まれています。*/dev/hdisk1* というパスはブロック・デバイスであるため、使用できません。ただし、これと同一の記憶領域には、*/dev/rhdisk1* というパスを使用して、キャラクター・デバイスとしてアクセスできます。

```
janeadministator% ls -l /dev/hdisk1
brwxrwxrwx 1 root root 49 Apr 23 2007 /dev/hdisk1

janeadministrator% ls -l /dev/rhdisk1
crwxrwxrwx 1 root root 53 Apr 23 2007 /dev/rhdisk1
```

Cluster Edition のデータベース・デバイスは、SCSI PGR (SCSI-3 Persistent Group Reservations) をサポートしている必要があります。Cluster Edition は、SCSI PGR を使用して、クラスタ・メンバシップの変更時のデータの一貫性を保証します。Sybase では、SCSI PGR をサポートしないディスク・サブシステム上のデータの一貫性は保証できません (Sybase はデータが破損しても容認されるようなテストおよび開発環境のためにこの構成をサポートしています)。

PGR は、SCSI-3 プロトコルの 1 つの機能です。ただし、比較的安価な SATA ディスクを使用している多くの SAN (Storage Area Network) でもこの機能が提供されます。使用しているシステムが SCSI-3 Persistent Group Reservation をサポートするかどうかについては、ストレージ・ベンダにお問い合わせください。

デバイスと I/O フェンシングの補足情報については、使用しているプラットフォーム用のインストール・ガイドを参照してください。

注意　Solaris の非グローバル・ゾーンでは、I/O フェンシングがサポートされていません。

プライベート・インストールと共有インストール

クラスタの設定時に、共有インストールを使用するかプライベート・インストールを使用するかを選択できます。

- 共有インストール ― ネットワーク・ファイル・システム (NFS) またはクラスタ・ファイル・システムを使用して作成した共有ファイル・システムが必要です。共有インストールを使用して作成されたクラスタは、単一の *$SYBASE* インストール・ディレクトリ、Adaptive Server ホーム・ディレクトリ、およびサーバ設定ファイルをサポートします。

- プライベート・インストール – インスタンスごとに、個別の *$SYBASE* インストール・ディレクトリ、Adaptive Server ホーム・ディレクトリ、およびサーバ設定ファイルをサポートします。サーバ設定ファイル間のパリティは、クォーラム・デバイス上のマスタ設定ファイルによって保持されます。

  注意 プライベート・インストールを使用する場合は、ディレクトリ・サービスとして LDAP を使用することをおすすめします。

クラスタ入力ファイルには、インストール・モードが共有かプライベートかを識別する、次のようなエントリが含まれています。

```
installation mode = private | shared
```

プライベート・インストールまたは共有インストールの設定手順については、sybcluster、Adaptive Server オンライン・ヘルプのマニュアル、および使用しているプラットフォーム用のインストール・ガイドを参照してください。また、プライベート・インストール・モードを使用するときにサーバ設定ファイルを保持する手順については、「プライベート・インストール・モード」(165 ページ) を参照してください。

Cluster Edition の Backup Server

Cluster Edition では、ダンプとロード操作を実行する複数の Backup Server を設定できます。次のように、クラスタの設定方法に応じて、dump および load コマンドに適切な Backup Server を経由させます。

- 専用 – クラスタ内の各インスタンスは特定の Backup Server に割り当てられます。
- ラウンドロビン – インスタンスは特定の Backup Server に割り当てられませんが、ダンプまたはロードを行うときには、クラスタによって Backup Server がそのジョブ用に状況によって割り当てられます。

クラスタが 1 つの Backup Server (SYB_BACKUP) を持つように選択できます。「第 5 章 クラスタ環境での Backup Server の使用」 を参照してください。

Cluster Edition によるノンクラスタ・エディションの拡張

Sybase は、Cluster Edition で対称型マルチプロセッシング (Symmetric Multiprocessing: SMP)、ノンクラスタ・サーバのサポートを拡張し、共有ディスク環境でも動作する Adaptive Server を導入しています。

複数のノンクラスタ・サーバをグループ化し、共有データベースの単一のシステム・ビューを作成することで、信頼性を向上させ、管理を容易にすることができます。

それぞれがノンクラスタ・サーバとして機能する疎結合された複数の Adaptive Server が連携することで、ユーザには単一のデータベース・システムのイメージが提供されます。

クラスタ内の Adaptive Server は、共有ディスク・デバイス上に存在する Adaptive Server データベースの単一のインストールを共同管理します。

Cluster Edition アーキテクチャには、主に次のような利点があります。

- 可用性の向上 – アーキテクチャの共有ディスク性とは、複数の他のクラスタ・メンバが故障した後でも、クラスタ・メンバが 1 つでも稼動していれば、アプリケーションが引き続き稼動できることを意味します。
- 単一管理 – データがすべてのインスタンスで共有されているため、クラスタのメンバシップの変更に応じてデータのパーティションを再設定する必要がありません。

注意　Cluster Edition は、分散アーキテクチャを可能にします。ノード間通信は、ノンクラスタ Adaptive Server とは異なり、共有メモリ (高速システム・メモリ) ではなくネットワーク間通信を通じて実行されます。ノード間メッセージングを最小化するアプリケーションを使用すると、Cluster Edition 環境で最適のパフォーマンスを得ることができます。

クラスタにおける相互接続ネットワークの使用

Cluster Edition を使用すると、クラスタ内のインスタンス間で 1 つまたは 2 つの相互接続ネットワークを構成できます。1 つの相互接続ネットワークでも十分ですが、2 つ構成することで、冗長性のある、より堅牢なクラスタを構成できます。相互接続ネットワークは、クラスタ内のすべてのインスタンス間に一連の論理リンクを形成します。これらのリンクは、インスタンス間のメッセージのやりとりに使用され、Cluster Edition によってモニタリングされます。障害の発生が検出されると、インスタンス間の別のネットワークにトラフィックがルート変更されます。

相互接続リンクでは、さまざまな原因による障害が発生する可能性があります。切断または壊れたケーブルなどの物理的障害、ネットワーク・インフラストラクチャ機器などの電源障害、ネットワーク・スタック内部のソフトウェア障害などです。Cluster Edition では、インスタンス間のトラフィックの流れをモニタリングすることでこれらの障害を検出します。各インスタンスが、さまざまなリンク経由で送信されたメッセージをモニタリングします。メッセージが受信されているかぎり、リンクは機能しているとみなされます。

ノードに故障がある場合やリンクが停止中の場合、インスタンスはメッセージを送らないことがあります。クラスタ内の特定のインスタンスから応答がない場合、Cluster Edition によってアクティブ・プロービング・メカニズムが開始され、そのインスタンスをサポートしているノードとネットワーク・リンクを使用して通信できるかどうか判断されます。このメカニズムにより、非活動期間に誤ってリンク障害がトリガされ、切り替えイベントが実施されるようなことがないように保証されています。

あるリンクに動作不可能というフラグが付いた場合は、定期的にリンクの確立が試行され、手動による対処なしにリンクを回復させて通常の動作が再開できるようにします。

注意 リンクのモニタリングは、Cluster Edition 内に複数のネットワークが定義されると、自動的に実装されます。

インスタンス間のリンクのモニタリング

monCIPCLinks モニタリング・テーブルは、クラスタ内のインスタンス間のリンク状態をモニタリングします。monCIPCLinks には、各リンクに関し「パッシブ」と「アクティブ」という 2 つの状態があります。

注意 論理クラスタとクラスタ内の各インスタンスは、異なる状態になることがあります。

- 論理クラスタは、たとえば、クラスタがオフラインかオンラインかを判断するような全体のグローバルな状態を保持しています。
- 各インスタンスは、論理クラスタがその特定のインスタンスに関して把握した状態を表す状態も保持しています。

クラスタの「状態」の詳細については、「クラスタとインスタンスの状態」(88 ページ) を参照してください。

「パッシブ」状態は、リンクを介して送信される日常的なメッセージのモニタリングに使用されます。Cluster Edition は、リンク上にメッセージ・トラフィックがない場合にアクティブ状態を収集します。リンクのステータスは、「状態」として表され、各状態には継続時間 (ミリ秒) が関連付けられています。この状態には、「Up」、「Down」、「In doubt」があります。インスタンス間でメッセージが送信されない場合、状態は「In doubt」になります。

クラスタが正常に動作している場合は、通常のノード間トラフィックを使用してリンクの状態が判断されます。これは、パッシブ・モニタリングと呼ばれ、リンクのパッシブ状態が保持されます。非活動期間に、状態を判断するためのモニタリングがクラスタ内で実行されると、定義された状態が古い、または信頼できない可能性があります (非活動期間に「Up」であると判断された状態が実は「Down」であるにもかかわらず、非活動期間であるために monCIPCLinks テーブルの結果セットにそのことが表示されないなど)。この停止中の状態は、[PassiveState] カラムに「In doubt」と表示されます。いったん、リンクが「In doubt」にマークされると、アクティブ・リンク状態モニタリングがトリガされ、[ActiveState] カラムに表示される値が有効になります。

アクティブ状態とパッシブ状態には、それぞれ継続時間が関連付けられており、その状態が最後に更新された時期を表しています。リンクの状態を維持するのに十分な量のトラフィックが通常発生している場合は、アクティブの状態が更新されず、この状態に関連する継続時間の値が大きくなります。値が大きくなると、それに関連する状態は、リンクの真の状態を正確に表していない可能性があります。

インスタンスがメッセージを送信していない場合は、[PassiveState] が「In doubt」としてリストされますが、[ActiveState] には、「Up」、「In doubt」、または「Down」のいずれかが表示されます。

次の例は、2 ノードのクラスタ (両方のリンクが動作中で、相互にトラフィックが流れている) を示します。すべてのリンクの [PassiveStateAge] が 0 であるため、出力結果はリンク状態を正確に表しているものと考えられます。

```
InstanceID    LocalInterface     RemoteInterface         PassiveState
      PassiveStateAge    ActiveState    ActiveStateAge

------------   --------------     ------------            --------------
      --------------     ------------   ---------------
2             ase2               ase2                    Up
      0                  Up             10300
2             blade2             blade1                  Up
      0                  Up             0
1             ase1               ase2                    Up
      0                   In doubt       17900

1             blade1             blade2                  Up
      0                   Up             100
```

次の例は、同じく 2 ノードのクラスタですが、プライマリの相互接続ネットワークに障害が発生した状態を表します。ネットワークの終了ポイント「ase1」と「ase2」間のリンクに関する [PassiveState] の値は「In doubt」で、[PassiveStateAge] は「大きい」値になっています (つまり、[ActiveState] がリンクの真の状態を表していることを示します)。[ActiveState] の値は、より最近のもので、リンクが「Down」状態であることを示しています。

```
InstanceID    LocalInterface     RemoteInterface         PassiveState
      PassiveStateAge    ActiveState    ActiveStateAge

------------   --------------     ------------            --------------
      ---------------     ------------   ----------------
2              ase2               ase1                    In doubt
      13500              Down           700
2              blade2             blade1                  Up
      0                  Up             700
1              ase1               ase2                    In doubt
      13600              Down           400
1              blade1             blade2                  Up
      0                   Up             400
```

注意 リンクの障害時とアクティブな状態がリンクの状態を正確に反映するまでの時間の間には、わずかに遅延が発生します。

[ActiveStateAge] に「大きい」値が表示されている状態は、リンクが古く、値が正確でない可能性があるので、すべて無視します。リンク状態が古く、[ActiveStateAge] の値が「大きい」場合、メッセージの送信が行われないとアクティブ・モニタリングがトリガされますが、リンクの状態はまだ判断されていません。

注意 クラスタ入力ファイルでプライマリ相互接続ネットワークとセカンダリ相互接続ネットワークの両方を設定する場合、両方の相互接続ネットワークを実行するまで、クラスタを再起動しないでください。

推奨される展開シナリオ

一般的に、Cluster Edition は、使用しているプラットフォーム向けのリリース・ノートに一覧表示されている機能を含むシナリオを除いては、この項で説明したシナリオをサポートしています。

クラスタ・データベース・アーキテクチャの採用を検討しているほとんどのユーザには、次のような目的があります。

- 可用性の向上 － 1 つのノードに障害が発生しても、クラスタ内のその他のノードが動作を続行し、データベースは引き続き利用可能になります。
- 管理性の向上 － 複数のアプリケーションとデータベースを単一のクラスタに統合できるので、管理の複雑さをなくし、スケールメリットを享受できます。
- スケーラビリティの向上 － 複数のノードがサポートされるので、単一ノードの環境における制限を超えて、クラスタ・データベースを拡張できます。一般的に「垂直スケーラビリティ」と表現され、より多くの CPU、メモリ、ホスト・バス・アダプタ (HBA)、ネットワーク・インタフェース・カード (NIC) などを追加することで、単一ノードの処理能力を増強します。
- 総所有コストの削減 － 独自仕様でなく、業界標準のハードウェア上にソフトウェアが展開されるので、購入、維持、サポートのコストが削減されます。

OLTP アプリケーション用の HA フェールオーバ

現在、高可用性 (HA) 機能を使用した運用環境を使用している場合は、アクティブ － パッシブ (スタンバイ) またはアクティブ － アクティブ (コンパニオン) 構成の Sybase Failover、またはオペレーティング・システムが提供するクラスタウェア (たとえば、Sun Cluster や Veritas Cluster) による現在の環境を、次に説明する Cluster Edition シナリオのいずれかで置き換えます。

アプリケーションのサービスレベル・アグリーメントと財務的制約に応じて、自社の要求を最もよく満たすフェールオーバのシナリオを選択し、設定します。

- *1:1* アクティブ－パッシブ

  クラスタ・ノードとインスタンスが、1 台ずつのそれぞれに対応するアイドル (パッシブ)・ノードとインスタンス (アクティブ・ノードでの障害の発生を待機) とペアで設定されます。これは、複数の障害を含むあらゆるフェールオーバ・シナリオにおいてサービスの低下が許されない、極端な環境でのみコスト効率のよいシナリオです。

- *1:1* アクティブ−アクティブ

  クラスタ・ノードとインスタンスがそれぞれペアでセットアップされます。各ペアは、他方に障害が発生した場合に備えてお互いを「コンパニオン・モード」でモニタリングする一方で、個別のアプリケーションとデータベース (インスタンス) にサービスを提供します。このシナリオは、現状の Sybase HA オプションを模倣したもので、リソースは完全に利用されますが、単一インスタンスで両方の負荷を処理するのに十分な能力を提供できるように、通常処理時のリソースの処理能力を低く余裕を持たせた状態 (50 % 未満) に維持しないかぎり、フェールオーバ処理中のサービス・レベルが低下します。

- *N:1 (*単一のパッシブ・スタンバイ・ノードによってカバーされる *N* 個のアクティブ・ノード*)*

  このシナリオでは、任意の数のアクティブ・インスタンスをモニタリングする単一のパッシブ・スタンバイ・ノードとインスタンスが提供されます。これは、最もコスト効率のよいシナリオです。ただし、複数のノードに障害が発生し、複数のインスタンスが単一のノードにフェールオーバされてしまうと、このシナリオが容認できなくなるほどのサービス・レベルの低下につながる恐れがあります。最も起こりそうなフェールオーバのシナリオの中でサービス・レベルの低下を緩和できるように、パッシブ・ノードの処理能力 (たとえば、CPU やメモリ) を追加して構成するようにしてください。

- *N:M (M* 個のパッシブ・スタンバイ・ノードによってカバーされる *N* 個のアクティブ・ノードおよびインスタンス*)*

  このモデルでは、任意の数 (M 個) のパッシブ・スタンバイ・インスタンスが、任意の数 (N 個) のアクティブ・インスタンスをモニタリングします。このオプションでは、最も典型的な障害のシナリオをカバーしつつ、コストを削減できます。パッシブ・スタンバイ・ノードの数を選択する要因として考えられるのは、通常、コスト、システムレベルの平均故障間隔 (MTBF) の統計、複数ノードの障害シナリオにおいてサービス・レベルが低下した状態で運用した場合のビジネスへの影響などです。ほとんどの障害シナリオでは、このようなサービス・レベルの低下を緩和するために、パッシブ・ノードに処理能力を追加して構成できます。

DSS レポート・アプリケーション向けの水平スケーラビリティ

主に、多くのユーザからクエリが要求されるような大規模レポートや意思決定支援システム (DSS) で構成される、オンライン操作を行うアプリケーションを使用している場合は、すべての Adaptive Server インスタンスが同一のアプリケーションにサービスを提供するような複数ノード・クラスタを作成することを検討してください。

このシステムを編成する場合は、次の点に注意してください。

- 負荷が 1 から *N* ノードに拡張する場合のユーザ、クエリ、応答時間のスケーラビリティ。
- インスタンス間におけるクライアントの負荷分散、クライアントとインスタンスの処理能力およびパフォーマンスとの関係。

OLTP アプリケーション向けの水平スケーラビリティ

オンライン・トランザクション処理 (OLTP) アプリケーションなど、頻繁に読み書きを行うアプリケーションでは、データベース・システムの ACID プロパティを維持するためにリソース競合が発生し、これまでスケーラビリティに難題をもたらしてきました。共有ディスクのクラスタ技術を使用する OLTP アプリケーションの水平スケーラビリティでは、共有ディスクのクラスタ内のサーバ・インスタンス間で必要とされるデータの一貫性およびメッセージ送信により、さらなる課題が発生しています。現在のコンピュータとネットワークは、共有ディスクのクラスタ上で OLTP アプリケーションを無限に拡張できる構造にはなっていません。ノード数と負荷、およびユーザ数が増えるに従って、ノード間でバッファの一貫性を保つために必要なノード間メッセージの量が飛躍的に増加します。

共有ディスクのクラスタ上で OLTP の負荷を増やす最適な方法は、アプリケーションとデータを相互排他的なセットに分けること (つまり、データを別々のデータベースに分けること) によって、複数のサーバ・インスタンス間で処理の調整が行われるのを避け、アプリケーション用のデータに同じインスタンスからアクセスするようにすることです。そのためには、分割されたインスタンス間でログとデータが競合しないようにするために、データベース・レベルで「データ」をどのように分割するかを入念に考慮してください。論理クラスタと負荷管理を採用した単一のインスタンスを使用することで、セグメント化された「データ」へのアクセスを容易にできます。

論理クラスタを使用すると、異なるアプリケーションまたは負荷に対して明確にインスタンスを割り当てたり、これらのデータ・グループが単一のクラスタの下で論理的に動作するようにしたりできます。これによって、インスタンス間のアクセスを減らし、インスタンスごとに専用のテンポラリ・データベースと組み合わせることで、Cluster Edition 内でデータの可用性を継続的にサポートする OLTP アプリケーションの展開を容易にできます。

Cluster Edition における新しいクライアント技術

Cluster Edition は、単一のシステムとしてアクセス可能なシステムをサポートします。つまり、クラスタを構成する複数のインスタンスが、クライアントには単一のシステムとして表示されます。新しいクライアント・テクノロジにより、クライアントは個々のインスタンスとの物理的な接続を維持しながら、クラスタに論理的に接続できます。この論理的な接続により、Adaptive Server はクライアントをクラスタ内のさまざまなインスタンスにリダイレクトし、高可用性フェールオーバ・データをクライアントに動的に提供できます。「第 2 章 クライアント・アプリケーションとクライアント／サーバの対話」および『新機能 Open Server 15.0 および SDK 15.0 Microsoft Windows、Linux、UNIX 版』を参照してください。

新しいクライアント技術には、次のようなものがあります。

- ログイン・リダイレクト － クライアントがクラスタ内の他のインスタンスに再接続する技術。
- 接続マイグレーション － 確立した接続がクラスタ内の他のインスタンスに移動する技術。
- 高可用性フェールオーバの拡張 － クライアントが最初に利用可能なインスタンス、または最も負荷の低いインスタンスを発見するまで、複数回フェールオーバできる機能。

複写のサポート

Cluster Edition では、Replication Server と RepAgent スレッドを使用した複写をサポートしています。クラスタ・データベースは、Sybase のクラスタ・システムにおいて、送信元または送信先のいずれにもなることができます。RepAgent の設定や複写対象テーブルへのマーク付けなど、すべてのタスクをクラスタ内の任意のインスタンスから実行できます。複写ステータスは、クラスタ全体で一貫しています。

RepAgent の設定

クラスタ・システムでプライマリ・データベースを設定する場合、指定するサーバ名はクラスタ名にしてください。クラスタ名は、select @@servername を使用して表示できます。

sp_config_rep_agent の構文では、クラスタ名またはインスタンス名は必要ありません。デフォルトで、Cluster Edition とノンクラスタ Adaptive Server エディションの両方で、select @@servername の値が考慮されます。Cluster Edition では、この文は現在のクラスタ名を返します。次に例を示します。

```
1> select @@servername
2> go

----------------------------
MYCLUSTER
```

RepAgent の起動

デフォルトでは、RepAgent はコーディネータ上で起動します。

ただし、クラスタ内の任意のインスタンス上で起動するように設定することもできます。たとえば、プライマリ・データベース pdb の RepAgent が常に「ase2」インスタンス上で起動するように設定するには、次のように入力します。

```
sp_config_rep_agent pdb, "cluster instance name", "ase2"
```

設定後、新しい設定を有効にするには、sp_start_rep_agent を使用して RepAgent を再起動してください。

RepAgent が常にコーディネータ上で起動するデフォルトの動作に戻すには、次のように入力します。

```
sp_config_rep_agent pdb, "cluster instance name",
"coordinator"
```

インスタンスは、そのノード上で起動すると、データベースがそのノード上で起動するように設定されているかどうかをチェックします。起動するように設定されていて、データベースが自動的に起動するようにマーク付けされている場合は、RepAgent が起動します。

コーディネータが起動すると、特定のインスタンス上で起動するように設定されていないすべての RepAgent が起動されます。コーディネータで障害が発生するか、適切に停止された場合、RepAgent は新しいコーディネータで起動します。

RepAgent がコーディネータ以外の特定のインスタンス上で起動するように設定されている場合で、そのインスタンスが停止するか、または障害が発生した場合、RepAgent はコーディネータ上で起動します。

注意　Cluster Edition では、未サポートの dbcc logtransfer インタフェースを必要とする Adaptive Server Enterprise Replicator はサポートされていません。

第 2 章

クライアント・アプリケーションとクライアント／サーバの対話

この章では、クライアント／サーバの対話、およびクラスタをサポートするために Open Client/Client-Library を呼び出すアプリケーションを変更する方法について説明します。共有ディスク・クラスタ環境での isql の使用方法についても説明します。

注意 DB-Library は、共有ディスク・クラスタ内のインスタンスに接続できますが、Sybase 共有ディスクまたは高可用性機能はサポートしません。

Cluster Edition とともに出荷されるバージョンの Open Client™、jConnect™ for JDBC™、ODBC、OLE DB、ADO.NET は以下をサポートします。

- ログイン・リダイレクト – ログインを確認する前に、着信クライアント接続を別のインスタンスにリダイレクトするインスタンスの機能。ログイン・リダイレクトはログイン手順中に行われます。クライアント・アプリケーションは、リダイレクトされた通知を受け取りません。
- 接続マイグレーション – 既存のクライアントが、クラスタ内の、1 つのインスタンスから別のインスタンスに転送されるときに行われます。マイグレーションが行われるとき、および接続基準の詳細については、「接続マイグレーション」(28 ページ) を参照してください。
- 高可用性フェールオーバの拡張 – 高可用性フェールオーバの拡張設定では、「高可用性対応」クライアントが接続するとき、それらクライアントへのフェールオーバ・アドレスのリストを Adaptive Server が提供します。これによって、高可用性対応クライアントまたはアプリケーションは、接続されているインスタンスが停止した場合、複数回フェールオーバできます。

これらのクライアントでは、interfaces ファイルまたはディレクトリ・サービスに HAFAILOVER エントリは必要ありません。ただし、interfaces ファイルまたはディレクトリ・サービスに HAFAILOVER エントリがある場合、クライアントは Adaptive Server がフェールオーバ・アドレス、または接続先サーバのリストを送信するまでこのエントリを使用し続けます。クライアントは常に、Adaptive Server が提供する最新のリストを使用します。

ログイン・リダイレクトと接続マイグレーションを実装する場合、クライアント・ライブラリの最新のコピーをアプリケーションが使用することを確認します。

- アプリケーションが共有ライブラリにリンクされる場合は、ライブラリ検索パス内で、新しいクライアント・ライブラリを古いクライアント・ライブラリの前に記入します。
- アプリケーションが静的にリンクされている場合、アプリケーションを再リンクします。

接続マイグレーションを有効または無効にするには、CS_PROP_MIGRATABLE 接続プロパティを使用します。CS_PROP_MIGRATABLE は、デフォルトでオンです。『Client-Library リファレンス・マニュアル』を参照してください。

フェールオーバの拡張を実装する場合、クライアント・ライブラリの最新のコピーをアプリケーションが使用すること、および高可用性を有効にしてあることを確認します。高可用性の有効化の詳細については、『高可用性システムにおける Sybase フェールオーバの使用』を参照してください。

jConnect、ODBC、OLE DB、ADO.NET ドライバについてフェールオーバ機能を使用する場合の詳細については、『新機能 Open Server 15.0 および SDK 15.0 Microsoft Windows、Linux、UNIX 版』を参照してください。

Open Client

ログイン・リダイレクト、接続マイグレーション、高可用性フェールオーバの拡張がこれらのバージョンについてサポートされています。

- ログイン・リダイレクト － Open Client バージョン 15.0
- 高可用性拡張 － OCS バージョン 15.0 ESD #3
- マイグレーション － OCS バージョン 15.0 ESD #8

これらのバージョンをサポートする SDK コンポーネントは OpenClient/Client-Library です。

Client-Library アプリケーションにおけるフェールオーバの有効化

既存のアプリケーションは Cluster Edition に接続できます。ただし、既存の高可用性 (HA) 機能を使用するためには、アプリケーション・コードを変更する必要があることがあります。

- 既存の Adaptive Server HA 機能に基づいた既存の HA アプリケーションの場合、アプリケーション・コードの変更は必要ありません。
- 既存の非 HA アプリケーションは、コード変更がないか、またはわずかな変更で Cluster Edition における HA 機能のいくつかの利益を得られることがあります。しかし、これらの場合、フェールオーバは透過的になりません。フェールオーバが最初に検出された時点で、アプリケーションはエラー・メッセージを受け取ります。フェールオーバを開始するために、ユーザはバッチまたはトランザクションを再実行する必要があります。

  非 HA アプリケーションについてフェールオーバを有効にするには、次のようにします。

  - isql の場合、Adaptive Server に接続するときに -Q オプションを指定します。
  - Client-Library とリンクするアプリケーションの場合、フェールオーバを有効にする、対応する接続プロパティを設定します。

フェールオーバをユーザに透過的にするには、アプリケーションは、積極的にフェールオーバ・エラー・ステータスを確認し、バッチまたはトランザクションを自動的に再実行する必要があります。

いずれの場合でも、アプリケーションが使用する Client-Library バージョンを更新し、クラスタ関連 HA 機能を使用する必要があります。

Client-Library アプリケーション内でフェールオーバを有効にするには、次の手順に従います。

1 ct_config または ct_con_props Client-Library API 呼び出しを使用し、コンテキストまたは接続レベルのどちらかで、CS_HAFAILOVER プロパティを設定します。

```
ct_config (context, action, CS_HAFAILOVER, buf, buflen,
&outlen);
ct_con_props(connection, action, CS_HAFAILOVER, buf,
buflen, &outlen);
```

CS_HAFAILOVER プロパティの詳細については、『Client-Library/C リファレンス・マニュアル』を参照してください。

2 ダウンしているインスタンスに接続しようとする場合、動作はノンクラスタ Adaptive Server と同じです。Client-Library は、interfaces ファイル内のインスタンス名に関するすべてのクエリ・エントリのうち 1 つが動作するか、または試みるものがなくなるまで接続を試みます。すべてのインスタンスについて、クライアント側の interfaces ファイル内に query 行を記入します。アプリケーションは interfaces ファイルの一連のクエリ・エントリによってクラスタに接続できます。interfaces ファイルの詳細については、『Cluster Edition Installation Guide』を参照してください。

3 フェールオーバが正常に完了すると、Client-Library は CS_RET_HAFAILOVER という名前の戻り値を発行します。これは、以下を含む、いくつかの Client-Library API 呼び出しに特有のものです。

```
ret = ct_results(cmd, result_type)
ret = ct_send(cmd)
```

同期接続 (これはサーバ応答を要求するルーチンであり、応答を受信するまでブロックします) の間、API 呼び出しから CS_RET_HAFAILOVER が返されます。非同期接続 (サーバ応答が CS_PENDING をただちに返すことを要求するルーチン) では、これらの API は CS_PENDING を発行し、callback 関数は CS_RET_HAFAILOVER を返します。リターン・コードに従って、要求された処理を実行し、コンテキストを設定し、実行すべき次のコマンドを送信できます。

4 Cluster Edition とともに出荷された Open Client SDK のバージョン以降の Open Client SDK を使用してアプリケーションを再構築します。

高可用性向けにアプリケーションを設定する場合の詳細については、『高可用性システムにおける Sybase フェールオーバの使用』を参照してください。

クライアント／サーバの対話

このセクションで説明する機能は、Open Client 15.0 ライブラリを使用するものであり、自動的に有効化されます。

ログイン・リダイレクト

ログイン・リダイレクトは、ログイン時に負荷を考慮して、クライアントが別のインスタンスにログインするように指示された場合に発生します。

クライアント・リダイレクトのために追加設定を行う必要はありません。

ログイン・リダイレクトは、Adaptive Server Workload Manager が論理クラスタ設定およびクラスタの現在の負荷に基づいて、着信接続を特定のインスタンスに送信する場合に使用されます。

ダウンしているインスタンスに接続しようとする場合、動作はノンクラスタ Adaptive Server と同じです。クライアントは、接続が正常に完了するまで、指定サーバのディレクトリ・サービス内のすべてのエントリを試みます。このため、ディレクトリ・サービス内のサーバ・エントリには、クラスタ内のすべてのインスタンスに関する接続情報が記入されている必要があります。

次の例にあるのは、クラスタ “mycluster” 内で実行中のマシン “blade1”、“blade2”、“blade3”、“blade4” 上の Adaptive Server である “ase1”、“ase2”、“ase3”、“ase4” です。

```
ase1
   query tcp ether blade1 19786
ase2
   query tcp ether blade2 19786
ase3
   query tcp ether blade3 19786
ase4
   query tcp ether blade4 19786
mycluster
   query tcp ether blade1 19786
   query tcp ether blade2 19786
   query tcp ether blade3 19786
   query tcp ether blade4 19786
```

たとえば、クライアントは、クラスタ “mycluster” に接続する場合、最初に “ase1” インスタンスに接続しようとします。“ase1” がダウンしている場合は、interfaces ファイル内にある “ase2” などの次のエントリへの接続を試みます。接続が正常に完了した後、Workload Manager は負荷ルールに基づいてクライアントを別のインスタンスにリダイレクトすることがあります。

インスタンスは interfaces ファイルに指定された順序で接続を試みますが、ホストまたはネットワークに到達できない場合、またはこれらがダウンしている場合、接続の試みが失敗するまでにかなりの時間がかかることがあります。接続情報にログイン・タイムアウトを追加して、再試行できます。

先の例で、クライアント接続についてデフォルトより短いログイン・タイムアウト間隔を指定すると、クライアントはインスタンス “ase2” により素早い接続を試みることができます。

詳細については、『ASE ユーティリティ・ガイド』の isql -l パラメータの説明、および『Client-Library リファレンス・マニュアル』の CS_LOGIN_TIMEOUT プロパティの説明を参照してください。

ログイン・リダイレクト用の接続プロパティ

接続プロパティを設定してログイン・リダイレクトを設定します。

- CS_PROP_REDIRECT − ログイン・リダイレクトを有効および無効にします。
- CS_DS_RAND_OFFSET − ct_connect のディレクトリ・サービス・ルックアップからランダムに取得した最初のクエリ・エントリの作成を無効または有効にします。デフォルトで、このプロパティは false に設定されます。

『Client-Library リファレンス・マニュアル』を参照してください。

接続マイグレーション

接続マイグレーションは、既存のクライアントが、クラスタ内の、1 つのインスタンスから別のインスタンスに転送されるときに行われます。たとえば、現時点で接続されているインスタンスがメンテナンス用にダウンするため接続がマイグレートする場合や、負荷分散のためにマイグレートする場合があります。転送はクライアント・アプリケーションに透過的です。接続マイグレーションによって、クラスタは、論理クラスタ全体で負荷のバランスを取ることができます。

接続マイグレーションによって、Workload Manager は、管理的フェールオーバ、フェールバック、論理クラスタのオフラインへの移行の間に、インスタンス間で既存の接続を適切に移動できます。Workload Manager は動的負荷分散用にマイグレーションを使用できます。この間、負荷をより均等に分散させるために、いくつかの既存の接続がインスタンスの間でマイグレートされます。

接続マイグレーションは、インスタンスが Open Client 15.0. クライアント・ライブラリを使用するとき、自動的に有効になります。接続マイグレーションのために追加設定を行う必要はありません。

マイグレーションとフェールオーバの違い

マイグレーションは、Adaptive Server が要求する計画的で、制御されたイベントです。フェールオーバは、Adaptive Server クラッシュまたはネットワーク切断の後に発生する予期しないイベントです。

アプリケーションがマイグレーションを認識することはなく、ユーザはマイグレーション用のコードを作成する必要はありません。ただし、フェールオーバ・サポート用のアプリケーションは明確にコード化する必要があります。

マイグレーションは、新しいインスタンスにおいてすべてのセッションのコンテキストをリストアします。一方、フェールオーバではアプリケーションが自身のコンテキストをリストアします。

マイグレーションが行われるとき

Workload Manager は、要求をクライアントに正常に送信できるときに、マイグレーションを開始できます。特に、マイグレーションは以下のときに行われる可能性があります。

- ログインを完了した接続については、次のようなとき。
  - インスタンスがクライアントから新しいバッチを受信した後で、しかもバッチが解析および実行される前。
  - インスタンスがクライアント・バッチの処理を完了した後で、しかも最終的な完了をクライアントに送信する前。
  - インスタンスがクライアントの代わりにバッチを実行していないとき。
- Workload Manager のアルゴリズムに従う。Workload Manager は、負荷分散のため、または現在のインスタンスがメンテナンス用にダウン中であるため、特定のクライアントをマイグレートする場合があります。
- 接続のコンテキストがマイグレーションを調整するとき。Workload Manager は、マイグレーション用の接続を対象にしますが、これらの接続がマイグレートするのはコンテキストで許容されるときのみです。特に、トランザクション内部でマイグレーションが行われることはありません。

注意　Java を実行している接続については、接続のマイグレーションはサポートされません。「クラスタ環境での Java の使用」(168 ページ) を参照してください。

マイグレートされたコンテキスト

ソース・インスタンスは、マイグレーションを正常に完了した時点でクライアントのすべてのコンテキストを送信先インスタンスに伝達し、送信先インスタンスはコンテキストを取得します。

クライアントのすべてのコンテキストは、マイグレーションが完了した後にリストアされるので、マイグレーションはクライアントに完全に透過的になります。しかし、マイグレートされた接続は新しい spid を取得します。

クライアントのコンテキストは次の要素で構成されます。

- 現在のデータベースの名前
- マイグレーションが事前バッチ・モードで実行される場合は、コマンドの保留中のバッチ
- クライアントのログイン・レコード
- クライアントの言語と文字セット

- クライアントの機能
- モニタ・カウンタと統計
- 役割
- set オプション
- トレース・フラグ 3604 および 3605

マイグレーションの基準

マイグレーションは非同期イベントです。タスクにマイグレートするようにという要求が発行され、タスクがクワイス (静止) 状態に到達したときにのみタスクはマイグレートします。Cluster Edition の場合、クワイス (静止) 状態は次のようになります。

- クエリ・バッチを実行していない
- 開いているトランザクションがない
- セッションレベルのテンポラリ・テーブルがない
- 宣言されたカーソルがない
- 最初の接続以降、パスワードを変更していない
- set user または set proxy を実行していない
- 論理プロセス・マネージャを使用して、エンジンにバインドしていない
- ASE Active Messaging を使用していない
- インバウンド・サイト・ハンドラと関連づけられた論理接続でない
- シングルユーザ・モードでデータベースを開いたままにしていない

コンテキスト・マイグレーション

Adaptive Server は、既存の接続を別のインスタンスにマイグレートするとき、現在のデータベースや set オプションのような、既存の接続のコンテキストもいくつかマイグレートする必要があります。

クライアント・マイグレーションの設定

idle migration timer 設定パラメータと session idle timer 設定パラメータは、アイドル状態のクライアントがマイグレートするタイミングを制御します。この 2 つのパラメータの合計によって、マイグレーションの完了が予測される時間 (秒単位) の上限が決まります。

idle migration timer のデフォルト設定は 60 秒です。session idle timer のデフォルト設定は 600 秒です。

idle migration timer を 0 に設定すると、インスタンスは、マイグレーション要求を送信した直後に、マイグレーション要求を発行した接続を閉じます。session idle timer を 0 に設定すると、インスタンスは、idle migration timer が期限切れになる前に完了していないアイドル・マイグレーションを無効にします。

session idle timer を大きな値に設定すると、アイドル状態の接続が正常にマイグレートする可能性が増大します。しかし、インスタンスは、クライアントをマイグレートするコンテキストを長時間保持する必要があります。また、idle migration timer と session idle timer の両方を 0 に設定するとマイグレーションは無効になります。『システム管理ガイド 第 1 巻』を参照してください。

次の例で、idle migration timer と session idle timer のさまざまな設定について説明します。

両方のパラメータをデフォルトに設定した場合

- 非同期通知を処理できないアイドル状態のクライアントに対してインスタンスがマイグレーション要求を発行し、idle migration timer の期間切れの前にユーザまたはアプリケーションがクライアントでコマンドを発行した場合、クライアントはただちにマイグレートし、コマンドは送信先インスタンスで実行されます。

  「先読み」クライアントは、アプリケーションが要求する前に、保留中のデータをネットワークから読み込みます。

- 以下がすべて当てはまる場合、インスタンスはクライアントへの接続を閉じます。
  - idle migration timer と session idle timer の両方がデフォルト値に設定されている。
  - インスタンスがアイドル状態の非先読みクライアントに対してマイグレーション要求を発行。
  - idle migration timer 持続期間中のアイドル状態のクライアント。
- インスタンスが接続を閉じた後、最初の 600 秒 (idle migration timer のデフォルト値) 間にクライアント上でコマンドが発行されると、クライアントは正常にマイグレートし、ターゲット・インスタンス上でコマンドが実行されます。
- 以下が当てはまる場合、インスタンスはクライアントへの接続を閉じます。
  - idle migration timer と session idle timer の両方がデフォルト値に設定されている。
  - インスタンスがアイドル状態の非先読みクライアントに対してマイグレーション要求を発行。
  - idle migration timer の期間切れの前に、クライアントに対してコマンドが発行されていない。

- 以下が当てはまる場合、インスタンスはマイグレーションを無効にします。
  - idle migration timer と session idle timer の両方がデフォルト値に設定されている。
  - インスタンスがアイドル状態の非先読みクライアントに対してマイグレーション要求を発行。
  - idle migration timer の期間切れの前に、クライアントに対してコマンドが発行されていない。

    コマンドが発行されるので、クライアントはマイグレーション要求を検出するとただちにマイグレートしようとします。しかし、インスタンスはマイグレーション要求を拒絶し、クライアントは初期インスタンスを継続しようとします。インスタンスへの接続は閉じられるので (idle migration timer が期間切れ)、HA 機能がある場合、クライアントは HA フェールオーバを試みます。HA 機能がない場合、クライアントはアプリケーションに切断を報告します。

idle migration timer を 0 に、session idle timer を 600 に設定した場合

インスタンスがアイドル状態の非先読みクライアントに対してマイグレーション要求を発行する場合、インスタンスは、マイグレーション要求を送信した後、ただちにクライアントへの接続を閉じます。最初の 600 秒の間にクライアントに対してコマンドが発行されると、クライアントは正常にマイグレートします。そうでない場合は、前の項で説明したのと同じように失敗します。

idle migration timer を 60 に、session idle timer を 0 に設定した場合

インスタンスは、アイドル状態の非先読みのクライアントに対しマイグレーション要求を発行します。idle migration timer が期間切れになる前にクライアントに対してコマンドが発行されると、クライアントは正常にマイグレートします。しかし、idle migration timer が期間切れになる前にコマンドが発行されなくても、session idle timer が 0 に設定されているので、クライアントはマイグレートできます。

idle migration timer と session idle timer を 0 に設定した場合

両方のパラメータが 0 に設定されている場合、どちらのインスタンスもクライアントをマイグレートしません。

高可用性フェールオーバの拡張

Adaptive Server は接続時に「HA 対応」クライアントに対してフェールオーバ・アドレスのリストを提供します。これにより、接続先のインスタンスが使用不可能になるたびに、高可用対応クライアントまたはアプリケーションで複数回フェールオーバを行うことができます。フェールオーバ・リストがクライアントに送信されていない場合、クライアントは interfaces ファイルの HAFAILOVER エントリ情報を使用します。

次の例の場合、インスタンスが高可用性拡張リストを送信する前の、ログイン中にネットワーク障害があった場合でも、HA 対応クライアントはフェールオーバできます。

```
ase1
        query tcp ether blade1 19786
ase2
        query tcp ether blade2 19786

mycluster
        query tcp ether blade1 19786
        query tcp ether blade2 19786
        hafailover mycluster
```

クライアント・アプリケーションではクラスタのインスタンスへの接続を確立できるまでクエリの各行を試みるため、HAFAILOVER エントリはクラスタ・エイリアスをサーバ名に使用します。

フェールオーバの拡張には、Open Client 15.0、ESD #3 以降が必要です。Cluster Edition のクライアント・ライブラリには ESD #8 が含まれています。

Open Client は拡張フェールオーバに CS_PROP_EXTENDEDFAILOVER プロパティを使用します。詳細は、『Client-Library/C リファレンス・マニュアル』を参照してください。

HA フェールオーバとクラスタ内フェールオーバの違い

クライアント側からは、高可用性を有効にするということは、ネットワーク障害の間にアプリケーションがエラー・コード CS_RET_HAFAILOVER を受信し、クライアント・ライブラリが自動的にサーバに再接続することを意味します。高可用性では、アプリケーションは最初にオリジナルのサーバに再接続しようとし、接続が失敗した場合はセカンダリ・サーバ (プライマリ・サーバ用の interfaces ファイル・エントリに設定されている) を試みます。Cluster Edition とともに出荷される Open Client ライブラリを使用する場合、インスタンスはフェールオーバ・ターゲット・リストをクライアントに送信します。アプリケーションは、このリストを使用し、インスタンスが失敗したときにどこに接続するか決定します。

高可用性を使おうと、Cluster Edition を使おうと、フェールオーバはどの時点でも行われる可能性があります。

クラスタ環境での *isql* の使用

デフォルトで、isql を使用し、共有ディスク・クラスタ環境で Adaptive Server インスタンスに接続できます。しかし、共有ディスク・クラスタ内で Adaptive Server インスタンスに接続し、高可用性フェールオーバまたはクライアント用の高可用性フェールオーバの拡張をオンにするには、-Q オプションを指定して、isql を開始する必要があります。

注意 isql -Q は高可用性機能拡張を使用できますが、透過的ではありません。インスタンスが失敗したとき、isql はエラーを受信し、ユーザはバッチまたはトランザクションを再実行する必要があります。

『ユーティリティ・ガイド』を参照してください。

クラスタ環境でのリモート・プロシージャ・コールの使用

リモート・プロシージャ・コール (RPC) によって、クライアントは、リモート・サーバでストアド・プロシージャを開始できます。リモート・サーバは、ローカル・クライアントによって要求されたかのようにコンテキスト内でシステム・プロシージャを実行し、ネットワーク経由でオリジナルのサーバに結果を返送します。

サーバからサーバへの RPC に加え、Cluster Edition には 3 つの追加クラスの RPC があります。最初のクラスには RPC が含まれており、ここではリモート・サーバはクラスタです。2 番目のクラスには RPC が含まれており、ここではローカル、すなわち発信サーバはクラスタ・インスタンスです。3 番目のクラスには RPC が含まれており、ここではローカル・サーバとリモート・サーバの両方が同じクラスタのインスタンスです。

初期のバージョンの Adaptive Server では、cis rpc handling のデフォルト値は 0 でした。Cluster Edition では、デフォルト値は 1 であり、サイト・ハンドラの代わりにデフォルトの RPC 処理メカニズムとして、RPC 処理がコンポーネント統合サービス (CIS：Component Integration Service) を使用します。

クラスタ・インスタンスは、インスタンス名ではなくクラスタ名を使用して、自身をリモート・サーバに識別します。*@@servername* グローバル変数はクラスタの名前を返します。

リモート・サーバがクラスタである RPC

Cluster Edition は、クラスタを使用して単一システム・イメージをサポートします。リモート・サーバの RPC 要求を送信するサーバへの影響が最小になるからです。

RPC を送信しているサーバが cis rpc handling を使用している場合、クラスタはインバウンド要求を通常のクライアント接続と見なします。Workload Manager は、設定されたルート指定ルールに基づいて、適切なインスタンスおよび論理クラスタに RPC をルート指定しようとします。開始サーバがログイン・リダイレクトをサポートすることを示すかぎり、Workload Manager のルールはログイン・リダイレクトを指示することがあります。Adaptive Server 15.0 以降は、RPC 要求についてログイン・リダイレクトをサポートします。

開始サーバがサイト・ハンドラを使用する場合、クラスタの Workload Manager はバイパスされ、RPC は、接続を受け入れたインスタンスのシステム論理クラスタ内で実行されます。

ローカル・サーバがクラスタである RPC

Cluster Edition は、サイト・ハンドラ経由でアウトバウンド RPC をサポートせず、デフォルトで CIS RPC 処理を使用します。クラスタからのアウトバウンド RPC はクラスタの名前を使用して識別されるので、個別のインスタンスからではなく、クラスタからの RPC を受け入れるようリモート・サーバを設定します。

ローカル・サーバとリモート・サーバが同じクラスタ内のインスタンスである RPC

クラスタ内のインスタンスはクラスタ間通信用に RPC を使用する場合があります。クラスタが開始されるとき、Adaptive Server は sysservers テーブルに各クラスタ・インスタンスを自動的に追加し、クラスタ定義内になくなった sysservers からクラスタ・インスタンスを削除します。インスタンスが実行時に動的に追加および削除された場合にもこれは行われます。これらの sysservers エントリは、cluster instance ステータス・ビットを設定します。クラスタ内 RPC は特定のインスタンス向けなので、sysservers エントリは enable login redirection ステータス・ビットは設定しません。

sp_serveroption

sp_serveroption システム・プロシージャには、enable login redirection オプションと cluster instance オプションがあります。

ログイン・リダイレクトの有効化

sp_serveroption enable login redirection は、着信 RPC 要求がクラスタ内の別のインスタンスに送信されたかどうか判別します。構文は次のとおりです。

```
sp_serveroption instance_name, 'enable login redirection', [ true | false ]
```

各パラメータの意味は次のとおりです。

- *instance_name* − enable login redirection を設定しているインスタンスの名前です。このインスタンスは、sysservers に記入する必要があり、すべてのインスタンスについて自動的に記入されます。
- true − インスタンスが着信 RPC 要求をクラスタ内の別のインスタンスにリダイレクトできるという意味です。
- false − インスタンスが RPC 要求をリダイレクトできないという意味です。

デフォルトで、enable login redirection は有効です (sp_serveroption を実行するには sa_role が必要です)。

クラスタ・インスタンス

cluster instance は、インスタンス情報を格納する sysservers エントリを識別します。*instance_name* は追加中のインスタンスの名前です。

```
sp_serveroption instance_name, 'cluster instance', [ true | false ]
```

デフォルトで、*cluster_instance* はリモート・サーバごとに無効です (false に設定されます)。

Cluster Edition は、ローカル・クラスタ内のインスタンスの sysservers ローを自動的に管理します。cluster instance フラグを手動で設定またはクリアする必要はありません。

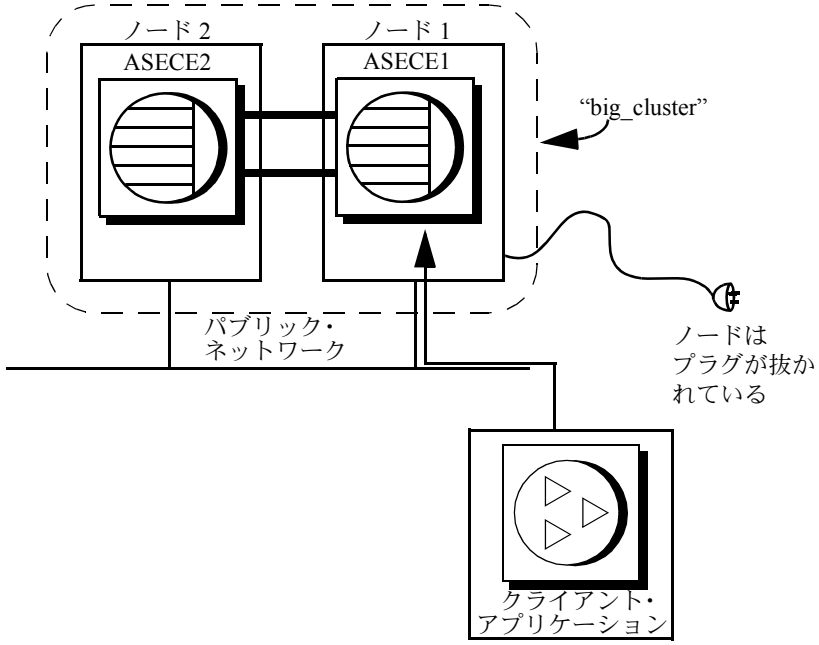
ノードの電源が停止したときのクライアントの再接続

ネットワーク・ケーブルがマシンから取り除かれた場合、またはクライアントが接続されているノードの電源が停止した場合、クライアント側ソケットは到達不能になります。クライアント・ソケットはサーバからの応答を待機するか、またはクラスタが送信オペレーションを発行するのを待機しますが、結果は得られません。

図 2-1 に示す状況では、クライアント・アプリケーションは “big_cluster” に接続されています。このクラスタは “Node1” と “Node2” で構成されており、“Node1” では “ASECE1” が実行されており、“Node2” では “ASECE2” が実行されています。クライアント・アプリケーションは、“Node1” で実行中のインスタンス “ASECE1” に接続されます。

“Node1” の電源が落とされると、クライアント・アプリケーションはノードからの連絡を待ちます。この状況を回避するには、クライアント・アプリケーションに、指定時間の後にノードがダウンすると想定させる以外の方法はありません。そのようにすると、クライアント・アプリケーションはクラスタの別のノードに接続します。

図 2-1: プラグが抜かれているノード



オペレーティング・システム・ネットワークが、クラッシュを検出し、クライアントを切断し、接続のリモート側からソケットにフェールオーバします。

クラスタがいつホストを失うか、またはインスタンスを実行中のノードからパブリック・ネットワークがいつ切断されるかの検出に必要な時間を短縮するために、以下のことができます。

- TCP keepalive を、クライアントを実行中のホストの適切な値に設定する。
- クライアント・アプリケーションの timeout 値を設定する。

TCP *keepalive* 値への短い値の設定

TCP keepalive パラメータは、最終的にクライアント・ソケットを失敗済みとマーク付けします。しかし、TCP keepalive 値のデフォルト値は長時間なので (システムによっては、2 時間に設定されている場合もあります)、3 時間以上たってからクライアント側ソケットがフェールオーバすることがあります。keepalive を小さい値 (数秒) に設定するのは、大組織には現実的でない場合があります。しかし keepalive を、HAFAILOVER 機能で動作する、各自のサイトに適切な期間に設定することはできます。

クライアント・マシンで TCP keepalive を設定します。ご使用のオペレーティング・システムに従って、適切な値はさまざまです。詳細については、クライアントのオペレーティング・システムのマニュアルを参照してください。

クライアントのタイムアウトをテストしている場合、表 2-1 の最初の 2 カラムのパラメータ値を数分に設定し、3 カラム目のパラメータ値を小さい数字に設定します。

表 2-1: TCP keepalive の設定

オペレーティング・システム	接続をプローブする前に待機する総時間パラメータ	パラメータ用のプローブ間の総時間	接続を削除する前に接続をプローブする最大総時間または試みのパラメータ
Solaris	N/A	tcp_keepalive_interval ミリ秒単位で測定	N/A
Linux	tcp_keepalive_time 秒単位で測定	tcp_keepalive_intvl 秒単位で測定	tcp_keepalive_probes 絶対数で測定
Windows XP	KeepAliveTime 秒単位で測定	KeepAliveInterval 秒単位で測定	TCPMaxDataRetransmissionsions 絶対数で測定
HP-UX	tcp_time_wait_interval ミリ秒単位で測定	tcp_keepalive_interval ミリ秒単位で測定	tcp_keepalive_kill ミリ秒単位で測定

クライアントのタイムアウト値の設定

Client-Library プログラム接続用に設定できるタイムアウト・プロパティには、次の 2 つがあります。

- CS_LOGIN_TIMEOUT － 到達不能なホストに接続するために、クライアントがどのくらい待機する必要があるか決定します。
- CS_TIMEOUT － コマンドが完了するのをクライアントがどのくらい待機するか決定します。

タイムアウト・イベントの設定に基づいて、クライアントは失敗するか、または別のノードにフェールオーバします。

Client-Library CS_TIMEOUT パラメータを設定し、どのくらい待機してからタイムアウトするか決定できるように設定できます。

ノードの電力が突然切れてしまった間、クライアントがフェールオーバできるように、CS_TIMEOUT パラメータと CS_LOGIN_TIMEOUT パラメータ、または isql -t パラメータと -l パラメータを設定する必要があります。

Client-Library パラメータの CS_TIMEOUT と CS_LOGIN_TIMEOUT の詳細については『Client-Library/C リファレンス・マニュアル』を参照してください。CS_HAFAILOVER の詳細については、「クライアント／サーバの対話」(26 ページ) を参照してください。

isql -t パラメータと -l パラメータの詳細については、『Adaptive Server Enterprise ユーティリティ・ガイド』を参照してください。

第 3 章

クラスタ環境でのセキュリティの使用

この章では、クラスタ環境におけるセキュリティの設定について説明します。

監査の詳細については、「監査の追加」(307 ページ) を参照してください。

クラスタ環境での暗号化カラムの使用

次の処理はできません。

- ローカル・テンポラリ・データベースでの暗号化キーの作成
- ローカル・テンポラリ・データベースでのシステム暗号化パスワードの設定

『暗号化カラム・ユーザーズ・ガイド』を参照してください。

クラスタ環境での SSL の使用

Cluster Edition では、ディレクトリ・サービス・エントリで指定されたサーバ名を、SSL ハンドシェイクを実行するために、SSL サーバ証明書が使用する共通名とは異なる名前にできます。これにより、SSL 証明書の共通名の完全修飾ドメイン名 (たとえば、*server1.bigcompany.com*) を使用でき、複数のサーバについて同じ証明書を使用できます。

interfaces ファイルに共通名を追加するには、次のフォーマットを使用します。

```
ase1
   master tcp ether host_name port_number ssl="CN='common_name'"
   query tcp ether host_name port_number ssl="CN='common_name'"
ase2
   master tcp ether host_name port_number ssl="CN='common_name'"
   query tcp ether host_name port_number ssl="CN='common_name'"
```

```
ase3
   master tcp ether host_name port_number ssl="CN='common_name'"
   query tcp ether host_name port_number ssl="CN='common_name'"
mycluster
   query tcp ether host_name port_number ssl="CN='common_name'"
   query tcp ether host_name port_number ssl="CN='common_name'"
   query tcp ether host_name port_number ssl="CN='common_name'"
```

ここで、*common_name* はクラスタノードの完全修飾ドメイン名です。*common_name* には空白を入れることができます。Interfaces ファイルに定義されたインスタンスは、同じ共通名を使用することも、使用しないこともあります。

注意 master データベースに追加できる SSL 証明書は 1 つのみです。クラスタ内の各インスタンスは同じディスクを共有するので、SSL サーバ証明書にすべて同じパスを使用します。すべてのインスタンスで同じ共通名を使用することをおすすめします。

たとえば、次はクラスタ mycluster 用の interfaces ファイル・エントリのサンプルです。

```
ase1
   master tcp ether blade1 19786 ssl="CN='ase1.big server 1.com'"
   query tcp ether blade1 19786 ssl="CN='ase1.big server 1.com'"
ase2
   master tcp ether blade2 19886 ssl="CN='ase1.big server 1.com'"
   query tcp ether blade2 19886 ssl="CN='ase1.big server 1.com'"
ase3
   master tcp ether blade3 19986 ssl="CN='ase1.big server 1.com'"
   query tcp ether blade3 19986 ssl="CN='ase1.big server 1.com'"
mycluster
   query tcp ether blade1 19786 ssl="CN='ase1.big server 1.com'"
   query tcp ether blade2 19886 ssl="CN='ase1.big server 1.com'"
   query tcp ether blade3 19986 ssl="CN='ase1.big server 1.com'"
```

sp_listener での共通名の指定

sp_listener にはパラメータ CN=*common_name* が含まれており、SSL 証明書の共通名を指定できます。構文は次のとおりです。

```
sp_listener 'command','[protocol:]machine_name:port_number:
"CN=common_name"', 'engine_number'
```

プロトコルとして sscltcp を指定する場合にのみ、CN=*common_name* を使用します。これが含まれている場合は、指定の *common_name* を使用して、SSL 証明書内の *common_name* を検証します。CN=*common_name* が含まれていない場合、Adaptive Server は *server_name* を使用して SSL 証明書の共通名に照らして検証します。CN=*common_name* は、証明書内の共通名エントリと一致する必要があります。証明書に完全修飾ドメイン名を含める場合、このドメイン名は CN=*common_name* と一致する必要があります。

属性名 “CN” は大文字と小文字を区別しませんが、共通名の属性値は大文字と小文字を区別します。たとえば、属性名は、“CN”、“Cn”、“cn” にできます。

たとえば、これは共通名 `ase1.big server 1.com` を指定します。

```
sp_listener 'start','ssltcp:blade1:17251:"CN=ase1.big server 1.com"','0'
```

sp_listener の詳細については、『ASE リファレンス・マニュアル』を参照してください。

ディレクトリ・サービスとしての LDAP の使用

Adaptive Server では、ディレクトリ・サービスを使用してネットワーク上でのクライアントと RPC との接続を確立しています。この章では、LDAP ディレクトリ・サービスを使用して接続を確立する方法について説明します。

LDAP (Lightweight Directory Access Protocol) は、ディレクトリ・サービスへの業界標準のアクセス方法です。ディレクトリ・サービスを使用すると、コンポーネントは LDAP サーバから情報を DN (識別名) で検索できます。LDAP サーバは、企業またはネットワーク上で使用されるサーバ、ユーザ、ソフトウェアの情報を格納したり管理したりします。

LDAP サーバは、Adaptive Server やクライアントを実行しているプラットフォームとは別のプラットフォームに配置できます。LDAP は、クライアントとサーバが交換するメッセージの通信プロトコルと内容を定義します。メッセージとは、読み取り、書き込み、クエリのクライアント要求やサーバの応答など、データ・フォーマット情報を含むオペレータです。

LDAP サーバは、次の情報を格納したり取り出したりします。

- Adaptive Server に関する情報 (IP アドレス、ポート番号、ネットワーク・プロトコルなど)
- セキュリティ・メカニズムとフィルタ
- 高可用性コンパニオン・サーバ名

LDAP サーバの設定時に、次のアクセス制限を指定できます。

- 匿名認証 － すべてのユーザがあらゆる情報にアクセスできます。
- ユーザ名とパスワードによる認証 － Adaptive Server は、UNIX プラットフォームのユーザ名とパスワードを使用します。
  - *$SYBASE/$SYBASE_OCS/config/libtcl.cfg* (32 ビット・プラットフォーム)
  - *$SYBASE/$SYBASE_OCS/config/libtcl64.cfg* (64 ビット・プラットフォーム)

ユーザ名とパスワードによる認証のプロパティによって、LDAP サーバとのセッション接続が確立され、終了します。

注意 ユーザ認証のために LDAP サーバに渡されるユーザ名とパスワードは、Adaptive Server へのアクセスに使用するユーザ名とパスワードとはまったく異なります。

LDAP サーバが *libtcl.cfg libtcl64.cfg* ファイルまたは *libtcl64.cfg* ファイル (まとめて *libtcl*.cfg* ファイル) に指定されているとき、サーバ情報は、*libtcl*.cfg* ファイルのディレクトリ・サービスの順序付けられたリストを使用して検索されます。情報がここで見つからない場合は、interfaces ファイルが検索されます。

Cluster Edition の場合、interfaces ファイルはクォーラム・ファイルで設定されることがあります。クォーラム・ファイルで interfaces ファイルが指定される場合、Cluster Edition は、*libtcl*.cfg* ファイルに指定されたディレクトリ・サービスを無視します。

1 台のサーバで複数のディレクトリ・サービスがサポートされている場合には、*libtcl*.cfg* で指定されている順序で検索されます。検索順は dataserver コマンドライン・オプションでは指定できません。「複数のディレクトリ・サービス」(48 ページ) を参照してください。

LDAP ディレクトリ・サービスと Sybase interfaces ファイルの違い

LDAP サーバで使用するために、LDAP ドライバでディレクトリ・サービスを実装します。LDAP ディレクトリは、以下の要素を実現するインフラストラクチャです。

- 従来の Sybase interfaces ファイルのネットワークベース版
- ユーザ、ソフトウェア、リソース、ネットワーク、ファイルなどの情報を階層構造で表した単一のビュー

表 3-1 に、Sybase interfaces ファイルと LDAP サーバの相違点を示します。

表 3-1: interfaces ファイルと LDAP ディレクトリ・サービスの違い

interfaces ファイル	ディレクトリ・サービス
プラットフォーム固有	プラットフォームに依存しない
Sybase インストール環境ごとに異なった構造	統一された階層構造
マスタ・エントリとクエリ・エントリが別々に存在する	各サーバの 1 つのエントリにクライアントとサーバの両方がアクセスできる
サーバのメタデータを格納できない	サーバのメタデータを格納できる

LDAP ディレクトリ・サービスは、Sybase interfaces ファイルより多くの属性をサポートしています。この属性には、サーバのバージョンやサーバのステータスなどを含めることができます。属性のリストについては、表 3-2 を参照してください。

注意　LDAP だけが、リエントラント・ライブラリでサポートされています。LDAP ディレクトリ・サービスを使用してサーバに接続する場合は、isql ではなく、isql_r を使用してください。

表 3-2 に、Sybase LDAP ディレクトリ・エントリのリストを示します。

表 3-2: Sybase LDAP ディレクトリの定義

属性名	値のタイプ	説明
ditbase	*interfaces* ファイルまたは *libtcl.cfg*	オブジェクト・ツリーの DIT ベース。*libtcl.cfg* ファイルが指定された場合は、*interfaces* ファイルは無視される。*libtcl.cfg* ファイルは、指定された接続用に ct_con_prop() で上書きできる。
dn	文字列	識別名。オブジェクトを識別するユニークな名前にする必要がある。
sybaseVersion	整数	サーバのバージョン番号。
sybaseServername	文字列	サーバの名前。
sybaseService	文字列	サービスの種類：Sybase Adaptive Server、Sybase SQL Server、または ASE。
sybaseStatus	整数	ステータス。1 = アクティブ、2 = 停止、3 = 失敗、4 = 不明。
sybaseAddress	文字列	各サーバのアドレス。次の項目を含む。 • プロトコル：TCP、NAMEPIPE、SPX DECNET (大文字と小文字を区別して入力する) • アドレス：そのプロトコル・タイプの有効なアドレス **注意**　dscp は、この属性をトランスポート・タイプとトランスポート・アドレスに分割します。 • フィルタ：なし、ssl、または ssl="CN=common_name"
sybaseSecurity (オプション)	文字列	セキュリティ OID (オブジェクト ID)。
sybaseRetryCount	整数	この属性は、CS_RETRY_COUNT にマッピングされる。CS_RETRY_COUNT は、ct_connect がサーバ名と対応するネットワーク・アドレスのシーケンスをリトライする回数を指定する。

属性名	値のタイプ	説明
sybaseRetryDelay	整数	この属性は、CS_LOOP_DELAY にマップされる。CS_LOOP_DELAY は、ct_connect がアドレスのすべてのシーケンスをリトライするまでの遅延時間を秒単位で指定する。
sybaseHAservername (オプション)	文字列	フェールオーバ保護用のセカンダリ・サーバ。

従来の interfaces ファイルは、TCP 接続のフェールオーバ・マシンで次のように表示されます。

```
looey
     master tcp ether huey 5000
     query tcp ether huey 5000
     hafailover secondary
```

次の例は、TCP 接続の LDAP エントリとフェールオーバ・マシンを示します。

```
dn: sybaseServername=foobar, dc=sybase,dc=com
objectClass: sybaseServer
sybaseVersion: 1500
sybaseServername: foobar
sybaseService: ASE
sybaseStatus: 4
sybaseAddress: TCP#1#foobar 5000
sybaseRetryCount: 12
sybaseRetryDelay: 30
sybaseHAServernam: secondary
```

LDAP ディレクトリ・サービスへのすべてのエントリは、エンティティと呼ばれます。各エンティティは DN (識別名) を持ち、それぞれの DN に基づいて階層ツリー構造内に格納されます。このツリーは、ディレクトリ情報ツリー (DIT) と呼ばれます。クライアント・アプリケーションは、DIT ベースを使用してエンティティの格納場所を指定します。「libtcl*.cfg ファイル」(45 ページ) を参照してください。

上記の例のエントリは、“foobar” という名前の Adaptive Server がポート番号 5000 の TCP 接続で受信していることを示します。`TCP` と `5000` の間にある 1 という値は、エントリが QUERY エントリと MASTER エントリの両方に使用されているという意味です。この値は 1 つの LDAP ディレクトリ・サービスについて必ず 1 になる必要があります。このエントリは、12 (回) のリトライ回数と 30 (秒) のリトライ遅延時間も指定しています。サーバが応答するアドレスをクライアントが検出すると、クライアントとサーバ間でログイン・ダイアログが開始されます。

UNIX での Sybase の LDAP ディレクトリ・スキーマのリストは、*$SYBASE/$SYBASE_OCS/config* にあります。

同じディレクトリに、*sybase-schema.conf* ファイルもあります。このファイルには、同じスキーマが格納されていますが、Netscape 固有の構文を使用します。

LDAP では各属性の複数のエントリをサポートしているので、各アドレス属性は単一サーバのアドレス (プロトコル、アクセス・タイプ、アドレスを含む) を持つ必要があります。表 3-2 の sybaseAddress を参照してください。

次の例は、異なる接続プロトコルの 2 つのアドレスで受信している Windows サーバの LDAP エントリを示します。

```
sybaseAddress = TCP#1#TOEJAM 4444
sybaseAddress = NAMEPIPE#1#¥pipe¥sybase¥query
```

注意 アドレス・フィールドの各エントリは、# 文字で区切ります。

このエントリは、dsedit を使用して編集できます。「ディレクトリ・サービスへのサーバの追加」(47 ページ) を参照してください。

すべての Sybase 製品でプラットフォームに関係なく互換性を保つため、プロトコルおよびアドレス属性フィールドはプラットフォームと製品に依存しないフォーマットにしてください。

libtcl.cfg* ファイル

libtcl.cfg* ファイルを使用して、LDAP サーバ名、ポート番号、DIT ベース、ユーザ名、パスワードを指定し、LDAP サーバへの接続を認証します。

libtcl.cfg* ファイルの目的は、設定情報 (Open Client/Open Server と Open Client/Open Server ベース・アプリケーション用のドライバ、ディレクトリ、セキュリティ・サービスなど) を提供することです。設定情報については、dsedit などの 32 ビット・ユーティリティは *libtcl.cfg* ファイルを検索し、64 ビット・アプリケーションは *libtcl64.cfg* ファイルを使用します。

32 ビットと 64 ビットのアプリケーション間の互換性を確保するには、*libtcl.cfg* ファイルと *libtcl64.cfg* ファイルの両方を編集する必要があります。

デフォルトの *libtcl.cfg* ファイルは、*$SYBASE/$SYBASE_OCS/config* にあります。

LDAP を *libtcl.cfg* ファイルで指定した場合は、interfaces ファイルは使用されません。

注意 起動時に -l オプションを使用する Open Client/Open Server アプリケーションは、*libtcl.cfg* ファイルを上書きして interfaces ファイルを使用します。

最も単純なフォームでは、*libtcl.cfg* ファイルは次のフォーマットになります。

```
[DIRECTORY]
ldap=libsybdldap.dll  ldapurl
```

ここでは、*ldapurl* は次のように定義されています。

```
ldap://host:port/ditbase
```

次の LDAP エントリは上記と同じ属性を使用していますが、匿名接続であり、LDAP サーバが読み込み専用アクセス可能な場合にだけ動作します。

```
ldap=libsybdldap.dll ldap://seashore/d=sybase,dc=com
```

libtcl.cfg ファイルでユーザ名とパスワードを LDAP URL への拡張機能として指定すると、接続時にパスワード認証が有効になります。

LDAP ディレクトリ・サービスの有効化

ディレクトリ・サービスを使用するには、次の手順に従います。

1 ベンダ提供のマニュアルに従って、LDAP サーバを設定します。

2 使用しているプラットフォームの UNIX ロード・ライブラリ・パス環境変数に、LDAP ライブラリのロケーションを追加します。

3 ディレクトリ・サービスを使用するように *libtcl.cfg* ファイルを設定します。

標準的な ASCII テキスト・エディタを使用して、次のように修正します。

- *libtcl.cfg* ファイルの *[DIRECTORY]* エントリにある LDAP URL 行の行頭から、コメント・マーカのセミコロン (;) を削除します。
- *[DIRECTORY]* エントリに LDAP URL を追加します。サポートされている LDAP URL 値については、表 3-3 を参照してください。

警告！ LDAP URL は、1 行で記述する必要があります。

```
file libtcl.cfg:
ldap=libsybdldap.so ldap://host:port/ditbase??scope??bindname=
username?password

file libtcl64.cfg
ldap=libsybdldap64.so
ldap://host:port/ditbase??scope??bindname=username?password
```

次に例を示します。

```
[DIRECTORY]
ldap=libsybdldap.so ldap://huey:11389/dc=sybase,dc=com??one??
bindname=cn=Manager,dc=sybase,dc=com?secret
```

“*one*” は、DIT ベースの 1 つ下のレベルのエントリを取得する検索のスコープを示します。

表 3-3 に、*ldapurl* 変数のキーワードの定義を示します。

表 3-3: ldapurl 変数

キーワード	説明	デフォルト
host (必須)	LDAP サーバを実行しているマシンのホスト名または IP アドレス	なし
port	LDAP サーバが受信しているポート番号	389
ditbase (必須)	デフォルトの DIT ベース	なし
username	認証するユーザの DN (識別名)	NULL (匿名認証)
password	認証されるユーザのパスワード	NULL (匿名認証)

4　必要なサード・パーティ・ライブラリが、適切な環境変数で指定されていることを確認します。Netscape LDAP SDK ライブラリは、*$SYBASE/$SYBASE_OCS/lib3p* または *lib3p64* にあります。UNIX ロード・ライブラリ・パス環境変数で、このディレクトリを指定する必要があります。

5　dscp または dsedit を使用して、LDAP サーバにエントリを追加します。「ディレクトリ・サービスへのサーバの追加」(47 ページ) を参照してください。

ディレクトリ・サービスへのサーバの追加

警告！ ほとんどの LDAP サーバには、ディレクトリ・エントリを追加するための ldapadd ユーティリティがありますが、代わりに dsedit を使用することをおすすめします。LDAP サーバには、一般のツールからは提供されない組み込みのセマンティック・チェックがあるからです。

各サーバ・エントリは、一連の属性で構成されています。サーバ・エントリを追加または修正するときは、サーバ属性についての情報を要求されます。属性のいくつかはデフォルトで提供されますが、その他はユーザが入力する必要があります。提供されたデフォルト値は、角カッコ “[ ]” で囲まれて表示されます。受け入れられる値については、表 3-2 (43 ページ) を参照してください。

❖ *dsedit* によるサーバ・エントリのディレクトリ・サービスへの追加

LDAP URL を *libtcl.cfg* ファイルに追加してから、LDAP サーバ・エントリの追加、削除、変更を行ってください。「libtcl*.cfg ファイル」(45 ページ) を参照してください。

dsedit を使用してディレクトリ・サービスにサーバを追加するには、次の手順に従います。

1　*SYBASE.csh* または *SYBASE.sh* ファイルを source コマンドで使用し、環境変数を設定します。

2　cd を実行して、*$SYBASE/$SYBASE_OCS/bin* に移動します。

3 dsedit を実行します。

4 サーバの一覧から [LDAP] を選択して、[OK] をクリックします。

5 [Add New Server Entry] をクリックします。

6 次のように入力します。

- サーバ名 – 必須。
- セキュリティ・メカニズム – オプション。高可用性フェールオーバ・サーバを使用している場合は、その名前を入力します。

7 [Add New Network Transport] をクリックして次の操作を実行します。

- トランスポート・タイプを選択します。
- ホスト名を入力します。
- ポート番号を入力します。
- (オプション) SSL フィルタ文字列を入力します。

8 [OK] を 2 回クリックして、dsedit を編集します。

サーバ・エントリを表示するには、Netscape で URL `http://host:port/ditbase??one` を入力します。

次に例を示します。

```
ldap://huey:11389/dc=sybase,dc=com??one
```

注意 Microsoft Internet Explorer では、LDAP URL は認識されません。

dscp の詳細については、Sybase の Webサイト (http://www.sybase.com/support/manuals) で、11.1.x Generic Collection の『Open Client/Server 設定ガイド』を参照してください。

複数のディレクトリ・サービス

LDAP サービスは、どのようなタイプでも (実際のサーバであっても、その他の LDAP サービスへのゲートウェイであっても)、LDAP サーバと呼ばれます。

高可用性フェールオーバ保護用に複数のディレクトリ・サービスを指定できます。リストにあるディレクトリ・サービスのすべてが LDAP サーバである必要はありません。

次に例を示します。

```
[DIRECTORY]
ldap=libsybdldap.so ldap://test:389/dc=sybase,dc=com
ldap=libsybdldap.so ldap://huey:11389/dc=sybase,dc=com
```

この例で、*test:389* への接続が失敗した場合、接続は *huey:11389* 上の LDAP サーバにフェールオーバします。DIT ベースのフォーマットはベンダによって異なります。

注意　詳細については、Sybase の Web サイト (http://www.sybase.com/support/manuals) で『Open Client Client-Library/C プログラマーズ・ガイド』と『Open Client Client-Library/C リファレンス・マニュアル』を参照してください。

パスワードの暗号化

libtcl.cfg ファイルのエントリは、人間が判読できるフォーマットになっています。Sybase では、基本的なパスワードの暗号化のために pwdcrypt ユーティリティを提供しています。pwdcrypt は、キーボード入力を行うと、パスワードと置換される暗号値を生成する単純なアルゴリズムです。pwdcrypt は *$SYBASE/$SYBASE_OCS/bin* にあります。

$SYBASE/$SYBASE_OCS ディレクトリから、次のように入力します。

```
bin/pwdcrypt
```

要求されたら、パスワードを 2 度入力します。

pwdcrypt が暗号化されたパスワードを生成します。次に例を示します。

```
0x01312a775ab9d5c71f99f05f7712d2cded2i8d0ae1ce78868d0e8669313d1bc4c706
```

標準的な ASCII テキスト・エディタを使用して、暗号化されたパスワードをコピーして *libtcl.cfg* ファイルに貼り付けます。暗号化の前に、ファイル・エントリが次のように表示されます。

```
ldap=libsybdldap.so
ldap://seashore/dc=sybase,dc=com??one??bindname=uid=Manager,dc=sybase,
dc=com?password
```

パスワードを、暗号化した文字列に置き換えます。

```
ldap=libsybdldap.so
ldap://seashore/dc=sybase,dc=com??one??bindname=uid=Manager,dc=sybase,dc=com?
0x01312a775ab9d5c71f99f05f7712d2cded2i8d0ae1ce78868d0e8669313d1bc4c706
```

警告！　パスワードが暗号化された場合でも、ファイル・システム・セキュリティを使用してパスワードを保護してください。

パフォーマンス

LDAP サーバを使用した場合のパフォーマンスは、interfaces ファイルを使用した場合よりも遅くなることがあります。これは、LDAP サーバがネットワークに接続してデータを取り出すのに時間がかかるためです。この接続は Adaptive Server を起動したときに行われるので、パフォーマンスに違いがある場合はログイン時にわかります。通常のシステム負荷では、パフォーマンスの低下を感じることはありません。高いシステム・ロードで接続が多い場合、特に短時間に繰り返し接続するような場合は、LDAP サーバを使用したときと従来の interfaces ファイルを使用したときとでは、全体のパフォーマンスに違いができることがあります。

interfaces ファイルから LDAP へのマイグレート

interfaces ファイルを使用する既存のサーバを、LDAP を使用するサーバに直接アップグレードすることはできません。Adaptive Server の以前のバージョンを Adaptive Server バージョン 15.0 にアップグレードするには、使用しているプラットフォームの『インストール・ガイド』を参照してください。

サーバをアップグレードしたら、サーバを設定して LDAP サービスを使用できます。

1 サーバを停止します。クラスタの開始と停止については、『Cluster ユーザーズ・ガイド』を参照してください。共有メモリ・サーバについては、Adaptive Server 15.0 の『設定ガイド』の「第 2 章 サーバの起動と停止」を参照してください。

2 *$SYBASE/$SYBASE_OCS/config/libtcl.cfg* または *libtcl64.cfg* ファイルを編集して、ディレクトリ・サービスを追加します。「LDAP ディレクトリ・サービスの有効化」(46 ページ) を参照してください。

3 dsedit または dscp を使用して、クラスタ・サーバのサーバ、クラスタ、インスタンスのエントリをディレクトリ・サービスに追加します。「ディレクトリ・サービスへのサーバの追加」(47 ページ) を参照してください。

4 qrmutil を使用して、クラスタ・クォーラム・ファイル内のインタフェース・ディレクトリ属性がクラスタ定義とインスタンス定義について空であることを確認します。たとえば、クォーラム・ファイル内の値を表示するには、次のように入力します。

```
$SYBASE/$SYBASE_ASE/bin/qrmutil --quorum-dev=path_to_your_quorum --
display=config
```

インスタンスについて、またはクラスタについて *interface_dir* 属性用にパスが定義されている場合、値をリセットしてください。パスの値でこの属性を指定すると、インスタンスは interfaces ファイルを使用し、*libtcl.cfg* ファイルと *libtcl64.cfg* ファイルの情報を上書きします。

たとえば、以下の qrmutil コマンドを使用して、*interface_dir* 属性の値をリセットします。*interface_dir* の値は 2 つの一重引用符です。つまり、空の文字列です。

```
$SYBASE/$SYBASE_ASE/bin/qrmutil --quorum-dev=path_to_your_quorum --
interface_dir=''

$SYBASE/$SYBASE_ASE/bin/qrmutil --quorum-dev=path_to_your_quorum --
instance=name_of_instance_to_reconfig --interface_dir=''
```

5　サーバまたはクラスタを再起動します。

共有ディスク・クラスタでの LDAP ディレクトリ・サービスの使用

Cluster Edition は、LDAP ディレクトリ・サービスを使用して、クラスタとインスタンスのエントリを指定します。クラスタのクォーラム・ファイル内で *interface_dir* 属性について空の文字列を指定してください。dataserver パラメータ -i *interfaces_path* は、interfaces ファイルへのパスを指定するときは使用しないでください。

interface_dir の値を指定しないと、Cluster Edition は、*$SYBASE/OCS-15_0/config/libtcl64.cfg* (64 ビット・サーバとクライアント) または *$SYBASE/OCS-15_0/config/libtcl.cfg* (32 ビット・サーバとクライアント) に定義されたディレクトリ・サービスの順序付けられたリストを使用します。サーバは、*libtcl64.cfg* に定義されたディレクトリ・サービスを検索した後、デフォルト・ロケーションの interfaces ファイルを検索します。

Open Client アプリケーションは、LDAP ディレクトリ・サービスを使用して、クラスタとインスタンス・サーバ・エントリを格納できます。たとえば、2 つのインスタンス (“ase1” と “ase2”) のある “mycluster” という名前のクラスタの場合、interfaces は次のようになります。

```
ase1

     master tcp ether blade1 10945
     query tcp ether blade1 10945
ase2

     master tcp ether blade2  10955
     query tcp ether blade2  10955
mycluster

     query tcp ether blade1 10945
     query tcp ether blade2 10955
```

dsedit または dscp を使用して、サーバ名 “ase”、“ase2”、“mycluster” について同等の LDAP ディレクトリ・サービス・エントリを LDAP ディレクトリ・サービスに追加する必要があります。「ディレクトリ・サービスへのサーバの追加」(47 ページ) を参照してください。dsedit と dscp の詳細については、『ユーティリティ・ガイド』を参照してください。

クライアントは、クラスタ名 (この例では、“mycluster”)、またはインスタンス特定のサーバ名 (“ase1” または “ase2”) を使用して、クラスタ内のインスタンスに接続できます。

SSL を使用するノンクラスタ Adaptive Server にクライアントが SSL を使用して接続する場合は、interfaces ファイルのポート番号の後に SSL フィルタが配置されます。ディレクトリ・サービスには、dsedit または手動編集を使用して追加できる共通名が含まれます。通常、1 つの共通名のある 1 つの SSL 証明書が、クラスタ全体について使用されます。インスタンスごとに 1 つではありません。「クラスタ環境での SSL の使用」(39 ページ) を参照してください。

次の例では、SSL フィルタをクラスタ “mycluster” の interfaces ファイル・エントリに追加します。

```
mycluster

        query tcp ether blade1 10945 ssl="cn=mycluster.domain.com"
        query tcp ether blade2 10955 ssl="cn=mycluster.domain.com"
```

LDAP ディレクトリ・サービスに追加されるエントリは、SSL フィルタ `ssl="cn=mycluster.domain.com"` を使用して共通名を指定してください。

たとえば、次の dscp セッションは、クラスタ “mycluster” について上記のエントリ例を追加します。

```
%  dscp

>> open ldap

ok

Session 1 ldap>> add mycluster
Service: [ASE]
Transport Type: [tcp]
Transport Address: blade1 10945 ssl="cn=mycluster.domain.com"
Transport Type: [tcp]
Transport Address: blade2 10955 ssl="cn=mycluster.domain.com"
Transport Type: [tcp]
Transport Address:
Security Mechanism [] :
HA Failoverserver:
Retry Count:
Retry Delay:
Added mycluster

Session 1 ldap>> read mycluster

DIT base for object:  dc=domain,dc=com
Distinguish name: sybaseServername=mycluster, dc=domain,dc=com
Server Entry Version: 15001
Server Name: mycluster
Server Service: ASE
```

```
Server Status: 4 (Unknown)
Server Address:
Transport Type: tcp
Transport Address: yellowstar 2521 ssl="cn=mycluster.domain.com"
Transport Type: tcp
Transport Address: yellowstar 2525 ssl="cn=mycluster.domain.com"

Session 1 ldap>> quit
```

第 4 章

クラスタ環境でのモニタリング・テーブルの使用

この章では、クラスタ Adaptive Server 用のモニタリング・テーブルについて、これらを設定、管理する方法について説明します。これらのテーブルのカラムやデータ型の詳細については、『リファレンス・マニュアル：テーブル』を参照してください。

クラスタ用モニタリング・テーブル

クラスタ環境では、モニタリング・テーブルは、クラスタ全体の結果を返すのではなく、インスタンスごとにレポートを出します。これによって、クラスタ全体のプロセスとクエリのアクティビティを監視して、複数のインスタンス上で開かれる可能性のあるオブジェクトの統計およびクラスタ内の各インスタンスのリソース使用状況をより良く理解できます。たとえば、あるテーブルを監視するクエリを実行する場合、このテーブルはクラスタ内の複数のインスタンスによって開かれたり、アクセスされたりしている可能性があるので、このテーブルの記述子 (および関連する統計) はインスタンス上のメモリ内にある可能性があります。クラスタの統計は累積されません。全インスタンスの統計結果は、各インスタンスから収集されたローのある、共用体化された結果として返されます。各インスタンスは、結果集合内で、InstanceID カラムのローによって識別されます。

システム・ビューの設定

system_view は、セッションに特定の設定であり、これを使用して、モニタリング・テーブル、sysprocesses、sp_who、その他のコマンドからの応答を問い合わせる監視データの範囲を制御できます。system_view を cluster に設定すると、モニタリング・テーブルについてのクエリは、クラスタ内のアクティブな全インスタンスからデータを返します。system_view を instance に設定すると、モニタリング・テーブルについてのクエリは、クライアントが接続されているインスタンス上でアクティブなプロセスまたはオブジェクトについてのみデータを返します。

set コマンドを使用して、セッションの範囲を設定します。

```
set system_view {instance | cluster | clear}
```

各パラメータの意味は次のとおりです。

- instance ローカル・インスタンスについての統計のみを返します。クラスタ間要求は、クラスタ内の他のインスタンスには送信されません。
- cluster クラスタ内の全インスタンスについての統計を返します。
- clear システム・ビューを設定済みデフォルトに返します。

次の例では、セッション設定を修正するので、モニタリング・テーブルについてのクエリは、クライアントが接続されているインスタンスについてのデータのみを返します。

```
set system_view instance
```

次の例では、セッション設定を修正するので、モニタリング・テーブルについてのクエリはクラスタ用のデータを返します。

```
set system_view cluster
```

次の例では、システム・ビュー現在の設定をクリアし、デフォルト設定のシステム・ビューを返します。

```
set system_view clear
```

モニタリング・テーブルに問い合わせるとき、またはモニタリング・テーブル RPC を呼び出すときに InstanceID を指定しないと、インスタンスは、現在の system_view 設定を使用します。

セッション・システム・ビューはホスト論理クラスタから継承されます。*@@system_view* グローバル変数を選択して、現在のシステム・ビューを決定します。

モニタリング・テーブルの設定

モニタリング・テーブルの設定パラメータを使用して、クラスタ全体またはインスタンスのみの動作を決定します。デフォルトで、モニタリング・テーブルの全設定値がクラスタ全体に適用されます。

メッセージ・パイプの管理

次のパラメータは、履歴モニタリング・テーブル用データの格納に使われるメモリを、クラスタとインスタンスがどのように管理するかを決定します。

- deadlock pipe max messages
- errorlog pipe max messages
- sql text pipe max messages
- plan text pipe max messages
- statement pipe max messages

これらのパラメータは、クラスタ用にグローバルに設定することも、インスタンスごとに個別に設定することもできます。これらのパラメータはパイプ用のメモリを割り付けます。インスタンスはメモリを動的にパイプに追加できますが、メモリを動的にパイプから削除することはできないので、パラメータのサイズを減少させる場合は、新しいパイプのサイズを有効にするためにインスタンスを再起動する必要があります。

以下は、パラメータのサイズを決定するいくつかのアルゴリズムです。

- 個別のインスタンスで、各パイプ設定に必要なメモリは次のとおりです。

  configuration_value X *number_of_engines*

- 各パイプ設定用にメモリをグローバルに設定する場合は次のとおりです。

  configuration_value X *number_of_engines* X *number_of_instances*

- インスタンスごとにパイプ構成の値を異なるように設定した場合、クラスタに必要なメモリの総量は次のとおりです。

  (*instance_1_value* X *number_of_engines*) + (*instance_2_value* X *number_of_engines*) +.. . + (*instance_n_value* X *number_of_engines*))

RPC 用の変更

InstanceID をパラメータとして指定せずに RPC を起動した場合、モニタリング・テーブルはシステム・ビュー設定を使用して、統計をレポートする方法を決定します。システム・ビュー設定を instance に設定している場合、すべてのデータ収集がローカルになります。システム・ビュー設定を cluster に設定している場合、モニタリング・テーブルはインスタンスと通信し、論理クラスタではなく、クラスタ内の全インスタンスを形成します。

Cluster Edition に固有のモニタリング・テーブル

モニタリング・テーブルの中には、クラスタ Adaptive Server とノンクラスタ Adaptive Server 両方についての情報を提供するものがあります。表 4-1 に、クラスタ Adaptive Server のみの情報を提供するモニタリング・テーブルの一覧を示します。

表 4-1: Cluster Edition のためのモニタリング・テーブル

monClusterCacheManager	monTempdbActivity
monPCM	monSysLoad
monDBRecovery	monLogicalCluster
monDBRecoveryLRTypes	monLogicalClusterinstance
monCMSFailover	monLogicalClusterRoute
monFailoverRecovery	monLogicalClusterAction
monCLMObjectActivity	monProcessMigration
monCIPC	monWorkloadProfile
monCIPCEndpoints	monWorkloadRaw
monCIPCLinks	monWorkloadPreview
monCIPCMesh	

全インスタンスについて同一の情報を返すモニタリング・テーブル

表 4-2 で、全インスタンスについて同一の情報を返すモニタリング・テーブルについて説明します。

表 4-2: 全インスタンスについての同一の情報を含むモニタリング・テーブル

テーブル名	説明
monMon	メタデータ・ビューはすべてのインスタンスについて同一です。
monTableColumns	メタデータ・ビューはすべてのインスタンスについて同一です。
monTableParameters	メタデータ・ビューはすべてのインスタンスについて同一です。
monTables	メタデータ・ビューはすべてのインスタンスについて同一です。
monWaitClassInfo	説明のリストはすべてのインスタンスについて同一です。
monWaitEventInfo	説明のリストはすべてのインスタンスについて同一です。

特定のインスタンス情報を返すモニタリング・テーブル

表 4-3 に、InstanceID カラムを含むモニタリング・テーブルの一覧を示します。

表 4-3: InstanceID カラムのあるモニタリング・テーブル

monCachePool	monDataCache
monCachedProcedures	monDeviceIO
monDeadLock	monErrorLog
monEngine	monIOQueue
monLicense	monLocks
monOpenDatabases	monNetworkIO
monOpenPartitionActivity	monOpenObjectActivity
monProcess	monProcedureCache
moProcessLookup	monProcessActivity
monProcessObject	monProcessNetIO
monProcessSQLText	monProcessProcedures
monProcessWaits	monProcessStatement
monResourceUsage	monProcessWorkerThread
monSysPlanText	monState
monSysStatement	monSysSQLText
monSysWorkerThread	monSysWaits
monCachedObject	

第 5 章

クラスタ環境での Backup Server の使用

この章では、クラスタでの Backup Server の使用に関して説明します。

ダンプ中にクラスタに参加するノード

従来のバージョンの Cluster Edition では、データベースまたはトランザクション・ログのダンプ中にノードがクラスタに参加またはクラスタから離脱することはできませんでした。ダンプ中に起動されたインスタンスはダンプが終了するまで待機してから起動を完了し、ダンプが完了するまで待機してシャットダウンしました。

Cluster Edition 15.0.3 以降では、ノードおよびノード上で実行中のインスタンスは、他のインスタンスが dump database または dump transaction を実行中に、クラスタに参加したりクラスタから離脱したりできます。

ノードがクラスタに参加したりクラスタから離脱したりできるのは、dump database のフェーズ 1 の間のみです。フェーズは次のように dump database コマンド中に表示されます。

```
Backup Server: 3.43.1.1: Dump phase number 1 completed.
Backup Server: 3.43.1.1: Dump phase number 2 completed.
Backup Server: 3.43.1.1: Dump phase number 3 completed.
```

dump database のフェーズ 2 またはフェーズ 3 でノードが参加または離脱しようとした場合は、dump database が終了するまで待機し参加します。

ノードは、通常のシャットダウン時にはデータベース・ダンプ中にクラスタから正常に離脱できます。

クラスタ用に複数の Backup Server を設定すれば、dump database または dump transaction 中にノードがクラスタに参加またはクラスタから離脱できるようにするために追加設定を行う必要はありません。

複数の Backup Server

15.0.3 より古いバージョンの Cluster Edition ではクラスタ用に 1 つの Backup Server を使用していましたが、これが dump コマンドと load コマンドのボトル・ネックになる可能性がありました。

Cluster Edition バージョン 15.0.3 以降では、クラスタは次の方式のいずれかで複数の Backup Server を使用できます。

- 専用方式 − 各インスタンスは特定の Backup Server に割り当てられます。
- ラウンドロビン方式 − dump または load コマンドを使用するとき、Cluster Edition が Backup Server の使用状況に合わせてインスタンスをグループ内の Backup Server に割り当てます。
- SYB_BACKUP という名前の 1 つの Backup Server

Cluster Edition はすべての Backup Server の情報を sysservers に格納します。任意のインスタンスから dump または load コマンドを発行するとき、Cluster Edition は sysservers で SYB_BACKUP という名前の Backup Server エントリを検索します。$DEDICATED または $ROUNDROBIN の srvnetname エントリを指す SYB_BACKUP エントリが検出されると、適切な Backup Server によって dump または load オペレーションが実行されます。

たとえば、“snack” クラスタが専用方式を使用するように設定され、“cupcake” という名前のインスタンスが含まれているとします。SYB_BACKUP エントリは “$dedicated” を指しているため、クラスタは cupcake_BS Backup Server を使用してダンプまたはロードを行います。

```
srvid    srvstatus    srvname         srvnetname    srvclass    srvsecmech
srvcoss           srvstatus2
-------  ---------    ------------   -----------   ----------   ------------
     ----------    ----------
      1          8      SYB_BACKUP    $dedicated            7           NULL
           NULL              2
      2          8         cupcake       cupcake            7           NULL
                1000          4
      3          8      cupcake_BS    cupcake_BS            7           NULL
                 0           2
      4          8          cookie        cookie            7           NULL
                1000          4
     5           8        cookie_BS      cookie_BS          7          NULL
                 0           2
```

snack クラスタがラウンドロビン方式を使用するように設定されている場合は、cupcake インスタンスは使用できる最初の Backup Server でダンプおよびロードを行います。

アップグレードしたときは、Cluster Edition はすべての Backup Server のエントリを sysservers に追加します。

クラスタからインスタンスを削除すると、Backup Server のエントリは sysservers から削除されます。

複数の Backup Server を使用するように Cluster Edition を設定する

sp_addserver を使用して、Backup Server を追加します。

```
sp_addserver instance_name_BS, NULL, instance_name
```

たとえば、cookie_BS エントリを sysservers に追加するには、次のように入力します。

```
sp_addserver cookie_BS, NULL, cookie
```

各 Backup Server に interfaces ファイルのエントリを挿入してから起動する必要があります。interfaces ファイルの編集の詳細については、『設定ガイド』の「第 5 章 ネットワークを介する通信の設定」を参照してください。

専用方式の使用

Cluster Edition を専用方式に設定するには、次のように入力します。

```
sp_addserver 'SYB_BACKUP', NULL, '$DEDICATED'
```

専用 Backup Server 方式に設定されているクラスタ内の各インスタンスは Backup Server を含んでいる必要があります。

isql を使用してインスタンス “cookie” に接続し、dump または load コマンドを発行すると、インスタンス cat は sysservers 内で '$DEDICATED' キーワードを指す SYB_BACKUP エントリを検出します。指定された Backup Server (‘cookie_BS’) が実行中でない場合は、Cluster Edition はこの Backup Server を自動的に再起動します。

ラウンドロビン方式の使用

Cluster Edition をラウンドロビン方式に設定するには、次のように入力します。

```
sp_addserver 'SYB_BACKUP', NULL, '$ROUNDROBIN'
```

Cluster Edition がラウンドロビン方式を使用するように設定すると、Cluster Edition は作業量が最も少ない Backup Server を dump または load コマンドを実行する Backup Server として選択します。

たとえば、isql からインスタンス Inst1 に接続し、このセッションから dump コマンドを発行した場合、インスタンス Inst1 は sysservers の SYB_BACKUP エントリをチェックし、キーワード $ROUNDROBIN を指していることがわかります。Inst1_BS が他のタスクを実行中でビジーである場合は、Cluster Edition は次の Backup Server である Inst2_BS に移動します。Inst2_BS でジョブが実行されておらず、使用できる場合は、この Backup Server はダンプを実行し、インデックスが増加して、この Backup Server がビジーで作業を実行できないことを示します。Inst2_BS がビジーである場合は、Cluster Edition は次の Backup Server に移動し、空いている Backup Server が見つかるまでこれを繰り返します。

一方、Inst2_BS に接続できなかった場合に、Backup Server を自動的に再起動するか、または使用可能な次の Backup Server に移動するかは、enable backupserver HA パラメータによって決まります。エラーが報告されるのは、設定されているすべての Backup Server が使用できない場合だけです。

enable backupserver HA を true に設定し、ラウンドロビン方式を使用するように Backup Server を設定した場合は、すべてのノードがクラスタ内のどの Backup Server でも起動できる必要があります。これは、コマンドを発行した最初のノードが失敗した場合に、他のノードがダンプまたはロード作業を実行できるようにするために必要です。すべてのノードがどの Backup Server でも起動できるようにするには、interfaces ファイルにすべての Backup Server のエントリを含める必要があります。

たとえば、Big_Cluster がラウンドロビン方式を使用しており、インスタンスとして Node_1、Node_2、および Node_3 があるとします。Node_1 が失敗した場合に、ラウンドロビン方式で代わりのノードとして Node_2 または Node_3 のどちらかのノードを選択するには、インスタンス Node_1 の interfaces ファイル・エントリに、Node_2 および Node_3 で実行している Backup Server のエントリが含まれている必要があります。

```
Node_1_BS
     master tcp ether Node_1 5004
     query tcp ether Node_1 5004
     master tcp ether Node_2 5006
     query tcp ether Node_2 5006
     master tcp ether Node_3 5008
     query tcp ether Node_3 5008

Node_2_BS
     master tcp ether Node_1 5004
     query tcp ether Node_1 5004
     master tcp ether Node_2 5006
     query tcp ether Node_2 5006
     master tcp ether Node_3 5008
     query tcp ether Node_3 5008

Node_3_BS
     master tcp ether Node_1 5004
     query tcp ether Node_1 5004
     master tcp ether Node_2 5006
     query tcp ether Node_2 5006
     master tcp ether Node_3 5008
     query tcp ether Node_3 5008
```

dump または load コマンドを発行するノードで、Backup Server を手動で起動することもできます。

Backup Server の起動と停止

sybcluster および Adaptive Server プラグインは、これらのユーティリティを使用して Backup Server を作成するときに Backup Server を自動的に起動します。sybcluster を使用した Backup Server の起動および停止については、『ユーティリティ・ガイド』を参照してください。Adaptive Server プラグインを使用した Backup Server の起動については、オンライン・ヘルプを参照してください。

startserver コマンドを使用して Backup Server を手動で起動します。startserver コマンドでは、Backup Server に runserver ファイルがあることが必要です。このファイルにより、MOUSE_BS Backup Server が起動されます。

```
startserver -f RUN_MOUSE_BS
```

startserver コマンドの詳細については、『**ユーティリティ・ガイド**』を参照してください。runserver ファイルについては、『設定ガイド』の「第 2 章 サーバの起動と停止」を参照してください。

Backup Server をシャットダウンするには、次のように shutdown *server_name* コマンドを使用します。

```
shutdown MOUSE_BS
```

SYB_BACKUP Backup Server を指定して、すべてのアクティブな Backup Server をシャットダウンします。

```
shutdown SYB_BACKUP
```

メディアへのバックアップ

Cluster Edition のオブジェクトのバックアップは、ノンクラスタ Adaptive Server でのオブジェクトのバックアップと同じです。詳細については、『システム管理ガイド 第 2 巻』を参照してください。

ストアド・プロシージャの変更

sp_addserver

sp_addserver は、Backup Server の方式を設定するためのパラメータとしてキーワード $DEDICATED および $ROUNDROBIN を追加します。「複数の Backup Server を使用するように Cluster Edition を設定する」(63 ページ) を参照してください。

sp_dumpoptimize と sp_helpserver

sysservers の SYB_BACKUP エントリがキーワード $DEDICATED または $ROUNDROBIN を指すとき、Cluster Edition はクラスタ内のすべてのアクティブな Backup Server に対してこれらのシステム・プロシージャを実行します。

- sp_dumpoptimize
- sp_helpserver

sp_volchanged

sp_volchanged を使用すると、特定の Backup Server 名を含めることができます。構文は次のとおりです。

sp_volchanged *session_id*, *devname*, *action*[, *fname* [, *vname*] [, *backup_server_name*]]

たとえば、ボリュームの変更要求を Backup Server Inst3_BS で行うには、次のようにします。

```
sp_volchanged 5, "db1.dmp", "proceed", "Inst3_BS"
```

$DEDICATED 方式または $ROUNDROBIN 方式を使用する Backup Server の名前が含まれていない場合は、sp_volchanged を使用するとエラーが発生します。

第 6 章

負荷の管理

この章では、負荷の管理方法および Cluster Edition にアクセスするアプリケーションをフェールオーバさせる方法について説明します。

Cluster Edition の Workload Manager は、ビジネス・アプリケーションのそれぞれが最も効率的に性能を発揮できるように、負荷管理とフェールオーバをカスタマイズできます。論理クラスタは、Workload Manager による作業環境の個別化が可能になるコンテナです。

論理クラスタは、物理共有ディスク・クラスタ内の、1 つまたは複数のインスタンスを抽象的に表したものです。各論理クラスタは、一連のインスタンスで実行され、フェールオーバ用の一連のインスタンスを保有できます。ルート指定ルールにより、着信接続はクライアントから指定されるアプリケーション、ユーザ・ログイン、またはサーバ・エイリアスに基づいて、特定の論理クラスタに送られます。論理クラスタをバインド接続に制限したり、認証された接続から論理クラスタへのアクセスを許可したりするルールもあります。

物理クラスタの上に論理クラスタを作成することで、同一のシステム上にあるアプリケーションを分割し、別々の負荷を処理するようにできます。負荷はアプリケーション・レベルで管理されます。つまり、着信接続、フェールオーバ・ポリシー、負荷分散方式、計画ダウン時間を、各アプリケーションがどのようにシステムを使用するかに応じて管理できます。

システム管理者は、この章で説明する、Sybase Central の Adaptive Server プラグインまたはコマンドライン・オプションを使用して負荷を管理します。システム管理者は次のことを実行します。

- 論理クラスタの設定と管理。これには、論理クラスタの作成と削除、クラスタに対するインスタンスの追加と削除、フェールオーバ・ルールの変更、クラスタまたはクラスタ内のインスタンスの起動と停止、ルート指定ルールの設定などが含まれます。
- 現在の相対的な負荷を決定するのにシステムが使用する、負荷プロファイルの選択または設定。
- クラスタ内のインスタンスと各インスタンスの負荷のモニタリング。

論理クラスタ・リソース

論理クラスタには、物理クラスタから次のリソースが割り当てられます。

- インスタンス － 論理クラスタの基本インスタンスです。つまり、論理クラスタが起動すると起動し、フェールバックによりリストアされます。
- フェールオーバ・リソース － 1 つ以上の基本インスタンスに障害が発生した場合に、論理クラスタが実行されるインスタンスの順序リストです。物理クラスタのインスタンスは、フェールオーバ・リソースとして使用できます。負荷管理機能により、リソースをグループ化し、フェールオーバの順序と優先度を指定できます。

システム論理クラスタ

共有ディスク・クラスタを作成すると、そのクラスタ用に Adaptive Server が自動的にシステム論理クラスタを作成します。システム論理クラスタは、デーモンに論理クラスタ表示を提供し、特定のタスクの管理を可能にします。これは、物理クラスタを表し、その中のすべてのインスタンスが含まれています。名前も物理クラスタと同じになります。チェックポイントやハウスキーピングなどのすべてのバックグラウンド・タスクがシステム論理クラスタ上で実行されます。システム論理クラスタには、特別なルールが適用されます。新しいシステム論理クラスタには、作成時にオープン・プロパティが付与されます。これにより、バインドされていないすべての接続がこのクラスタにルート指定されます。

システム管理者は、通常はシステム論理クラスタとはやりとりを行いません。ただし、次に挙げる内容は、システム論理クラスタに適用されます。

- ルート指定ルール。たとえば、システム管理者が使用するログインをシステム論理クラスタにルート指定できます。
- オープン・プロパティ
- システム・ビューの設定
- 負荷プロファイル
- ログイン分散モード

次に挙げる内容は、システム論理クラスタには適用されません。

- フェールオーバのリソースとルール
- 次のコマンド：
  - 論理クラスタのリソースの作成または削除
  - インスタンスのマイグレート、フェールオーバ、フェールバック
  - 論理クラスタまたはインスタンスの状態の変更
  - フェールオーバ・モード設定の変更
  - 起動モード設定の変更
  - 下方ルーティング・モードの設定

論理クラスタの設定

論理クラスタを設定するには、さまざまなオプションが用意されています。正常に機能する論理クラスタを設定する基本手順は、次のとおりです。

1　論理クラスタを作成します。

2　インスタンスを追加します。

3　ルート指定ルールを割り当てます。

4　論理クラスタを起動します。

次の例では、“SalesLC”、“HRLC”、“CatchallLC” という 3 つの論理クラスタを物理クラスタ “mycluster” 用に作成します。

論理クラスタの作成

注意 Adaptive Server プラグインを使用して論理クラスタを作成することもできます。「論理クラスタの追加」(241 ページ) を参照してください。

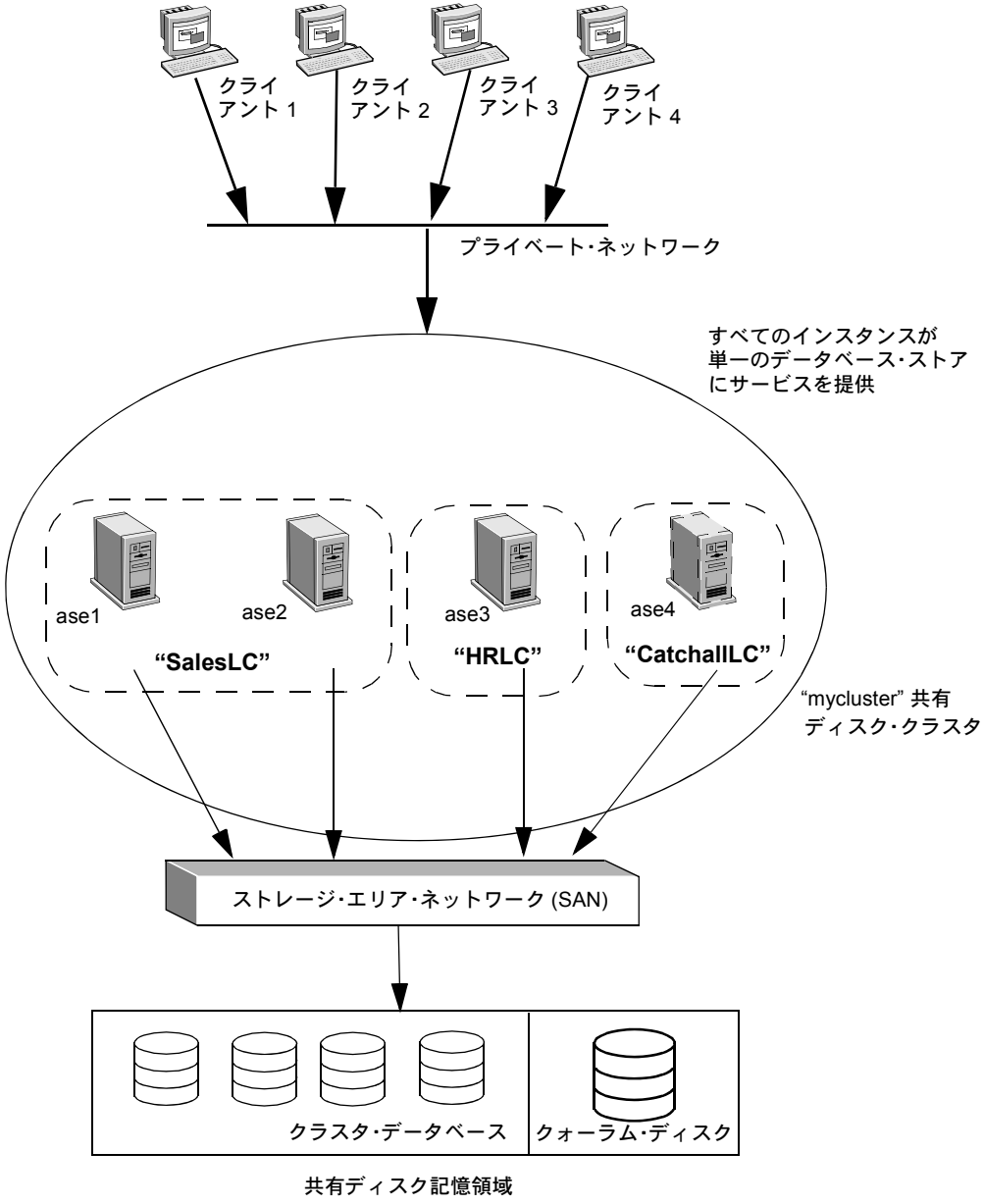
sp_cluster logical, "create" を使用して論理クラスタを作成します。

たとえば、“ase1”、“ase2”、“ase3”、“ase4” の 4 つのインスタンスを含む “mycluster” という物理クラスタがあるとします。次の 3 つの論理クラスタを作成します。

- “SalesLC” − 営業部門のアプリケーションとログインを処理する。
- “HRLC” − 人事部門のアプリケーションとログインを処理する。
- “CatchallLC” − 後で open 論理クラスタに使用する。

“SalesLC”、“HRLC”、“CatchallLC” を作成するには、次のように入力します。

```
sp_cluster logical, "create", SalesLC
sp_cluster logical, "create", HRLC
sp_cluster logical, "create", CatchallLC
```


クライアント 1
クライアント 2
クライアント 3
クライアント 4
プライベート・ネットワーク
すべてのインスタンスが単一のデータベース・ストアにサービスを提供
ase1
ase2
ase3
ase4
“SalesLC”
“HRLC”
“CatchallLC”
“mycluster” 共有ディスク・クラスタ
ストレージ・エリア・ネットワーク (SAN)
クラスタ・データベース
クォーラム・ディスク
共有ディスク記憶領域

論理クラスタへのインスタンスの追加

sp_cluster logical, "add" を使用して、クラスタにインスタンスを追加します。

必須の “SalesLC” に、次のように 2 つのインスタンスを追加します。

```
sp_cluster logical, "add", SalesLC, instance, ase1
sp_cluster logical, "add", SalesLC, instance, ase2
```

“HRLC” に、次のように単一のインスタンスを追加します。

```
sp_cluster logical, "add", HRLC, instance, ase3
```

“CatchallLC” に、次のように単一のインスタンスを追加します。

```
sp_cluster logical, "add", CatchallLC, instance, ase4
```

論理クラスタへのルートの追加

sp_cluster logical, "add" を使用して、クライアントをターゲットの論理クラスタにルート指定します。「ルート指定ルールの割り当て」(73 ページ) を参照してください。

たとえば、アプリケーション “field_sales” と “sales_reports” を “SalesLC” にルート指定するには、次のように入力します。

```
sp_cluster logical, "add", SalesLC, route, application,
"field_sales;sales_reports"
```

インターネット経由で複数の販売アプリケーションによって使用されるログイン名 “sales_web_user” をルート指定するには、次のように入力します。

```
sp_cluster logical, "add", SalesLC, route, login,
sales_web_user
```

人事アプリケーションを使用するすべてのクライアントを “HRLC” にルート指定するには、次のエイリアス・ルートを入力します。

```
sp_cluster logical, "add", HRLC, route, alias, HR_SERVER
```

注意 クライアントのディレクトリ・サービスに各サーバ・エイリアスを含め、クエリ・エントリで物理クラスタが受信するアドレスを指定するようにしてください。

たとえば、クライアントのディレクトリ・サービス用にエイリアスを設定するには、次のようにします。

```
ase1
     query tcp ether blade1 19786

ase2
     query tcp ether blade2 19786
```

```
ase3
     query tcp ether blade3 19786

ase4
     query tcp ether blade4 19786

mycluster
     query tcp ether blade1 19786
     query tcp ether blade2 19786
     query tcp ether blade3 19786
     query tcp ether blade4 19786

HR_SERVER
     query tcp ether blade1 19786
     query tcp ether blade2 19786
     query tcp ether blade3 19786
     query tcp ether blade4 19786
```

構文と使用法の詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。

論理クラスタの起動

論理クラスタを起動して、オンライン状態にするには、sp_cluster logical, “online” を使用します。

たとえば、“SalesLC” と “HRLC” を起動するには、次のように入力します。

```
sp_cluster logical, "online", SalesLC
sp_cluster logical, "online", HRLC
```

ルート指定ルールの割り当て

共有ディスク・クラスタへのクライアント接続は、それぞれが 1 つの論理クラスタに関連付けられています。この関連付けはルート指定ルールに基づいています。また、ルート指定ルールが存在しないという事実にも基づくことができます。その場合、open 論理クラスタへルート指定されます。

ターゲット論理クラスタがオフラインか、クライアントがログイン・リダイレクトをサポートしていないために、論理クラスタに接続をリダイレクトできない場合、接続はターゲット論理クラスタの down-routing モードに従って処理できます。down-routing ルールの指定については、「down-routing モード」(76 ページ)を参照してください。

ルート指定ルール

ルート指定ルールは、クライアントの着信接続を適切な論理クラスタにリダイレクトします。いったん接続が論理クラスタにルート指定されると、そのルートに入力された Adaptive Server タスクは論理クラスタ管理機能によって管理できます。

ルート指定ルールまたはバインドには 3 種類あります。各バインドでは、TDS ログイン・レコードの中の名前が使用されます。

バインド・ルートは、最高位から最低位へ優先順位によってリストされています。そのため、ログイン・ルートはアプリケーション・ルートよりも優先され、アプリケーション・ルートはエイリアス・ルートよりも優先されます。

- ログイン・ルート － Adaptive Server ログインと論理クラスタの関係を確立します。
- アプリケーション・ルート － アプリケーション名と論理クラスタの関係を確立します。
- エイリアス・ルート － サーバ名を論理クラスタに関連付けます。アプリケーションは、interfaces ファイルに含まれるサーバ・エイリアスから論理クラスタを選択できます。これらのエイリアスは、ユニークなサーバ名を使用します。

  エイリアス・エントリは、物理クラスタ内の任意の場所を指すことができます。Workload Manager は、ログイン・リダイレクト経由でそのエイリアスを正しいインスタンスに送信します。

論理クラスタ属性の設定

すべての論理クラスタには、一連の属性 (またはプロパティ) があります。これにより、論理クラスタの動作の異なる側面が制御されます。各属性にはデフォルト値があります。デフォルト値を使用することも、使用しているアプリケーション環境に最適になるように値を変更することもできます。

論理クラスタ属性の現在の設定を表示するには、sp_cluster logical, "show" を使用します。「論理クラスタに関する情報の表示」(84 ページ) を参照してください。

sp_cluster logical を使用して、次の属性を管理します。

- open － 特定のルート・プランを持たないクライアントがダイレクトされる論理クラスタを指定します。
- system view － sp_who または sp_lock のようなモニタリング・ツールや情報ツール、およびモニタリング・テーブルがクラスタ内の 1 つのインスタンスを記録するか、クラスタ全体を記録するかを指定します。

- start-up モード − 論理クラスタが自動または手動のどちらで起動されるかを指定します。
- down-routing モード − ルート指定ルールで使用される論理クラスタが利用できない場合、クライアント接続をどのようにルート指定するかを指定します。
- failover モード − フェールオーバ・インスタンスがいつ、どのようにオンラインになるかを指定します。
- fail-to-any − 任意のインスタンスをフェールオーバ・リソースに使用するか、特定のインスタンスのみをフェールオーバ・リソースに使用するかを指定します。
- load profile − 論理クラスタ内のインスタンス上の相対的な負荷を判断するための一連の加重測定基準を測定します。
- login distribution モード − 論理クラスタが複数のインスタンスに及ぶ場合、Adaptive Server にどのように接続を分散させるかを指定します。
- action release − 論理クラスタのアクション (online、offline、failover、および failback) の完了またはキャンセル後に、これらのアクションを手動または自動で解放してクリアします。
- gather モード − 事前定義されたアクション (online、add route、alter route、または drop route) のいずれかが発生したときに、接続のグループを別の論理クラスタに手動または自動で収集します。
- 役割 − 論理クラスタが “system” と “open” のどちらであるかを示し、どちらでもない場合は "none" を示します。

open 論理クラスタ

明示的なルート指定ルールによって論理クラスタにルート指定されていないすべての接続は、現在の open 論理クラスタにルート指定されます。新しくクラスタを作成すると、システム論理クラスタは自動的に open 論理クラスタに指定されます。open 属性を別の論理クラスタに再設定することもできます。ただし、各物理クラスタに存在できる open 論理クラスタは、1 つのみです。

新しい open 論理クラスタを指定するには、sp_cluster logical, "set" を使用します。たとえば、“CatchallLC” を open 論理クラスタに指定するには、次のように入力します。

```
sp_cluster logical, "set", CatchallLC, "open"
```

新しい論理クラスタ用に open 属性を再設定すると、Adaptive Server によって古い open 論理クラスタの open 属性が自動的にオフになります。

open プロパティを down-routing モードと共に使用して、1 つ以上のインスタンスを特定の論理クラスタ用に排他的に使用するように予約できます。

down-routing モード

ルート指定ルールは、クライアントの着信接続を適切な論理クラスタにリダイレクトします。「ルート指定ルールの割り当て」(73 ページ) を参照してください。ただし、ルート指定ルールでは、ターゲット論理クラスタがオフラインの場合、またはリダイレクトが必要なのに接続がリダイレクトをサポートしていない場合、接続がどのように処理されるかは指定されません。

注意 Client-Library のプロパティである CS_PROP_REDIRECT は、クライアント接続がログイン・リダイレクトをサポートするかを決定します。デフォルトでは、CS_PROP_REDIRECT の値は true で、クライアント接続はログイン・リダイレクトをサポートします。『Client-Library/C リファレンス・マニュアル』を参照してください。

---

down-routing モードを指定し、ルート指定ルールが適用されない場合に接続をリダイレクトできます。

また、この属性を使用し、特定の接続用にインスタンスを予約することもできます。「リソース予約」(77 ページ) を参照してください。

down-routing モードは、sp_cluster logical, "set" を使用して設定します。値は次のとおりです。

- system − ルート指定不可能な接続をシステム論理クラスタに送信します。system によって、システム論理クラスタがすべてのインスタンスについて必ずオンラインとなり、最高の可用性が確保されます。デフォルトの設定です。
- open − ルート指定不可能な接続を、open 論理クラスタに送信します。open 論理クラスタに接続を送信できない場合、または接続がリダイレクトをサポートしていない場合、Adaptive Server は open 論理クラスタの down-routing モードを適用します。
- disconnect − ルート指定不可能な接続を切断します。この設定をした場合、ターゲット論理クラスタ内のインスタンスによってサービスできないクライアントを切断することによって、リソースを保持できます。「リソース予約」(77 ページ) を参照してください。

たとえば、“SalesLC” に対して down-routing モードを open に設定するには、次のように入力します。

```
sp_cluster logical, "set", SalesLC, down_routing, "open"
```

この例では、“SalesLC” が利用不可能な場合、その論理クラスタにバインドされているクライアントは open 論理クラスタにルート指定されます。

リソース予約

open プロパティを disconnect の down-routing モードと共に使用して、1 つ以上のインスタンスを特定の論理クラスタ用に排他的に使用するように予約できます。たとえば、“HRLC” 論理クラスタを排他的に使用するためにインスタンス “ase3” を予約するには、次の手順に従います。

1　“ase3” を含まない論理クラスタに open プロパティを設定します。

2　open 論理クラスタの down-routing モードを disconnect に設定し、リダイレクトをサポートしないクライアントであってもそれにアクセスできないようにします。

“HRLC” を指定するルート指定ルールを持つ接続のみが “ase3” に接続できます。

system-view 属性

system-view 属性は、sp_who や sp_lock のようなシステムのストアド・プロシージャを使用する場合、またはモニタリング・テーブルやフェイク・テーブルを表示する場合に、クラスタがどのように表示されるかを制御します。Adaptive Server が現在のインスタンスまたは物理クラスタに関する情報を表示するように system_view 属性を設定できます。

たとえば、インスタンスにシステム・ビューを設定するには、次のように入力します。

```
sp_cluster logical, "set", SalesLC, system_view, instance
```

クラスタにシステム・ビューを設定するには、次のように入力します。

```
sp_cluster logical, "set", SalesLC, system_view, cluster
```

論理クラスタ・レベルでシステム・ビューを設定すると、論理クラスタ内の接続用のデフォルト値が提供されます『リファレンス・マニュアル：プロシージャ』を参照してください。

注意　system-view 属性は、管理ツールの 1 つです。その現在値は、論理クラスタ、データベース、またはデータベース・オブジェクトをアプリケーションが把握する方法には影響を与えません。

start-up モード

start-up モード属性は、クラスタが起動するときに論理クラスタを自動的に起動するか、管理者がそれを手動で起動するかを指定します。

- 自動モードでは、クラスタの再起動後、論理クラスタの最初の基本インスタンスがオンラインになって初めて、その論理クラスタが起動します。そのため、論理クラスタはフェールオーバ・インスタンスがオンラインになっても、自動的にはオンラインになりません。これはデフォルト値です。
- 手動モードでは、管理者がオンライン・コマンドを実行しないと、論理クラスタはオンラインになりません。

たとえば、“SalesLC” の start-up モードを手動に変更するには、次のように入力します。

```
sp_cluster logical, "set", SalesLC, startup, manual
```

failover モード

failover モードは、論理クラスタがフェールオーバ・インスタンス上でいつ、どのように実行されるかを指定します。

failover モードは、sp_cluster logical, "set" を使用して指定します。値は次のとおりです。

- instance − オンライン・インスタンス (基本インスタンスまたはフェールオーバ・インスタンスのいずれか) に障害が発生すると、フェールオーバ・インスタンスによって 1:1 ベースで置き換えられます。たとえば、論理クラスタ “SalesLC” が、インスタンス “ase1” と “ase2” 上で実行され、フェールオーバ・インスタンスとして “ase3” と “ase4” があり、failover モードが “instance” であるとします。ここで、“ase1” に障害が発生すると、“ase3” がオンラインとなり、クラスタは “ase2” と “ase3” (または、2 つのフェールオーバ・インスタンスの相対的な負荷によっては “ase4”) 上で実行されます。instance がデフォルト値です。
- group − すべての基本インスタンスに障害が発生した場合のみ、基本インスタンスが置き換えられ、すべてのフェールオーバ・インスタンスがオンラインになるように指定します。“SalesLC” の failover モードは “group” であるとします。“ase1” に障害が発生しても、クラスタは “ase2” 上で実行し続けます。オンラインになるフェールオーバ・インスタンスはありません。ただし、“ase1” と “ase2” の両方に障害が発生すると、クラスタは “ase3” と “ase4” 上で実行します。

また、複数のフェールオーバ・グループを指定できるので、最初のフェールオーバ・セット内のインスタンスに障害が発生しても、フェールオーバ・インスタンスのもう 1 つのセットをオンラインにすることができます。

たとえば、“SalesLC” の failover モードを “group” に設定するには、次のように入力します。

```
sp_cluster logical, "set", SalesLC, failover, "group"
```

fail_to_any 属性

fail_to_any 属性は、論理クラスタがクラスタ内のどのインスタンスにもフェールオーバできるか、指定したフェールオーバ・インスタンスのみにフェールオーバできるかを決定します。この属性は、指定したフェールオーバ・インスタンスをオンラインにできない場合にのみ、重要になります。

fail_to_any 属性は、sp_cluster logical, "set" を使用して設定します。値は次のとおりです。

- true – クラスタ内の他のインスタンスがオンラインで利用可能なかぎり、システムが他のインスタンスをフェールオーバ・インスタンスとして機能するように選択するように指定します。これはデフォルト値です。
- false – 指定したフェールオーバ・インスタンスのみ使用できるように指定します。

たとえば、“SalesLC” の fail_to_any 属性をオフにするには、次のように入力します。

```
sp_cluster logical, "set", SalesLC, fail_to_any, false
```

load profile 属性

Adaptive Server は、負荷プロファイルを使用して、論理クラスタ内の各インスタンスに負荷スコアを与えます。この負荷スコアは、Workload Manager が負荷をうまく分散するために接続を他のインスタンスにダイレクトするタイミングを決定します。Sybase が提供するテスト済みの負荷プロファイルの例を使用するか、独自のものを設定できます。詳細については、「負荷プロファイル」(97 ページ) を参照してください。

login distribution モード

login distribution モードでは、複数のインスタンスを持つ論理クラスタ内でどのように接続が分散されるかを指定できます。login distribution モードは、単一インスタンスの論理クラスタには影響を与えません。

値は次のとおりです。

- affinity – ターゲット論理クラスタがインスタンス上で実行中であるかぎり、接続を受け入れているそのクラスタがそのまま保持されるように指定します。
  - 負荷プロファイルで負荷スレッショルドを指定している場合で、インスタンス上の負荷が高すぎる場合、Workload Manager は、論理クラスタ内のより負荷の低いインスタンスに接続をリダイレクトします。
  - ターゲット論理クラスタがインスタンス上で実行中でない場合は、Workload Manager は論理クラスタ内の最も負荷の低いインスタンスに接続をリダイレクトします。

- round-robin – 論理クラスタをホストしているインスタンス間で着信接続がラウンドロビン方式で分散されるように指定します。たとえば、“SalesLC” が “ase1” と “ase2” で実行中の場合、Workload Manager は最初の接続を “ase1” に送信し、次の接続を “ase2” に接続するといった動作になります。負荷スコアはアルゴリズムに組み込まれません。

注意 Cluster Edition では、sa_role を使用する接続に対しては負荷ベースのリダイレクト (affinity または round-robin) が実行されません。ただし、sa_role の接続は、別のインスタンス上で実行されている論理クラスタにそれらがルート指定される場合は、リダイレクトされます。

Sybase による推奨

短く頻繁な接続を使用するトランザクション指向のアプリケーションには affinity モードを使用することをおすすめします。アプリケーション・サーバが Adaptive Server への接続の持続的なプールを確立するような、大部分が読み込みであるアプリケーションには round-robin モードを使用することをおすすめします。

アクション解放

アクション解放では、すべての論理クラスタのアクション (online、offline、failover、および failback) の完了またはキャンセル後に、これらのアクションを手動または自動でクリアします。論理クラスタのアクションの自動クリアを有効または無効にする構文は、次のとおりです。

```
sp_cluster logical, "set", "lc_name", "action_release",
    {"automatic" | "manual"}
```

論理クラスタのすべてのアクションを手動でクリアする構文は、次のとおりです。

```
sp_cluster logical, "action", "lc_name", "release", "all"
```

gather モード

gather モードを使用すると、システム上で条件を満たすすべての接続、つまり open 論理クラスタに接続する接続を収集して、定義済みのルート指定ルールを使用して他の論理クラスタに移動できます。gather モードでは、online、add route、alter route、drop route などの事前定義されたアクションが発生したときに自動的に接続を移動するか、または手動で接続を収集できます。Adaptive Server は、影響を受ける論理クラスタのルート指定ルールに一致するすべての接続を検索し、それらを指定した論理クラスタにマイグレートします。

手動で接続を収集し、指定した論理クラスタに移動する構文は、次のとおりです。

```
sp_cluster logical, "gather", "lc_name"
```

接続の自動収集を有効または無効にする構文は、次のとおりです。

```
sp_cluster logical, "set", "lc_name", "gather", {"automatic" | "manual" }
```

「接続のマイグレート」(87 ページ) を参照してください。

Roles

論理クラスタの役割の値には、“system”、“open”、および “none” があります。

- System － 指定された論理クラスタがシステム論理クラスタであることを示します。「システム論理クラスタ」(68 ページ) を参照してください。
- Open － 指定した論理クラスタが特定のルート・プランを持たない場合に、クライアントがダイレクトされる論理クラスタであることを示します。論理クラスタを open 論理クラスタにするには、「open 論理クラスタ」(75 ページ) を参照してください。
- None －論理クラスタが、システム論理クラスタでも open 論理クラスタでもないことを示します。

論理クラスタの現在の役割を表示する構文は、次のとおりです。

```
sp_logical cluster, "show", "lc_name"
```

フェールオーバの設定

物理クラスタのインスタンスは、フェールオーバ・リソースとして使用できます。論理クラスタのフェールオーバ・ルールは、インフラストラクチャ、ロック・リマスタリング、またはリカバリには影響を与えません。論理クラスタのフェールオーバは、次を指定することで設定できます。

- フェールオーバ・リソース － フェールオーバが行われる特定のインスタンスまたはインスタンス・グループ。
- failover モード － フェールオーバが論理クラスタの個別のメンバに対して行われるか、クラスタ全体に対してのみ行われるかを指定します。「failover モード」(78 ページ) を参照してください。
- fail_to_any 属性 － フェールオーバが指定したフェールオーバ・リソースに対してのみ実行されるか、それとも指定したフェールオーバ・リソースが利用不可能な場合に他のインスタンスに対して実行されるかを指定します。「fail_to_any 属性」(79 ページ) を参照してください。

論理クラスタを作成すると、fail_to_any 属性のデフォルト設定により、基本インスタンスに障害が発生すると、即時に別のインスタンスで置き換えられるようになります。これは、多くのサイトで妥当な、最も単純なフェールオーバ方式です。

フェールオーバ・リソースをさらに詳細に制御する必要のあるサイトでは、デフォルト設定を変更し、フェールオーバが特定のインスタンスまたはインスタンス・グループに対して実行されるようにできます。

論理クラスタごとに、最大 31 のフェールオーバ・グループを作成できます。フェールオーバ・インスタンスをグループ化することで、特定のフェールオーバ・グループを優先的に扱うことができます。たとえば、グループ 2 に含まれるインスタンスより以前に、グループ 1 に含まれるインスタンスが考慮されるようにできます。Workload Manager によって、グループ内のフェールオーバ・インスタンスが負荷に応じて選択されます。負荷の低いインスタンスが最初に選択されます。

インスタンスがメンバになることができるフェールオーバ・グループは、論理クラスタごとに 1 つのみです。このため、インタンス “ase4” が “SalesLC” のフェールオーバ・グループ 1 に属する場合、“SalesLC” のフェールオーバ・グループ 2 には属することはできません。ただし、“ase4” が “SalesLC” のフェールオーバ・グループ 1、“HRLC” のフェールオーバ・グループ 2、“CatchallLC” の基本インスタンスに同時に属することはできます。

Workload Manager がフェールオーバ・インスタンスを有効化する場合、まずはグループ 1、次にグループ 2 というように、フェールオーバの条件が満たされるまで検索されます。設定されているフェールオーバ・リソースを有効化できない場合、Workload Manager は、fail_to_any パラメータ設定をチェックします。fail_to_any が true の場合、Workload Manager は利用可能な任意のインスタンスを使用して、フェールオーバを完了しようとします。fail_to_any が false の場合、フェールオーバは行われません。

フェールオーバ・リソースの追加

フェールオーバ・リソースを論理クラスタに追加するには、sp_cluster logical, "add" を使用します。フェールオーバ・リソースは、システム論理クラスタには追加できません。

sp_cluster logical, "add" を使用してフェールオーバ・リソースを追加するたびに、Adaptive Server は 1 つ以上のフェールオーバ・グループを作成します。

1 つ以上のフェールオーバ・インスタンスをセミコロンで区切って追加すると、Adaptive Server はすべてのインスタンスを単一のグループに配置します。

たとえば、“ase3” をフェールオーバ・グループとして “SalesLC” に追加する場合は、次のように入力します。

```
sp_cluster logical, "add", SalesLC, failover, ase3
```

フェールオーバ・インスタンスは、既存のフェールオーバ・グループにも追加できます。たとえば、“ase3” がフェールオーバ・グループ 1 のメンバであるとします。“ase4” をフェールオーバ・グループ 1 に追加するには、次のように入力します。

```
sp_cluster logical, "add", SalesLC, failover, ase4, "1"
```

フェールオーバ・グループ ID などのフェールオーバ・リソース情報を表示するには、sp_cluster logical, “show” を使用します。

論理クラスタの管理

この項では、論理クラスタの管理方法について説明します。

ユーザ・タスクと論理クラスタ

各 Adaptive Server タスク (SPID) は、1 つの論理クラスタ内で実行されます。sysprocesses 内の lcid カラムは、一定のタスクをホストしている論理クラスタの ID です。この ID を lc_name() 組み込み関数に渡して、対応する論理クラスタの名前を判断できます。

個別のタスクが lc_name() 組み込み関数を実行して、現在の論理クラスタを判断することがあります。

Workload Manager スレッドの管理

Workload Manager スレッドとは、各インスタンス上で実行されるシステムサービス・スレッドです。インスタンスが開始すると、自動的に Workload Manager スレッドが生成されます。このスレッドはほとんどの場合スリープ状態ですが、論理クラスタのアクションの処理、作業負荷測定基準の収集、各インスタンス上の負荷の計算、すべてのインスタンスへの負荷情報の送信、その他の管理タスクを実行するために、定期的にウェイクアップします。

sp_who とプロセスをモニタリングするためのその他の Adaptive Server 機能を使用し、sysprocesses と monProcesses にクエリすることで、Workload Manager に関する情報を表示できます。

Workload Manager が使用できる最大メモリ量を表すデフォルト値を変更することもできます (「Workload Manager のメモリ要件の設定」(83 ページ) を参照)。それ以外では、Workload Manager のメンテナンスは必要ありません。

Workload Manager のメモリ要件の設定

Workload Manager が使用できる最大メモリ量を指定するには、“workload manager cache size” 設定パラメータを変更します。Workload Manager が使用するすべてのメモリは、workload manager cache size によって指定されるメモリ・プールに含まれます。

接続のマイグレーションは、このプールからメモリを使用します。設定されている論理クラスタ、ルート、負荷プロファイルはそれぞれ、このプールのメモリを使用します。failover や failback などのコマンドによるアクションは、このプールからメモリを使用し、アクションが解放されるまで使用し続けます。

次のガイドラインに基づいて、メモリ使用量を予想します。

- 現在マイグレート中の接続ごとに 4 メモリ・ページ
- 3 つの論理クラスタ用に 1 ページ
- 2 つの負荷プロファイル用に 1 ページ
- 30 ルート用に 1 ページ
- 12 アクション用に 1 ページ

sp_configure を使用して、メモリ・プールの最大値を 2K ページ単位で設定します。workload manager cache size の値を 2K ページで 100 設定するには、次のように入力します。

```
sp_configure "workload manager cache size", 100
```

workload_manager_size は動的に設定されるので、サーバを再起動する必要はありません。デフォルト値は 80 (160KB) です。

ほとんどのインストール環境では、デフォルト値で十分です。多くの同時接続を使用する論理クラスタをマイグレートすることが予想される場合は、メモリ・プールのサイズを増やす必要がある場合があります。

論理クラスタに関する情報の表示

論理クラスタに関する情報を表示するには、次のことを実行できます。

- 組み込み関数 lc_name( )、lc_id( )、instance_name( )、および instance_id( ) を使用する。
- グローバル変数 *@@clustername*、*@@instancename*、および *@@instanceid* を使用する。
- モニタリング・テーブルにクエリする。
- sp_cluster logical, show を使用する。

モニタリング・テーブルへのクエリ

次のモニタリング・テーブルは、論理クラスタ、負荷、負荷プロファイルに関する情報を提供します。

- monLogicalCluster − システムの論理クラスタの要約情報を提供。
- monLogicalClusterInstance − システムの論理クラスタ内の各インスタンスに関する情報を提供。
- monLogicalClusterRoute − 設定済みルートの情報を提供。
- monLogicalClusterAction − システムの論理クラスタ内のアクションに関する情報を提供。

- monWorkload − 負荷スコアごとに各インスタンスの負荷プロファイルを提供。
- monWorkloadProfile − 各負荷プロファイルの情報を提供。

Transact-SQL コマンドを使用して、これらのテーブルから情報をクエリできます。各モニタリング・テーブルの完全な説明については、『リファレンス・マニュアル：テーブル』を参照してください。

sp_cluster logical, "show" の使用

次の目的で sp_cluster logical, "show" を使用できます。

- 特定の論理クラスタまたはすべての論理クラスタに関する概要情報を表示する。たとえば、“SalesLC” の情報を表示するには、次のように入力します。

  ```
  sp_cluster logical, "show", SalesLC
  ```

  すべての論理クラスタに関する情報を表示するには、次のように入力します。

  ```
  sp_cluster logical, "show"
  ```

- アクションについての情報を表示する。たとえば、完了したすべてのアクションの情報を表示するには、次のように入力します。

  ```
  sp_cluster logical, "show", NULL, action, complete
  ```

  “SalesLC” に関して、キャンセルされたアクションの情報を表示するには、次のように入力します。

  ```
  sp_cluster logical, "show", SalesLC, action, cancelled
  ```

  “SalesLC” に関して、アクティブなアクションの情報を表示するには、次のように入力します。

  ```
  sp_cluster logical, "show", SalesLC, action, active
  ```

- 設定済みルートの情報を表示する。特定のアプリケーション、ログイン、エイリアス、またはこれらの組み合わせについてクエリできます。

  “SalesLC” への “sales_web_user” ログイン・ルートの情報を表示するには、次のように入力します。

  ```
  sp_cluster logical, "show", SalesLC, route, login,
  sales_web_user
  ```

sp_cluster logical, "show" の構文と使用法の詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。

論理クラスタの作成と削除

- 論理クラスタを作成するには、sp_cluster logical, "create" を使用します。たとえば、“FinanceLC” を作成するには、次のように入力します。

  ```
  sp_cluster logical, "create", FinanceLC
  ```

- 論理クラスタを削除するには、sp_cluster logical, "drop" を使用します。論理クラスタは、削除する前にオフラインまたは非アクティブにしてください。sp_cluster logical, "drop" によって、クラスタとクラスタに関連するすべてのルート、リソース、属性が削除されます。

  たとえば、“FinanceLC” を削除するには、次のように入力します。

  ```
  sp_cluster logical, "drop", FinanceLC, cluster
  ```

論理クラスタへのリソースの追加

論理クラスタにリソースを追加するには、sp_cluster logical, "add" を使用します。次のリソースを追加できます。

- 基本インスタンス – 「論理クラスタへのインスタンスの追加」(72 ページ) の例を参照。
- フェールオーバ・インスタンス – ÅuÉtÉFÅ[ÉãÉIÅ[ÉoÅEÉäÉ\Å[ÉXÇÃí«â¡Åv(82 ÉyÅ[ÉW) の例を参照。

論理クラスタからのリソースの削除

論理クラスタから 1 つ以上のリソースを削除するには、sp_cluster logical, "drop" を使用します。基本インスタンスまたはフェールオーバ・インスタンスを削除するには、まずそれらをオフラインにしてください。

次のリソースを削除できます。

- 基本インスタンス – たとえば、インスタンス “SalesLC” を “ase2” から削除するには、次のように入力します。

  ```
  sp_cluster logical, "drop", SalesLC, instance, ase2
  ```

- フェールオーバ・インスタンス – たとえば、インスタンス “ase3” および “ase4” を “SalesLC” から削除するには、次のように入力します。

  ```
  sp_cluster logical, "drop", SalesLC, failover, "ase3;ase4"
  ```

ルートの追加、移動、削除

- ルートを追加するには、sp_cluster logical, "add" を使用します。たとえば、ログイン “accounting” および “projects” を “SalesLC” に追加するには、次のように入力します。

```
sp_cluster logical, "add", SalesLC, route, login,
"accounting;projects"
```

- ある論理クラスタから別の論理クラスタにルートを移すには、sp_cluster logical, “alter” を使用します。たとえば、エイリアス “SalesLC” を使用して “My_LC” へのルートを作成し、そのルートを “My_LC” から “Your_LC” へ移すには、次のように入力します。

```
sp_cluster logical, "add", My_LC, route, alias,
    SalesLC
sp_cluster logical, "alter", Your_LC, route, alias,
    SalesLC
```

- ルートを削除するには、sp_cluster logical, "drop" を使用します。たとえば、“SalesLC” からログイン “projects” を削除するには、次のように入力します。

```
sp_cluster logical, "drop", SalesLC, route, login,
projects
```

接続のマイグレート

sp_cluster connection “migrate” を使用すると、次のことを実行できます。

- 接続 (または別のタスク) が実行中の論理クラスタまたはインスタンスをマイグレートする。
- アプリケーションまたはログインを、それがルート指定されていない論理クラスタまたはインスタンスへマイグレートする。

たとえば、次の例は、spid が 73 の接続を SalesLC 論理クラスタにマイグレートします。

```
sp_cluster connection, "migrate", SalesLC, NULL, "73"
```

『リファレンス・マニュアル：プロシージャ』を参照してください。

sp_cluster logical を使用して、事前定義されたイベントが発生した場合に、他の論理クラスタに手動または自動でマイグレートするように設定するか、または他の論理クラスタに対して接続のグループを収集することができます。システムまたは論理クラスタ上の完全に条件を満たす接続すべてを、定義済みのルート指定ルールを使用して指定した論理クラスタに「収集」できます。Cluster Edition は、この論理クラスタのルート指定ルールに一致するすべての接続を検索し、それらを指定した論理クラスタにマイグレートします。

構文は次のとおりです。

```
sp_cluster logical, 'gather', lc_name
sp_cluster logical 'set', lc_name, 'gather', 'automatic | manual'
```

次の例は、定義済みの接続をすべて SalesLC 論理クラスタに収集します。

```
sp_cluster logical, 'gather', SalesLC
```

次の例は、SalesLC 論理クラスタ用に収集を「手動」に設定します。

```
sp_cluster logical 'set', SalesLC, 'gather' 'manual'
```

『リファレンス・マニュアル：プロシージャ』を参照してください。

フェールオーバ、フェールバック、計画ダウン時間の管理

アクションが次のうちのいずれかの場合、sp_cluster logical, "action" を使用して、論理クラスタの状態を手動で変更できます。

- failover
- failback
- online
- offline
- deactivate

クラスタとインスタンスの状態

論理クラスタとクラスタ内の各インスタンスは、異なる状態になることがあります。

- 論理クラスタは、たとえば、クラスタがオフラインかオンラインかを判断するような全体のグローバルな状態を保持しています。
- さらに、論理クラスタは、それが特定のインスタンスに関して把握した状態を表す、インスタンスの状態も保持しています。たとえば、オンラインの論理クラスタは、その基本インスタンス上ではオンラインで、そのフェールオーバ・インスタンス上ではオフラインである場合があります。実際は動作中のインスタンス上で論理クラスタがオフラインの可能性があるので、この状態は実際の Adaptive Server の状態とは無関係です。

ユーザが認識できる状態には 5 種類あります。これらの状態は、インスタンスの状態に加え、論理クラスタの状態にも当てはまります。表 6-1 は、グローバル・レベルおよびインスタンス・レベルでの各状態を示します。

表 6-1: 論理クラスタの状態

状態	グローバル・レベル	インスタンス・レベル
online	論理クラスタがオンラインで、1 つ以上のインスタンス上で実行中である。	オンラインの論理クラスタが、現在のインスタンス上の接続を受け入れ、管理している。
offline	論理クラスタがいずれのインスタンス上でも実行していない。	論理クラスタが現在のインスタンス上で実行していない。そのため、インスタンスでは接続を受け入れたり、リソースを使用したりできない。
inactive	offline の状態と同様、論理クラスタがいずれのインスタンス上でも実行していない。非アクティブな論理クラスタは、自動的には起動されず、フェールオーバには参加しない。クラスタは、deactivate コマンドを使用してのみ、この非アクティブ状態になる。いったん非アクティブになると、online コマンドを使用してのみ、クラスタはオンラインになる。	論理クラスタが現在のインスタンス上で実行していない。そのため、インスタンスでは接続を受け入れたり、リソースを使用したりできない。さらに、非アクティブなインスタンスはフェールオーバできず、自動起動後にオンラインにならない。deactivate コマンドを使用してのみ、この状態になる。
failed	offline の状態と同様、論理クラスタがいずれのインスタンス上でも実行していない。論理クラスタは、そのアクティブなインスタンスが shutdown with nowait であるか、または障害時用のリソースが利用できない場合にシステムに障害が発生すると、failed の状態に移行する。	論理クラスタが現在のインスタンス上で実行していない。そのため、インスタンスでは接続を受け入れたり、リソースを使用したりできない。nowait を使用して停止させるか、システムの障害時に failed の状態になる。
time_wait	online と offline、または online と inactive の間の遷移状態。online の論理クラスタは、offline または inactive に移行する前に time_wait の状態になる。このとき、新しい接続があれば down-routing モードに従ってルート指定され、既存の接続はマイグレートされるか切断される。	online と offline、または online と inactive の間の遷移状態。あるインスタンスに対して論理クラスタが online の場合、offline または inactive に移行する前に time_wait の状態になる。time_wait の状態では、新しい接続は論理クラスタまたはインスタンスにルート指定できないが、既存の接続はマイグレートか切断されるまで実行され続ける。

状態の変更方法

クラスタとインスタンスの状態は、次の方法で変更できます。

- 手動による変更。online、offline、failover、failback の各コマンドを使用して状態の変更を実行します。action コマンドを使用することもできます。
- 自動による変更。これはシステム変更の結果です。

クラスタまたはインスタンスの初期状態は、状態変更が有効かどうか、さらに最終的な状態も決定できます。表 6-2 は、異なるアクションの手動実行と状態の関係を示します。状態はロー、アクションはカラムで説明されています。各セルは、初期状態の論理クラスタまたはインスタンスにアクションが適用された場合の新しい状態を表します。

表 6-2: アクションと状態の関係

	offline	online	time_wait	failed	inactive
オンライン	online			online	online
オフライン		offline/time_wait		offline	offline
フェールバック・インスタンス	online			online	online
フェールバック・クラスタ		online/time_wait		offline	offline
フェールオーバ・インスタンス	online			online	online
フェールオーバ・クラスタ		online/time_wait		offline	offline
キャンセル・アクション			online		
wait の変更			time_wait		
非アクティブ化	inactive	inactive/time_wait		inactive	

システムの変更の結果として、状態が変わることもあります。表 6-3 は、システム変更による、クラスタまたはインスタンスの状態に対する影響を示します。

表 6-3: アクションと状態の関係

	offline	online	time_wait	failed	inactive
インスタンスがクラスタに参加	online (自動起動が設定されている場合)			online (自動起動が有効化されている場合)	
適切な停止		time_wait			
システムの障害		failed	failed		
nowait を使用した停止		failed	failed		
フェールオーバ選択	online				

注意 論理クラスタの状態は、クラスタ全体の再起動後は保持されません。たとえば、自動起動モードの論理クラスタに offline コマンドを実行したとします。このクラスタは、再起動後にはオンラインの状態になります。

非同期のコマンドと論理クラスタの状態

sp_cluster logical コマンドの deactivate、failback、failover、offline は、非同期です。これらのコマンドは、トランザクションが存在する可能性のあるオンラインのインスタンスを停止します。これらのトランザクションは、インスタンスが実際にオフラインになったり、非アクティブになったりする前に処理する必要があります。したがって、これらのコマンドは後で完了させることができます。

これらのコマンドのいずれかを実行すると、ターゲット・インスタンスは time_wait 状態になり、新しい接続を受け入れなくなります。

各非同期コマンドには、既存のトランザクションを処理するための 3 つの “待機” オプションがあります。値は次のとおりです。

- wait − 既存の接続をたとえば 5 分間など、指定した期間存続させます。マイグレーションをサポートする接続は、静止状態になるとすぐにマイグレートされます。マイグレーションをサポートしない接続は、静止状態になると切断されます。HA 対応のクライアントは、フェールオーバし、HA 対応でないクライアントは切断されます。指定した期間以後に残存する接続は終了されます。
- until − 既存の接続をたとえば午後 12:30 などの指定した時刻まで存続させます。その他の点では、until と wait は接続を同じように処理します。
- nowait − 既存の接続を即時終了します。マイグレーションをサポートする接続は、すぐにマイグレートしないと終了されます。

注意 sp_cluster logical 非同期コマンドの実行時に待機オプションを指定しないと、Adaptive Server は無限に待機します。

インスタンスを使用する最後の接続が切断すると、インスタンスの状態は、time_wait から offline または inactive に変更されます。

アクション記述子の使用

アクション記述子を使用すると、アクションの追跡または変更ができます。

非同期コマンドが 1 つ以上のインスタンスを停止しようとすると、アクション記述子が生成されます。アクション記述子は、アクション、待機オプション、time_wait 状態であるターゲット・インスタンスを追跡します。アクション記述子に関する情報を表示するには、monLogicalCLusterAction テーブルをクエリするか、`sp_cluster logical, "show", NULL, action` を実行します。

アクションは、“active” または “complete” のいずれかです。アクションは、少なくとも 1 つのターゲット・インスタンスが time_wait 状態のままである場合にアクティブです。アクションは、すべてのターゲット・インスタンスが time_wait 状態ではなくなると完了します。

sp_cluster logical, action では、次のオプションを使用して、アクション記述子を管理できます。

- cancel – アクティブなアクションを終了します。そのアクションのために time_wait 状態であったインスタンスは、オンラインの状態に戻ります。既存の接続は、マイグレーションまたは終了のマークが付いていても存続します。

  アクションを作成したコマンドによってインスタンスがオンラインになった場合は、それらのインスタンスはそのままの状態になります。たとえば、アクションによって s1 から f1 へのフェールオーバがキャンセルされると、f1 はオンラインのままの状態になります。

- modify_wait – 待機オプション (「非同期のコマンドと論理クラスタの状態」(91 ページ) を参照) とアクティブなアクションに関連する時間を変更します。たとえば、10 分の待機時間を持つアクションが作成されている場合、modify_wait を使用して次のように変更できます。

  - 時間遅延を 20 分に変更
  - 時間遅延を実際の時刻である午後 4:30 に変更
  - 待機オプションを nowait に変更

- release – 完了したアクションを monLogicalClusterAction テーブルから削除します。

  完了したアクションは、monLogicalClusterAction テーブルに残るので、そのステータスを追跡できます。ただし、完了したアクションは、Workload Manager のキャッシュのメモリを消費します。アクションの完了後は、release コマンドを実行し、このメモリを解放します。

注意 アクション情報はメモリにのみ保存されます。クラスタ全体を再起動すると、すべてのアクションが monLogicalClusterAction テーブルからクリアされます。

例：フェールオーバのスケジューリングと再スケジューリング

クラスタまたはインスタンスに対して、管理用のフェールオーバを実行できます。クラスタまたはインスタンスは、事前に設定されたフェールオーバ・リソースにフェールオーバします。

次の例では、“SalesLC” クラスタを午前 2 時にスケジューリングしてフェールオーバさせます。後でアクションを追跡したり変更したりできるように、アクション・ハンドルを出力する構文も追加します。

```
declare @out_handle varchar(15)

execute
```

```
sp_cluster logical, "failover", SalesLC, cluster, NULL, until,
"02:00:00", @handle = @out_handle output
```

コマンドがアクション・ハンドル “1234” を出力し、SalesLC が time_wait 状態に入るとします。新しい接続はすべて、フェールオーバ・リソースにマイグレートされます。午前 2 時以降に残っている接続があれば終了され、“SalesLC” は offline 状態に入ります。

すべての接続をすぐにマイグレートする必要があることが判明したとします。アクション・ハンドルを使用して、即時フェールオーバを再スケジューリングできます。次のように入力します。

```
sp_cluster logical, "failover", SalesLC, modify_time, "1234",
nowait
```

failover、failback、online、offline、deactivate の使用

failover

failover は、論理クラスタの基本リソースからそのフェールオーバ・リソースへの手動によるフェールオーバを開始します。フェールオーバ・リソースは、事前に sp_cluster logical, "add" を使用して設定する必要があります。部分クラスタ・フェールオーバを開始する場合、フェールオーバする基本リソースのリストと基本インスタンスのフェールオーバ先となるフェールオーバ・リソースのリストを指定します。

たとえば、論理クラスタの一部を、事前に設定した一連のフェールオーバ・リソースにフェールオーバできます。ここで、“SalesLC” はインスタンス “ase1” および “ase2” で実行中です。“SalesLC” を “ase2” で実行したままにし、“ase1” を事前に定義したフェールオーバ・リソース “ase3” にフェールオーバするには、次のように入力します。

```
sp_cluster logical, "failover", SalesLC, instance, ase1, ase3
```

この例では、no wait オプションが指定されており、デフォルトで無限待機が指定されます。

failback

failback は、failover を元に戻します。論理クラスタのフェールオーバ・リソースからその基本リソースへの手動によるフェールバックを開始します。部分クラスタ・フェールオーバを開始する場合、フェールバックするフェールオーバ・リソースのリストとフェールオーバ・インスタンスのフェールバック先となる基本リソースのリストを指定します。

次の例では、“ase1” で実行中の “SalesLC” が “ase1” で実行されるように段階的にフェールバックします。2 分の待機を指定します。

```
declare @out_handle varchar(15)

execute
sp_cluster logical, "failback", SalesLC, instance, ase3, ase1,
wait, "00:02:00", @handle = @out_handle output
```

online

online は、論理クラスタまたは論理クラスタ内のインスタンスを起動し、それらをオンライン状態にします。

たとえば、“SalesLC” を “ase1” 上で起動するには、次のように入力します。

```
sp_cluster logical, "online", SalesLC, ase1
```

その他の例については、「論理クラスタの起動」(73 ページ) を参照してください。

offline

offline は、オンラインまたはアクティブ状態にある論理クラスタまたはインスタンスを停止します。

たとえば、“SalesLC” をオフラインにし、5 分間待機し、アクション・ハンドルのアクションをローカル変数に保存するには、次のように入力します。

```
declare @out_handle varchar(15)

execute
sp_cluster logical, "offline", SalesLC, cluster, wait,
00:05:00, @handle=@out_handle output
```

deactivate

deactivate は、クラスタまたはインスタンスを非アクティブな状態にすること以外は、offline と同じです。「offline」(94 ページ) を参照してください。

負荷の分散

各インスタンスには、現在のクラスタ上の負荷を計算し、その情報をクラスタ内の他のインスタンスに送信する役割を果たす Workload Manager スレッドが存在します。Workload Manager は、システムサービス・スレッドです。インスタンスが起動すると生成されます。

Adaptive Server は、負荷測定アルゴリズムを使用し、インスタンスごとに負荷スコアを計算します。この負荷スコアは、単位のない数字で、他の負荷スコアと比較することで相対的な負荷を決定できます。そのため、クラスタにわたって負荷スコアを比較したり、特定のクラスタ用に異なる時間で比較したりできます。負荷スコアは、他の負荷スコアと比較して初めて意味があります。

負荷測定基準

Workload Manager は、15 秒ごとに負荷スコアを再計算します。その際、その負荷スコア (および結果として得られる統計データ) を使用して、クラスタ内のインスタンスすべてに対する相対的な負荷を比較します。Workload Manager は、この統計データを使用してモニタリング・テーブルに値を入力します。統計データは、インスタンスごとのデータです。Workload Manager が SPID ごとの統計データを追跡することはありません。

workload_metric を使用して、インスタンスのユーザ測定基準の値を更新できます。この値は、インスタンスの合計負荷スコアの計算に使用されます。workload_metric はユーザ定義のストアド・プロシージャ、トリガ、または外部でトリガされるスクリプトに埋め込むことができます。インスタンスは、負荷スコアを 15 秒ごとに再計算します。

負荷スコアの計算時、Adaptive Server ではシステム定義の測定基準が 5 つ、オプションでユーザ定義の測定基準が 1 つ考慮されます。

- ユーザ接続 − 使用できるリソースに基づいた、新しい接続を受け入れるインスタンスの能力。
- CPU 利用率 − 追加の作業を処理するインスタンスの能力。
- 実行キュー長 − システム上の実行可能なタスクの数。実行キューの長さは、処理バックログを測定するものであり、相対的な応答時間の良いインジケータです。
- I/O 負荷 − 未処理の非同期 I/O。

- エンジンの不足 – クラスタ内のインスタンス間のオンライン・エンジン数の違い。

  注意 エンジンの不足は、クラスタ内のインスタンスが異なる数のエンジンを持っている場合にのみ測定可能です。このシナリオでは、エンジンの不足は、相対的な最大容量の測定基準を負荷スコアに追加します。

- ユーザ測定基準 – ユーザ環境に特有のオプションの測定基準。workload_metric を使用して、値を提供する。「ユーザ測定基準の作成」を参照してください。

負荷スコアは、次の式で計算します。

```
ConnectionsWeight  x  (ConnectionsLoad)/100
+                        CPUWeight x (CPULoad)/100
+      RunQueueWeight x (RunQueueLoad)/100
+ UserDefinedWeight x (UserDefinedLoad)/100
+                EngineWeight x (EngineLoad)/100
+                             IOWeight x (IOLoad)/100
-----------------------------------------------------------------
=                                                  Load score
```

ユーザ測定基準の作成

workload_metric 組み込み関数を使用して、サイト特有の測定基準を負荷測定アルゴリズムに追加できます。典型的な例では、外部モニタを使用して応答時間をモニタリングし、アルゴリズムに応答時間の値を挿入することが考えられます。

Adaptive Server は、システム提供の測定基準を負荷アルゴリズムに含める前に標準化します。互換性を保つために、ユーザ提供の測定基準も正規化してください。たとえば、許容可能な最大応答時間が 5 秒で、測定された応答時間が 2 秒の場合、正規化された値は 40 であり (5 に対して 2 は 40%)、この値を workload_metric を使用して負荷アルゴリズムに入力できます。

負荷測定基準の重み付け

負荷スコアの各要素は、その重要度に応じて重み付けされます。測定基準値は正規化され、その結果は合計され、インスタンスごとに全体的な負荷スコアが計算されます。ほとんどのサイトでは、Sybase が提供するデフォルト値で十分ですが、サイト特有の測定基準を含める場合は、sp_cluster profile を使用して重みを調整してください。

重み付け係数の入力と調整は、Sybase Central または sp_cluster を使用して行います。システム管理者はこの係数を確認して、負荷プロファイルをカスタマイズします。

負荷スレッショルド

Adaptive Server は、各インスタンスの負荷スコアを使用して次の内容を決定します。

- 受信接続を最適に分散する方法 − ログイン・リダイレクト。
- 既存の接続をマイグレートするかどうか − 動的負荷分散。

負荷分散は、複数のインスタンス上で実行中の論理クラスタに対してのみ実行されます。Sybase の負荷分散方式では、1 つのインスタンスが過負荷状態で他のインスタンスが利用可能な場合に作業がリダイレクトされます。この方式では、完全にバランスの取れた負荷スコアを維持しようとはしません。その結果、Adaptive Server は、ログイン・リダイレクトおよび動的負荷分散に関する負荷スレッショルドを保持します。負荷スレッショルドとは、現在のインスタンスと論理クラスタ内の最小負荷インスタンスの、それぞれの負荷の差異 (割合) です。その値が満たされてから、Adaptive Server はログインをリダイレクトするか、または既存の接続をマイグレートします。

Adaptive Server は、ログイン・リダイレクトと接続マイグレーションには、別々の負荷スレッショルドを保持します。通常、ログイン・リダイレクトのスレッショルドは、接続マイグレーションのスレッショルドよりも低くなります。sp_cluster profile を使用して、負荷プロファイルを作成する際に、負荷スレッショルドを設定できます。

ヒステリシス値

負荷スレッショルドにより、Adaptive Server は過負荷状態のインスタンスからより負荷の少ないインスタンスへ接続をリダイレクトするタイミングを判断します。ヒステリシス値は、スレッショルド値は満たされているものの、実際の負荷スコアが低すぎてマイグレーションを要求できない場合に、マイグレーションされないようにします。

たとえば、現在のインスタンスの負荷スコアが 2 で、負荷の最も低いインスタンスの負荷スコアが 1 であるとします。割合の差異は 100% で、負荷スレッショルドを満たしますが、実際の負荷スコアが低すぎてマイグレーションは適切ではありません。

負荷プロファイル

負荷プロファイルは、負荷スコア・システムの設定可能なすべての部分を単一の名前付きエンティティに集約します。

同じ物理クラスタ内の異なる論理クラスタに対して別々の負荷プロファイルを割り当てることができるので、同じ物理クラスタ内にさまざまな負荷を持つ複数のアプリケーションが存在できます。また、複数の論理クラスタ内で同じ負荷プロファイルを共有することもできます。

たとえば、1 つの共有ディスク・クラスタで、主に読み取り専用の DSS ベースの論理クラスタと頻繁な書き込みを処理する OLTP 論理クラスタを同時にホストできます。これらの 2 つのクラスタ用のログイン・リダイレクトと接続マイグレーションに対する最適なスレッショルドは、かなり異なる可能性があります。論理クラスタごとに特定の負荷プロファイルを割り当てる機能によって、各クラスタはより効率的に動作できます。

注意 Adaptive Server は、論理クラスタを考慮せずに、インスタンスごとの負荷統計を収集します。そのため、2 つの論理クラスタが同じインスタンス上で実行されると、それらはそのインスタンスに関して同じ未加工データを持つことになります。ただし、各論理クラスタは、独自の負荷プロファイルに従ってそのデータを解釈し使用します。

Sybase は、OLTP 環境向けに作成されている、事前設定されたプロファイルを提供しています。ユーザは、sp_cluster profile を使用して、独自の負荷プロファイルを作成することもできます。

サンプル負荷プロファイルの使用

Sybase は、次の 2 つのサンプル負荷プロファイルを提供しています。

- sybase_profile_oltp － OLTP 環境用に設定されています。これは、負荷ベースのログイン分散と動的負荷分散を無効にすることで、同じインスタンス上ですべての接続を保持しようとします。再キューの深さに重点が置かれており、応答時間の予測に優れています。
- sybase_profile_dss － 主に読み込み専用の DSS 環境用に設定されています。これは、負荷ベースのログイン分散と動的負荷分散を使用して、複数のインスタンスに負荷を分散しますが、CPU 使用率とユーザ接続のバランスに重点が置かれています。

表 6-4 に、sybase_profile_oltp と sybase_profile_dss の測定基準を一覧表示します。

表 6-4: サンプル負荷プロファイルの測定基準

負荷測定基準	“sybase_profile_oltp”	“sybase_profile_dss”
プロファイル ID	1	2
ユーザ接続	0	40
CPU 利用率	10	40
実行キューの長さ	70	0
I/O 負荷	20	10
エンジンの不足	0	10
ユーザによる重み付け (不使用)	0	0
ログイン・スレッショルド	0	20
動的スレッショルド	0	50
ヒステリシス	20	20

独自の負荷プロファイルの作成と設定

独自の負荷プロファイルを作成して設定するには、次の手順に従います。

1　空の負荷プロファイルを作成します。

2　個別の測定基準の重みおよびスレッショルドを指定し、負荷プロファイルを構築します。

3　負荷プロファイルを論理クラスタに関連付けます。

負荷プロファイルの作成

sp_cluster profile, "create" を使用して空の負荷プロファイルを構築します。たとえば、プロファイル “my_profile” を作成するには、次のように入力します。

```
sp_cluster profile, "create", my_profile
```

負荷プロファイルの構築

次を指定して負荷プロファイルを構築します。

- 負荷プロファイルを構成する各測定基準の重み
- 負荷分散スレッショルド

負荷プロファイルの測定基準の重み指定

sp_cluster profile, "set" を使用して、負荷プロファイル内に含まれる測定基準ごとに重みを設定します。

- ユーザ接続
- CPU ビジー
- 実行キューの長さ
- I/O 負荷
- エンジンの不足
- ユーザ測定基準 (オプション、「ユーザ測定基準の作成」(96 ページ) を参照)

負荷測定基準の説明については、「負荷測定基準」(95 ページ) を参照してください。

0 ～ 255 の値を使用し、各測定基準を個別に設定します。たとえば、“my_profile” に重みを設定するには、次のように入力します。

```
sp_cluster profile, "set", my_profile, weight, "user
connections", "0"

sp_cluster profile, "set", my_profile, weight,
"cpu busy", "20"

sp_cluster profile, "set", my_profile, weight, "run queue",
"30"

sp_cluster profile, "set", my_profile, weight,
"io load", "10"

sp_cluster profile, "set", my_profile, weight, "engine
deficit", "10"

sp_cluster profile, "set", my_profile, weight, "user metric",
"30"
```

負荷分散スレッショルドの指定

sp_cluster profile, "set" を使用して、負荷分散スレッショルドを 0 ～ 100 の値に設定します。値をゼロ (0) にすると、負荷分散の機能が無効になります。次の項目に別々の負荷スレッショルドを設定できます。

- ログイン・リダイレクト
- 動的負荷分散
- ヒステリシス値

たとえば、“my_profile” 内で動的負荷分散をオフにするには、次のように入力します。

```
sp_cluster profile, "set", my_profile, threshold, "dynamic",
"0"
```

“my_profile” のログイン・リダイレクト・スレッショルドを 30 に設定するには、次のように入力します。

```
sp_cluster profile, "set", my_profile, threshold, "login",
"30"
```

“my_profile” のヒステリシス値を 20 に設定するには、次のように入力します。

```
sp_cluster profile, "set", my_profile, threshold,
"hysteresis", "20"
```

負荷プロファイルと論理クラスタとの関連付け

負荷プロファイルを論理クラスタに関連付けるには、sp_cluster logical, “set” を使用します。たとえば、プロファイル “my_profile” を “SalesLC” に関連付けるには、次のように入力します。

```
sp_cluster logical, "set", SalesLC, load_profile, my_profile
```

負荷プロファイルの変更

各論理クラスタは、システム負荷プロファイルかユーザ作成の負荷プロファイルのいずれかに関連付けられます。負荷プロファイルを変更するには、論理クラスタに新しい負荷プロファイルを関連付けます。

たとえば、“SalesLC” の負荷プロファイルを “my_profile” から “sybase_profile_oltp” に変更するには、次のように入力します。

```
sp_cluster logical, "set", SalesLC, load_profile,
sybase_profile_oltp
```

その後、古い負荷プロファイルを削除できます。たとえば、“my_profile” を削除するには、次のように入力します。

```
sp_cluster profile, "drop", my_profile
```

トラブルシューティング

Sybase は、Workload Manager のトラブルシューティングに使用できる、いくつかのトレース・フラグを提供しています。

表 6-5: Workload Manager 用のトレース・フラグ

トレース・フラグ	説明
16403	論理クラスタへの接続および論理クラスタからの切断を通じて、SPID をトレースする。
16404	ルートとルート解決をトレースする。
16406	クライアント・リダイレクトとマイグレーションをトレースする。
16414	オンラインからオフラインへの移行などを含む、論理クラスタの状態マシンにおける変更をトレースする。

第 7 章

クラスタ・キャッシュ構成

この章では、Cluster Edition 環境における名前付きデータ・キャッシュの設定と使用のメカニズムについて説明します。

クラスタ・キャッシュ設定は、複数の名前付きキャッシュがアプリケーションのニーズに応じて、ローカルまたはグローバル・キャッシュを持つように定義します。この機能を使用すると、クラスタ・インスタンスにローカル・キャッシュを持たせることができます。オブジェクトはローカルまたはグローバル・キャッシュにバインドできます。複数のバッファ・プール・サポートは、大容量 I/O を円滑に行うことで、名前付きキャッシュ・サポートへのアクセス・パフォーマンスを向上させます。

ユーザは、アプリケーションを分割し、そのアプリケーションを対象とする特定のインスタンスに対して、アプリケーション・データのアクセスをローカライズすることもできます。

グローバル・キャッシュ

グローバル・キャッシュは、クラスタのすべてのインスタンスに対して定義されます。グローバル・キャッシュの場合、キャッシュ・サイズやバッファ・プール設定のような属性は sysconfigures テーブルであり、クラスタのすべてのインスタンスは、このエントリから読み込まれ、該当するインスタンスのキャッシュを作成します。

グローバル・キャッシュのキャッシュ・サイズ、バッファ・プール設定などの属性を、インスタンス固有のものに変更できます。特定のインスタンスにローカル設定がある場合、キャッシュはそれらを使用して作成されます。インスタンスにローカル定義がない場合、グローバル定義を使用してキャッシュを作成します。つまり、ローカル定義があるインスタンスは、グローバル定義および設定を上書きします。

注意 ローカル・キャッシュおよびグローバル・キャッシュのサイズは動的に増大させることができますが、動的に縮小することはできません。

ローカル・キャッシュ

アプリケーションはクラスタの各インスタンスにローカル・キャッシュを定義して、インスタンスの固有のニーズに合わせ、キャッシュをオブジェクトにバインドできます。グローバル定義は、クラスタのそれぞれのインスタンス固有のキャッシュに必要ありません。

ローカル・キャッシュはインスタンス固有です。インスタンスやインスタンスが属する論理クラスタのニーズに設定を合わせることができます。特定のアプリケーションが通常特定のインスタンスで実行される場合は、クラスタ内のインスタンス間でアプリケーションを分割することをお勧めします。これにより、特定のアプリケーションのアクセス・パターンに対して特別に設定できるようになるため、ローカル・キャッシュの最大限の利益を得られます。

オブジェクトは、任意の特定のインスタンスのローカルまたはグローバルのいずれかの 1 つのキャッシュにのみバインドされます。オブジェクトをキャッシュにバインドしない場合、またはフェールオーバが発生し、インスタンス固有のキャッシュが設定されない場合、デフォルト・データ・キャッシュが使用されます。効果的なアクセスを支援するために、Adaptive Server はすべてのインスタンスのバインド情報を管理しています。

注意 任意のインスタンスのローカル・キャッシュ定義は、そのインスタンスのグローバル定義を上書きします。

名前付きデータ・キャッシュの作成と設定

sp_cacheconfig はグローバル・データ・キャッシュとローカル・データ・キャッシュの両方を作成して設定します。Adaptive Server には、デフォルト・データ・キャッシュと呼ばれる 1 つのグローバル・キャッシュが含まれています。

名前付きキャッシュに関する情報の取得

以下のように入力すると、キャッシュに関する情報を表示できます。

```
sp_cacheconfig
go

Cache Name           Status           Type  Config Value   Run Value
------------------ ------ -------------- ------------ ------------
default data cache Active Global,Default   0.00 Mb         8.00 Mb
------------ ------------       Total      0.00 Mb         8.00 Mb
================================================================================
Cache: default data cache, Status: Active, Type: Global,Default
        Config Size: 0.00 Mb, Run Size: 8.00 Mb
        Config Replacement: strict LRU, Run Replacement: strict LRU
        Config Partition: 1, Run Partition: 1
IO Size    Wash Size       Config Size     Run Size      APF Percent

-------- ------------- ------------ ------------ -----------
2 Kb        1638 Kb         0.00 Mb         8.00 Mb           10
(return status = 0)
```

新しいキャッシュの作成

データ・キャッシュの最大サイズは、システム上で使用可能なメモリ容量によってのみ制限されます。新しいキャッシュの作成に必要なメモリは、Adaptive Server のグローバル・メモリから取得されます。

キャッシュは、次のように作成されます。

- デフォルトのウォッシュ・サイズが定義される。
- 非同期プリフェッチ・サイズは、グローバル非同期プリフェッチ制限の値に設定される。
- デフォルトのバッファ・プールのみが存在する。

sp_cacheconfig

定義

新しい名前付きキャッシュを作成します。この構文は、ノンクラスタ Adaptive Server の拡張です。構文の末尾にローカル構成のインスタンス名を指定する追加オプションがあります。インスタンス名を指定しない場合、設定はグローバルになります。

構文

```
sp_cacheconfig "[cachename [,cache_size [P|K|M|G]"
[,logonly | mixed ] [,strict | relaxed ] ]
        [, "cache_partition = [1|2|4|8|16|32|64]"]
[, "instance instance_name"]
```

パラメータ

cachename

作成または設定するデータ・キャッシュの名前です。キャッシュ名はユニークな名前にしてください。最大 30 文字まで指定できます。有効な Adaptive Server 識別子である必要はありません。スペースや他の特殊文字も指定できます。

cache_size

作成するデータ・キャッシュのサイズです。すでにそのキャッシュが存在する場合は、データ・キャッシュの新しいサイズを指定します。最小キャッシュ・サイズは、サーバの論理ページ・サイズの 256 倍です。サイズをページ単位で指定する場合は P を使用します。バイト数で指定する場合は、K (キロバイト)、M (メガバイト)、G (ギガバイト) を使用します。デフォルトは K です。メガバイトとギガバイトによる指定では、浮動小数点数を指定できます。キャッシュ・サイズは論理ページ・サイズの倍数で指定します。

logonly | mixed − キャッシュのタイプを指定します。

strict | relaxed − キャッシュ置換方式を指定します。

cache_partition − キャッシュ内に作成するパーティションの数を指定します。

想定例 次の例では、MYCLUSTER という名前の共有ディスク・クラスタを使用することを前提としています。このクラスタには、以下の 2 つのインスタンスが含まれています。

- SALES_INSTANCE
- HR_INSTANCE

名前付きキャッシュの作成 インスタンス SALES_INSTANCE に固有の 100M のサイズの log_sales という名前付きキャッシュを作成できます。インスタンス SALES_INSTANCE 上で sp_cacheconfig を実行すると、以下のような出力が得られます。

```
sp_cacheconfig 'log_sales','100M','instance SALES_INSTANCE'
go


 Cache Name              Status Type            Config Value Run Value
------------------------ ------ -------------- ------------ ------------
default data cache        Active Global,Default       0.00 Mb    8.00 Mb
```

```
SALES_INSTANCE:log_sales  Active Mixed            100.00 Mb  100.00 Mb
                                                 ------------ ------------
                                               Total 100.00 Mb 108.00 Mb
===============================================================================
Cache: default data cache, Status: Active, Type: Global,Default
        Config Size: 0.00 Mb, Run Size: 8.00 Mb
        Config Replacement: strict LRU, Run Replacement: strict LRU
        Config Partition: 1, Run Partition: 1
IO Size   Wash Size    Config Size    Run Size    APF Percent
-------- ------------ ------------ ------------ -----------
2 Kb      1638 Kb         0.00 Mb    .00 Mb        10
===============================================================================
Cache: SALES_INSTANCE:log_sales, Status: Active, Type: Mixed
        Config Size: 100.00 Mb, Run Size: 100.00 Mb
        Config Replacement: strict LRU, Run Replacement: strict LRU
        Config Partition: 1, Run Partition: 1
IO Size     Wash Size  Config Size      Run Size   APF Percent
-------- ------------ ------------ ------------    -----------
2 Kb         20480 Kb    0.00 Mb        100.00 Mb    10
(return status = 0)
```

デフォルトで、isql 接続にはクラスタ・ビューがあります。インスタンス固有のすべてのキャッシュが、すべてのインスタンスで表示されます。たとえば、インスタンス HR_INSTANCE はキャッシュ log_sales に関する情報を表示します。これは、SALES_INSTANCE に対してインスタンス固有のキャッシュです。HR_INSTANCE にこのインスタンス固有のローカル・キャッシュとグローバル・キャッシュのリストのみを表示する場合、システム・ビューを instance に設定します。

インスタンス HR_INSTANCE 上で sp_cacheconfig を実行すると、以下のような出力が得られます。

```
set system_view instance
go

Cache Name        Status Type             Config Value   Run Value
------------------ ------ -------------- ------------ ------------
default data cache Active Global,Default    0.00 Mb     8.00 Mb
                                           ------------ ------------
                                      Total 0.00 Mb 8.00 Mb
===============================================================================
Cache: default data cache, Status: Active, Type: Global,Default
        Config Size: 0.00 Mb, Run Size: 8.00 Mb
        Config Replacement: strict LRU, Run Replacement: strict LRU
        Config Partition: 1, Run Partition: 1
IO Size     Wash Size    Config Size   Run Size    APF Percent
-------- ------------ ------------ ------------ -----------
2 Kb         1638 Kb       0.00 Mb      8.00 Mb       10
(return status = 0)
```

キャッシュ情報出力のインスタンス・レベルへの制限 インスタンス SALES_INSTANCE でキャッシュを表示するには、以下のように実行します。

```
sp_cacheconfig 'instance SALES_INSTANCE'
go
Cache Name                Status Type             Config Value   Run Value
------------------------ ------ -------------- ------------ ------------
default data cache Active Global,Default          0.00 Mb         8.00 Mb
SALES_INSTANCE:log_sales Active Mixed           100.00 Mb        100.00 Mb
                                                ------------ ------------
                                          Total 100.00 Mb 108.00 Mb
===============================================================================
Cache: default data cache, Status: Active, Type: Global,Default
        Config Size: 0.00 Mb, Run Size: 8.00 Mb
        Config Replacement: strict LRU, Run Replacement: strict LRU
        Config Partition: 1, Run Partition: 1
IO Size   Wash Size     Config Size   Run Size    APF Percent
-------- ------------ ------------ ------------ -----------
2 Kb       1638 Kb       0.00 Mb       8.00 Mb       10
===============================================================================
Cache: SALES_INSTANCE:log_sales, Status: Active, Type: Mixed
        Config Size: 100.00 Mb, Run Size: 100.00 Mb
        Config Replacement: strict LRU, Run Replacement: strict LRU
        Config Partition: 1, Run Partition: 1
IO Size      Wash Size  Config Size     Run Size   APF Percent
-------- ------------ ------------ ------------ -----------
2 Kb        20480 Kb       0.00 Mb    100.00 Mb        10
(return status = 0)
```

この出力は、インスタンス SALES_INSTANCE のローカル設定とグローバル設定の両方を表示します。

名前付きキャッシュの存在の確認 指定されたキャッシュがすでに存在するかどうかを確認します。インスタンス SALES_INSTANCE 上で sp_cacheconfig を実行すると、以下のような出力が得られます。

```
Cache Name               Status   Type    Config Value Run Value
------------------------ ------ ----- ------------ ------------
SALES_INSTANCE:log_sales Active Mixed    100.00 Mb   100.00 Mb
                                      ------------ ------------
                                 Total 100.00 Mb 100.00 Mb
===============================================================================
Cache: SALES_INSTANCE:log_sales, Status: Active, Type: Mixed
        Config Size: 100.00 Mb, Run Size: 100.00 Mb
        Config Replacement: strict LRU, Run Replacement: strict LRU
        Config Partition: 1, Run Partition: 1
IO Size    Wash Size    Config Size    Run Size    APF Percent
-------- ------------ ------------ ------------ -----------
2 Kb       20480 Kb       0.00 Mb       100.00 Mb    10
(return status = 0)
```

Adaptive Server 構文を使用したグローバル・キャッシュの作成 このキャッシュはすべてのインスタンスで作成され、メモリはグローバル・キャッシュのすべてのインスタンスに割り付けられます。グローバル・キャッシュ tempdb_cache を作成するには、以下のようにインスタンス SALES_INSTANCE で sp_cacheconfig を実行します。

```
sp_cacheconfig 'tempdb_cache', '100M'
go
Cache Name              Status Type             Config Value   Run Value
----------------------- ------ -------------- ------------ ------------
default data cache Active Global,Default            0.00 Mb   8.00 Mb
tempdb_cache Active Global,Mixed                  100.00 Mb    100.00 Mb
SALES_INSTANCE:log_sales Active Mixed             100.00 Mb   100.00
Mb                                               ------------ ------------Total 200.00 Mb
208.00 Mb
===============================================================================
Cache: default data cache, Status: Active, Type: Global,Default
        Config Size: 0.00 Mb, Run Size: 8.00 Mb
        Config Replacement: strict LRU, Run Replacement: strict LRU
        Config Partition: 1, Run Partition: 1
IO Size    Wash Size    Config Size   Run Size     APF Percent
-------- ------------- ------------ ------------ -----------
2 Kb      1638 Kb         0.00 Mb      8.00 Mb         10
===============================================================================
Cache: tempdb_cache, Status: Active, Type: Global,Mixed
        Config Size: 100.00 Mb, Run Size: 100.00 Mb
        Config Replacement: strict LRU, Run Replacement: strict LRU
        Config Partition: 1, Run Partition: 1
IO Size   Wash Size     Config Size   Run Size      APF Percent
-------- ------------- ------------ ------------ -----------
2 Kb        20480 Kb     0.00 Mb      100.00 Mb     10
===============================================================================
Cache: SALES_INSTANCE:log_sales, Status: Active, Type: Mixed
        Config Size: 100.00 Mb, Run Size: 100.00 Mb
        Config Replacement: strict LRU, Run Replacement: strict LRU
        Config Partition: 1, Run Partition: 1
IO Size    Wash Size    Config Size    Run Size    APF Percent
-------- ------------- ------------ ------------ -----------
2 Kb        20480 Kb      0.00 Mb       100.00 Mb    10
(return status = 0)
```

単一のグローバル設定と複数のローカル設定での名前付きキャッシュの作成

すべてのキャッシュ操作を任意のインスタンスから実行できます。たとえば、SALES_INSTANCE で tempdb_cache という大きな名前付きキャッシュを作成するには、インスタンス HR_INSTANCE に接続してから、以下のように実行します。

```
sp_cacheconfig 'tempdb_cache','150M', 'instance
SALES_INSTANCE'
```

インスタンス SALES_INSTANCE で sp_cacheconfig を実行すると、以下のような出力が得られます。

```
Cache Name          Status Type           Config Value   Run Value
------------------ ------ -------------- ------------ ------------
default data cache Active Global,Default 0.00 Mb        8.00 Mb
tempdb_cache Active Global,Mixed       100.00 Mb     100.00 Mb
SALES_INSTANCE:log_hr Active Mixed       150.00 Mb     150.00 Mb
SALES_INSTANCE:tempdb_cache Active Mixed 150.00 Mb     150.00 Mb
                                  ------------ ------------
                                Total 350.00 Mb 408.00 Mb
===============================================================================
Cache: default data cache, Status: Active, Type: Global,Default
        Config Size: 0.00 Mb, Run Size: 8.00 Mb
        Config Replacement: strict LRU, Run Replacement: strict LRU
        Config Partition: 1, Run Partition: 1
IO Size     Wash Size      Config Size   Run Size APF Percent
-------- ------------ ------------ ------------ -----------
2 Kb         1638 Kb        0.00 Mb         8.00 Mb     10
===============================================================================
Cache: tempdb_cache, Status: Active, Type: Global,Mixed
        Config Size: 100.00 Mb, Run Size: 150.00 Mb
        Config Replacement: strict LRU, Run Replacement: strict LRU
        Config Partition: 1, Run Partition: 1
IO Size   Wash Size    Config Size     Run Size   APF Percent
-------- ------------ ------------ ------------ -----------
2 Kb       30720         Kb 0.00Mb    150.00 Mb        10
===============================================================================
Cache: SALES_INSTANCE:log_sales, Status: Active, Type: Mixed
        Config Size: 100.00 Mb, Run Size: 100.00 Mb
        Config Replacement: strict LRU, Run Replacement: strict LRU
        Config Partition: 1, Run Partition: 1
IO Size    Wash Size    Config Size    Run Size  APF Percent
-------- ------------ ------------ ------------ -----------
2 Kb      20480 Kb       0.00 Mb      100.00 Mb     10
===============================================================================
Cache: SALES_INSTANCE:tempdb_cache, Status: Active, Type: Mixed
        Config Size: 150.00 Mb, Run Size: 150.00 Mb
        Config Replacement: strict LRU, Run Replacement: strict LRU
        Config Partition:           1, Run Partition:   1
IO Size  Wash Size      Config Size   Run Size     APF Percent
-------------------------------------------------------------
  2 Kb  30720 Kb        0.00 Mb        150.00 Mb        10
```

```
(return status = 0)
```

注意　名前付きキャッシュ tempdb_cache のローカル設定は、グローバル設定を上書きします。

たとえば、システム・ビューを cluster に設定した場合、Adaptive Server では名前付きキャッシュのすべての設定が表示されることがあるため、その場合は、そのインスタンスで有効でない設定の実行値を無視する必要があります。たとえば、SALES_INSTANCE には有効なローカル設定であるキャッシュ tempdb_cache があります。したがって、グローバル設定の実行値を無視する必要があります。

同様に、HR_INSTANCE には有効なグローバル設定があります。したがって、インスタンス HR_INSTANCE の SALES_INSTANCE に関連する temdb_cache のローカル設定の実行値を無視する必要があります。

既存のキャッシュへのメモリの追加　メモリを追加するには、『リファレンス・マニュアル：プロシージャ』に示された構文を使用します。割り付けた追加メモリは、Adaptive Server のページ・サイズのプールに追加されます。たとえば、プールの最小サイズは論理ページ・サイズが 2K のサーバでは 2K になります。キャッシュが分割されている場合は、追加のメモリは各パーティションに均等に分けられます。

注意　あるインスタンスで既存のグローバル・キャッシュにメモリを追加することに失敗しても、少なくとも他の 1 つのインスタンスで成功すれば、サーバはクラスタ全体のレベルで成功したと見なします。したがって、グローバル・キャッシュの場合は異なる実行値を持つことができますが、キャッシュに対しては 1 つの設定値しか持つことができません。sp_cacheconfig では、現在のインスタンスの syscurconfigs エントリから、グローバル・キャッシュの実行値を表示します。

tempdb_cache のサイズを 200MB に増大させるには、インスタンス HR_INSTANCE で、以下を実行します。インスタンス HR_INSTANCE 上で sp_cacheconfig 'tempdb_cache' を実行すると、以下のような出力が得られます。

```
    sp_cacheconfig 'tempdb_cache','200M'

Cache Name   Status Type          Config Value   Run Value
------------ ------ ------------ ------------ ------------
tempdb_cache Active Global,Mixed   200.00 Mb    200.00 Mb
                                 ------------ ------------
                               Total 200.00 Mb 200.00 Mb
================================================================================
Cache: tempdb_cache, Status: Active, Type: Global,Mixed
        Config Size: 200.00 Mb, Run Size: 200.00 Mb
        Config Replacement: strict LRU, Run Replacement: strict LRU
```

```
        Config Partition: 1, Run Partition: 1
IO Size     Wash Size   Config Size    Run Size   APF Percent
-------- ------------ ------------ ------------ -----------
2 Kb         40960 Kb     0.00 Mb      200.00 Mb     10
(return status = 0)
```

「sp_cacheconfig」(106 ページ) に説明されているように、インスタンス・オプションを使用してローカル・キャッシュのキャッシュ・サイズを増大させることもできます。

新規キャッシュに対する領域の割り付け Adaptive Server が要求したメモリの大きさを割り付けることができない場合、すべての使用可能なメモリを割り付け、エラー・メッセージを発行し、動的に割り付けることができるメモリの大きさがキロバイトで示します。

ただし、このメモリは Adaptive Server を再起動するまで割り付けられません。Adaptive Server はメモリが使用できないか、リソースの制限のために領域が不足していることを通知します。システム管理者は、これらの原因が一時的なものであることを確認する必要があります。この状態が解消されない場合は、それ以降の再起動に失敗することがあります。

たとえば、最大メモリが 700MB で、tempdb_cache が 100MB の場合、サーバの論理メモリの合計は 600MB となり、100MB を tempdb_cache に追加しようとすると、追加のメモリは最大メモリに収まります。ただし、90MB しか割り付けることができない場合は、その量が動的に割り付けられますが、設定ファイル内のキャッシュのサイズ・フィールドは、100MB に更新されます。その後、再起動すると、Adaptive Server はすべてのデータ・キャッシュに対して一度にメモリを取得するため、tempdb_cache は 100MB になります。

キャッシュの縮小 キャッシュ・サイズを縮小するときは、Adaptive Server を再起動します。たとえば、tempdb_cache のサイズを 100M に縮小するには、以下を使用します。インスタンス HR_INSTANCE 上で sp_cacheconfig 'tempdb_cache' を実行すると、以下のような出力が得られます。

```
sp_cacheconfig 'tempdb_cache', '100M'
go

Cache Name   Status Type           Config Value   Run Value
------------ ------ ------------ ------------ ------------
tempdb_cache Active Global,Mixed      100.00 Mb   200.00 Mb
------------ ------------      Total 100.00 Mb 200.00 Mb
================================================================================
Cache: tempdb_cache, Status: Active, Type: Global,Mixed
        Config Size: 100.00 Mb, Run Size: 200.00 Mb
        Config Replacement: strict LRU, Run Replacement: strict LRU
        Config Partition: 1, Run Partition: 1
IO Size    Wash Size   Config Size    Run Size    APF Percent
-------- ------------ ------------ ------------ -----------
2 Kb      40960 Kb      0.00 Mb       200.00 Mb       10
(return status = 0)
```

Adaptive Server を再起動し、HR_INSTANCE 上でコマンドを実行すると、次の出力が得られます。

```
sp_cacheconfig 'tempdb_cache'
go

Cache Name    Status Type        Config Value  Run Value
------------ ------ ------------ ------------ ------------
tempdb_cache Active Global,Mixed 100.00 Mb     100.00 Mb
                                 ------------ ------------
                                 Total 100.00 Mb 100.00 Mb
================================================================================
Cache: tempdb_cache, Status: Active, Type: Global,Mixed
        Config Size: 100.00 Mb, Run Size: 100.00 Mb
        Config Replacement: strict LRU, Run Replacement: strict LRU
        Config Partition: 1, Run Partition: 1
IO Size    Wash Size    Config Size   Run Size     APF Percent
-------- ------------- ------------ ------------ ------------
2 Kb        20480 Kb      0.00 Mb     100.00 Mb       10
(return status = 0)
```

名前付きキャッシュの削除　名前付きキャッシュを完全に削除するには、そのサイズを 0 にリセットします。

```
sp_cacheconfig 'tempdb_cache','0'
```

名前付きキャッシュにオブジェクトがバインドされ、Adaptive Server がエラー・メッセージを発行する場合は、名前付きキャッシュを削除できません。

名前付きキャッシュに複数の設定がある場合、設定ファイルのキャッシュに対応するエントリが削除されます。同様に、sysconfigures のキャッシュに対応するエントリも削除されます。インスタンスが次に再開されるときに、そのキャッシュが削除されます。キャッシュにグローバルまたはローカルの設定 1 つだけある場合、キャッシュ・エントリは設定ファイルからも sysconfigures からも削除されません。このエントリは、クラスタを再開するか、名前付きキャッシュの新しい設定を作成することによって削除されます。

インスタンス固有の設定を削除した場合、グローバル設定が存在すれば、名前付きキャッシュはグローバル設定に復元します。インスタンス SALES_INSTANCE 上で sp_cacheconfig を実行すると、以下のような出力が得られます。

```
sp_cacheconfig 'tempdb_cache', '0', 'instance SALES_INSTANCE'
go

Cache Name     Status   Type       Config Value   Run Value
------------ ------ ------------ ------------ ------------
tempdb_cache Active Global,Mixed     100.00 Mb    100.00 Mb
                                   ------------ ------------
                                  Total 100.00 Mb 100.00 Mb
================================================================================
Cache: tempdb_cache, Status: Active, Type: Global,Mixed
        Config Size: 100.00 Mb, Run Size: 100.00 Mb
        Config Replacement: strict LRU, Run Replacement: strict LRU
```

```
        Config Partition: 1, Run Partition: 1
IO Size   Wash Size     Config Size      Run Size   APF Percent
-------- ------------ ------------ ------------ -----------
2 Kb      20480 Kb       0.00 Mb     100.00 Mb     10
(return status = 0)
```

キャッシュ・タイプの変更 トランザクション・ログによってのみ使用するためにキャッシュを予約するには、キャッシュ・タイプを logonly に変更します。この変更は動的です。HR_INSTANCE で logonly キャッシュを作成するには、以下のように入力します。インスタンス HR_INSTANCE 上で sp_cacheconfig 'log_hr' を実行すると、以下のような出力が得られます。

```
sp_cacheconfig 'log_hr','logonly','instance HR_INSTANCE'


Cache Name           Status Type   Config Value    Run Value
------------------ ------ -------- ------------ ------------
HR_INSTANCE:log_hr Active Log Only 150.00 Mb       150.00 Mb
                                  ------------ ------------
                             Total 150.00 Mb 150.00 Mb
================================================================================
Cache: HR_INSTANCE:log_hr, Status: Active, Type: Log Only
        Config Size: 150.00 Mb, Run Size: 150.00 Mb
        Config Replacement: strict LRU, Run Replacement: strict LRU
        Config Partition: 1, Run Partition: 1
IO Size     Wash Size    Config Size Run Size     APF Percent

-------- ------------ ------------ ------------ -----------
2 Kb        30720 Kb     0.00 Mb      150.00 Mb         10
(return status = 0)
```

キャッシュ置換方式の設定 キャッシュがテーブルまたはインデックス専用で、システムが安定状態のときにそのキャッシュでバッファの置換がほとんど発生しない、またはまったく発生しない場合は、リラックス LRU (Least Recently Used) 置換方式を設定できます。この方式を使用すると、バッファの置換がほとんど発生しない、またはまったく発生しない場合にキャッシュのパフォーマンスが向上し、ほとんどのログ・キャッシュのパフォーマンスも向上します。リラックス置換方式は、以下のように設定します。

```
sp_cacheconfig 'log_sales','relaxed','instance SALES_INSTANCE'
go


Cache Name               Status Type   Config Value Run Value
------------------------ ------ ----- ------------ ------------
SALES_INSTANCE:log_sales Active Mixed 100.00 Mb     100.00 Mb
                                     ------------ ------------
                                Total 100.00 Mb 100.00 Mb
================================================================================
Cache: SALES_INSTANCE:log_sales, Status: Active, Type: Mixed
        Config Size: 100.00 Mb, Run Size: 100.00 Mb
        Config Replacement: relaxed LRU, Run Replacement: strict LRU
        Config Partition: 1, Run Partition: 1
```

```
IO Size     Wash Size Config Size     Run Size    APF Percent
-------- ------------ ------------ ------------ -----------
2 Kb         20480 Kb 0.00 Mb         100.00 Mb          10
(return status = 0)
```

注意 キャッシュ置換方式の設定は動的には行われません。Adaptive Server を再起動する必要があります。

複数のバッファ・プールの設定と使用

sp_poolconfig を使用して、複数のバッファ・プールを作成できます。

sp_poolconfig

説明

複数のバッファ・プールを作成します。

構文

sp_poolconfig *cache_name* [, "*mem_size* [P¥K¥M¥G]", "*config_pool*K"
[, "affected_poolK"]] [, 'instance instance_name']

パラメータ

- *cache_name* − 既存のデータ・キャッシュの名前。
- *mem_size* − 作成するメモリ・プールのサイズ、または指定した I/O サイズの既存のプールの新しい総サイズ。プールの最小サイズは、論理サーバ・ページの 256 倍です。サイズ単位として、P (ページ)、K (キロバイト)、M (メガバイト)、G (ギガバイト) を指定できます。デフォルトの単位は、キロバイトです。
- *config_pool* − メモリが割り付けられる、または削除されるメモリ・プール内の I/O サイズ。有効な I/O サイズは、論理ページ・サイズの倍数で、論理ページ・サイズの大きさの最大 8 倍までです。
- *affected_pool* − メモリの割り付けを解除する、またはウォッシュ・サイズおよびプリフェッチ制限などのプールの属性が変更されるメモリ・プール内で実行される I/O のサイズ。*affected_pool* が指定されなければ、メモリは最小の論理ページ・サイズのメモリ・プールから使用されます。

例

名前付きキャッシュに対する 4K プールの作成　インスタンス SALES_INSTANCE 上で sp_poolconfig 'tempdb_cache' を実行すると、以下のような出力が得られます。

```
sp_poolconfig 'tempdb_cache','25M','4K'
go
Cache Name    Status Type        Config Value  Run Value
------------ ------ ------------ ------------ ------------
```

```
tempdb_cache Active Global,Mixed   100.00 Mb    100.00 Mb
(1 row affected)
                                   ------------ ------------
                               Total 100.00 Mb 100.00 Mb
===============================================================================
Cache: tempdb_cache, Status: Active, Type: Global,Mixed
        Config Size: 100.00 Mb, Run Size: 100.00 Mb
        Config Replacement: strict LRU, Run Replacement: strict LRU
        Config Partition: 1, Run Partition: 1
IO Size    Wash Size    Config Size   Run Size   APF Percent
-------- ------------- ------------ ------------ -----------
2 Kb      15360 Kb       0.00 Mb      75.00 Mb      10
4 Kb       5120 Kb      25.00 Mb     25.00 Mb      10
(return status = 0)
```

ローカル・キャッシュのプール設定の作成 名前付きキャッシュ 'log_hr' の 8K プールを作成できます。インスタンス HR_INSTANCE 上で sp_poolconfig 'tempdb_cache' を実行すると、以下のような出力が得られます。

```
sp_poolconfig 'log_hr','50M','8K','instance HR_INSTANCE'
go

Cache Name           Status Type    Config Value   Run Value
------------------ ------ -------- ------------ ------------
HR_INSTANCE:log_hr Active Log Only   150.00 Mb    150.00 Mb
(1 row affected)
                                   ------------ ------------
                               Total 150.00 Mb 150.00 Mb
===============================================================================
Cache: HR_INSTANCE:log_hr, Status: Active, Type: Log Only
        Config Size: 150.00 Mb, Run Size: 150.00 Mb
        Config Replacement: strict LRU, Run Replacement: strict LRU
        Config Partition: 1, Run Partition: 1
IO Size    Wash Size    Config Size    Run Size    APF Percent
-------- ------------- ------------ ------------ -----------
2 Kb        20480 Kb      0.00 Mb       100.00 Mb     10
8 Kb        10240 Kb    50.00 Mb         50.00 Mb    10
(return status = 0)
```

注意 グローバル・キャッシュの場合、バッファ・プールのインスタンス固有の設定はありません。インスタンス・オプションは、プール設定が必要なローカル・キャッシュ設定を検出するために使用されます。

バッファ・プール間でのメモリの移動

新しい 8K プールを作成し、デフォルト・プールからではなく、4K プールからメモリを使用するには、以下のように実行します。

```
sp_poolconfig 'tempdb_cache','8M','8K','4K'
go
sp_poolconfig 'tempdb_cache'
go

Cache Name                  Status Type   Config Value   Run Value
--------------------------- ------ ----- ------------ ------------
tempdb_cache        Active Global,Mixed   100.00 Mb      100.00 Mb
(1 row affected)
                                          ------------ ------------
                                  Total 100.00 Mb 100.00 Mb
===================================================================================
Cache: tempdb_cache, Status: Active, Type: Global,Mixed
       Config Size: 100.00 Mb, Run Size: 100.00 Mb
       Config Replacement: strict LRU, Run Replacement: strict LRU
       Config Partition: 1, Run Partition: 1
IO Size    Wash Size    Config Size    Run Size   APF Percent
-------- ------------ ------------ ------------ -----------
8 Kb        1632 Kb      8.00 Mb     8.00 Mb         10
2Kb        15360 Kb     0.00 Mb      75.00 Mb        10
4 Kb       3480 Kb     17.00 Mb      17.00 Mb        10
(return status = 0)
```

プールのウォッシュ・サイズの変更

ウォッシュ・サイズとは、メモリ・プールに対して Adaptive Server がダーティ・ページをディスクに書き込むキャッシュ内の位置です。

```
sp_poolconfig cache_name, 'affected_poolK ', 'wash=size[P|K|M|G]'
        [, instance 'instancename']
```

名前付きキャッシュ “log_hr” の 8K プールのウォッシュ・サイズを 12480K に変更するために、インスタンス HR_INSTANCE 上で sp_poolconfig 'log_hr' を実行すると、以下のような出力が得られます。

```
sp_poolconfig 'log_hr','8K','wash=12480K','instance HR_INSTANCE'
go

Cache Name          Status Type     Config Value   Run Value
------------------ ------ -------- ------------ ------------
HR_INSTANCE:log_hr Active Log Only   150.00 Mb   150.00 Mb
(1 row affected)
                                   ------------ ------------
                                Total 150.00 Mb 150.00 Mb
===================================================================================
Cache: HR_INSTANCE:log_hr, Status: Active, Type: Log Only
```

```
        Config Size: 150.00 Mb, Run Size: 150.00 Mb
        Config Replacement: strict LRU, Run Replacement: strict LRU
        Config Partition: 1, Run Partition: 1
IO Size     Wash Size   Config Size  Run Size     APF Percent
-------- ------------- ------------ ------------ -----------
2 Kb        20480 Kb       0.00 Mb   100.00 Mb        10
8 Kb        12480 Kb      50.00 Mb   50.00 Mb         10
(return status = 0)
```

プールのローカル非同期プリフェッチ率の変更

ローカル非同期プリフェッチとは、プールのバッファのうち、非同期プリフェッチでキャッシュに読み込まれたが、まだ使用されていないバッファを保持するために使用できるバッファの比率です。プールの非同期プリフェッチ率を変更するには、以下のように実行します。

sp_poolconfig *cache_name*, "*affected_poolK* ",
"local async prefetch limit=*percent* "

名前付きキャッシュ log_sales のローカル同期プリフェッチを変更するために、インスタンス SALES_INSTANCE 上で sp_poolconfig 'log sales' を実行すると、以下の出力が得られます。

```
sp_poolconfig 'log_sales','2K','local async prefetch limit=20','instance
          SALES_INSTANCE'
go
Cache Name                Status Type   Config Value Run Value
------------------------- ------ ----- ------------ ------------
SALES_INSTANCE:log_sales Active Mixed   100.00 Mb     100.00 Mb
(1 row affected)
                                      ------------ ------------
                                   Total 100.00 Mb 100.00 Mb
================================================================================
Cache: SALES_INSTANCE:log_sales, Status: Active, Type: Mixed
        Config Size: 100.00 Mb, Run Size: 100.00 Mb
        Config Replacement: relaxed LRU, Run Replacement: relaxed LRU
        Config Partition: 1, Run Partition: 1
IO Size    Wash Size    Config Size   Run Size    APF Percent
-------- ------------- ------------ ------------ -----------
2 Kb       20480 Kb       0.00 Mb     100.00 Mb        20
(return status = 0)
```

全ページのページ分割時における非同期書き込み

ページ分割時に dump database コマンドがアクティブである場合、Cluster Edition は分割によって生成される新しいページで同期書き込みを発行します。

バッファ・プールの削除

サイズを 0 に設定することにより、バッファ・プールを削除できます。このプールからのメモリは、デフォルト・プールに追加されます。インスタンス SALES_INSTANCE で、名前付きキャッシュ tempdb_cache からプール 4k を削除するには、以下の手順に従ってください。

```
sp_poolconfig 'tempdb_cache','0','4K'
go
sp_poolconfig 'tempdb_cache'
go

Cache Name                    Status Type Config Value Run Value
----------------------------- ------ ----- ------------ ------------
tempdb_cache            Active Global,Mixed  100.00 Mb    100.00 Mb
(1 row affected)
                                           ------------ ------------
                                    Total 100.00 Mb 100.00 Mb
===============================================================================
Cache: tempdb_cache, Status: Active, Type: Global,Mixed
        Config Size: 100.00 Mb, Run Size: 100.00 Mb
        Config Replacement: strict LRU, Run Replacement: strict LRU
        Config Partition: 1, Run Partition: 1
IO Size    Wash Size   Config Size    Run Size    APF Percent
-------- ------------ ------------ ------------ -----------
8Kb       1632 Kb       8.00 Mb     8.00 Mb     10
2Kb       18840 Kb      0.00 Mb    92.00 Mb     10
(return status = 0)
```

オブジェクトの名前付きキャッシュへのバインド

sp_bindcache は、データベース、テーブル、インデックス、テキスト・オブジェクト、イメージ・オブジェクトをキャッシュに割り当てます。エンティティをキャッシュにバインドするには、次の条件を満たす必要があります。

- 名前付きキャッシュが存在していて、そのステータスが “Active” であること。
- データベースまたはデータベース・オブジェクトが存在すること。
- 格納されたデータベースからのテーブル、インデックス、オブジェクトのみをバインドできること。
- トランザクション・ログ・テーブル syslogs を含むシステム・テーブルをバインドするには、データベースはシングルユーザ・モードであること。
- データベースをバインドするには、master データベースから実行すること。

- データベース、ユーザ・テーブル、インデックス、テキスト・オブジェクト、イメージ・オブジェクトは、“Mixed” タイプのキャッシュにバインドすること。syslogs テーブルは、“Log Only” タイプのキャッシュにのみバインドできる。
- キャッシュにオブジェクトをバインドするには、オブジェクトまたはデータベースを所有するか、システム管理者ステータスであること。
- キャッシュへのオブジェクトのバインドは動的に行われること。

注意 ローカル・システム・テンポラリ・データベースでのキャッシュのバインドやバインド解除は動的ではなく、バインドを有効にするためには所有者インスタンスを再起動する必要があります。その他のテンポラリ・データベースでのキャッシュのバインドとバインド解除は、グローバル・システム・テンポラリ・データベースを含めて、動的に行われます。

オブジェクトのバインドのための構文

構文

```
sp_bindcache cache_name, dbname
        [, [owner. ] tablename [, indexname |”text only'] ]
```

パラメータ

owner − テーブルが “dbo” により所有されている場合は、任意です。

例

データベース SALES を名前付きキャッシュ sales_cache にバインドするには、以下のように入力します。

```
sp_bindcache 'sales_cache', 'SALES'
```

使用法

キャッシュのバインドは、クラスタ全体に有効です。インスタンス固有キャッシュのバインドはありません。オブジェクトをローカル・キャッシュにバインドする場合、インスタンスに対して設定されるキャッシュを持つインスタンスはそのキャッシュを使用し、その他のすべてのインスタンスは default data cache を使用します。

注意 sp_bindcache の完全なマニュアルについては、『リファレンス・マニュアル：プロシージャ』を参照してください。

バインドされたキャッシュに関する情報の取得

キャッシュ名を指定して sp_helpcache を実行すると、そのキャッシュと、キャッシュにバインドされているエンティティに関する情報が表示されます。

次に例を示します。

```
sp_helpcache 'sales_cache'
--------------------------
Cache Name         Config Size Run Size   Overhead

------------------ ----------- ---------- ----------
sales_cache:SALES_INSTANCE 50.00 Mb 50.00 Mb 6.39 Mb
(1 row affected)
--------------Cache Binding Information: --------
Cache Name Entity Name Type Index Name
Status
---------- ----------- ---- ----------
sales_cache SALES     database
V
(return status = 0)
```

sp_helpcache 構文の詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。

キャッシュ・バインドの解除

キャッシュのバインドを解除するためのコマンドは 2 種類あります。

- sp_unbindcache はキャッシュから 1 つのエンティティのバインドを解除します。
- sp_unbindcache_all は、キャッシュにバインドされているすべてのオブジェクトのバインドを解除します。

構文

sp_unbindcache dbname [, [owner.] tablename
[, indexname | “text only”]]

例

データベース 'sales' のキャッシュのバインドを解除するには、以下のように実行します。

```
sp_unbindcache 'SALES'
```

使用法

- オブジェクトに対するキャッシュのバインドを解除すると、現在のメモリ内にあるすべてのページがそのキャッシュからクリアされます。
- キャッシュにバインドされたシステムまたはリモートのローカル・テンポラリ・データベースがある場合、名前付きキャッシュ上で sp_unbindcache_all は実行できません。代わりに、sp_unbindcache を使用してこれらのデータベースをそれぞれバインド解除してから、sp_unbindcache_all を実行します。

設定ファイルの変更

設定ファイルの名前付きキャッシュ・セクションには、インスタンス情報が含まれています。グローバル・キャッシュのキャッシュ・セクションは、ノンクラスタ Adaptive Server 出力に類似しています。

以下の定義は、ノンクラスタ Adaptive Server 環境の設定ファイルから取得されます。

```
[Named Cache:tempdb_cache]
    cache size = 100M
    cache status = mixed cache
    cache replacement policy = DEFAULT
    local cache partition number = DEFAULT

[2K I/O Buffer Pool]
    pool size = DEFAULT
    wash size = DEFAULT
    local async prefetch limit = DEFAULT
```

ローカルの名前付きキャッシュのフォーマット

以下は、ローカルの名前付きキャッシュのフォーマットを示します。

```
[Named Cache:log_sales]
    [Instance: SALES_INSTANCE]
    cache size = 100M
    cache status = mixed cache
    cache replacement policy = DEFAULT
    local cache partition number = DEFAULT

[2K I/O Buffer Pool]
    [Instance: SALES_INSTANCE]
    pool size = DEFAULT
    wash size = DEFAULT
    local async prefetch limit = DEFAULT
```

設定ファイルでグローバル定義が最初に宣言され、次にローカル定義が宣言されるようにする必要があります。そうしないと、サーバは起動しません。たとえば、名前付きキャッシュのインスタンス固有のプール設定は、対応するインスタンス固有のキャッシュ設定がない場合は、許可されません。以下は、間違った例です。

```
[Named Cache:tempdb_cache]
cache size = 100M
cache status = mixed cache
cache replacement policy = DEFAULT
local cache partition number = DEFAULT
[2K I/O Buffer Pool]
pool size = DEFAULT
wash size = DEFAULT
local async prefetch limit = DEFAULT
```

```
[instance: SALES_INSTANCE]
pool size = DEFAULT
wash size = 40960K
local async prefetch limit = DEFAULT
```

ローカル・キャッシュ・エントリの追加行

ローカル・キャッシュとバッファ・プール定義には、追加行 ([Instance: SALES_INSTANCE]) があり、設定がインスタンス SALES_INSTANCE に属することを示します。名前付きキャッシュがグローバル設定とローカル設定の両方を持つ場合、設定ファイルのキャッシュ・セクションには以下のように示されます。

```
[Named Cache:tempdb_cache]
    cache size = 100M
    cache status = mixed cache
    cache replacement policy = DEFAULT
    local cache partition number = DEFAULT
    [Instance: SALES_INSTANCE]
    cache size = 150M
    cache status = mixed cache
    cache replacement policy = DEFAULT
    local cache partition number = DEFAUL

[2K I/O Buffer Pool]
    pool size = DEFAULT
    wash size = DEFAULT
    local async prefetch limit = DEFAULT
    [Instance: SALES_INSTANCE]
    pool size = DEFAULT
    wash size = 40960k
    local async prefetch limit = DEFAULT
```

グローバル設定を持つ削除された名前付きキャッシュ

名前付きキャッシュが削除される場合、キャッシュ設定エントリは以下のようになります。

```
[Named Cache: tempdb_cache]
    cache size = 100M
    cache status = deleted
    cache replacement policy = DEFAULT
    local cache partition number = DEFAULT

[2K I/O Buffer Pool]
    pool size = DEFAULT
    wash size = DEFAULT
    local async prefetch limit = DEFAULT
```

ローカル設定を持つ名前付きキャッシュ

名前付きキャッシュがローカルで設定される場合、キャッシュ・セクションのエントリは以下のようになります。

```
[Named Cache: tempdb_cache]
[Instance: SALES_INSTANCE]
    cache size = 100M
    cache status = deleted
    cache replacement policy = DEFAULT
    local cache partition number = DEFAULT

[2K I/O Buffer Pool]
    [Instance: SALES_INSTANCE]
    pool size = DEFAULT
    wash size = DEFAULT
    local async prefetch limit = DEFAULT
```

有効な設定を持つ削除済みのエントリ

設定ファイルには、有効な設定を少なくとも 1 つ持つ削除済みのエントリを含めることはできません。したがって、以下のようなキャッシュ・セクションのエントリを含めることはできません。

```
[Named Cache: tempdb_cache]
    cache size = 100M
    cache status = mixed cache
    cache replacement policy = DEFAULT
    local cache partition number = DEFAULT
    [Instance: SALES_INSTANCE]
    cache size = 150M
    cache status = deleted
    cache replacement policy = DEFAULT
    local cache partition number = DEFAULT

[2K I/O Buffer Pool]
    pool size = DEFAULT
    wash size = DEFAULT
    local async prefetch limit = DEFAULT
    [Instance: SALES_INSTANCE]
    pool size = DEFAULT
    wash size = 40960K
    local async prefetch limit = DEFAULT
```

グローバル設定が存在する場合のローカル設定の作成

グローバル設定が存在するときに名前付きキャッシュのローカル設定を作成する場合、すべてのプール・エントリは、ローカル設定に対して複写されます。たとえば、次のようなグローバル設定があるとします。

```
[Named Cache: tempdb_cache]
    cache size = 100M
    cache status = mixed cahe
    cache replacement policy = DEFAULT
    local cache partition number = DEFAULT

[2K I/O Buffer Pool]
    pool size = DEFAULT
    wash size = DEFAULT
    local async prefetch limit = DEFAULT

[4K I/O Buffer Pool]
pool size = 25.0000m
wash size = DEFAULT
local async prefetch limit = DEFAULT
```

このグローバル設定でサイズ 120M のローカル設定を作成する場合、設定ファイルのキャッシュ・セクションは以下のようになります。

```
[Named Cache:tempdb_cache]
    cache size = 100M
    cache status = mixed cache
    cache replacement policy = DEFAULT
    local cache partition number = DEFAULT
    [Instance:SALES_INSTANCE]
    cache size = 120M
    cache status = mixed cache
    cache replacement policy = DEFAULT
    local cache partition number = DEFAULT


[2K I/O Buffer Pool]
    pool size = DEFAULT
    wash size = DEFAULT
    local async prefetch limit = DEFAULT
    [Instance:SALES_INSTANCE]
    pool size = DEFAULT
    wash size = DEFAULT
    local async prefetch limit = DEFAULT

[4K I/O Buffer Pool]
    pool size = 25.0000m
    wash size = DEFAULT
    local async prefetch limit = DEFAULT
    [Instance:SALES_INSTANCE]
    pool size = 25.0000m
    wash size = DEFAULT
    local async prefetch limit = DEFAULT
```

制限事項

- 名前付きキャッシュにローカル設定のみがあり、それにオブジェクトがバインドされている場合、グローバル・キャッシュ設定を作成することはできません。エラー 19817 が表示されます。次に例を示します。

```
sp_cacheconfig 'SALES', '200M', 'instance SALES_INSTANCE'
go
sp_bindcache 'SALES','SALES_DB'
go
sp_cacheconfig 'SALES', '300m'
go
-------
Error 19817, "New configuration on cache 'SALES' is not permitted."
```

  このような状況を避けるには、キャッシュ “SALES” にバインドされたすべてのオブジェクトのバインドを解除します。キャッシュ “SALES” の新しい設定を作成し、オブジェクトをもう一度バインドします。

- 物理ロックのデッドロックを回避するには、Cluster Edition のインデックス・ページで大規模なプール・サポートを使用しないようにします。

第 8 章

テンポラリ・データベースの使用

この章ではローカル・テンポラリ・データベースとグローバル・テンポラリ・データベースについて取り上げ、その作成と管理方法、ログインやアプリケーションを 1 つのテンポラリ・データベースや複数のテンポラリ・データベースのグループにバインドする方法について説明します。また、プライベート・デバイスについても取り上げ、その作成と管理方法、それをローカル・ユーザ・テンポラリ・データベースに使用する方法についても説明します。

テンポラリ・データベースは、テンポラリ・テーブルやその他の一時的な作業領域に必要な記憶領域を提供します。Cluster Edition はローカル・テンポラリ・データベースとグローバル・テンポラリ・データベースの両方をサポートします。ローカル・テンポラリ・データベースは所有インスタンスだけでアクセスでき、主に、# テーブル、ワーク・テーブル、フェイク・テーブルなどのセッション固有のプライベート・テンポラリ・オブジェクトを格納するために使用されます。グローバル・テンポラリ・データベースは、クラスタのすべてのインスタンスによってアクセスでき、現在のセッションを超えて存在するテンポラリ・オブジェクトを格納するために使用されます。

Cluster Edition は以下をサポートしています。

- ローカル・システム・テンポラリ・データベース
- ローカル・ユーザ・テンポラリ・データベース
- グローバル・ユーザ・テンポラリ・データベース
- グローバル・システム・テンポラリ・データベース (dbid が 2)

テンポラリ・データベースは、次のデータベース・オプション・セットを持つ model データベースから継承されます。

- select into/bulkcopy
- trunc log on chkpt

guest ユーザはテンポラリ・データベースに自動的に追加され、すべてのユーザに create table パーミッションが付与されます。

テンポラリ・データベースのタイプ

ローカル・テンポラリ・データベース

ローカル・テンポラリ・データベースには、次の 2 種類があります。

- ローカル・ユーザ・テンポラリ・データベース
- ローカル・システム・テンポラリ・データベース

Cluster Edition を使用すると、クラスタの各インスタンスに対してテンポラリ・データベースを作成できます。インスタンス特有のテンポラリ・データベースは、ローカル・テンポラリ・データベースと呼ばれます。ローカル・テンポラリ・データベースを所有するインスタンスは、所有者インスタンスと呼ばれます。

各ローカル・テンポラリ・データベースは特定のインスタンスにバインドされ、そのインスタンスからのみアクセスできます。クラスタの各インスタンスにローカル・システム・テンポラリ・データベースを作成する必要があります。ローカル・ユーザ・テンポラリ・データベースの作成は任意です。

ローカル・ユーザ・テンポラリ・データベース

各インスタンスに対して複数のローカル・ユーザ・テンポラリ・データベースを作成でき、アプリケーションまたはログインを個々のローカル・ユーザ・テンポラリ・データベースまたは複数のテンポラリ・データベースのグループにバインドできます。

ローカル・システム・テンポラリ・データベース

ローカル・システム・テンポラリ・データベースは、各インスタンスに必要なデフォルトのテンポラリ・データベースです。クラスタが設定されたとき、または新しいインスタンスがクラスタに追加されたときにローカル・システム・テンポラリ・データベースを設定します。インスタンスは、インスタンスのローカル・ユーザ・テンポラリ・データベースを作成し、使用しないかぎり、このデータベースのすべてのセッション固有のテンポラリ・オブジェクト (# テーブルやワーク・テーブルなど) を格納します。ローカル・テンポラリ・データベースは共有記憶領域に作成します。「テンポラリ・データのプライベート・デバイスの使用」(140 ページ) を参照してください。

ノンクラスタ Adaptive Server 環境では、システム・テンポラリ・データベース tempdb (dbid 2) は default テンポラリ・データベース・グループに追加されます。Cluster Edition では、ローカル・システム・テンポラリ・データベースはインスタンスの default グループの一部ではありません。ローカル・システム・テンポラリ・データベースは、現在のインスタンスの default グループが空であり、他のバインドが指定されていない場合にのみ、セッションに割り当てられます。

注意　Cluster Edition の場合、セッションに割り当てられたデフォルトのテンポラリ・データベースは、システムの tempdb (dbid が 2) ではなく、インスタンスのローカル・テンポラリ・データベースになります。ノンクラスタ環境で、デフォルトで割り当てられたテンポラリ・データベースをアクションによってシステムの tempdb (dbid が 2) と想定し、割り当てられたローカル・テンポラリ・データベースにそれらのアクションが適用されるようにアプリケーションを修正する必要があります。

たとえば、ノンクラスタ Adaptive Server では、アプリケーションはデフォルトのテンポラリ・データベースのログを、次のコマンドを使用してトランケートします。

```
dump tran tempdb with truncate_only
```

Cluster Edition では、デフォルトで割り当てられるテンポラリ・データベースのログを次のコマンドでトランケートするように、アプリケーションに変更を加えます。

```
declare @tempdbname varchar(30)
select @tempdbname = db_name(@@tempdbid)
dump tran @tempdbname with truncate_only
```

セッションのテンポラリ・データベース用の個別のユーザ・ログ・キャッシュ

Cluster Edition は、ユーザ・ローカル・テンポラリ・データベース、システム・ローカル・テンポラリ・データベース、グローバル・テンポラリ・データベース (tempdb) などの複数のテンポラリ・データベースをサポートします。

Cluster Edition は、セッションのテンポラリ・データベース用のみの個別のユーザ・ログ・キャッシュをサポートしますが、他のテンポラリ・データベース用についてはサポートしません。

セッションのテンポラリ・データベースには次のものがあります。

- ユーザ・ローカル・テンポラリ・データベース
- システム・ローカル・テンポラリ・データベース
- クラスタが tempdb 設定モードの場合は、グローバル・テンポラリ・データベース

tempdb 設定時に、ローカル・システムまたはローカル・ユーザ・テンポラリ・データベースが存在しない場合、セッションのデフォルトのテンポラリ・データベースはシステムの tempdb (データベース ID は 2) になります。設定時には、システムの tempdb 用の別のユーザ・ログ・キャッシュを使用することはできません。

グローバル・テンポラリ・データベース

マスタ・デバイスが作成されると、システムによってグローバル・システム・テンポラリ・データベース (dbid 2) が自動的に作成されます。グローバル・テンポラリ・データベースはさらに追加で作成できます。

グローバル・テンポラリ・データベースにはクラスタの任意のインスタンスからアクセスでき、同じまたは別のインスタンスの複数のセッションにわたって共有できるテンポラリ・オブジェクトを作成できます。グローバル・テンポラリ・データベースは特定のセッションに割り当てることができないため、グローバル・テンポラリ・データベースの中で # テーブルやワーク・テーブルなどのテンポラリ・オブジェクトを作成することはできません。

グローバル・テンポラリ・データベースは、次の点を除き、通常のユーザ・データベースと同じです。異なる点は、クラスタが起動するたびに作成し直さなければならないことです。グローバル・テンポラリ・データベースは、標準的なロギング、ログのフラッシュ、I/O 動作、実行時のロールバック機能を備えています。これらは主に、Adaptive Server の過去のバージョンとの下位互換性を保つためにサポートされています。

グローバル・テンポラリ・データベースのすべてのオブジェクトは、クラスタを停止したときに失われます。ただし、インスタンスが失敗した場合、インスタンス・フェールオーバ・リカバリ手順によってグローバル・テンポラリ・データベースのオブジェクトも同様に回復するため、失敗したインスタンスのセッション中に作成されたオブジェクトを引き続き、回復されたインスタンスで使用できます。グローバル・テンポラリ・データベースは、クラスタのインスタンスが 1 つでもアクティブであるかぎり、共有可能なテンポラリ・ワーク・スペースを引き続き存続させます。

注意 Adaptive Server 12.5.0.3 以降に開発したアプリケーションでは、ユーザが作成したテンポラリ・データベース内に共有可能なテーブルを作成できます。それらの既存アプリケーションを Cluster Edition でも動作させるには、共有可能なテーブルを含むユーザ作成データベースを削除して、グローバル・テンポラリ・データベースを同じ名前で作り直す必要があります。

要約

表 8-1 は、これらのデータベースの基本的な特徴をまとめています。

表 8-1: テンポラリ・データベースの特徴

機能	ローカル・システム・テンポラリ・データベース	ローカル・ユーザ・テンポラリ・データベース	グローバル・ユーザ・テンポラリ・データベース	グローバル・システム・テンポラリ・データベース
サポートされているテンポラリ・データベース・オブジェクト	セッション固有のテンポラリ・オブジェクトと通常のデータベース・オブジェクトの両方。	セッション固有のテンポラリ・オブジェクトと通常のデータベース・オブジェクトの両方。	通常のデータベース・オブジェクトのみ。セッション固有のテンポラリ・オブジェクトはサポートしません。	通常のデータベース・オブジェクトのみ。セッション固有のテンポラリ・オブジェクト*はサポートしません。
リカバリ	所有インスタンスが再起動するときに作成し直されます。	所有インスタンスが再起動するときに作成し直されます。	クラスタを再起動したときに作成し直され、インスタンスが失敗したときにトランザクションが回復されます。	クラスタを再起動したときに作成し直され、インスタンスが失敗したときにトランザクションが回復されます。
アクセシビリティ	所有インスタンスのみからアクセスできますが、インスタンスは共有デバイス上に作成し、他のインスタンスから作成または削除できなければなりません。	所有インスタンスのみからアクセス可能。	任意のインスタンスからアクセス可能。	任意のインスタンスからアクセス可能。
作成	各インスタンスに対して 1 つ (必須)、ユーザが作成。	各インスタンスに対して 0 個以上、ユーザが作成。	クラスタに対して 0 個以上、ユーザが作成。	クラスタに対して 1 つずつ、システム生成 (dbid = 2)。
バインドの許可	許可されない。	許可される。	許可されない。	許可されない。
記憶領域 (共有デバイスまたはプライベート・デバイス)	共有記憶領域のみ。	共有記憶領域とローカル記憶領域の両方。	共有記憶領域のみ。	共有記憶領域のみ。

* tempdb 設定モード中に、グローバル・システム・テンポラリ・データベースは、ブート・コーディネータに対してローカル・システム・テンポラリ・データベースのように動作し、セッション固有のテンポラリ・オブジェクトと通常のデータベース・オブジェクトの両方をサポートします。

テンポラリ・データベースの作成

Cluster Edition では、各クラスタにグローバル・システム・テンポラリ・データベースがあり、クラスタの各インスタンスにローカル・システム・テンポラリ・データベースがあります。他のテンポラリ・データベースは任意です。グローバル・システム・テンポラリ・データベースは、Adaptive Server をインストールするときに作成されます。この項では、他のテンポラリ・データベースを作成する方法を説明します。

同じようなサイズのテンポラリ・データベースを作成し、アプリケーションがテンポラリ・スペースの要件に関係なく各インスタンスにアクセスできるようにすることをおすすめします。

ローカル・システム・テンポラリ・データベースの作成

クラスタの初期起動時と、それ以降クラスタにインスタンスを追加した場合は、必ずローカル・システム・テンポラリ・データベースを各インスタンスに作成します。ローカル・システム・テンポラリ・データベースは共有デバイスに作成します。どのインスタンスにおいても、ローカル・システム・テンポラリ・データベースの作成または削除は可能ですが、アクセスできるのは所有インスタンスからのみです。

ローカル・システム・テンポラリ・データベースは、Adaptive Server プラグインまたは sybcluster を使用して作成します。詳細については、プラットフォームの『インストール・ガイド』を参照してください。

ローカル・ユーザ・テンポラリ・データベースとグローバル・ユーザ・テンポラリ・データベースの作成

ローカル・ユーザ・テンポラリ・データベースとグローバル・ユーザ・テンポラリ・データベースはいつでも作成できます。

ローカル・ユーザ・テンポラリ・データベースは所有インスタンスから作成します。たとえば、ase1 上でローカル・テンポラリ・データベース local_temp1 を作成するには、次のように入力します。

```
create temporary database local_temp1
    for instance ase1 on default = 50
```

グローバル・ユーザ・テンポラリ・データベースは、クラスタの任意のインスタンスから作成できます。たとえば、ase1 上でグローバル・テンポラリ・データベース global_tempdb1 を作成するには、ase1 にログインし、次のように入力します。

```
create global temporary database global_tempdb1
    on default = 50
```

注意　create database の for instance *instance_name* 句は任意です。この句を含めない場合、Adaptive Server は現在のインスタンスのテンポラリ・データベースを作成します。完全な構文の説明は、『リファレンス・マニュアル：コマンド』を参照してください。

テンポラリ・データベースへのユーザとアプリケーションのバインド

sp_tempdb を使用して、ログインおよびアプリケーションをテンポラリ・データベースやテンポラリ・データベース・グループにバインドしたり、ノンクラスタ Adaptive Server と Cluster Edition の両方にテンポラリ・データベース・グループを作成したりできます。Cluster Edition では、ローカル・テンポラリ・データベースとグローバル・テンポラリ・データベースの両方をサポートするため、ログインやアプリケーションをローカル・ユーザ・テンポラリ・データベースまたはテンポラリ・データベース・グループにバインドできますが、グローバル・テンポラリ・データベースにはバインドできません。

default テンポラリ・データベース・グループはシステムが生成し、常に存在します。これは、sp_tempdb を使用してローカル・テンポラリ・データベースを追加しないかぎり、空です。

注意　Cluster Edition では、システム・テンポラリ・データベース tempdb (dbid 2) はグローバル・システム・テンポラリ・データベースで、default グループのメンバではありません。この動作は、ノンクラスタ Adaptive Server 内のシステム・テンポラリ・データベース tempdb (dbid 2) とは異なります。ノンクラスタ Adaptive Server では、システム・テンポラリ・データベース tempdb はデフォルトでは default グループのメンバです。

sp_tempdb ユーザ・インタフェースにはオプション unbindall_gr があります。これは、ノンクラスタ Adaptive Server または Cluster Edition のいずれかで指定されたグループとのすべてのバインドを削除します。さらに、unbind オプションには、Cluster Edition に固有の *instance_name* パラメータがあります。

テンポラリ・データベース・グループの作成と管理

テンポラリ・データベース・グループの作成と管理は、Adaptive Server のすべてのエディションの場合と同じです。

注意 Cluster Edition では、ローカル・システム・テンポラリ・データベースは、default グループのメンバではありません。

- テンポラリ・データベース・グループを作成するには、sp_tempdb create を使用します。たとえば、tempdbgroup1 という名前のテンポラリ・データベースを作成する場合、以下のように入力します。

  ```
  sp_tempdb "create", "tempdbgroup1"
  ```

- テンポラリ・データベース・グループを削除するには、sp_tempdb drop を使用します。たとえば、tempdbgroup1 という名前のテンポラリ・データベース・グループを削除する場合、以下のように入力します。

  ```
  sp_tempdb "drop", "tempdbgroup1"
  ```

- テンポラリ・データベース・グループにデータベースを追加するには、sp_tempdb add を使用します。たとえば、データベース local_tempdb1 を tempdbgroup1 というテンポラリ・データベース・グループに追加するには、以下のように入力します。

  ```
  sp_tempdb "add", "local_tempdb1", "tempdbgroup1"
  ```

- テンポラリ・データベース・グループからデータベースを削除するには、sp_tempdb remove を使用します。たとえば、データベース local_tempdb1 を tempdbgroup1 というテンポラリ・データベース・グループから削除するには、以下のように入力します。

  ```
  sp_tempdb "remove", "local_tempdb1", "tempdbgroup1"
  ```

sp_tempdb の構文と使用法の詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。

バインド対象

バインドには次のような条件があります。

- ユーザのログインまたはアプリケーションは複数のローカル・ユーザ・テンポラリ・データベースにバインドできますが、1 インスタンスあたり 1 つのデータベースに限られます。

  たとえば、「sa」を「ase1」の local_tempdb1 にバインドし、「ase2」の local_tempdb2 にバインドできますが、「sa」を「ase1」の local_tempdb3 にバインドすることはできません。

- ユーザ・ログインまたはアプリケーションは 1 つのデータベース・グループにのみバインドできます。テンポラリ・データベース・グループはすべてのインスタンスから参照できます。

  たとえば、isql を default グループにバインドする場合、isql を別のテンポラリ・データベース・グループにバインドすることはできません。

- ユーザ・ログインまたはアプリケーションは個々のテンポラリ・データベースまたはテンポラリ・データベース・グループにバインドできますが、両方にバインドすることはできません。

  たとえば、「sa」を「ase1」上の local_tempdb1 にバインドした場合、「sa」を default グループにバインドすることはできません。

セッションのバインドが行われる方法

セッションが開始されると、Adaptive Server はバインドが適切であるかどうかを確認します。複数のバインドが適切である場合、Cluster Edition は以下のアルゴリズムに基づいて、どのバインドを使用するかを判別します。1 セッションあたり 1 つのバインドのみがサポートされます。

1　現在のログインにバインドが存在する場合、そのバインドが使用されます。それ以外の場合、Cluster Edition は現在のアプリケーションのバインドを検索します。両方が見つかった場合は、ログインのバインドが使用されます。ログインのバインドは、アプリケーションのバインドよりも優先されます。

  たとえば、ユーザ「sa」がインスタンス「ase1」の tempdb1 にバインドされ、isql が「ase1」の tempdb2 にバインドされている場合、ユーザ「sa」が isql を使用して「ase1」のセッションを開始すると、「sa」ユーザのバインドのみが使用されます。

  注意　最初のバインドが失敗すると、Cluster Edition は自動的に default グループのメンバを割り当てます。Cluster Edition は別のバインドを検索しません。

2　データベースにバインドされる場合は、データベースが割り当てられます。データベース・グループにバインドされる場合は、現在のインスタンスのメンバ･データベースがラウンドロビン方式で選択されて割り当てられます。所有者インスタンスを使用して設定されたセッションに対してのみ、グループのメンバであるローカル・テンポラリ・データベースを割り当てることができます。それらをグループ・バインドを使用して別のインスタンスのセッションに割り当てることはできません。

3　バインドが見つからなかったり、割り当てができない場合は、default グループからラウンドロビン方式で選択されたメンバに対して割り当てが試行されます。

4 default グループのメンバに割り当てを実行できない場合、ローカル・システム・テンポラリ・データベースに割り当てが行われます。

バインドの作成と管理

sp_tempdb bind は、ログインまたはアプリケーションをローカル・テンポラリ・データベースまたはデータベース・グループにバインドします。たとえば、「sa」ログインを default グループにバインドするには、以下のように入力します。

```
sp_tempdb "bind", "LG", "sa", "GR", "default"
```

ase1 の local_tempdb1 に isql をバインドするには、以下のように入力します。

```
sp_tempdb "bind", "AP", "isql", "DB", "local_tempdb1"
```

sp_tempdb unbind は、1 つまたは複数のバインドを削除します。Cluster Edition の場合、このコマンドには、特定のローカル・テンポラリ・データベースへのバインドを削除する instance_name パラメータがあります。

たとえば、「ase2」の local_tempdb2 から isql をバインド解除するには、以下のように入力します。

```
sp_tempdb "unbind", "AP", "isql", NULL, "ase2"
```

注意 isql がデータベース・グループにバインドされている場合、上記のコマンドではグループのバインドが削除されます。複数のデータベースのバインドが isql に対して存在する場合、上記のコマンドでは「ase2」のバインドのみが削除されます。つまり、他のインスタンスのテンポラリ・データベースへのバインドは影響を受けません。

データベース・グループまたはテンポラリ・データベースからユーザ・ログインまたはアプリケーションのバインドを解除するには、ログインまたはアプリケーション名のみを指定して unbind パラメータを使用します。たとえば、「sa」ログインのバインドを削除するには、以下のように入力します。

```
sp_tempdb "unbind", "lg", "sa"
```

特定のグループへのすべてのログインまたはアプリケーションのバインドを解除するには、sp_tempdb unbindall_gr を使用します。たとえば、tempdbgroup1 へのすべてのバインドを解除するには、次のように入力します。

```
sp_tempdb "unbindall_gr", "tempdbgroup1"
```

特定のデータベースへのすべてのログインまたはアプリケーションのバインドを解除するには、sp_tempdb unbindall_db を使用します。たとえば、localtempdb1 へのすべてのバインドを解除するには、次のように入力します。

```
sp_tempdb "unbindall_db", "localtempdb1"
```

グループおよびバインド情報の表示

既存のグループ、グループ・メンバ、ログインおよびアプリケーションのバインドのリストを表示するには、sp_tempdb show を使用します。たとえば、テンポラリ・データベース・グループ「tempdbgroup1」のメンバとその所有者インスタンスを表示するには、以下のように入力します。

```
sp_tempdb "show", "gr" "tempdbgroup1"
```

テンポラリ・データベースを割り当てたアクティブなセッションのリストを表示するには、sp_tempdb who を使用します。たとえば、テンポラリ・データベース「localtempdb1」を割り当てたアクティブなセッションを表示するには、次のように入力します。

```
sp_tempdb "who", "localtempdb1"
```

テンポラリ・データベースの削除

テンポラリ・データベースを削除する場合、いくつかの制約が適用されます。

最後のローカル・システム・テンポラリ・データベース以外のすべての削除

所有者インスタンスが実行中の場合、ローカル・システム・テンポラリ・データベースは常に使用中になるため、所有者インスタンスの実行中にローカル・システム・テンポラリ・データベースを削除することはできません。

ローカル・システム・テンポラリ・データベースを削除するには、インスタンスを停止してから、別のインスタンスからローカル・システム・テンポラリ・データベースを削除します。

最後のローカル・システム・テンポラリ・データベースの削除

最後のインスタンスのローカル・システム・テンポラリ・データベースを削除する場合は、次の手順に従います。

1　sp_tempdb_markdrop を使用して、削除するデータベースをマークします。たとえば、ase3_tdb1 が最後のローカル・システム・テンポラリ・データベースの場合、次のコマンドでマークします。

```
sp_tempdb_markdrop ase3_tdb1
```

2　最後のインスタンスを停止し、再起動します。「drop」というマークが付いたローカル・システム・テンポラリ・データベースは使用されません。

3　このインスタンスからテンポラリ・データベースを削除します。

テンポラリ・データベースの制約

Cluster Edition の場合

- 別のインスタンスが所有するローカル・テンポラリ・データベースにアクセスしようとすると、このエラーが発生します。

```
Error Number : 969

You can access database '<local-tempdbname>' only from its
owner instance '<owner-instance-name>'. You cannot access
local temporary databases from non-owner instances except
to use CREATE DATABASE and DROP DATABASE with local system
temporary databases.
```

  一般的には、所有者インスタンスからローカル・テンポラリ・データベースにアクセスするストアド・プロシージャやコマンドを実行する必要があります。代わりに次の方法を使用して、リモート・インスタンスのローカル・テンポラリ・データベースで操作を実行することもできます。

  - connect to *instance_name* を使用して、任意のローカル・テンポラリ・データベースにアクセスする前に所有者インスタンスに接続します。
  - sp_remotesql を使用して、リモート・インスタンスに Transact-SQL 文を送信します。たとえば、インスタンス ase1 が所有するローカル・テンポラリ・データベース local_tempdb_ase1 上でクエリを実行するには、次のように入力します。

```
sp_remotesql "ase1", "select * from local_tempdb_ase1..sysobjects"
```

  - ストアド・プロシージャの場合、インスタンス名を指定するプロシージャを実行できます。たとえば、インスタンス「ase1」に対する local_tempdb_ase1 上で sp_spaceused を実行します。

```
ase1.local_tempdb_ase1.dbo.sp_spaceused
```

    注意 db_instanceid を使用して、ローカル・テンポラリ・データベースの所有者インスタンス ID を決めます。

- ノンクラスタ Adaptive Server のすべてのデータベース上で操作を実行する一部のストアド・プロシージャとコマンドは、Cluster Edition のリモート・ローカル・テンポラリ・データベースの操作をスキップする場合があります。これらの操作がすべてのデータベースで確実に実行されるようにするには、所有者インスタンスからスキップされたそれぞれのデータベースに対して操作をもう一度実行します。

  たとえば、データベース名を指定せずに sp_dbcc_faultreport を実行すると、リモート・ローカル・テンポラリ・データベースが障害レポートからスキップされます。

- ローカル・テンポラリ・データベースのユーザに対するユーザ定義ロールの付与または取り消しはできません。

- ローカル・テンポラリ・データベースのオブジェクトに対するパーミッションの付与または取り消しはできません。
- ローカル・テンポラリ・データベースを任意のログインのデフォルト・データベースとして使用することはできません。使用しようとすると、sp_addlogin と sp_modifylogin は失敗します。
- テーブルが同じローカル・テンポラリ・データベースに存在しないかぎり、ローカル・テンポラリ・データベースのカラムを参照する参照整合性制約を含めることはできません。create table、alter table、create schema は、別のデータベースからローカル・テンポラリ・データベースのカラムへの参照を作成しようとすると、失敗します。
- テーブルが同じローカル・テンポラリ・データベースに存在しないかぎり、ローカル・テンポラリ・データベースにある暗号化キーでカラムを暗号化することはできません。create table と alter table は、別のデータベースに常駐するローカル・テンポラリ・データベースとテーブルにある暗号化キーでカラムを暗号化しようとすると、失敗します。
- どのローカル・システム・テンポラリ・データベースにも、データベース・リカバリ順序は指定できません。ローカル・システム・テンポラリ・データベースはユーザが作成するものですが、常にシステム・データベースで回復されます。
- sp_configure を使用して、ローカル・テンポラリ・データベース・コンテキストの中から設定オプションを変更することはできません。

ローカル・データベースのプライベート・デバイス・サポート

Cluster Edition では、ローカル・ディスクやファイル・システム・デバイスなどの廉価な記憶領域を使用して、インスタンス特有のテンポラリ・スペースのニーズを満たすことができます。プライベート・デバイスとして作成されたそのような記憶領域を使用して、プライベート・デバイスにインスタンス特有のローカル・ユーザ・テンポラリ・データベースを作成できます。ただし、ローカル・システム・テンポラリ・データベースのプライベート・デバイスは使用しないでください。ローカル・システム・テンポラリ・データベースは共有記憶領域に作成する必要があります。

Cluster Edition では、プライベート・デバイスは 1 つのクラスタ・インスタンスのみによって所有され、アクセスされます。そのデバイスはプライベートで、*logical* 属性を持ちます。デバイスへの物理パスは、非所有者インスタンスからアクセスできる場合と、できない場合があります。

sysdevices テーブルで、プライベート・デバイスはビットマップの status2 カラムによって識別され、デバイスを所有するインスタンスはデバイスに属するローの instanceid カラムによって識別されます。

インスタンスは多くのプライベート・デバイスを所有できますが、プライベート・デバイスは 1 つのインスタンスによって所有されます。

プライベート・デバイスは、デバイスを所有するインスタンス上でローカル・ユーザ・テンポラリ・データベースに対してのみ使用できます。それをデフォルト記憶領域として使用することはできません。プライベート・デバイスで sp_diskdefault を呼び出すと、ストアド・プロシージャは失敗し、エラー・メッセージが表示されます。

プライベート・デバイスのミラーリング、ミラーリング解除、または再ミラーリングはできません。

テンポラリ・データのプライベート・デバイスの使用

Cluster Edition のテンポラリ・データの最適なパフォーマンスを実現するために、ローカル・ユーザ・テンポラリ・データベースには高速なローカル・ディスクを使用することをおすすめします。ローカル・システム・テンポラリ・データベースは共有記憶領域に作成する必要があります。テンポラリ・データの共有記憶領域の使用を制限するには、次の手順に従います。

1 ローカル・ディスクに 1 つまたは複数のプライベート・ディスクを作成します。

2 プライベート・デバイスに 1 つまたは複数のローカル・ユーザ・テンポラリ・データベースを作成します。

3 これらのローカル・ユーザ・テンポラリ・データベースを default テンポラリ・データベース・グループに追加します。

明示的なテンポラリ・データベースのバインディングがないユーザ接続は、default グループからメンバ・データベースに割り当てられます。高速ローカル・ディスクに作成されたローカル・ユーザ・テンポラリ・データベースをすべてのユーザ接続が使用する場合、共有記憶領域に作成されたローカル・システム・テンポラリ・データベースの使用は内部のシステム・タスクに制限されます。

disk init を使用したプライベート・デバイスの作成

プライベート・デバイスを作成するには、disk init を使用します。disk init には、インスタンスにプライベートなデバイスをマークする、オプションの instance パラメータがあります。

次の例は、cluster1_1 インスタンスに対してプライベートな *private_dev1* という名前のデバイスを作成します。

```
disk init
name = "private_dev1",
physname = "/usr/u/sybase/data/private_dev1.dat",
vdevno = 2, size = 5102, instance = "cluster1_1"
```

disk init コマンドは、クラスタのどのインスタンスからも実行できます。このコマンドは、所有インスタンスに内部的に発行されます。ただし、このコマンドを処理するには、所有インスタンスが起動し、実行中であることが必要です。そうでない場合、コマンドは失敗し、プライベート・デバイスは作成されません。

disk reinit を使用したプライベート・デバイスの再初期化

master データベースのリストア手順の一部として、disk reinit を使用して master データベースの sysdevices テーブルのプライベート・デバイス・エントリをリストアします。disk reinit には、インスタンスに対してプライベートであるデバイスをマークする、オプションの instance パラメータがあります。

この例は、private_dev1 を master データベースの sysdevices テーブルにあるインスタンス cluster1_1 のプライベート・デバイスとしてリストアします。

```
disk reinit
name = "private_dev1"
physname = "/usr/u/sybase/data/private_dev1.dat"
vdevno = 2, size = 5120, instance = "cluster1_1"
```

このコマンドは、クラスタのどのインスタンスからも実行できます。所有インスタンスが起動し、実行中である場合、このコマンドは所有インスタンスに対して発行されます。

所有インスタンスが起動しておらず、実行中でもない場合、disk reinit はそのデバイスに対応するローを sysdevices に挿入しますが、インスタンスが実行中になるまでそれをアクティブにしません。これは、デバイスを disk refit の発行前にリストアする必要がある場合に便利です。

注意　実行中でないインスタンスのプライベート・デバイスをリストアするときは、渡すパラメータについて特に注意を払ってください。このような場合、Adaptive Server はプライベート・デバイスの物理パスを検証できません。ユーザが有効な物理パスを示しているものと想定します。

---

sp_dropdevice を使用したプライベート・デバイスの削除

プライベート・デバイスを削除するには、sp_dropdevice を使用します。このプロシージャは、クラスタのどのインスタンスからも実行できます。ただし、所有インスタンスが実行中でない場合、プライベート・デバイスの削除は失敗します。sp_dropdevice の使用方法の詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。

sp_helpdevice を使用したプライベート・デバイス情報の表示

sp_helpdevice は、デバイスがプライベートであるかどうかを報告します。デバイスがプライベートの場合、sp_helpdevice はプライベート・デバイスを所有するクラスタのインスタンスの名前を出力します。

次の例では、'regular_dev' は通常の共有可能デバイスである一方、'private_dev1' はインスタンス 'cluster1_1' によって所有されるプライベート・デバイスです。

```
1> sp_helpdevice
2> go
 device_name
        physical_name
        description
        status cntrltype vdevno      vpn_low     vpn_high

 ------------------------
        ------------------------------------------------------------------------------------
---------

         ------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
------------

          ------ --------- ----------- ----------- -----------
 master
        /remote/percy_dev/user1/sdc-DBs/master.dat
        file system device, special, dsync on, directio off, default disk,
        physical disk, 512.00 MB, Free: 351.00 MB
            3         0           0           0      262143
 private_dev1
        /percy_dev/user1/sdc-DBs/private_dev1.dat
        file system device, special, private 'cluster1_1', MIRROR DISABLED, non
        serial writes, dsync off, directio on, physical disk, 2.00 MB
            2050         0           1           0        1023
 regular_dev1
        /percy_dev/user1/sdc-DBs/dev1.dat
        file system device, special, dsync off, directio on, physical disk, 2.0
        0 MB
         2         0           2           0        1023
 tapedump1
        /dev/nst0
        unknown device type, disk, dump device
            16         2           0           0       20000
 tapedump2
        /dev/nst1
        unknown device type, tape,        625 MB, dump device
            16         3           0           0       20000

(5 rows affected)
(return status = 0)
```

disk refit がデバイスで保留中の場合は、sp_helpdevice の出力にそれが反映されます。

次の例では、'regular_dev' は通常の共有可能デバイスである一方、'private_dev1' と 'private_dev2' はインスタンス 'cluster1_1' によって所有されるプライベート・デバイスです。さらに、'private_dev1' で disk refit が保留されています。

```
1> sp_helpdevice
2> go
 device_name
        physical_name
        description
        status cntrltype vdevno      vpn_low     vpn_high

 -------------------------

         ------------------------------------------------------------------------------
----------

         ------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
----------------------------------------

         ------ --------- ----------- ----------- -----------
 master
        /remote/percy_dev/user1/sdc-DBs/master.dat
        file system device, special, dsync on, directio off,
        default disk, physical disk, 512.00 MB, Free: 351.00 MB
            3         0           0           0      262143
 private_dev1
        /percy_dev/user1/sdc-DBs/private_dev1.dat
        file system device, special, private 'cluster1_1',
        REFIT PENDING, MIRROR DISABLED, nonserial writes, dsync off, directio
        on, physical disk, 2.00
MB
        2050         0           1           0        1023
 private_dev2
        /percy_dev/user1/sdc-DBs/private_dev2.dat

         file system device, special, private 'cluster1_1', MIRROR DISABLED, non
         serial writes, dsync off, directio on, physical disk, 2.00
 MB
        2050         0           3           0        1023
 regular_dev1
        /percy_dev/user1/sdc-DBs/dev1.dat
        file system device, special, dsync off, directio on, physical disk, 2.00  MB
        2         0           2           0        1023
 tapedump1
        /dev/nst0
        unknown device type, disk, dump device

             16         2           0           0       20000
```

```
 tapedump2
        /dev/nst1
        unknown device type, tape,         625 MB, dump device
        16          3            0           0        20000

(6 rows affected)
(return status = 0)
```

create database と alter database のプライベート・デバイスでの使用

プライベート・デバイスはクラスタの 1 つのインスタンスのみに属しているため、プライベート・デバイスを所有するインスタンスのローカル・ユーザ・テンポラリ・データベースにのみ使用できます。プライベート・デバイスを使用してデータベースやそのログを作成したり、拡張したりする場合、そのデバイスを所有するインスタンスのローカル・ユーザ・テンポラリ・データベースのためでないかぎり、create database または alter database は失敗し、エラー・メッセージが表示されます。

disk refit の使用

クラスタにプライベート・デバイスがない場合、disk refit は通常通りに動作します。

クラスタに 1 つまたは複数のプライベート・デバイスがある場合、disk refit は 2 段階の操作になります。

- 第 1 段階は、コマンドを開始したインスタンスにアクセス可能なデバイスにのみ disk refit を実行します。つまり、通常の共有可能なデバイスと、そのインスタンスによって所有されるプライベート・デバイスが対象になります。
- 第 2 段階は、disk refit をクラスタのインスタンスのプライベート・デバイスにのみ、一度に 1 インスタンスずつ実行します。disk refit は、これらの段階が両方とも正常に終了したときにのみ完了します。

❖ **クラスタにプライベート・デバイスがあるときのディスクの実行**

disk refit を実行する前に、sysdevices テーブルを正しくリストアします。sysdevices テーブルで欠落しているデバイス・エントリがある場合は、disk reinit コマンドを使用して通常のデバイスまたはプライベート・デバイスを追加します。

sysdevices のリストアの詳細については、『システム管理ガイド 第 2 巻』を参照してください。

1. シングルユーザ・モードでクラスタを再起動します。これは、-m オプションでクラスタのインスタンスを再起動することによって行います。シングルユーザ・モードはクラスタで実行中の 1 つのインスタンスのみを必要とし、クラスタへのその他のアクティブな接続は必要ありません。トレース・フラグは 3608 に設定します。
2. インスタンスで disk refit コマンドを発行します。これによって、sysdatabases と sysusages のすべてのローを削除し、sysdevices のデバイスからその部分を再構築します。それらはクラスタのすべてのインスタンスによってアクセスできる正常な共有可能なデバイスであるか、このインスタンスにプライベートなものです。第 1 段階の最後に、クラスタが停止します。

   Adaptive Server は、第 1 段階の最後に、disk refit が完了したかどうかを示すメッセージを出力します。完了していない場合、Adaptive Server は、クラスタの他のどのインスタンスにプライベート・デバイスがあり、第 2 段階を実行する必要があるかをユーザに通知するメッセージも出力します。

   sp_refit_admin 'status' を使用して、disk refit がまだ実行されていないプライベート・デバイスを持つインスタンスを検索することもできます。

   出力の例を次に示します。

```
1> sp_refit_admin 'status'
2> go
You need to run DISK REFIT on the following instances to refit their private
devices.

 instance
    device

status
-------------------------------------------------------------
    -----------------------------------------------------------

          ---------------------------------------------------------------

-------------------------------------------------------------------------

-------------------------------------------------------------------------

-------------------------------------------------------------------------

-------------------------------------------------------------------------

-------------------------------------------------------------------------

-------------------------------------------------------------------------

----------------------

cluster1_1

private_dev1

REFIT
```

```
PENDING

(1 row affected)

(return status = 0)
```

3 プライベート・デバイスを持ち、disk refit をこれから実行する必要がある各インスタンスに対し、次のように実行します。

- -m オプションとトレース・フラグ 3608 を使用して、インスタンスを起動します。
- そのインスタンスに disk refit を発行します。

disk refit は、このインスタンスが所有するプライベート・デバイスの sysdatabases と sysusages のローを再構築します。インスタンスの第 2 段階が正常に完了した場合、クラスタは停止します。

4 disk refit の実行中にエラーが発生しなかった場合でも、sysdatabases と sysusages の整合性を検査してから、通常の操作を再開する必要があります。「トラブルシューティング」(201 ページ) には、disk refit で見つかった共通の問題を確認し、修正するためのサンプルの SQL が掲載されています。

5 通常の操作を再開します。

第 9 章

クラスタ環境での Job Scheduler の実行

この章では、共有ディスク・クラスタ環境で Job Scheduler を実行する方法について説明します。Job Scheduler をインストール、開始、設定する方法の詳細については、『Job Scheduler ユーザーズ・ガイド』を参照してください。

Cluster Edition を使用して、共有ディスク・クラスタ環境で Job Scheduler を実行できます。1 つの Job Scheduler は、クラスタ内のすべてのインスタンスから発生するすべてのジョブ・スケジューリング要求にサービスを提供します。Job Scheduler はブート・コーディネータで実行されます。

クラスタの起動後、Job Scheduler を手動で起動した場合、クラスタはコーディネータを選択して Job Scheduler をホストします。

Job Scheduler のインストールと設定

Job Scheduler を実行中のコーディネータ・インスタンスは「Job Scheduler インスタンス」とも呼ばれます。Job Scheduler を設定し起動するには、『Job Scheduler ユーザーズ・ガイド』の「第 2 章 Job Scheduler の設定と実行」の指示に従ってください。Job Scheduler を Cluster Edition でインストールおよび設定するための指示については、ご使用のプラットフォームの『インストール・ガイド』を参照してください。

クラスタ環境での Job Scheduler の実行

Job Scheduler インスタンス以外のインスタンスでユーザが Job Scheduler アクションを実行すると、インスタンスは、Job Scheduler インスタンスに要求を出します。Job Scheduler は要求を処理し、要求を出しているインスタンスに応答を返送します。

Job Scheduler のシャットダウン

共有ディスク・クラスタ内で Job Scheduler をシャットダウンするのは、ノンクラスタ Adaptive Server 設定で Job Scheduler をシャットダウンするのと同じです。

Adaptive Server Sybase Central Plug-in からシャットダウンするには、次の手順に従います。

1 Adaptive Server インスタンスで [定期ジョブ] フォルダを右クリックします ([定期ジョブ] フォルダは、各インスタンスの下に表示され、各インスタンスはプラグインからすべての Job Scheduler コマンドを実行します)。

2 [管理] を選択し、[Job Scheduler 管理] ダイアログを開きます。

3 [タスク設定] タブで [停止] を選択します。

コマンド・ラインで以下を実行します。

```
use sybmgmtdb
go
sp_js_wakeup "stop_js", 1
go
```

定期ジョブのリダイレクト

Job Scheduler エージェントが定期ジョブを実行するためにインスタンスに接続しようとしたもののインスタンスがビジーである場合、Workload Manager は接続を別のインスタンスにリダイレクトできるので、クラスタ内のインスタンスは定期ジョブを実行できます。これは自動的に行われるので、ユーザが Job Scheduler を再設定する必要はありません。

一般に、Workload Manager がログイン・リダイレクトを処理します。接続がリダイレクト方法に影響を及ぼすには、論理クラスタ・レベルでルールを設定します。ただし、定期ジョブをリダイレクトしない場合は、リダイレクトに関する新しいジョブ・プロパティを設定します。

- Adaptive Server プラグインから － [ジョブ・プロパティ] ダイアログ・ボックスの [リダイレクトの許可] オプションをオフにします。

  デフォルトの動作では、Workload Manager は、接続が行われたときに存在した負荷ルールに従って、定期ジョブ接続をリダイレクトします。

- コマンド・ラインから － ジョブを作成または修正するときに、リダイレクト・プロパティ no_conn_redirection を設定します。たとえば、sp_sjobcreate を使用して find_old_logins という名前のジョブについてプロパティを設定するには、次のように入力します。

  ```
  sp_sjobcreate @name='jname=find_old_logins',
  @option='jcmd=exec
  sp_find_old_logins,jproperties=no_conn_redirection=true'
  ```

第 10 章

インスタンスのリカバリ

この章では、単一のクラスタ・インスタンスのエラーと複数のクラスタ・インスタンスのエラーの両方に対するサポートについて説明します。15.5 以前の Cluster Edition では単一のインスタンス・エラーのみがサポートされますが、Sybase バージョン 15.5 以降では、同時に発生した複数のエラーがサポートされます。

単一のインスタンスのリカバリ

Cluster Edition にはフェールオーバ・リカバリが備わっており、その間、データベースはオンラインのままで使用継続できます。ただし、失敗したインスタンスによって変更された、リカバリの必要があるデータを要求すると、フェールオーバ・リカバリ・プロセスがデータを一貫性のある状態にするまで、ユーザ・プロセスはブロックされます。

Cluster Edition は以下をサポートしています。

- クラスタ起動時に発生するリカバリ。これは、サーバのコールド再起動を実行したときに発生するノンクラスタ Adaptive Server リカバリと同じです。
- フェールオーバ・リカバリ。これは、他のインスタンスが共有ディスク・クラスタ内の同じデータベースを使用している間に、エラーが発生したインスタンスによって変更されたデータをリカバリします。この章では、「フェールオーバ」とは、クライアント接続フェールオーバではなく、インスタンス・フェールオーバのことです。

リカバリは、クラスタ・メンバシップ・サービスがエラーを検出すると開始されます。コーディネータ・インスタンスのリカバリ・イベント・ハンドラは、システム・データベースをリカバリします。次に、回復されたインスタンス (コーディネータ・インスタンスを含む) によってユーザ・データベースが同時にリカバリされます。各インスタンスの PCM スレッドの 1 つがリカバリ・タスクを実行します。コーディネータ・インスタンスのその他のスレッドは、他のインスタンスと同じように、その他のアクティビティを続行します。

トレース・フラグ 3483 がオンの場合、分散型のフェールオーバ・リカバリは無効となり、ユーザ・データベースは、コーディネータ・インスタンスのリカバリ・イベント・ハンドラによって逐次リカバリされます。

エラーが発生したインスタンスによって変更されたデータにアクセスを試みるプロセスは、リカバリが完了するまでブロックされます。

Cluster Edition は、データベース ID (dbid) で指定されている順序でユーザ・データベースをリカバリし、インスタンスのフェールオーバ・リカバリの実行中は、ユーザ定義のリカバリ順を使用しません。

単一トランザクション・ログ

クラスタにはデータベースごとに 1 つのログがあり、論理的に分割されています。Cluster Edition のログ・レコード・マーカにはインスタンス ID が含まれています。これは、特にフェールオーバ・リカバリ時に特定のインスタンスによってログされたログ・レコードを内部スキャンします。チェックポイント・ログ・レコードには値ゼロを持つインスタンス ID が必ずあるため、ログ・スキャンで指定されるインスタンス ID に関係なく、常にログ・スキャンに含まれます。

複数の同時発生エラー

バージョン 15.5 以降では、Cluster Edition は複数の同時発生インスタンス・エラーをサポートしています。単一のクラスタ・ビュー内で複数のインスタンスがエラーになっても、クラスタはオンラインのままで、単一のインスタンスがエラーになったときと同じようにフェールオーバ・リカバリが行われます。

Cluster Edition には設定パラメータ cluster redundancy level があり、データベース管理者は、このパラメータを使用してクラスタでのリカバリ可能な同時発生インスタンス・エラーの最大数を設定できます。

複数の同時発生フェールオーバの有効化

エラーが発生したインスタンスのリカバリが可能な数とは、クラスタ冗長レベル (CRL) のことで、cluster redundancy level の値を決定する設定パラメータの名前でもあります。CRL は、同時に障害が発生しても許容されるインスタンスの最大数であり、この数値内であれば、リカバリして他のアクティビティを同時に続行できます。エラーが発生したインスタンスの数が設定パラメータによって指定されている最大数を超えると、クラスタはシャット・ダウンします。

CRL が 1 よりも大きい数値に設定されると、クラスタ内でのロックの複製も増加します。CRL は物理的なクラスタ・レベルにおけるもので、フェールオーバ論理クラスタ設定よりも常に優先されます。つまり、CRL がクラスタ内の物理的なインスタンスの数を超えることはありません。

cluster redundancy level はデフォルト値 1 を持つ静的なクラスタ・パラメータです。最小許容値は 1 で、最大許容値は *cluster.cfg* またはクォーラム・ファイルに指定されている 'maximum number of instances' よりも 1 つ小さい値になります。1 つまたは複数のインスタンスがクラスタからなくなっても、cluster redundancy level の値は変わりません。

cluster redundancy level は sp_configure パラメータです。

```
sp_configure 'cluster redundancy level', config_value
```

フェールオーバ・リカバリでは、他のインスタンスが同一のデータベースを使用している間に、エラーが発生したインスタンスによって変更されたデータをリカバリします。CRL が 1 の場合 (デフォルト)、Adaptive Server はクラスタ内の各ロックのコピーを少なくとも 2 つ保持します。そのため、1 つのインスタンスでエラーが発生しても、ロックのいずれかのコピーがそのままの状態を保つため、Adaptive Server はエラーが発生したインスタンスによって変更されたデータにマークを付けることができます。同様に、CRL が "n" の場合、Adaptive Server にはクラスタの各ロックのコピーが少なくとも "n+1" 個あり、同時発生した "n" 個のフェールオーバに対してフェールオーバ・リカバリを実行できます。

クラスタが起動するためには、cluster redundancy level の値が *cluster.cfg* またはクォーラム・ファイルで指定されている maximum number of instances の値より少なくとも 1 少ない値である必要があります。そのため、次のいずれかに設定した場合は、クラスタが起動できません。

- cluster redundancy level の値と等しいかそれより小さい maximum number of instances の値。
- maximum number of instances の値に等しいかまたはそれより大きい cluster redundancy level の値。

クラスタを起動する前に cluster redundancy level を設定します。ランタイム時にその値を変更する場合は、クラスタを再起動する必要があります。

パフォーマンスの制限とクラスタ冗長レベル

CRL の値が増加すると、クラスタがサポートできる同時発生したインスタンス・エラーの数も増えます。ただし、CRL 値が増加するということはクラスタ内に複数のロックのコピーが存在することになるため、メッセージング・トラフィックが増える原因にもなります。この冗長レベルを維持するには、オーバヘッドの増加が必要です。

たとえば、CRL が 1 より大きい場合は、number of locks や cache size などの他の設定パラメータにもより多くのリソースが必要になります。これはつまり、number of locks の値が同じでも、ノンクラスタ環境よりも max memory を増やす必要があるということです。ユーザに意識させることなく Adaptive Server がオーバヘッド値の計算を実行するとしても、CRL は大量のリソースを必要とします。

複数のインスタンス・エラーが頻発しない限りは、cluster redundancy level を 1 に設定することをおすすめします。

リカバリ・アルゴリズム

ノンクラスタ Adaptive Server での内部データベース・リカバリ処理は、Cluster Edition でのリカバリ処理と類似しています。単一および複数のインスタンス・エラー・リカバリで、同じフェーズが必要になります。

表 10-1: クラスタ・サーバとノンクラスタ・サーバでのリカバリ手順

ノンクラスタのリカバリ・フェーズ	Cluster Edition のリカバリ・フェーズ
1) リカバリ可能なログ境界の見積もり − リカバリ可能なログは、リカバリ・チェックポイントに記録されている最も古いアクティブ・トランザクションからログの末尾までです。	1) リカバリ可能なログ境界を見積もります (ノンクラスタ Adaptive Server と同じ)。
2) 分析 − リカバリ可能なログを、最も古いアクティブ・トランザクションからログの末尾までスキャンし、未完了のトランザクションなどとしてリカバリ情報を作成します。この情報は、再実行パス、取り消しパス、および取り消し後パスで使用されます。	2) リカバリ対象のデータベースのログを分析します (ノンクラスタ Adaptive Server と同じ)。
3) 再実行 − リカバリ可能なログをスキャンし、必要に応じてログ・レコードに指定されている操作を再度実行します。	3) 再実行 − 未完了トランザクションのロックが取得されます。
	4) 補正ログ・レコードの領域を予約し、エラーが発生したインスタンスにある未完了のトランザクションを取り消します。
	5) エラーが発生したインスタンスでエラー発生前に取得されたすべてのロックを解放します。未完了トランザクションについて手順 3 で取得されたロックは除きます。

ノンクラスタのリカバリ・フェーズ	Cluster Edition のリカバリ・フェーズ
4) 取り消し － ログの末尾から最も古い未完了トランザクションの開始地点まで操作し、未完了トランザクションのログ・レコードによって指定されている操作を取り消します。Adaptive Server は、トランザクションごとに補正ログ・レコードのログを記録します。	6) 取り消し － 未完了トランザクションの論理的なロックが、完了したものとして解放されます。
	7) 空き領域の情報を書き込みます (スレッショルド・マネージャ・リカバリ)。
5) 取り消し後 － データベースのチェックポイント・レコードのログを記録し、空き領域量情報を書き込み (スレッショルド・リカバリ)、キャッシュをクリアします。	8) 取り消し後 － リカバリ・インスタンス上のすべてのダーティ・バッファをフラッシュします。インスタンスのフェールオーバ・リカバリ後、Cluster Edition はチェックポイントを実行しません。

注意『システム管理ガイド 第 2 巻』の「第 11 章 バックアップおよびリカバリ・プランの作成」の章を参照してください。

第 11 章

補足トピック

この章では、クラスタ・アーキテクチャの要素について詳しく説明します。クラスタおよびノンクラスタ Adaptive Server の両方で使用できる機能の、クラスタ環境に固有の側面について多くの項目で説明しています。

ノンクラスタ Adaptive Server のサブシステムの多くは、共有ディスク・クラスタ環境で稼動し、その他は Cluster Edition 専用に開発されています。これらのサブシステムは以下のとおりです。

- ロック・マネージャ
- バッファ・マネージャ
- クラスタ・プロセス間通信 (CIPC)
- リカバリ (起動時刻とフェールオーバの処理)

ロック

データベースのデータおよびメタデータのすべてのオブジェクトにアクセス可能で、クラスタ内のどのインスタンスでもキャッシュできます。その結果、これらのオブジェクトは、実行時のグローバルなデータ構造体や変数と同様に、分散ロックとキャッシュの一貫性が強化されます。

クラスタ・ロック・マネージャ (CLM) は、グローバルにアクセスとキャッシュができるオブジェクトを共有するための分散ロック・サービスを提供します。CLM はロックの要求を受けてロックを作成してキューに入れます。また、インスタンスから共有ディスク・クラスタ環境のグローバル・オブジェクトへとアクセス権のロック要求を調整します。

ロックにはタスク、トランザクション、またはインスタンスの所有権を含めることができます。ローカルのロック・マネージャで管理されるロックには、タスクまたはトランザクションの所有権があります。CLM で管理されるロックには、インスタンスの所有権があります。これらは、そのインスタンスのすべての処理やトランザクション間で共有され、別のインスタンスが競合するロック・モードを要求するまで維持されます。

デッドロック

Adaptive Server のノンクラスタ・エディションで使用される全ページ・ロックでは、サーバがページのロックを取得できなかった場合、ロックを待機する (デッドロックを引き起こす可能性がある) のではなく、再取得を試みます。設定パラメータ deadlock retries は、Adaptive Server が再試行する回数を決定します。

Cluster Edition には、ノード間のページの一貫性を確保する物理ロックが含まれています。ただし、デッドロック検出メカニズムには物理ロックが含まれていないため、サーバがデッドロックを検出できません。

Cluster Edition はロックの取得に失敗すると再試行します。ただし、サーバが deadlock retries の値を超えた場合は、ロックを待機してデッドロックが未検出になるリスクを負うのではなく、エラーを発生するかクエリをロール・バックします。

保持ロック

Cluster Edition では、保持ロックも使用されています。これはクラスタ内のインスタンスに付与されるクラスタ全体のロックで、インスタンス上のすべてのプロセス間で所有権が共有されます。保持ロックの所有権は、別のインスタンスが競合するロック・モードを要求するか、ロックに関連付けられているリソースが別のインスタンスによって要求されるときまで維持されます。保持ロックを使用すると、ロックを取得して解放する必要性が減るので、インスタンス内メッセージングが減少します。

Cluster Edition では数種類の保持ロックが使用されています。

すべてのロックが保持ロックです。

クラスタのロック要求とタスク要求のステータス

Cluster Edition では、タスクのロック要求がクラスタのロック要求をトリガしますが、いったんクラスタのロックが要求されると、クラスタのロック要求はタスクの要求ステータスから独立します。

たとえば、ロックを要求しているタスクが、クラスタ・レベルでロックが付与される前に終了するとします。これは、ロックがタイムアウトになったりタスクが強制終了された場合に起こる可能性があります。このような場合、クラスタのロックを所有しているインスタンスの sp_lock の出力には、ロックを待機しているタスクがない場合でも、クラスタのロック解放を所有しているインスタンスがそれをダウングレードするまでは、ブロックしているプロセスを示す “-blk” が引き続き表示されます。

メモリ

Cluster Edition は、スタンドアロンの SMP Adaptive Server よりも多くのメモリを必要とします。この追加のメモリでノード間のメッセージングやクラスタのロックをサポートします。

- 設定パラメータ CIPC regular message pool size を使用して、ノード間メッセージング用のメモリを設定できます。デフォルト値は 8MB です。
- 分散キャッシュの一貫性をサポートするため、Cluster Edition ではインスタンスの起動時にバッファのキャッシュとシステム記述子のロックが自動的に設定されます。

  データ・キャッシュ内の各バッファには自動的に物理ロックが設定されます。各物理ロックのオーバヘッドは、2KB ページのデータベースで物理ロック 1 個あたり約 24.6%、4KB ページのデータベースで約 12.3% です。したがって、100M のデータ・キャッシュの場合、物理ロックに必要な追加のメモリ・オーバヘッドは、2KB ページで 24MB、4KB ページで 12MB になります。

  システム記述子の平均オーバヘッドは、開いているオブジェクト 1 個につき約 1.5KB です。パフォーマンスを最大化するため、lock hashtable size パラメータは 1 バケット 8 ロックに自動調整されます。大きいキャッシュ構成には、lock hashtable size が自動的に数メガバイトに調整される場合があります。

スレッショルド

セグメントの空き領域が Adaptive Server が管理するスレッショルドを下回ると、ユーザ作成のストアド・プロシージャが実行されます。

すべてのデータベースには、ログ・セグメントにラストチャンス・スレッショルドがあります。これは、トランザクション・ログのバックアップとトランケートに必要な空きログ・ページ数の見積もりです。ラストチャンス・スレッショルド (LCT) は、定義されている使用可能領域で、ログ・セグメントに指定された空きページ数です。

LCT は、領域が指定のスレッショルドを下回った場合に取るアクションも定義します。このスレッショルドはデータベース内の空き領域をモニタして、トランザクション・ログの領域が足りなくなるのを防いでいます。

セグメントにスレッショルドを追加するときに、このスレッショルドのモニタに使用されるストアド・プロシージャを指定する必要があります。デフォルトでは、ラストチャンス・スレッショルドに sp_thresholdaction が使用されます。

ユーザには、クラスタ処理のスレッショルド管理とノンクラスタ Adaptive Server のスレッショルド管理が区別できません。ただし、以下に説明するように、dbcc コマンドにいくつか変更があります。

dbcc thresholds の出力

dbcc thresholds はセグメントの構造を出力します。Cluster Edition では、セグメントの構造自体が変更されたため、出力が異なります。たとえば、セグメントのスレッショルドのポインタである sg_below と sg_above は、dbt_thresholds 配列で 2 つのスレッショルドの位置を指定する数字です。

dbcc dbtable 出力

dbcc dbtable は dbt_thrmgr_info を出力します。これにはクラスタ・スレッショルドの管理データが含まれ、テーブルのセグメント構造を出力します。

remap オプションを使用する *dbcc dbrepair*

remap を使用する dbcc dbrepair：

- ディスク・マップを dbtable で再構築する
- セグメントの空きページ数を再計算する
- セグメントのスレッショルド・レベルと合計空きページ数に従って各セグメントのスレッショルドを設定する

dbcc dbrepair を remap と一緒に使用して、セグメント番号またはセグメント名を指定できますが、再マップするセグメントは一度に 1 つしか指定できません。

再マップするセグメントを指定しない場合は、データベース内のすべてのセグメントが再マップされます。dbcc dbrepair の最後にオプションのパラメータを入力して、再マップするセグメントを指定できます。

dbcc dbrepair にはオプション fixalloc が含まれています。remap と fixalloc の両方を使用するには、remap を fixalloc の前に追加します。Adaptive Server バージョン 15.1.1 では、このコマンドの最後にセグメント番号またはセグメント名を指定できます。この例では、pubs2 データベースのログ・セグメントを次のように指定します。

```
dbcc dbrepair(pubs2,"remap","fixalloc", -1,"logsegment")
```

pubs2 データベースのログ・セグメントを fixalloc を使用せずに再マップするには、次のように入力します。

```
dbcc dbrepair(pubs2, "remap",NULL, -1,"logsegment")
```

pubs2 データベースのすべてのセグメントを再マップするには、次のコマンドを使用します。

```
dbcc dbrepair(pubs2, "remap")
```

dbcc dbrepair と remap オプションは、sp_addsegment、sp_dropsegment、sp_modifysegment、sp_extendsegment、sp_logdevice、および sp_placeobject によって使用されます。これらのストアド・プロシージャと dbcc dbrepair の詳細については、『ASE リファレンス・マニュアル：コマンド』の最新バージョンを参照してください。

newthreshold オプションを使用する *dbcc dbrepair*

newthreshold オプションを使用する dbcc dbrepair：

- データベースのスレッショルドを systhresholds から dbtable->dbt_thresholds にロードします。
- セグメントのスレッショルドのレベルとセグメントの空きページ数に従って、各セグメントのスレッショルド・ポインタを設定します。
- セグメント名またはセグメント番号を指定し、systhresholds のすべてのスレッショルドを dbt_thresholds に読み込みます。dbt_thresholds セグメントのスレッショルドのみが、dbt_thresholds セグメントに属するスレッショルドとセグメントの空きページ数に従って設定されます。

dbcc dbrepair と newthreshold は、sp_addthreshold、sp_dropthreshold、sp_modifythreshold によって使用されます。これらのストアド・プロシージャと dbcc dbrepair の詳細については、『Adaptive Server Enterprise リファレンス・マニュアル：コマンド』の最新バージョンを参照してください。

クラスタ・プロセス間通信

クラスタ・プロセス間通信 (CIPC) はクラスタ・アーキテクチャに不可欠のサブシステムです。ほとんどのクラスタ・サブシステムは、CIPC を使用してクラスタ内の他のインスタンスと通信しています。

注意 Cluster Edition を快適に実行するには、高速な CIPC ネットワークが必要です。

クラスタ内のインスタンスで実行しているタスクは、CIPC を使用してクラスタ内の他のインスタンスの他のタスクと通信します。クラスタ内のすべてのインスタンスを接続するには、クラスタ入力ファイルで指定してクォーラム・デバイスに保存した CIPC 情報が使用されます。入力ファイル内で、CIPC はプライマリ相互接続とオプションのセカンダリ相互接続をサポートしています。セカンダリ相互接続がある場合は、バッファ・キャッシュの転送に使用して、プライマリ相互接続の負荷を軽減します。

分散チェックポイント

チェックポイントは、ノンクラスタ Adaptive Server と Cluster Edition の両システムでリカバリを制御しますが、トランザクション情報を提供するプロセスが異なります。Cluster Edition では、検出される最も古いアクティブ・トランザクションは、クラスタ全体で最も古いアクティブ・トランザクションであり、ダーティ・バッファはすべてのクラスタ・インスタンスでフラッシュされます。

どちらのプロセスでも、Adaptive Server はチェックポイント・レコードがログに記録されるときに最も古いアクティブ・トランザクションを検出し、それをカットオフ・ポイントに使用して、チェックポイント・レコードをログに書き込みます。次に、ダーティ・バッファが現在のフラッシュ・シーケンス番号までフラッシュされます。このチェックポイント・レコードは、ダーティ・バッファがすべてフラッシュされてからリカバリ・チェックポイントとして登録されます。

クォーラム・デバイスのハートビート

クラスタ内の各インスタンスは、定期的にクォーラム・デバイスに「チェックイン」してクォーラム・デバイスのハートビートを作成します。これによって、インスタンスはクラスタのステータスをモニタできます。インスタンスはデバイスのハートビートを以下の目的で使用します。

- クォーラム・デバイスにアクセス可能かどうかを確認する。インスタンスがクォーラム・デバイスにハートビートを書き込めなかった場合は、ストレージ・エリア・ネットワーク (SAN) とのリンクが失われたか、クラスタ・デバイスからブロックされている可能性があります。
- 起動中にクォーラム・ディスクからハートビート値を読み込む。設定した時間が経過してもハートビートの変化が検出されない場合は、クラスタが実行していないと判断されます。

クォーラム・ハートビートはインスタンスのエラー検出には使用されません。

クォーラム・デバイスのハートビートの設定

ハートビートが発生する間隔と、クォーラム・デバイスが機能していないと想定されるまでインスタンスがクォーラムのハートビート検出を試みる回数を、それぞれ設定パラメータの quorum heartbeat interval と quorum heartbeat retries で指定できます。

quorum heartbeat interval

設定パラメータ quorum heartbeat interval は、クォーラムのハートビート間の秒数を指定します。デフォルトは 5 で、各インスタンスは 5 秒間隔でクォーラムのハートビートを書き込みます。最小値は 1 秒で、最大値は 60 秒です。

ほとんどのサイトでは、この設定パラメータを調整する必要はありません。quorum heartbeat interval を小さい数に設定すると、ハートビートのオーバヘッドが増加しますが、失われたディスク・リンクの検出が加速されるので、ブロックされたり、SAN リンクを失った場合にインスタンスが迅速に終了します。quorum heartbeat interval を大きい値に設定すると、ハートビートのオーバヘッドが減少しますが、失われたディスク・リンクの検出が遅れます。ハートビートがもたらすオーバヘッドの量は、ディスク・サブシステムのパフォーマンスによって異なります。

quorum heartbeat retries

設定パラメータ quorum heartbeat retires は、クォーラム・デバイスが実行していないとインスタンスが判断して終了する前に、クォーラムのハートビート検出を何回試みるかを指定します。デフォルトは 2 です (インスタンスがクォーラムのハートビート検出に最初の 2 回失敗したら、連続 3 回失敗した後で終了します)。最小値は 0 で、インスタンスが最初のハートビート検出に失敗した時点で終了します。最大値は 32,768 です。

この値を小さくすると、クォーラム・デバイスへのアクセスが失われたときに迅速にフェールオーバするので、アプリケーションのリカバリ時間が早まる可能性があります。この値を大きくすると、アプリケーションのリカバリが低下するので、一時的なディスク・アクセスの問題がインスタンス障害を引き起こす可能性が減ります。

InfiniBand の使用

Adaptive Server 15.0.1 以降は、クラスタ内のノード間の内部通信に使用される InfiniBand (IPoIB) をサポートしています。

注意 InfiniBand は Cluster Edition の HPIA システムではサポートされていません。

InfiniBand を使用するには

- ホスト・チャネル・アダプタ (HCA) を設定します。
- InfiniBand ソフトウェア・スタックを使用します。
- Linux の場合は Cluster Edition に OpenFabrics の OFED 1.2 が必要です。
- InfiniBand ソフトウェアは Solaris 10 に付属しています。

注意 Solaris 10 には InfiniBand が含まれています。InfiniBand のインストールと設定については、オペレーティング・システムのマニュアルを参照してください。

バッファ領域の設定

Cluster Edition を快適に実行する (特に高速な相互接続で) には、十分なバッファ領域が必要です。Linux および Solaris システムでのバッファ領域のデフォルト値は、ギガビット・イーサネットと InfiniBand には小さすぎます。これらを変更して十分なネットワーク・パフォーマンスを確保する必要があります。

バッファ領域の設定

Linux

下記のようなコマンドを使用して、バッファ領域をシステムで適切なサイズに設定します。

```
/sbin/sysctl -w net.core.rmem_max=value
/sbin/sysctl -w net.core.wmem_max=value
```

ほとんどのプラットフォームで、rmem_max のデフォルト値は約 128KB に設定されていますが、Cluster Edition には小さすぎます。rmem_max を、プラットフォームが許可する最大値に増やします (最初は目安として 1 メガバイトが推奨されます)。

```
/sbin/sysctl -w net.core.rmem_max=1048576
```

Solaris

下記のようなコマンドを使用して、バッファ領域をシステムで適切なサイズに設定します。

```
ndd -set /dev/udp udp_max_buf value
```

クラスタでの InfiniBand の設定

ホスト・チャネル・アダプタ (HCA) を設定すると、*/etc/hosts* ファイルに対応するホスト名と IP アドレスが含まれます。クラスタ設定中に Adaptive Server プラグインまたは sybcluster がこの情報を要求した場合は、このホスト名または IP アドレスを使用します。クラスタを手動で設定する場合は、IP アドレスを追加します。

プライベート・インストール・モード

クラスタをインストールして作成するとき、クラスタを共有インストール・モードで設定するか、プライベート・インストール・モードで設定するかを選択します。

- 共有インストール － ネットワーク・ファイル・システム (NFS) またはクラスタ・ファイル・システムを使用して作成された共有ファイル・システムをサポートしています。共有インストールを使用して作成されたクラスタは、単一の *$SYBASE* インストール・ディレクトリ、Adaptive Server ホーム・ディレクトリ、およびサーバ設定ファイルをサポートします。

- プライベート・インストール – インスタンスごとに別の *$SYBASE* インストール・ディレクトリ、Adaptive Server ホーム・ディレクトリ、およびサーバ設定ファイルをサポートしています。サーバ設定ファイル間のパリティは、クォーラム・デバイス上のマスタ設定ファイルによって管理されます。

  注意 プライベート・インストールを使用する場合は、ディレクトリ・サービスとして LDAP を使用することをおすすめします。

インストール・モードがプライベートの場合は、クラスタ入力ファイルに自動的に次のエントリが含まれます。

```
installation mode = private
```

インストール・モードが共有の場合は、クラスタ入力ファイルに自動的に次のエントリが含まれます。

```
installation mode = shared
```

プライベートまたは共有のインストール・モードでクラスタをインストールして設定する手順については、使用しているプラットフォームのインストール・ガイドを参照してください。

サーバ設定ファイルの管理

設定ファイルのバージョン番号によって、サーバ設定ファイルの最新コピーを追跡します。この情報はクォーラム・デバイス上の各サーバ設定ファイルに保存されています。新しい設定ファイルがクォーラム・デバイスに追加されるたびに、バージョン番号が増え、Adaptive Server が新しいサーバ設定ファイルとバージョン番号を各インスタンスにダンプします。

設定ファイルの現在のバージョン番号を確認するには、次のように qrmutil を使用します。

```
qrmutil --quorum-dev=/dev/raw/raw101 --ase-config-version
```

構文の詳細については、『ユーティリティ・ガイド』を参照してください。

実行時の設定オプションの変更

インスタンスの実行中に、sp_configure を使用してサーバの設定オプションを更新できます。オプションが変わると、Adaptive Server は自動的にマスタ設定ファイルを更新してバージョン番号を増やし、sysconfigures テーブルを更新してから、設定ファイルの新しいコピーをダンプするようにすべてのインスタンスに指示します。

Cluster Edition で sp_configure を使用する場合

- 現在接続しているインスタンスにのみ適用されるインスタンス固有の値を設定します。
- "厳密にクラスタ全体の" 設定オプション、つまり、すべてのインスタンスで同じである設定オプションの小さいグループは、インスタンス固有にすることはできません。
- クラスタ全体の値を再設定しても、インスタンス固有の設定は上書きされません。

設定プロセス中にインスタンスが実行していない場合は、その設定ファイルとバージョン番号が更新されません。インスタンスの起動時に Adaptive Server に更新を検出させるか (「起動時にバージョン番号が一致しない場合」(168 ページ) を参照)、バージョン番号を確認して、次のようにクォーラム・デバイスから新しいコピーをダンプできます。

```
qrmutil --quorum-dev=/dev/raw/raw101 --ase-config-version

qrmutil --quorum-dev=/dev/raw/raw101 --ase-config-extract=mycluster.cfg
```

設定ファイルの手動変更

設定するインスタンスが実行していない場合は、設定ファイルを手動で変更できます。

1　設定ファイルを抽出します。次に例を示します。

```
qrmutil --quorum/dev/raw/raw101 --ase-config-extract=mycluster.cfg
```

2　設定ファイルを次のように編集します。

```
vi mycluster.cfg
```

3　インスタンスを再起動します。

インスタンスを再起動すると、Adaptive Server が自動的にバージョン番号を増やして新しいバージョン番号で sysconfigures テーブルを更新し、新しい設定ファイルをローカル・ディレクトリとクォーラム・デバイスに書き込んでから、クォーラム・デバイスから設定ファイルの新しいコピーをダンプするようにすべてのインスタンスに指示します。

注意　設定するインスタンスが実行していなくてもコーディネータが実行していれば、インスタンスに固有のパラメータを変更できますが、クラスタ全体のパラメータは変更できません。

起動時にバージョン番号が一致しない場合

各インスタンスの起動時に Adaptive Server がファイルのバージョン番号をチェックします。サーバ設定ファイルのバージョン番号が、クォーラム・デバイスに保存されているバージョン番号と一致する必要があります。ファイルのバージョン番号が一致しない場合、Adaptive Server はエラー・メッセージを出力し、クォーラム・ディスクからローカルのファイル・システムに設定ファイルの最新コピーを抽出して、インスタンスの起動をアボートします。

この時点で、最新の設定ファイルを使用してインスタンスを再起動するか、手動でファイルを編集してからインスタンスを再起動するかを決定できます。

- バージョン番号が一致し、設定ファイルに変更がない場合は、インスタンスを正常に再起動できます。
- 設定ファイルを編集した場合、バージョン番号は一致しますが、設定ファイルの内容が変更されます。インスタンスを再起動します。Adaptive Server が自動的にバージョン番号を増やして新しいバージョン番号で sysconfigures テーブルを更新し、新しい設定ファイルをローカル・ディレクトリとクォーラム・デバイスに書き込んでから、クォーラム・デバイスから設定ファイルの新しいコピーをダンプするようにすべてのインスタンスに指示します。

推奨される設定

Sybase では次の設定をおすすめします。

- サーバ設定ファイルの “config file version” パラメータを手動で変更しない。これはサーバが生成した番号のままにしてください。
- 設定ファイル・パラメータは起動インスタンスでのみ更新する。別のインスタンスで設定ファイル・パラメータを更新した場合は、そのインスタンスの起動時に新しい設定ファイルが生成され、すべてのユーザ変更が他のインスタンスから削除されます。

クラスタ環境での Java の使用

Adaptive Server は、Sun Java 2 Platform、Standard Edition (J2SE) などのサードパーティ Java 仮想マシン (JVM) をクラスタ環境でサポートしています。Sybase Java コンポーネントを Adaptive Server CE にインストールする方法については、使用しているプラットフォームのインストール・ガイドを参照してください。Adaptive Server で Java を使用する方法の詳細については、『Adaptive Server Enterprise における Java』を参照してください。

一般に、クラスタ環境でもノンクラスタ環境でも同様に Java を使用できます。相違点は以下のとおりです。

- Cluster Edition では、Java を実行している接続のマイグレーションはサポートされていません。

  接続に失敗した場合は、設定に従って、Java を実行している接続が別のインスタンスにフェールオーバします。フェールオーバ後、既存の SQL バッチ プロセスと Java は自動的に再実行されます。

- Adaptive Server は、JVM が共有かプライベートかによって、静的なデータを異なる方法で処理します。

  - Cluster Edition では、各ノードに独自の JVM があり、静的なデータはノード間で共有されません。ノードがダウンした場合は、JVM もダウンし、JVM 上のすべての作業が失われます。現存している SQL バッチ・プロセスを再適用する必要があります。

  - Adaptive Server 15.0.3 のノンクラスタ・エディションは、1 台の JVM をすべてのエンジンと接続で共有するので、共通のクラス・ローダからロードされたクラスで静的なデータを共有できます。

  - Adaptive Server 15.0.2 以前では、各クライアント接続に独自のプライベート Sybase JVM があるので、クライアント間で静的なデータはサポートされていません。

アーカイブ・データベースへの領域の追加

一般に、アーカイブ・データベースへのアクセスは、クラスタ Adaptive Server とノンクラスタ Adaptive Server で同様に処理されます。どちらの環境でも、アーカイブ・データベースの領域が不足すると、alter database コマンドを使用してアーカイブ・データベースに領域を追加できます。

クラスタ Adaptive Server では、アーカイブ・データベースを更新するノードから alter database を実行します。別のノードから alter database を実行した場合は、実際にアーカイブ・データベースを更新しているノードの番号とエラー・メッセージが出力されます。

この機能は、Adaptive Server バージョン 15.5 以降で使用できます。

共有ディスク・クラスタ上の分散トランザクション

バージョン 15.5 以降では、Adaptive Server のクラスタ・アーキテクチャで分散トランザクション管理 (DTM) がサポートされています。Cluster Edition の DTM には次のような特徴があります。

- XA-Server のような追加のサービスなしにリソース・マネージャ (RM) として機能する場合は、X/Open XA プロトコルに完全準拠しています。
- Adaptive Server のデータをリモート・プロシージャ・コール (RPC) とコンポーネント統合サービス (CIS) を介して更新するトランザクションに、一貫したコミットとロールバックを保証します。
- Adaptive Server Transaction Coordination (ASTC) メカニズムを使用している他の Adaptive Server インストールでコーディネートされた分散トランザクションの一部になることができます。
- ASTC メカニズムを使用して、複数の Adaptive Server インストール間の分散トランザクションをコーディネートできます。

注意 Cluster Edition は Microsoft 分散トランザクション・コーディネータ (MSDTC) 独自のプロトコルはサポートしていません。

共有ディスク・クラスタでの DTM の使用

一般に、分散トランザクションのユーザ・インタフェースは、Adaptive Server クラスタ環境とノンクラスタ環境で同じですが、この章で説明するクラスタ固有の問題に注意する必要があります。ノンクラスタ Adaptive Server で DTM を使用しているアプリケーションは、クラスタ Adaptive Server で同じアプリケーションを使用できます。『Adaptive Server 分散トランザクション管理機能の使用』を参照してください。

リソース・マネージャとしてのクラスタ

クラスタは複数の Adaptive Servers で構成されていますが、ユーザには 1 つのシステム・ビューを提供する必要があります。したがって、各アプリケーションは完全なクラスタを単一の RM として見なければならず、個々のインスタンスを別々の RM と見なすことはできません。

たとえば、クラスタでリモート・プロシージャ・コールを実行する場合は、RM としてクラスタ名を使用します。

```
exec cluster_name.dbname.owner.procedure_name
```

クラスタで XA トランザクションを実行するトランザクション・マネージャ (TM) が、XA 設定ファイルでクラスタ名を論理リソース・マネージャ (LRM) 名として使用することを確認します。インスタンス名を使用しないでください。XA 設定の詳細については、『XA インタフェース統合ガイド for CICS、Encina、TUXEDO』を参照してください。

同様に、jConnect を使用して XA トランザクションを実行するアプリケーションは、XADataSource オブジェクトの作成時にクラスタ名を使用する必要があります。次に例を示します。

```
XADataSource xads = (XADataSource)
    ctx.lookup("server_name=cluster_name");
```

クラスタは RM として名付けられますが、各トランザクションはクラスタ内の単一インスタンス (所有者インスタンス) で実行し、トランザクションの状態は、そのインスタンスでのみ管理されます。実行中のトランザクションはクラスタ内のインスタンス間でマイグレートできません。Transact-SQL 文とトランザクションのコミット処理はすべて、そのトランザクションを所有するインスタンスでのみ実行します。

非所有者インスタンスでの要求の処理

X/Open-XA プロトコルのような一部の分散トランザクション・メカニズムは、接続指向型ではなく、アプリケーションが他の接続でコマンドを発行できます。その結果、中断されたトランザクション・ブランチでの作業を再開する要求を、非所有者インスタンスで発行できます。同様に、コミットを処理する要求を非所有者インスタンスへの接続で発行できます。クラスタ Adaptive Server では、非所有者インスタンスで発行されたそのような要求は、特別な処理を必要とします。

非所有者インスタンスで要求が発行されると、クラスタ Adaptive Server はそのトランザクションの所有者インスタンスを特定してから、コネクティビティ・ライブラリを利用して接続を所有者インスタンスにマイグレートします。接続のマイグレートはアプリケーションに対して透過的に発生します。接続が正常にマイグレートした後、要求は所有者インスタンスで処理されます。

接続のマイグレートの詳細については、「第 2 章 クライアント・アプリケーションとクライアント／サーバの対話」を参照してください。

ASTC のトランザクションは接続マイグレーションを必要としません。ASTC メカニズムでは、Transact-SQL 文とトランザクションのコミット処理がすべて同じ接続で発行される必要があります。

インスタンス・エラーの処理

X/Open-XA プロトコルと ASTC プロトコルでは、準備されたトランザクションを TM の指示なしに RM が一方的にコミットしたりアボートしたりできません。RM は、トランザクションの終了ステータスが TM から通知されるまで、準備されたトランザクションを維持する必要があります。ただし、クラスタ内のインスタンスでトランザクション分岐が実行しているときに、エラーが発生することがあります。クラスタ・コーディネータはフェールオーバ・リカバリ中に、エラーが発生したインスタンスの準備されたトランザクションを再インスタンス化します。クラスタ・コーディネータでトランザクションが再インスタンス化されると、TM はそのトランザクションをコミットまたはロールバックできます。

注意 再インスタンス化されたトランザクションをコミットまたはロールバックするコマンドは、現在の所有者インスタンス以外のインスタンスで発行することもできます。このシナリオでは、次のように処理されます。

- XA トランザクションの場合 − サーバが接続を所有者インスタンスにマイグレートし、そこで要求が処理されます。
- ASTC トランザクションの場合 − commit または rollback コマンドを受け取るインスタンスはプロキシの役割を果たし、要求を所有者インスタンスに転送して処理し、応答を TM に送ります。

エラーが発生したインスタンスの準備されたトランザクションのみがフェールオーバ・リカバリ中に再インスタンス化されます。エラーが発生したインスタンスで実行中のトランザクション分岐がまだ準備されていない場合、クラスタ Adaptive Server ではフェールオーバ・リカバリ中にその作業がロールバックされます。

ASTC でコーディネートされたトランザクション

クラスタ Adaptive Server は、ノンクラスタ Adaptive Server と同様に、Adaptive Server の複数のインストールに渡る分散トランザクションを ASTC メカニズムを使用してコーディネートします。トランザクション・コーディネーション・サービスは、分散トランザクションを実行するクライアントに対して透過的です。ローカル・クライアント・トランザクションが RPC を実行するときや、CIS を使用してデータを更新するとき、コーディネーション・サービスはリモート作業用に新しいトランザクション名を作成し、従属するリモート・サーバにそのトランザクションを送信します。ローカル・クライアントがローカル・トランザクションをコミットまたはロールバックすると、Adaptive Server は従属サーバのそれぞれにその要求をコーディネートし、リモート・トランザクションが同じようにコミットまたはロールバックされることを保証します。

クラスタ Adaptive Server は特殊な ASTC_HANDLER タスクを実行して、接続の中断、リモート・サーバの障害、ローカル・サーバの障害などの問題を処理します。ASTC_HANDLER は、障害状態から回復した後、トランザクションの終了ステータスをリモート参加者に伝送します。ASTC_HANDLER のタスクはクラスタ内の各インスタンスで実行されます。障害が発生した場合、インスタンスの ASTC_HANDLER タスクは、そのインスタンスで開始した分散トランザクションのみの終了ステータスをリモート参加者に伝送します。クラスタ・コーディネータの ASTC_HANDLER タスクには、クラスタの前回の起動からのトランザクションのリモート参加者や、そのインスタンスがクラスタを出たときのインスタンスのリモート参加者を処理する追加作業があります。

接続マイグレーションの影響

XA トランザクションの場合、Adaptive Server クラスタは接続マイグレーションを使用して非所有者インスタンスの要求を処理します。

接続マイグレーションとパフォーマンス

接続マイグレーションには以下の操作が必要になるため、パフォーマンスに悪影響を与える可能性があります。

- 非所有者インスタンスの既存の接続を終了する
- トランザクションを実行するインスタンスへの新しい接続を作成する
- 既存の接続に関するコンテキスト情報を保存してから、接続マイグレーション後に所有者インスタンス上でこの情報をリストアする

接続マイグレーションによるパフォーマンスの低下を軽減するには

- 接続マイグレーションを行う必要がないように、論理クラスタと負荷管理機能を使用して XA トランザクションをクラスタ内の特定のインスタンスにバインドします。「第 6 章 負荷の管理」を参照してください。
- アプリケーションを変更して、不要な切断や再接続を防ぎます。

接続がマイグレートできない場合

場合によっては、マイグレーションができない状態にある接続でアプリケーションが要求を発行できます。「マイグレーションの基準」(30 ページ) を参照してください。

たとえば、アプリケーションがインスタンスに接続し、トランザクションに処理を実行する要求を発行する前に、接続を使用してテンポラリ・テーブルを作成した場合は、接続がマイグレートできません。それが非所有者インスタンスへの接続であれば、Adaptive Server は要求を処理できず、アプリケーションはプロトコル固有のエラー・コードを受け取ります。X/Open-XA プロトコルの場合は、TM がリターン・ステータスとして XAER_RMERR を受け取ります。これは、クラスタが XA コマンドを処理していないことを示します。トランザクションはロールバックされておらず、まだ所有者インスタンスで実行されています。

別の接続でコマンドを再発行して、このエラーに対処してください。XAER_RMERR は、多くのエラー状況で返される一般的なエラーです。関数 xact_connmigrate_check と xact_owner_instance を使用すると、XAER_RMERR エラーが、マイグレートできない接続によるものかどうかを確認できます。「設定とシステムの問題」(174 ページ) を参照してください。

マイグレートできない接続による XAER_RMERR エラーを防ぐには

- XA アプリケーション用に使用される接続を一般的な用途に使用しないでください。接続がマイグレートできない状態のままになる可能性があります。
- トレース・フラグ 3960 を使用して、マイグレートできない接続で非所有者インスタンスに発行された commit および rollback コマンドの XAER_RMERR エラーを防ぎます。トレース・フラグ 3960 を使用すると、非所有者インスタンスが TM と所有者インスタンス間のプロキシの役割を果たし、要求を TM から所有者インスタンスに転送してから、応答を TM に送ります。

設定とシステムの問題

DTM の管理と設定の概要については、『Adaptive Server 分散トランザクション管理機能の使用』を参照してください。この章では、クラスタ Adaptive Server 上の DTM に固有の設定と管理の問題について説明します。

- クラスタに DTM を設定するとき、クラスタ名を RM と指定します。ASTC RPC と TM の設定でアプリケーションがクラスタ名を使用することを確認してください。

- DTM に関連した設定パラメータ (enable DTM、enable xact coordination、strict dtm enforcement) はクラスタ Adaptive Server ではクラスタに静的です。これらのパラメータの値を変更した場合は、クラスタを再起動して変更を反映させる必要があります。
- syscoordinations および systransactions システム・テーブルの instanceid カラムは、トランザクションの所有者インスタンスを識別します。値がゼロ (0) の instanceid は、所有者インスタンスが停止しているか、エラーが発生したため、現在はクラスタ・コーディネータがトランザクションの所有者インスタンスであることを示します。
- クラスタ Adaptive Server では次の 2 つの関数が DTM をサポートしています。
  - xact_owner_instance(*XID*) − 外部トランザクションが実行しているインスタンスを返します。
  - xact_connmigrate_check(*XID*) − 現在の接続が外部トランザクションを処理できるかどうかを決定します。*XID* パラメータはオプションです。

    外部トランザクション名 (XID) を指定する場合
    - 次の場合に 1 を返します。
      - 所有者インスタンスへの接続の場合。
      - 非所有者インスタンスへの接続で、接続はマイグレートできます。
    - それ以外は 0 を返します。

    XID を指定しない場合
    - 接続がマイグレートできる場合は、1 を返します。
    - それ以外は 0 を返します。
- クラスタのインスタンスが分散トランザクション分岐を実行している場合は、正常な `shutdown` *`instance_name`* コマンドを使用してインスタンスを停止しないで、次のコマンドを使用してください。

  ```
  shutdown instance_name with nowait
  ```

  正常な `shutdown` *`instance_name`* コマンドは、インスタンスで実行している通常のユーザ・トランザクションはロールバックできますが、準備された状態の分散トランザクション分岐はロールバックできません。つまり、インスタンスにアクティブな分散トランザクションがない場合にのみ、正常なインスタンス shutdown を使用できます。

nowait オプションを使用してインスタンスを停止した場合は、フェールオーバ・リカバリがトリガされ、インスタンスの準備された分散トランザクション分岐がクラスタ・コーディネータで再インスタンス化されます。「「インスタンス・エラーの処理」(172 ページ)」参照。

注意 正常な `shutdown instance_name` コマンドは使用できませんが、正常な `shutdown cluster_name` コマンドは、クラスタのインスタンスで分散トランザクション分岐が実行している場合でも実行できます。

mount コマンドと *unmount* コマンドのサポート

Adaptive Server 15.5 以降のバージョンでは、mount database コマンドと unmount database コマンドを共有ディスク・クラスタで使用できます。このどちらかのコマンドが進行中にインスタンスでエラーが発生した場合は、コマンドがアボートされる可能性があります。インスタンスのフェールオーバ・リカバリが完了したら、mount database または unmount database を再発行してください。

sp_showplan の使用

sp_showplan を共有ディスク・クラスタの複数のノードで使用することはできません。sp_showplan はノードへの単一接続に固有のものです。

第 12 章

Veritas Cluster Server と Cluster Edition の使用

この章では、Veritas Storage Foundation for Sybase CE (SF for Sybase CE) 用に Cluster Edition を設定して使用する方法について説明します。

Cluster Edition バージョン 15.0.3 以降と Veritas SF for Sybase CE を統合すると、Cluster Edition で Veritas Storage Foundation の記憶領域管理テクノロジ、Cluster Server (VCS) アプリケーション、およびクラスタ管理機能を活用できます。

注意 Cluster Edition バージョン 15.0.3 と SF for Sybase CE を組み合わせると、クラスタが使用可能になり、データの整合性を維持できます。その他のバージョンの Veritas Storage Foundation には、統合に必要なコンポーネントが含まれていないので使用しないでください。

Cluster Edition と SF for Sybase CE を組み合わせることで、以下の機能を利用できます。

- Storage Foundation Cluster File System － Cluster Edition インストール・ファイル (*$SYBASE* にあるファイルとディレクトリ)、データベース・デバイス、クォーラム・デバイス、およびその他のアプリケーション・ファイルと共に使用できる汎用クラスタ・ファイル・システム。
- Cluster Volume Manager － 複数のクラスタ・ノードが共有する論理ボリュームを作成します。
- 動的マルチパス化 － 記憶領域の可用性とパフォーマンスを向上させます。

- サービス・グループ・ベースのアプリケーション管理 − モニタリングおよびフェールオーバ機能を提供します。複数のアプリケーション間での依存関係を作成できるので、アプリケーションが必要とするコンポーネント (ディスク・ボリュームなど) に障害が発生した場合、アプリケーション全体をフェールオーバできます。
- VCS エージェントおよび管理コンソールによる複数のクラスタ、アプリケーション、およびデータベース・サーバの統合管理。
- Veritas Volume Replicator を使用したハードウェア複写テクノロジおよびブロックレベル複写のサポート。

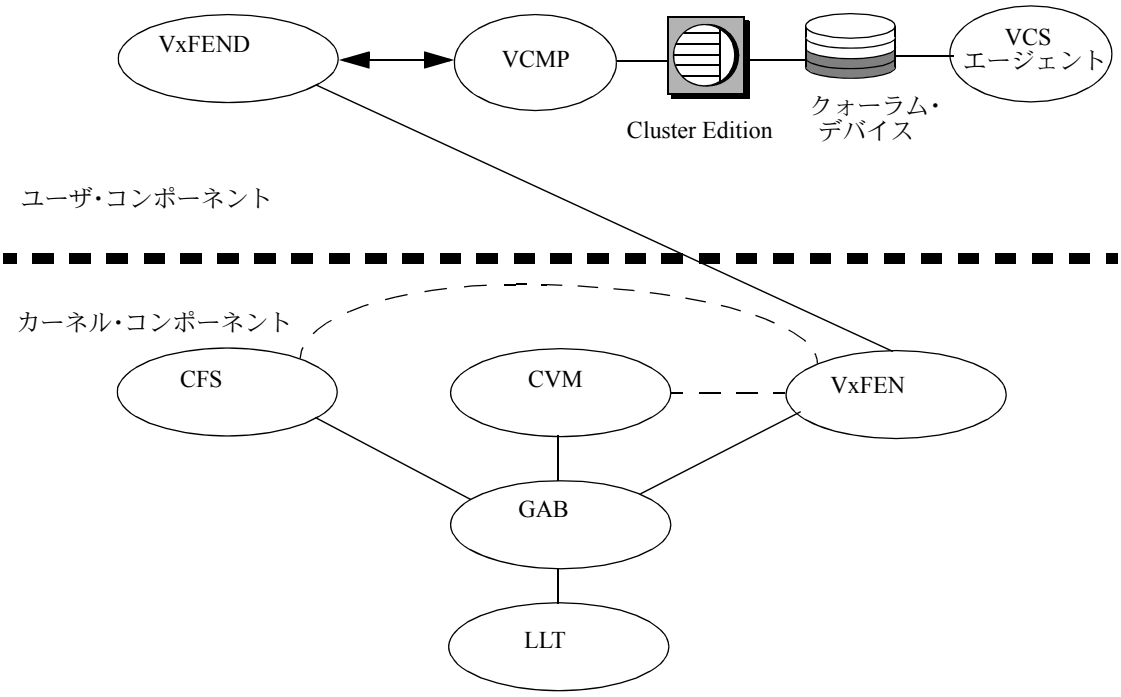
Cluster Edition と SF for Sybase CE には、メンバシップ管理と I/O フェンシングというクラスタ操作の 2 つの主要な要素が含まれています。

- メンバシップ管理 − Cluster Edition および SF for Sybase CE は、それぞれ独自のメンバシップ・マネージャを維持します。メンバシップ・マネージャは、以下の処理を行います。
  - クラスタのログインおよびログアウトのコーディネート
  - 障害の検出
  - 通信損出が発生したときにクラスタのどのメンバをアクティブにするかの決定 (アービトレーション)
  - 一貫したクラスタ・ビューの維持

  Veritas Cluster Membership プラグイン (VCMP) を使用すると、基盤となる Veritas メンバシップと Cluster Edition メンバシップを・サービスを同期することによって、2 つのメンバシップ・マネージャによるコーディネートの失敗、障害に対する異なるアービトレーション、およびクラスタのシャットダウンを回避できます。VCMP を使用すると、Cluster Edition は、Veritas メンバシップ・ビュー内のノードで実行するインスタンスを優先的にアービトレーションします。
- I/O フェンシング − メンバシップ・マネージャによって記憶領域の周囲にフェンスが構築され、書き込みを実行できるのは適切に動作するインスタンスまたはノードだけになるため、このように呼ばれます。I/O フェンシングを使用して、協調しないクラスタ・メンバによるデータ破損を防止します。Cluster Editions と SF for Sybase CE は、次のように I/O フェンシングをコーディネートします。
  - SF for Sybase CE は、すべてのフェンシングを管理および実行します。
  - Cluster Edition は、Cluster Edition メンバシップ変更のためにフェンシング・アクションが必要な SF for Sybase CE と通信できます。

Cluster Edition と SF for Sybase CE に含まれるコンポーネントを図 12-1 に示します。

図 12-1: クラスタ化された Veritas システム内のコンポーネント



以下のコンポーネントが Sybase によって提供されます。

- Cluster Edition － ノード上で実行するリレーショナル・データベース・サーバ。
- クォーラム・デバイス － クラスタの構成情報が含まれており、すべてのクラスタ・メンバによって共有されます。
- Veritas Cluster Membership プラグイン (VCMP) － メンバシップ変更メッセージを VxFEND から受信し、Cluster Edition のメンバシップ・サービスと通信します (Cluster Edition は、VCMP が許可するまでメンバシップ変更をブロックします)。

以下のコンポーネントが Veritas によって提供されます。

- LLT (Low Latency Transport) － VCS がクラスタで通信することを許可し、ハートビートおよび障害を検出します。
- GAB (Global Atomic Broadcast) － VCS メンバシップをコーディネートします。LLT が、障害イベント (ハートビートの損失など) に関する情報を GAB に提供します。
- CVM (Cluster Volume Manager) － メンバシップ情報を GAB から受信して VxFEN とコーディネートします。

- CFS (Cluster File System) － クラスタ内の複数のノードによって同時にマウントおよびアクセスできるファイル・システム。CFS は、分散ロック・マネージャを使用して、ノード間の一貫性を維持します。CVM は、SAN (Storage Area Network) と共に、基盤となる記憶領域を提供します。
- VxFEN (カーネル側 I/O フェンシング・コントロール) － VxFEN は、メンバシップ情報を GAB から受信し、必要に応じてフェンシングを実行します。
- VxFEND (ユーザ空間 I/O フェンシング・コントロール・デーモン) － VxFEN と通信するユーザ空間で実行するデーモン (このデーモンはカーネル内で実行するデーモンとは対照的です)。メンバシップ変更の際、VxFEN はメンバシップの変更を示すメッセージを VxFEND に送信します。VxFEND は、VCMP を介してメンバシップ変更に関する情報を Cluster Edition に通知します。
- VCS エージェント － Cluster Edition 用の VCS モニタリング・エージェント。VCS エージェントは、必要に応じて、Cluster Edition に障害が発生した場合にホスト・パニックをトリガします。また、VCS エージェントは、VCS で Cluster Edition を起動および停止するためにも使用されます。

サポートされているプラットフォーム、要件、および制限

Cluster Edition と SF for Sybase CE の組み合わせは、以下のプラットフォームでサポートされています。

- SPARC (Sun Sparc 64) 上の Solaris 10
- x86_64 (Linux AMD 64) 上の Redhat Enterprise Linux 5

必要なリリース・レベル、パッケージ、およびパッチに関する情報については、該当するプラットフォーム用の Cluster Edition のリリース・ノートおよびインストール・ガイドを参照してください。SF for Sybase CE の要件の完全なリストについては、『Veritas Storage Foundation for Sybase CE Release Bulletin』および『Installation and Configuration Guide』を参照してください (両書とも Veritas から提供されています)。

Cluster Edition は、native または vcs モードで実行します。native モードでは、Cluster Edition はクラスタ・メンバシップ・マネージャとコーディネートしません。vcs モードでは、クラスタ・メンバシップ・マネージャは Veritas クラスタウェアとコーディネートします。「メンバシップ・モード」(187 ページ) を参照してください。

Cluster Edition を vcs メンバシップ・モードで使用するには、次の手順に従う必要があります。

- SF for Sybase CE v5.0 をホスト・ノードにインストールして設定します。
- すべての Cluster Edition 記憶領域 (マスタ・デバイス、ユーザ・データベース・デバイス、およびクォーラム・デバイスを含む) が SF for Sybase CE 管理記憶領域からのものであることを確認します。これは、CVM ボリュームまたは CFS ファイルのいずれかです。
- Veritas LLT リンクと同じ物理ネットワーク用に Cluster Edition プライベート・ネットワークを設定します。「VCS 上での Cluster Edition のインストールおよび設定」(181 ページ) を参照してください。

Cluster Edition を vcs メンバシップ・モードで使用する場合は、以下の制限があります。

- 異なる Adaptive Server クラスタでパーティションが作成されている場合でも、1 つのハードウェア ノードで実行できるインスタンスは 1 つだけです。
- 1 つのクラスタで 4 つ以上のインスタンスまたはノードを設定することはできません。
- インスタンスに障害が発生した場合、ホスト・ノードがパニック再起動することがあります。このことを念頭において、関連付けられていない重要なアプリケーション (クラスタ化されていないその他の Adaptive Servers を含む) を同じノードで実行する場合は、ノードが再起動するときに、これらのアプリケーションをシャットダウンしてください。

VCS 上での Cluster Edition のインストールおよび設定

Cluster Edition を Veritas クラスタ・サブシステムにインストールして設定するには、Veritas のマニュアルおよび以下のセクションに記載されている手順を実行する必要があります。

Veritas から提供されている『SF Sybase CE Installation and Configuration Guide』の説明に従って次の手順 1 ～ 6 を実行します。

1. Veritas の SF for Sybase CE をインストールして設定します。
2. I/O フェンシングを Sybase モードで設定します。
3. Adaptive Server バイナリ・インストール (*$SYBASE*) 用の共有ディスク・グループ、ボリューム、およびマウント・ポイントを作成します。
4. Adaptive Server クォーラム・デバイス用の共有ディスク・グループ、ボリューム、およびマウント・ポイント (CFS を使用する場合) を作成します。
5. Adaptive Server データベース・デバイス用の共有ディスク・グループ、ボリューム、およびマウント・ポイント (CFS を使用する場合) を作成します。

6 “sybase” ユーザ (または同等のユーザ) に適切な所有権および手順 5 の記憶領域へのアクセスがあることを確認します。

このマニュアルの「Cluster Edition のインストール」(182 ページ)、「Storage Foundation 統合用の新しい Adaptive Server クラスタの作成」(183 ページ)、および「SF for Sybase CE を使用するための既存のクラスタの変換」(184 ページ) のセクションの説明に従って以下の手順 7 および 8 を実行します。

7 Adaptive Cluster Server Edition をインストールします。

8 Adaptive Server クラスタを作成するか、既存のクラスタを使用できるように準備します。

『SF Sybase CE Installation and Configuration Guide』の説明に従って次の手順 9 を実行します。

9 SF for Sybase CE インストーラを使用して、Adaptive Server クラスタ用の VCS サービス・グループを作成します。クラスタが vcs メンバシップ・モードになります。

注意 Cluster Edition を vcs メンバシップ・モードで設定した後、クラスタを VCS サービス・グループの一部として起動します。Cluster Edition が vcs モードであるときに VCS サービス・グループの外部で開始すると、Cluster Edition はハングしてシャットダウンします。

Cluster Edition のインストール

該当するプラットフォーム用のインストール・ガイドの「第 3 章 クラスタの作成と起動」で説明する手順を実行し、SF for Sybase CE を使用してクラスタを作成します。

注意 Cluster Edition は、VCS サービス・グループを作成する前にインストールします。

Sybase では次のことをおすすめします。

- プライベート・インストールではなく共有インストールで Cluster Edition を作成します。
- Storage Foundation Cluster File System (SFCFS) マウント・ポイント上にリリース・ディレクトリ (*$SYBASE*) を作成します。Cluster Edition をインストールする前に SF for Sybase CE をインストールします。Veritas から提供されている『SF Sybase CE Installation Guide』を参照してください。

Storage Foundation 統合用の新しい Adaptive Server クラスタの作成

sybcluster および Adaptive Server プラグインはいずれも、SF for Sybase CE がホスト・ノードにインストールされているかどうかを自動的に検出し、適切な記憶領域およびネットワーク相互接続を選択するためのガイドを提供します。クラスタを作成する際は、以下の点に注意してください。

- すべての記憶領域は、CVM 制御ボリュームまたは SFCFS ファイル・システムを使用して、SF for Sybase CE 管理デバイス上に配置する必要があります。以下の制限を適用してください。
  - CVM 制御ボリュームに直接アクセスする場合、(*/dev/vx/dsk/* でなく) */dev/vx/rdsk* を使用して、ロー (文字) デバイスとしてアクセスする必要があります。
  - ローカル・テンポラリ・データベース用に追加されたプライベート・デバイスをローカルの SF for Sybase CE 以外の記憶領域デバイスに配置できます。
  - Veritas Storage Foundation の Sybase CE バージョンから提供されている記憶領域以外は使用しないでください。Cluster Edition を vcs メンバシップ・モード用に設定する場合を除き、SF for Sybase CE によって提供されている記憶領域は使用しないでください。データベースが破損する可能性があるため、Adaptive Server は VxFEN 制御の I/O フェンシングを native モードでは実行できません。
- VCMP ソケット・パスを設定します。デフォルトでは、VCMP は */tmp/vcmp_socket* を使用して Veritas と通信します。
  a　*$SYBASEASE-15_0/install/RUN_instance name* にある runserver ファイルに以下を追加します。

```
export VCMP_SOCKET=new_vcmp_socket_path
```

次に例を示します。

```
export VCMP_SOCKET=/tmp/my_socket
```

  b　vxfend リソースを変更します。

```
hares -modify vxfend Arguments "%-m sybase -k new_vcmp_socket_path"
```

次に例を示します。

```
hares -modify vxfend Arguments "%-m sybase -k /tmp/my_socket"
```

- Cluster Edition のプライマリおよびセカンダリ・クラスタ相互接続は、Veritas LLT 管理ネットワーク上で実行する必要があります。LLT でどのネットワークを使用するかは、*/etc/lttab* ファイルを参照して確認します。以下の例では、LLT は *eth0* および *eth1* ネットワークで実行します。

```
[admin@sdcc2 ~]$ cat /etc/llttab
set-node sdcc2
set-cluster 53
link eth0 eth-00:19:b9:b0:73:13 - ether - -
link eth1 eth-00:19:b9:b0:73:15 - ether - -
```

注意 Cluster Edition の相互接続では、リンクの IP アドレスを設定する必要があります。LLT では IP アドレスが必要ないため、相互接続の個別設定が必要になることがあります。オペレーティング・システムのマニュアルを参照してください。

ifconfig コマンドを使用して、特定のネットワークに割り当てられている IP アドレスを確認します。このアドレスを使用して、プライマリまたはセカンダリ・クラスタ相互接続のいずれかを設定できます。次の例では、インタフェース *eth0* に 10.22.104.141 の IPv4 アドレスがあります。

```
[admin@sdcc2 ~]$ /sbin/ifconfig -a eth0
eth0   Link encap:Ethernet  HWaddr  00:19:B9:B0:73:13
       inet addr:10.22.104.141  Bcast:10.22.104.255  Mask:255.255.255.0
       inet6 addr: fd77:55d:59d9:168:219:b9ff:feb0:7313/64 Scope:Global
       inet6 addr: fe80::219:b9ff:feb0:7313/64 Scope:Link
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:19710010 errors:0 dropped:0 overruns:0 frame:0
       TX packets:11707065 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:1000
       RX bytes:7506123489 (6.9 GiB)  TX bytes:1839949672 (1.7 GiB)
       Interrupt:169 Memory:f8000000-f8012100
```

クラスタを作成した後、クラスタを vcs モードに戻します。

SF for Sybase CE を使用するための既存のクラスタの変換

SF for Sybase CE と連動するように Cluster Edition を変換する場合、Storage Foundation ソフトウェアがすべてのデータベース記憶領域を管理するようにし、Cluster Edition 相互接続が LLT リンク上で稼動するようにする必要があります。

データベース記憶領域の移動

既存のデータベースを Storage Foundation 以外の記憶領域から Storage Foundation 記憶領域に移動する方法は、データベース・デバイスの設定、サイズ、および数に応じてサイトごとに異なります。

相互接続の再設定

相互接続を再設定するには、次の手順に従います。

1　ifconfig コマンドを使用して、LLT リンクの適切な IP アドレスを確認します。

2　shutdown cluster コマンドを使用して、クラスタをシャットダウンします。

3　qrmutil を使用して相互接続を再設定します。適切なインスタンス名、プライマリ・アドレス、およびセカンダリ・アドレスを使用して、このコマンドをクラスタ内の各インスタンスに対して実行します。

```
qrmutil --quorum_dev=path_to_quorum_device --instance=instance_name
--primary-address=address_of_primary_interconnect
--secondary-address=address_of_secondary_interconnect
```

次の例では、192.168.0.1 のプライマリ・アドレスおよび 192.168.0.2 のセカンダリ・アドレスを使用するように “ase1” インスタンスを設定します。

```
qrmutil --quorum_dev=/sybase_cfs/quorum.dat --instance=ase1
--primary-address=192.168.0.1 --secondary-address=192.168.0.2
```

各インスタンスに手順 3 を実行する代わりに、現在の設定をファイルに抽出し (手順 4 参照)、各インスタンス用にファイルを編集して、--load-config パラメータを使用してクォーラム・デバイスでファイルを再ロードすることもできます。

4　qrmutil --extract-config パラメータを使用して、新しいクラスタ設定のバックアップを作成します。次の例では、クラスタ設定が *quorum.bak* ファイルにバックアップされます。

```
qrmutil --quorum_dev=/sybase_cfs/quorum.dat
--extract-config=/sybase_cfs/quorum.bak
```

5　Cluster Edition を再起動します。

VCS 制御下でのクラスタの管理

Cluster Edition と SF for Sybase CE を連動させるソフトウェアをアクティブにするには、Cluster Edition メンバシップ・モードを vcs に変更して、Cluster Edition を VCS の制御下に配置する必要があります。

VCS サービス・グループを作成する手順については、『SF Sybase CE Installation and Configuration Guide』の「Configuring a Sybase ASE CE cluster under VCS control using the SF for Sybase CE installer」を参照してください。

インスタンスの起動と停止

vcs メンバシップ・モードの Cluster Edition は、Transact SQL の shutdown cluster、shutdown cluster with nowait、sybcluster shutdown コマンド、および Adaptive Server プラグインの起動および停止の選択を認識しますが、VCS エージェントを使用してインスタンスを起動および停止することをおすすめします。

vcs メンバシップ・モードで実行するインスタンスは、VCS サービス・グループを使用して起動および停止してください。CS サービス・グループは、VCS エージェントを使用してインスタンスを起動および停止します。VCS エージェントは、VCS Cluster Manager (Java コンソール) または hagrp コマンド・ライン・ユーティリティを使用して制御します。

『SF Sybase CE Administrator's Guide』の「Administering the Sybase Agent」を参照してください。VCS エージェントおよびリソース管理の詳細については、『Veritas Cluster Server User's Guide』を参照してください。

インスタンスの追加および削除

クラスタへのインスタンスの追加、またはクラスタからのインスタンスの削除を行うには、次の手順を実行します。

- VCS へのホスト・ノードの追加、または VCS からのホスト・ノードの削除 (インスタンスを削除する場合はオプション)
- インスタンスの追加または削除

これらのタスクはいずれの順序でも実行できますが、インスタンスを起動するには、両方のタスクを完了する必要があります。「第 14 章 Adaptive Server プラグインによるクラスタの管理」および「第 15 章 sybcluster を使用したクラスタの管理」を参照してください。VCS クラスタにノードを追加する際の詳細については、『SF Sybase CE Administrator's Guide』の「Adding a node to SF Sybase CE clusters」を参照してください。

ユーザ接続数の増加

ユーザ接続数を増やす場合は、ファイル記述子の制限値も増やす必要があります。ただし、VCS で実行している Veritas は、あるユーザ・セッションから別のユーザ・セッションにファイル記述子の制限値の変更を反映しません。たとえば、次のコマンドを使用して現在のセッションでファイル記述子の制限値を変更するとします。

```
ulimit -n 8194
```

後続の UNIX セッションでは、ファイル記述子の数にデフォルト設定 (1024) が使用されます。その結果、Veritas VCS は、クラスタを管理してインスタンスをオンラインにする間に、Linux オペレーティング・システムに (新しい UNIX セッションを開始する) ユーザ sybase としてログインし、ファイル記述子の数にデフォルト値を使用します。

すべての sybase ユーザ・セッションに正しいファイル記述子の制限値を設定するには、以下の行を */etc/security/limits.conf* に追加します。

```
sybase hard nofile 8096
sybase soft nofile 8096
```

文字セットまたはソート順を変更する

文字セットまたはソート順を変更するには、次の手順に従います。

1　VCS コマンドを使用して Adaptive Server を停止します。

2　メンバシップ・モードを次のように native に変更します。

```
qrmutil -Q quorum_file --membership-mode="native"
```

クラスタ内の 1 つのインスタンスを手動または sybcluster を使用して再起動します。

3　文字セットを変更するには、charset を実行します。次に例を示します。

```
$SYBASE/ASE-15_0/bin/charset -Usa -P nocase.srt utf8
```

4　ソート順を変更するには、デフォルトの sortorder id 設定パラメータをリセットします。次に例を示します。

```
isql -Usa -P
sp_configure 'default sortorder id', 101, 'utf8'
```

5　isql からクラスタを停止します。

6　インスタンスを再起動します。Adaptive Server によって文字セットとソート順が再設定されます。Adaptive Server を停止します。

7　メンバシップ・モードを次のように VCS に変更します。

```
qrmutil -Q quorum_file --membership-mode='vcs'
```

8　VCS コマンドを使用してクラスタを再起動します。

メンバシップ・モード

Cluster Edition バージョン 15.0.3 以降には、native および vcs の 2 つのメンバシップ・モードが含まれています。メンバシップ・モードによって Cluster Edition の内部メンバシップ・マネージャ (CMS) が独立して機能するか (native モード、デフォルト設定)、Veritas クラスタウェアと連動するか (vcs モード) が決定します。

メンバシップ・モードの確認

現在のメンバシップ・モードは、*@@membershipmode* グローバル変数および sybcluster show membership mode を使用して確認できます。Cluster Edition が VCS クラスタ・サブシステムと連動している場合は、*@@membershipmode* および sybcluster は、“vcs” を返します。Cluster Edition が VCS クラスタ・サブシステムなしで稼働している場合は、“native” を返します。次に例を示します。

```
select @@membershipmode
vcs
```

メンバシップ・モードの変更

VCS サービス・グループを作成するとインスタンスは自動的に vcs モードに変更されるので、通常はメンバシップ・モードを変更する必要はありません。

メンバシップ・モードを変更するには、qrmutil --membership-mode オプションを使用します。クラスタはシャットダウンする必要があります。構文は次のとおりです。

qrmutil --[-Q | quorum-dev=*quorum_device* --membership-mode=*mode*

各パラメータの意味は次のとおりです。

- vcs − Cluster Edition が VCS クラスタ・サブシステムと連動していることを示します。
- native − Cluster Edition が VCS クラスタ・サブシステムなしで稼働していることを示します。

次に例を示します。

```
qrmutil --quorum-dev=/dev/vx/rdsk/quorum_dg/quorumvol --membership-mode=vcs
```

Cluster Edition を手動で設定して、VCMP をロードし、VCS 統合をアクティブにするには、クラスタ設定ファイルに「membership mode = vcs」という行を追加します。この行を追加した後、Cluster Edition は VCS サービス・グループの一部として起動します。Veritas クラスタ・サブシステムなしで Cluster Edition を設定するには、クラスタ設定ファイルに「membership mode = native」と入力します。

次の例では、メンバシップ・モードが vcs に設定されます。

```
[cluster]
name = isles
max instances = 4
primary protocol = udp
master device = /sybase_cfs/data/master.dat
config file = /sybase_cfg/ASE-15_0/ase.cfg
interfaces path = /sybase_cfg/ASE-15_0/ase.cfg
membership mode = vcs
```

membership mode = 行のパラメータとして vcs または native を指定しないと、または行そのものを含めないと、Cluster Edition は native モードで起動します。

マスタ・データベースを生成した後にクラスタ設定ファイルでメンバシップ・モードを変更しても何の影響もありません。マスタ・データベースを生成した後は、qrmutil を使用してメンバシップ・モードを変更します。

qrmutil --quorum-dev= *quorum_path* --membership-mode=[vcs|native]

障害シナリオの理解

Cluster Edition は、メンバシップ変更を VCS とコーディネートし、SF for Sybase CE を使用して I/O フェンシングを実行するので、Cluster Edition メンバシップ・マネージャは、VCS がそのメンバシップを再設定して I/O フェンシングを実行するのを待ちます。しかし、障害によっては、VCS メンバシップの再設定が必要ない場合でも、Adaptive Server で I/O フェンシングが必要になることがあります。I/O フェンシングを取得するために、Adaptive Server は、クォーラム・デバイスを介してフェンシング対象のノード上にある VCS エージェントにメッセージを送信します。メッセージを受信した VCS エージェントは、ノードでパニックを発生させ、その結果、VCS の再設定がトリガされます。

障害が発生したときの VCS システムの動作は、状況によって異なります。

- ホスト・ノードのクラッシュ － VCS によって検出されます。VCS は、クラスタ・メンバシップを再設定して、障害が発生したノードから I/O フェンシングを実行します。Cluster Edition は、VCS から新しいメンバシップを受信して、リカバリを実行します。
- ホスト・ノードのハング － VCS LLT タイムアウトが発生し、Cluster Edition ハートビートのタイムアウトが発生します。VCS は I/O フェンシングを再設定して実行します。Cluster Edition は、VCS から新しいメンバシップを受信して、リカバリを実行します。ハングしているノードは、共有記憶領域に書き込むことができず、最終的にパニックが発生します。
- インスタンスの障害またはハング － 残りのインスタンスがハートビートの障害を検出して、メンバシップ・アービトレーションを実行します。クラスタは、障害が発生したインスタンスを含むホスト・ノードにある VCS エージェントにメッセージを送信し、ホスト・ノード・パニックがトリガされます。この時点から、VCS は上記の「ホスト・ノードのクラッシュ」シナリオで説明したように動作します。
- インスタンスの shutdown with nowait － Cluster Edition は VCS メンバシップ・モードでの VCS メンバシップの変更をトリガしません。

- 記憶領域アクセスの損失 − マルチパスが設定されていて、少なくとも 1 つの記憶領域パスが可視の状態にある場合、Cluster Edition への影響はありません。ただし、記憶領域へのすべてのパスが失われた場合、VCS はホスト・パニックを発生させます。このホスト・パニックは、上記の「ホスト・ノードのクラッシュ」シナリオで説明したように処理されます。
- 相互接続の損失 − 応答は代替通信チャネルの可用性によって異なります。Adaptive Server と VCS は両方とも代替チャネルにトラフィックを切り替えます。Adaptive Server に可視チャネルがなく、VCS に可視チャネルがある場合、クラスタは上記の「インスタンスの障害またはハング」シナリオで説明したように動作します。VCS に可視チャネルがない場合は、上記の「ホスト・ノードのクラッシュ」シナリオで説明したように動作します。

VCS のトラブルシューティング

エラー・ログに含まれる情報は、問題の原因および解決方法の特定に役立ちます。エラー・ログのデフォルトの場所を以下に示します。

- Cluster Edition − *$SYBASE/$SYBASE_ASE/install/instance_name.log*
- Veritas − */var/VRTSvcs/log/engine_A.log*
- Linux − */var/log/messages*
- Solaris − */var/adm/messages*

注意 以下に示す問題の多くは、Veritas コマンド (hagrp や hares など) で解決できます。構文および使用方法については、Veritas のマニュアルを参照してください。

Cluster Edition が起動しない

Cluster Edition が起動しない場合、いくつかの原因が考えられます。

VCS が Cluster Edition を起動できない場合、または Cluster Edition が既に実行していることが検出された場合、Cluster Edition でリソース障害が発生します。hagrp -clear コマンドを使用して、リソース障害をクリアします。次の例は、“asegrp1” という VCS グループのリソース障害をクリアします。

```
hagrp -clear asegrp1
```

VCS は ps コマンドを使用して dataserver プロセスの場所を見つけます。dataserver バイナリの名前を変更しないでください。

hares -display を使用して、Cluster Edition リソースの *Home* 属性を確認します。*Home* は *$SYBASE* を指し、dataserver バイナリが *$SYBASE/ASE-15_0/bin* に存在する必要があります。*Home* が正しい値を指していない場合は、hares -modify コマンドを使用して値を変更します。

VCS は *RUN_server_name* ファイルを使用してインスタンスを起動します。*RUN_server_name* ファイルが存在しない場合、VCS は、*Home*、*Server*、および *Quorum_dev* 変数で定義されたパラメータを使用して、インスタンスを起動する実行文字列を構築します。

```
HOME/ASE-15_0/bin/dataserver -sserver_name -QQuorum_dev
```

リソース障害が発生した場合は、以下のことを確認してください。

- クラスタ内の各インスタンスに個別の *RUN_server_name* ファイルが存在する。
- インスタンスの名前が *RUN_server_name* ファイルにサフィックスとして含まれている。
- 各 *RUN_server_name* ファイルが実行可能である。
- 各 *RUN_server_name* ファイルに dataserver バイナリへの正しいパスが含まれている。
- *Quorum_dev* 属性値が既存のクォーラム・デバイスの正しいパスを使用している。*Quroum_dev* 値を参照するには、hares -display を使用します。以下の例では、VCS リソースの名前は “aseres1” です。

  ```
  hares -display aseres1 -attribute Quorum_dev
  ```

  パスを変更するには、hares -modify を使用します。

  ```
  hares -modify aseres1 Quorum_dev /sybase/quorum
  ```

- *Server* 属性はインスタンス名に対応します。*Server* 属性を確認するには、hares -display コマンドを使用します。以下の例では、VCS リソースの名前は “aseres1” です。

  ```
  hares -display aseres1 -attribute Server
  ```

  値を変更するには、hares -modify を使用します。

  ```
  ohares -modify aseres1 Server ase1 -sys hpcblade1
  ```

- hares -display を使用して VCS リソースが有効化されている。

  ```
  hares -display aseres1 -attribute Enabled
  ```

  必要な場合は、hares -modify を使用して変更します。

  ```
  hares -modify aseres1 Enabled 1
  ```

Veritas ログ:「Sybase home directory does not exist」

Sybase ホーム・ディレクトリ (*$SYBASE*) は CFS マウント・ディレクトリ上に存在する必要があるので、CFS マウント・ディレクトリが使用できない場合はインスタンスを起動できません。

$SYBASE ディレクトリが使用可能であることを確認するには、hagrp -dep を使用して、*ASE* グループが *cfsmount* グループに依存していることを確認します。

```
hagrp -dep
```

依存関係がない場合は、hagrp -link を使用して依存関係を再度確立します。

```
hagrp -link asegrp1 cfsmount1 online local firm
```

インスタンス・ログ:「failed to bind to socket」

このエラーは、インスタンスに VCMP ソケット・エラーがある場合に発生します。"sybase" ユーザは、ソケットに対する読み取りと書き込み、および VxFEN との通信を行うために、VCMP ソケット (*/tmp/vcmp_socket*) を所有している必要があります。

"sybase" ユーザが */tmp/vcmp_socket* を所有していることを確認するには、ls -l UNIX コマンドを使用します。

```
ls -l /tmp/vcmp_socket
```

"sybase" ユーザが */tmp/vcmp_socket* を所有していない場合は、UNIX chown コマンドを使用して所有権を変更します。

```
chown sybase /tmp/vcmp_socket
```

インスタンス・ログ:「Membership service failed to receive initial message from Veritas cluster membership after 301 seconds.Exiting...」

このエラーは、インスタンスで VCMP がタイムアウトしたために発生します。この原因および解決方法を次に示します。

- *ASE* グループをオンラインにしたときに VxFEND リソースが使用できない。VxFEND リソースが使用可能であることを確認するには、次の手順に従います。

  a VxFEND リソースをオフラインにします。

  ```
  hares -offline vxfend -sys <server name>
  ```

  b *ASE* グループをオンラインにします。

  ```
  hagrp -online ASE -sys <server name>
  ```

- 別のインスタンスがソケットを使用している。ノード上でその他の dataserver プロセスが実行していないことを確認するには、UNIX ps コマンドを使用します。

  ```
  ps -ef | grep dataserver
  ```

- VxFEND リソースが有効でない。VxFEND が有効であることを確認するには、hares -display を使用します。

  ```
  hares -display vxfend -attribute Enabled
  ```

  有効でない場合は、hares -modify を使用します。

  ```
  hares -modify vxfend Enabled 1
  ```

- *ASE* リソースが VxFEND リソースに依存していない。*ASE* リソースの依存関係を確認するには、hares -dep を使用します。

  ```
  hares -dep
  ```

  依存関係がない場合は、hares -link を使用して依存関係を再度確立します。

  ```
  hares -link aseres1 vxfend
  ```

- 別の Adaptive Server または VxFEND リソースがリソース起動と競合している。その他の Adaptive Server または VxFEND リソースが有効でないことを確認するには、hares -display を使用します。

  ```
  hares -display
  ```

  問題を修正するには、hares -modify を使用します。

  ```
  hares -modify ASE2 Enabled 0
  ```

インスタンス・ログ:「Failed to open quorum device '*device_path*'. OS error 13, 'Permission denied'」

このエラーは、VCS ロー・ボリュームのパーミッションに “sybase” ユーザのアクセスが含まれていないために発生します。“sybase” ユーザに VCS ボリュームの読み取りおよび書き込みパーミッションを付与します。

```
vxedit -g sybasedg set user=sybase group=sybase mode=660 quorum
vxedit -g sybasedg set user=sybase group=sybase mode=660 master
vxedit -g sybasedg set user=sybase group=sybase mode=660 sybasehome
```

インスタンス・ログ:「basis_dsizecheck: attempt to open device 'device_path' failed, system error is: Permission denied」

このエラーは、上記のエラーと同じ理由で発生します。同じ手順で問題を修正します。

インスタンス・ログ:「The configuration area in master device appears to be corrupt.」

Linux ログに以下のようなメッセージも表示されている場合があります。

```
Linux log: READ CAPACITY failed
Linux log: reservation conflict
Linux log: attempt to access beyond end of device
```

このエラーは、マスタ・データベースが壊れている場合、またはクラスタ化されたファイル・システムによってハングが発生した場合に発生します。この問題は、ディスク配列が Veritas 用に適切に設定されていないために発生します。問題を修正するには、次の手順に従います。

- ディスク配列ファームウェアが最新のものであることを確認します。
- (Linux のみ) Veritas Volume Manager のディスク初期化形式がスライスであることを確認します。
- 1 つのホスト用に予約されている LUN (論理ユニット番号) がその他のホストから認識できるようにディスク配列接続モード設定を修正して、SCSI-3 フェンシングが配列で適切に有効化されていることを確認します。一般的に、これらの設定は次のようになります。
  - Set Host Mode Option = 19 (ON)
  - Set Host Mode Option = 254 (ON)
  - Set Host Mode Option = 186 (ON)
  - Set Host Mode Option = 60 (ON)
  - Host Mode = 0A
  - Path Switch Mode (Active/Active) = 1 (Enabled)
  - No_RSV_Conf Mode = 1 (Enabled)
  - Persistent RSV Cluster Mode = 1 (Enabled)
  - Unique Reserve Mode = 1 (Enabled)

特定のオペレーティング・システムおよびハードウェアに固有の設定については、Veritas のマニュアルを参照してください。

Veritas ログ:「*Path* not found」

このエラーは、dataserver バイナリを起動する UNIX sh コマンドを VCS が見つけることができないために発生します。Veritas では *$PATH* 環境変数の最初の部分がトランケートされるので、Veritas が sh コマンドを見つけていることを確認します。*$PATH* の値を確認して、Veritas を再起動します。

1　*$PATH* の値を確認します。

```
echo $PATH
```

2　*$PATH* の値を変更して sh コマンドを含めます。

```
export PATH=/bin:/sbin:/usr/bin:/usr/sbin
```

3　Veritas を停止します。

```
Issue hastop -local
```

4　Veritas を再起動します。

```
hastart
```

起動後に VCS によってインスタンスがシャットダウンされ、リソース障害が発行される

hagrp -clear を使用して、リソース障害をクリアします。次のコマンドは、“asegrp1” という VCS グループのリソース障害をクリアします。

```
hagrp -clear asegrp1
```

このエラーが発生した場合、インスタンスによってエラー・ログにエラーが書き込まれることがあります。

- `Failed to identify instance` − インスタンスを起動するために VCS で間違った名前が使用されています。問題を修正するには、次の手順に従います。
  - 各 *RUN_server_name* ファイルにクォーラム・デバイスの正しいパスが含まれていること、および *server_name* の値がインスタンスの名前と同じであることを確認します。
  - qrmutil を使用して、インスタンス名とホスト名を中心にクォーラム設定を確認します。

    ```
    qrmutil -Qquorum_path --display=all
    ```

- `Shutdown with nowait detected` − インスタンスが実行しているかどうかが VCS で確認できません。VCS は、実行文字列内のサーバ名と共に UNIX ps コマンドを使用して、インスタンスが実行しているかどうかを確認します。ps コマンドで各インスタンスを見つけることができることを確認するには、次の手順に従います。

- (Solaris のみ) *RUN_server_name* ファイル内のインスタンス名が最初の 80 文字内にあることを確認します。
- hares -display を使用して、*ASE* リソースの *Server* 属性を確認します。*Server* 属性は、インスタンス名に対応している必要があります。次の例では、“aseres1” リソースのインスタンス名が表示されます。

  ```
  hares -display aseres1 -attribute Server
  ```

  必要な場合は、hares -modify を使用して値を変更します。

  ```
  hares -modify aseres1 Server ase1 -sys hpcblade1
  ```

  リソースの起動に時間がかかると、Veritas によって起動中に dataserver プロセスが停止され、リソース障害が発行されることがあります。このようなリソース障害を防止するために、以下の Sybase VCS モニタリング・パラメータを調整して、インスタンスの起動に対する許容時間を長くします。

  - *ASE* リソースのモニタ・チェック間の時間を調整します。次に例を示します。

    ```
    hatype -modify Sybase MonitorInterval 60
    ```

  - *ASE* リソースがオンラインであることを確認するモニタ・チェックの回数を調整します。次に例を示します。

    ```
    hatype -modify Sybase OnlineWaitLimit 10
    ```

  - *Sybase* リソースのリソース障害を発行するまでに、モニタ・チェックで障害を検知する回数を調整します。次に例を示します。

    ```
    hatype -modify Sybase ToleranceLimit 2
    ```

- `Engine exited with signal 11` － Linux プラットフォーム上のインスタンスに障害が発生しました。問題を修正するには、*RUN_server_name* ファイル内の *LD_POINTER_GUARD* 環境変数を再設定します。
  - SUSE プラットフォームの場合、次のコマンドを入力します。

    ```
    export LD_POINTER_GUARD=1
    ```

  - RedHat プラットフォームの場合、次のコマンドを入力します。

    ```
    export LD_POINTER_GUARD=0
    ```

VCS がインスタンス・リソースをシャットダウンできない

この問題は、VSC が isql を使用してインスタンスに接続できない場合に発生します。VCS は、isql を使用してインスタンスに接続し、インスタンスをシャットダウンします。isql は、接続パラメータの interfaces ファイルを使用します。次のことを確認します。

- interfaces ファイルがリリース・ディレクトリ (*$SYBASE*) で定義されていて、適切に設定されていること。
- 定義された各インスタンスの定義が interfaces ファイルに含まれていること。
- リリース・ディレクトリに *SYBASE.sh* または *SYBASE.csh* ファイルが含まれていて、正しく設定されていること。

VCS は SA および SApswd コマンドを isql と一緒に使用し、インスタンスに接続してインスタンスをシャットダウンします。hares -display を使用して、*ASE* リソースの SA および SApswd 属性を確認します。

```
hares -display aseres1 -attribute SA
hares -display aseres1 -attribute SApswd
```

必要な場合は、hares -modify を使用して属性を変更します。

```
hares -modify aseres1 SA sa
hares -modify aseres1 SApswd ""
```

VCS グループのリソース障害

cfsmount グループによるリソース障害が発見された場合は、次の手順に従います。

- hagrp -dep を使用して、*cfsmount* グループが *cvm* および *cvmvoldg* グループに依存していることを確認します。

  ```
  hagrp -dep
  ```

  hagrp -link を使用して、依存関係を修正します。

  ```
  hagrp -link cfsmount1 cvm online local firm
  hares -link cfsmount1 cvmvoldg1
  ```

- vxdg list を使用して、VCS マウントおよびボリューム・ディスク・グループがすべてのノードで有効であることを確認します。

  ```
  vxdg -o alldgs list
  ```

  vxdg deport および import を使用して、再度有効にします。

  ```
  vxdg deport disk_group_name
  vxdg -s import disk_group_name
  ```

- vxdisk list を使用して、VCS マウントおよびボリューム・ディスクがオンラインであることを確認します。

  ```
  vxdisk list
  hastop -local
  ```

  必要な場合は、各ノードの VCS を再起動します。

  ```
  /etc/init.d/vxfen stop
  /etc/init.d/vxfen start
  hastart
  ```

- VCS ボリュームが使用中である場合、CFS マウントで障害が発生することがあります。競合をクリアするために、クラスタ内のすべてのノードの再起動が必要になる場合があります。オペレーティング・システムのマニュアルを参照してください。

cvm_clus グループによるリソース障害が見つかった場合、hares -dep を使用して、このリソースが *cvm_vxconfigd* リソースに依存していることを確認します。

```
hares -dep
```

hares -link を使用して、依存関係を修正します。

```
hares -link cvm_clus cvm_vxconfigd
```

vxfsckd グループによるリソース障害が見つかった場合、hares -dep を使用して、このリソース・グループが *cvm_clus* リソースに依存していることを確認します。

```
hares -dep
```

hares -link を使用して、依存関係を修正します。

```
hares -link vxfsckd cvm_clus
```

VCS が起動しない

この問題は、ノードが VCS クラスタに参加できない場合に発生します。

このエラーが Solaris または Linux エラー・ログに表示された場合は、VCS が I/O フェンシングを起動できないためにノードがクラスタに参加できないことを示します。

```
CVMCluster:???:monitor:node - state: out of cluster
```

- vxdg list を使用して、すべてのノードで *vxfencoordg* ディスク・グループが有効であることを確認します。

  ```
  vxdisk list
  ```

  vxfencoordg ディスク・グループが有効でない場合は、グループを解放して、再度インポートします。

  ```
  vxdg deport vxfencoorddg
  vxdg -t import vxfencoorddg
  ```

- vxdisk list を使用して、*vxfencoordg* ディスクがオンラインであることを確認します。

  ```
  vxdisk list
  ```

  vxfencoordg ディスクがオンラインでない場合は、各ノードで VCS を再起動します。

  ```
  hastop -local
  /etc/init.d/vxfen stop
  /etc/init.d/vxfen start
  hastart
  ```

- vxfenadm -g を使用して、*vxfencoordg* ディスク上に予約が存在することを確認します。

  ```
  vxfenadm -g
  ```

  予約が存在しない場合は、VCS を停止し、vxfenclearpre を使用して VCS 予約をクリアした後で VCS を再起動します。

  ```
  hastop -local
  /etc/init.d/vxfen stop
  vxfenadm -g all -f /etc/vxfentab
  vxfenclearpre
  /etc/init.d/vxfen start
  hastart
  ```

  新しいノードが既存のクラスタに参加した場合、gapconfig -x を使用してグループ・メンバシップ設定を再度適用する必要が生じる場合があります。

  ```
  gabconfig -x
  ```

次のメッセージが Solaris または Linux のエラー・ログに表示された場合は、Veritas 設定デーモンが起動できないことを示します。

```
ERROR: IPC Failure: Configuration daemon is not accessible
```

設定デーモンをクリアして再起動します。

```
vxconfigd -k
"vxiod set 10
"vxconfigd -m disable
"vxdctl init
"vxdctl initdmp
"vxdctl enable
```

次のメッセージが VCS エラー・ログに表示された場合は、vxdg init を実行したときにディスク・グループの初期化が失敗したことを示します。

```
Device sda cannot be added to a CDS disk group
```

デフォルトで、vxdg はディスクを CDS として初期化します。vxdg コマンドに cds=off を付けて再度実行します。

```
vxdg init disk_group cds=off
```

第 13 章

トラブルシューティング

この章では、一般的なエラーに関するトラブルシューティング情報について説明します。

クラスタ環境の確認

Cluster Edition の使用時には、クラスタ環境の設定上の問題から発生する多くのエラーがあります。クラスタを設定する前に、次の操作を行うことをおすすめします。

- *$SYBASE* にある *SYBASE.csh* または *SYBASE.sh* ファイルを読み込んで環境変数を設定していることを確認します。
- 各ノードから dataserver -v を実行し、すべての必要なライブラリがホスト上にインストールされていることを確認します。

  欠落しているシステム・ライブラリがある場合、オペレーティング・システムのエラーが発生し、データ・サーバのバージョン番号が表示されません。先に進む前に、この問題を修正してください。エラーが生じることなく dataserver にバージョンの文字列が表示されれば、すべての必要なシステム・ライブラリがインストールされていると考えられます。
- クラスタの各ノードが各データベース・デバイスに読み書きできることを確認します。オペレーティング・システムの ls -l コマンドを使用して、dd オペレーティング・システム・ユーティリティでデバイスに読み書きできるかどうかをテストします。

  デバイスの読み取りをテストするには、次のように実行します。

  ```
  dd if=<device path> of=/dev/null count=x
  ```

  次のような結果が返されます。

  ```
  %dd if=/dev/raw/raw123 of=/dev/null
  count=10
  10+0 records in
  10+0 records out
  ```

  dd ユーティリティで、デバイスの書き込みも同様にテストできます。ただし、これはデバイスに保持するデータがない場合のみ、行う必要があります。
- ping ユーティリティを実行して、すべてのノード間の接続性を検証します。各ノードから、その他の各ノードのホスト名またはネットワーク・アドレスに ping を試みます。これは、使用する各ネットワークに対して行います。たとえば、設定でパブリック・ネットワークと 2 つのプライベート・ネットワークが使用されている場合、ping がノードとネットワーク・アドレスのすべての組み合わせに対して成功することを確認します。

sybcluster 'show cluster config' パラメータを使用して、各インスタンスで使用されるプライベート相互接続アドレスを判別します。たとえば、クラスタにノード node1 および node2 が含まれる場合、sybcluster は次のような情報を表示します。

```
SYBCE> show cluster config
** Cluster configuration for "SYBCE" **
    Interface Path "/sybce"
    Trace Flags:
    There were are no trace flags.
    Maximum Instances "4"
    Quorum "/dev/raw/raw23"
    Master Device "/dev/raw/ra24"
      logfile INSTANCE1 /sybce/ASE15_0/install/GATEST_INSTANCE1.log
      run_parameters INSTANCE1
      logfile INSTANCE2 /sybce/ASE-15_0/install/GATEST_INSTANCE2.log
      run_parameters INSTANCE2
    Primary Interconnect "udp"
      Server[1] INSTANCE1 node1_priv 49152 49171
      Server[2] INSTANCE2 node2_priv 49172 4919
```

このクラスタには、相互接続ネットワーク・アドレス `node1_priv` および `node2_priv` が含まれています。node1 から ping node2_priv を実行して、node2 のプライベート・ネットワークのアドレスが node1 からアクセス可能であることを確認します。node2 から ping node1_priv を実行して、node2 から node1 のプライベート・ネットワークに到達可能であることを確認します。

ping コマンドが失敗した場合、またはプライベート・ネットワークの問題を示すエラー・メッセージが返された場合は、次の点を確認します。

- */etc/hosts* ファイルに含まれる情報
- プライベート・ネットワークで使用されるネットワーク・ケーブル、ルーター、スイッチの状態
- sybcluster “show cluster config” コマンドにより報告されるクラスタ設定に指定された名前または IP アドレス

前のバージョンの *dataserver* バイナリを使用したクラスタの再開

次のような場合は、Cluster Edition が開始されず、「`Cluster is running with message version y. This version of ASE requires message version x`」というメッセージがエラー・ログに書き込まれます。

- 実行中の同じクラスタに dataserver バイナリの異なるバージョンを使用しようとした。
- クラスタをあらかじめ停止せずに、dataserver バイナリの前のバージョンを使用して EBF または ESD を適用した。

問題の解決

dataserver バイナリの古いバージョンを使用して、クラスタのインスタンスを開始してから、`shutdown cluster` を発行します。

dataserver バイナリの新しいバージョンを使用してクラスタを再開します。

dataserver バイナリの前のバージョンが使用できない場合、クォーラム・デバイスを作成し直すことにより、このエラー・メッセージを解決できます。

警告！ これらの手順は Adaptive Server の安全チェックを無視するため、遂行時には、実行中のインスタンスがないことを必ず確認してください。

---

1 クラスタのすべてのインスタンスが停止されていることを確認します。

2 qrmutil を使用してクラスタ入力ファイル情報を取り出します。

```
$SYBASE/ASE-15_0/bin/qrmutil --quorum_dev=path_to_quorum
--extract_config=quorum.out
```

3 dataserver バイナリを使用して Cluster Edition を開始し、クォーラム・デバイスを再構築します。

```
dataserver --quorum_dev=path_to_quorum --instance=instance_name
--buildquorum=force --cluster_input=quorum.out
```

4 クラスタを停止します。

5 通常の手順でクラスタを開始します。

警告！ --buildquorum または --cluster_input dataserver パラメータは、この手順にのみ使用されます。この後のクラスタやインスタンスの再開中には使用しないでください。

ディスク・デバイスのアクセス・エラー

Cluster Edition で使用されるすべてのディスク・デバイスは、クラスタのすべてのノードからアクセスできるように設定する必要があります。これらのデバイスへのパスはすべてのノードに対して同じでなければなりません。また、クラスタを開始するために使用されるアカウントには、すべてのディスク・デバイスに対する読み書きパーミッションが必要になります。

Cluster Edition がアクセスできるデバイスがないことを報告した場合、クラスタの各ノードから、次のことを確認します。

- クラスタを設定するときに指定したデバイス・パスに、クラスタのすべてのノードからアクセスできること。
- クラスタを開始するために使用されるアカウントに、これらのデバイスに対する読み書きパーミッションがあること。
- デバイス・パスが、クラスタのすべてのノードに対して同じであること。
- デバイスを参照するために使用されるすべてのシンボリック・リンクが正しいこと。

UNIX ls および ls -l コマンドを使用して、パスやファイルのパーミッションを検証します。UNIX dd ユーティリティを使用して、Sybase アカウントがデバイスに対して読み取りおよび書き込み可能であることを確認できます。

「クラスタ環境の確認」(202 ページ) を参照してください。

クラスタの停止の確認

クラスタが損傷していて、クォーラム・デバイスの一部のステータス情報が整合性のない状態で残されている場合、sybcluster ユーティリティで、クラスタが実行中であるかどうかを判断できない場合があります。クラスタの状態がわからない場合は、クラスタの各インスタンスを解析して、クラスタのステータスを判断します。

1　isql を使用して、クラスタの各インスタンスにログインします。

2　sybcluster 'show cluster status' を使用して、クラスタの各インスタンスのステータスを表示します。

3　sybcluster で、すべてのインスタンスの状態が `'Up'` であり、`heartbeat` が `'Yes'` であると報告されない場合、インスタンスは停止している可能性があります。

4　クラスタの各ノードで UNIX 'ps' コマンドを使用し、dataserver プログラムを表すプロセスがそのノードに設定された各インスタンスに対して実行中であるかどうかを判断します。

5 最初のインスタンスを開始した後で、sybcluster start cluster を発行して、クラスタのすべての残りのインスタンスを開始します。

インスタンスが実行中でないと判断した場合、次のようにクラスタのロックを解除して、再開します。

```
start instance instance_name unlock
```

sybcluster を使用したクラスタの作成がエラー -131 で失敗しました

sybcluster create cluster コマンドが、エラー -131 で失敗し、ロー・デバイスが定義された親ディレクトリにアクセスできないことを示すメッセージが発行されます。

```
INFO - Choosing the first instance to be created using the connected agent...
ERROR - Parent directory access error. The parent directory /dev/rdsk for the device can
not be accessed.  Please change the protection on the device and try again.
INFO - Create cluster error: -131
```

ロー・デバイスの名前のスペルが間違っている可能性があります。各ロー・デバイスの完全な名前をチェックし、正しいことを確認してください。

クラスタの作成が失敗し、$SYBASE ディレクトリにファイルが残されています

sybcluster create cluster コマンドまたは Sybase Central Adaptive Server プラグインの Create Cluster Wizard がエラー状態で終了すると、一部のファイルがクラスタ・インストール・ディレクトリに残されたままになることがあります。次の手順で、これらのファイルを削除してから、クラスタの作成をもう一度試みます。

1 インストール・ディレクトリの interfaces ファイルのクラスタのエントリを削除します。

2 これらの場所から各ノードの Unified Agent プラグイン・ディレクトリを削除します。

```
$SYBASE/UAF-2_5/nodes/node_name/plugins/cluster_name
```

3 *$SYBASE* ディレクトリから、名前の末尾に **.res* が付くリソース・ファイルを削除します。

4　*$SYBASE* ディレクトリから、名前の末尾に **.inp* が付くクラスタ定義ファイルを削除します。

5　*$SYBASE/ASE-15_0/install* から中間の *RUN_instance_name* ファイルを削除します。

6　*$SYBASE/ASE-15_0/install* からエラー・ログファイルを削除します。

7　*$SYBASE* からインスタンス設定ファイルを削除します。

別のクラスタ作成操作を開始する前に必要なのは、手順 1 と 2 のみです。ただし、他の手順を実行すると、インストール領域のファイル数を減らすことができます。

Unified Agent が開始されますが、*sybcluster connect* が失敗しました

Unified Agent の起動時または sybcluster コマンド実行時のネットワーク設定情報が誤っているために、sybcluster 'connect' コマンドが失敗した可能性があります。

- クラスタの各エージェントの Unified Agent ログ・ファイルを調べ、エージェントの RMI リスナを開いてエラーが報告されているかどうかを確認します。エージェント・ログ・ファイルは次のとおりです。

  $SYBASE/UAF-2_5/nodes/node_name/log/agent.log

- sybcluster の開始時に、ノード名と -F パラメーターの受信ポート番号を正しく指定します。

  ```
  sybcluster -U uafadmin -P -C MYCLUSTER
  -Fnode1:1234,node2:1234
  ```

ディスク・デバイスが使用中

Linux では一部のデバイスはデフォルトでディスクにバインドされます。クラスタを作成しようとして、これらのデバイスの 1 つを誤って指定した場合、disk init が失敗し、Create Cluster ウィザードまたは sybcluster でクラスタを作成できません。

Create Cluster ウィザードまたは sybcluster セッションを開始する前に、ディスク・デバイスが使用可能であることを確認します。

「クラスタ環境の確認」(202 ページ) を参照してください。

クラスタの結合のためのインスタンスの失敗

インスタンスは、相互参照コミュニケーションをサポートするために使用されるプライベート・ネットワークの問題が原因で、クラスタの結合に失敗することがあります。クラスタの 1 つまたは複数のインスタンスが開始しない場合、次を実行します。

- 結合できなかったインスタンスのエラー・ログを調べ、ネットワーク通信が失敗したことを示すメッセージがあるかどうか確認します。エラー・ログは、*$SYBASE/$SYBASE_ASE/install/instance_log* にあります。
- クラスタが使用するノードで使用されているすべてのプライベート・ネットワークが、クラスタのすべてのノードからアクセスできることを確認します。これには、ping ユーティリティを使用します。「クラスタ環境の確認」(202 ページ) を参照してください。
- UAF エージェント・ログのエラーをチェックします。

プライベート相互接続の失敗

プライベート相互接続が正しく設定されていないと、クラスタは開始されません。オペレーティング・システムの ping コマンドを使用し、プライベート相互接続が機能していることを確認します。ping が機能しない場合は、システム管理者に問い合わせて、種々のノード間の相互接続通信を有効にしてください。

「クラスタ環境の確認」(202 ページ) を参照してください。

クライアント接続フェールオーバが失敗しました

このエラーは、クライアントが IP アドレスを使用してクラスタに接続し、ノードがローカルの */etc/hosts* または DNS に含まれていないときに、クライアント接続フェールオーバが失敗すると発生します。

Adaptive Server は、フェールオーバ・インスタンス・アドレスがクラスタの interfaces ファイルにリストされているため、それらを送信します。クラスタの interfaces ファイルがホスト名としてインスタンス・ネットワーク・アドレスをリストする場合、Adaptive Server はホスト名をクライアント・アプリケーションに返します。ただし、クライアント・アプリケーションは DNS または */etc/hosts* ファイルを使用してクラスタ・インスタンスのホストの名前を解決するため、クライアントのホスト名が */etc/hosts* ファイルまたは DNS サーバにないと、フェールオーバは成功しません。

クラスタのすべてのノードがクライアント・システムの DNS サーバまたは */etc/hosts* ファイルにリストされていることを確認します。

クラスタの interfaces ファイルのホスト名ではなく、IP アドレスを使用してみてください。

クライアントが代替高可用性サーバへの再接続に失敗しました

DNS 名前解決に問題のあるクライアントは、接続先のインスタンスが失敗したときに、代替高可用性サーバへの再接続に失敗する場合があります。この場合、次のメッセージが生成されます。

```
Connection to Sybase server has been lost, connection to the
next available HA server also failed. All active transactions
have been rolled back.
```

DNS 名前解決の問題は、クライアントのホストが代替インスタンスの *hostname* を正しく解決できないときに発生します。このエラーが DNS 名前解決の問題によって生成されたものかどうかを調べるには、高可用性リストの各代替インスタンスに ping *hostname* コマンドを発行します。ここで、*hostname* は代替サーバが存在するホストです。*hostname* を解決できない場合、または間違った IP アドレスを解決する場合は、システム管理マニュアルに従って名前解決の問題を修正してください。

すべての接続が SSL を使用する場合、sybcluster が接続できません

すべての Adaptive Server 受信ポートが SSL を使用する場合、Adaptive Server は次のエラー・メッセージを発行します。

```
2008-03-20 10:42:46,260 ERROR [Timer-6]
GA1:GA1_1:SQLConnect:270:Login Failure - The user "sa" and the
entered password is not authorized to connect to the cluster.
```

クラスタの各インスタンスに非 SSL 接続があることを確認し、この接続が Unified Agent が使用する interfaces ファイルに含まれていることを確認します。

jConnect サンプルが HA を無効にします

Adaptive Server version 15.0 の Open Client、Open Server、ODBC、jConnect、ADO.NET の『What's New in ESD #12』ドキュメントに含まれる Cluster Edition のサンプル jConnect 接続文字列が間違っています。

ドキュメントのサンプル接続文字列は次の通りです。

```
URL="jdbc:sybase:Tds:server1:port1,server2:port2,...,serverN:
portN/mydb?JCONNECT_VERSION=6&PACKETSIZE=1024&DYNAMIC_PREPARE
=true&REQUEST_HA_SESSION=true"
```

この文字列は、JCONNECT_VERSION=6 パラメータが高可用性コンパニオン・サーバの機能性をエミュレートするために失敗します。クライアントが jConnect ドライバのフェールオーバ・プロパティを使用しようとしたときに、接続文字列で指定された hafailover サーバがないと、JDBC ドライバがクライアント Java 例外を発生させます。

これが、正しい接続文字列です。

```
URL="jdbc:sybase:Tds:server1:port1,server2:port2,...,serverN:
portN/mydb?&PACKETSIZE=1024&DYNAMIC_PREPARE=true&REQUEST_HA_S
ESSION=true"
```

PC-Client インストール − java.lang.NoClassDefFound エラー

Winzip ユーティリティではなく、MKS tar ユーティリティを使用して PC Client tar ファイルのパックを解凍すると、Windows プラットフォームで発生します。Windows プラットフォームでは、MKS tar ユーティリティはフルパス名を切り捨てるため、ファイルが欠落します。

UNIX プラットフォームでは、GNU tar ユーティリティを使用して Adaptive Server インストーラの tar を解凍しないと、この問題が発生します。

Windows プラットフォームの場合は、常に Winzip ユーティリティを使用してインストーラのパックを解凍します。UNIX プラットフォームの場合は、インストーラを解凍するために常に GNU tar ユーティリティを使用します。

クラスタ・エントリ *“name”* にサーバが含まれていません

クォーラム・デバイスが UAF エージェントにアクセスできない場合、sybcluster start cluster および start instance コマンドを発行したときに次のエラーが発生する可能性があります。

```
start cluster
ERROR - The cluster entry SDCDEMO did not contain any servers
start instance INSTANCE1
ERROR - The cluster entry SDCDEMO did not contain any servers
```

- UAF エージェントにクォーラム・デバイスを読み取るパーミッションがない可能性があります。適切なユーザ・アカウントで UAF エージェントを開始し、このアカウントに、クォーラム・デバイスへの読み書きの両方を行うためのパーミッションがあることを確認します。

- sybcluster および UAF エージェントが、クラスタの 1 つまたはすべてのノードのクォーラム・デバイスへの間違ったパスを使用しています。シンボリック・リンクを使用している場合は、すべてのリンクがすべてのノードで正しいことを確認します。
- 別のプロセスによって、UAF エージェントがクォーラム・デバイスを読み取れないようになっています。これは、ディスク・ストレージ・システムの機能上の問題または設定エラーが原因で発生します。

パスワードが変更された後で sybcluster はクラスタを管理できません

sybcluster は、UAF エージェントを使用してクラスタの操作に接続し、これを実行します (例、shutdown cluster)。そのためには、UAF エージェントはクラスタにログインする必要があります。UAF エージェントがログインするためには、正しい sa ログインとパスワードを使用する必要があります。

sa ログインおよびパスワードは、クラスタの UAF プラグイン *configure* ファイルの暗号化フォームに保存されます。UAF がクラスタに接続するために使用するログインとパスワードを変更するには、sybcluster set cluster login を使用します。

Adaptive Server のパスワードを変更するたびに、UAF エージェント・ログインも変更する必要があります。

次のいずれかを実行して、UAF がクラスタに接続するために使用するログインとパスワードを変更します。

- sybcluster set cluster login を使用して、UAF ログイン、パスワード、またはその両方を設定します。このコマンドを発行する前にクラスタに接続します。

  構文は次のとおりです。

  ```
  set cluster login sa-login [password sa-password
  ```

- Adaptive Server プラグインを使用して、UAF プラグインを変更します。
  a　クラスタに接続します。
  b　[サーバ・インスタンス] から、ツリー・ビューのウィンドウ枠でリストされているインスタンスを左クリックします。
  c　右ウィンドウ枠から、[エージェント属性] を選択します。
  d　[パスワード] ローをクリックします。
  e　現在のインスタンスを管理するエージェントの新しいパスワードを入力します。
  f　クラスタの各インスタンスに対して手順 c ～ e を繰り返します。

- UAF 設定ファイルを編集します。

  注意 *agent-plugin.xml* ファイルを編集する前に、すべてのクラスタ UAF エージェントを停止する必要があります。

  UAF Adaptive Server ログインとパスワードが、次に保存されます。

  ```
  $SYBASE/UAF-2_5/nodes/<node name>/plugins/
  <cluster name>/agent-plugin.xml
  ```

  次に例を示します。

```
<set-property property="ase.user" value="sa" /><set-property
property="ase.password"
value="REVTe1NZVUFGfWNvbS5zdW4uY3J5cHRvLnByb3ZpZGVyLlN1bkpDRXtTWVVBRn1jUEZPSkJo
TTZ2QT0=" />
```

  注意 パスワード・タグが欠落している場合、上記の手順 a ～ f を追加します。

  クリア・テキストでパスワードを入力します。passencrypt は、暗号化されたテキスト文字列を生成します。*agent-plugin.xml* ファイルの `password` タグの引用符の間に、文字列全体を入力します。

  暗号化されたパスワードを生成します。

  ```
  $SYBASE/UAF-2_5/bin/passencrypt
  ```

エージェントが見つかりません

sybcluster show agents では、すべてのエージェントが表示されません。

クラスタの複数のホスト・システム名が類似していることがあります。sybcluster -F を使用して、クラスタの各エージェントへの接続を指定している場合は、各ホスト・システムのスペリングが正しく、指定されたポート番号があり、ポート番号も正しいことを確認します。

Sybase Central が AMCP プラグインを登録できません

AMCP プラグイン (*amcplugin.jpr*) を登録しようとしたときに、次のメッセージを取得することがあります。

```
Could not read manifest.file
```

この場合、インストーラは *amcplugin.jar* ファイルの名前を *amcplugin.jar.installed* に変更して、既存のファイルの上書きを避けます。

コマンド行から *registerAMCP* を実行しようとすると、次のように表示されます。

```
Error: Unable to find the AMC Plugin binary.
Please check that $SYBROOT has been set correctly and that the
file
/AMCP/lib/amcplugin.jar' exists.¥n
```

amcplugin.jar.installed の名前を *amcplugin.jar* に変更します。

UAF プラグイン登録エラー

既存のクラスタ・パラメータ値に新しい設定パラメータとの整合性がないために、次のエラーが表示される場合があります。

```
2008-06-16 14:05:16,051 ERROR [main] Failed to register plugin
com.sybase.ase.cluster_15.0.1. Class
com.sybase.ua.plugins.ase.cluster.ASEClusterAgentPlugin not found. Ignored.
java.lang.ClassNotFoundException:
com.sybase.ua.plugins.ase.cluster.ASEClusterAgentPlugin
2008-06-16 14:05:16,052 INFO [main] Finished loading primordial services.
2008-06-16 14:05:16,063 WARN [main] Bootstrap completed with 1 error(s):
2008-06-16 14:05:16,064 WARN [main] Failed to register plugin
com.sybase.ase.cluster_15.0.1. Class
com.sybase.ua.plugins.ase.cluster.ASEClusterAgentPlugin not found. Ignored.
java.lang.ClassNotFoundException:
com.sybase.ua.plugins.ase.cluster.ASEClusterAgentPlugin
```

このエラーが表示された場合は、次のことが考えられます。

- UAF エージェントは開始されず、*agent.log* ファイルの末尾の近くに `Bootstrap completed with x error(s):` が表示される。
- 同じクラスタの作成、削除、再構成を何回も繰り返した。

このエラー・メッセージは、バックグラウンド・プロセスとして *uafstartup.sh* が実行されない場合、*agent.log* ファイルまたはターミナル・ウィンドウに表示されます。次のいずれかを行います。

- エラーを引き起こしている可能性がある古いパラメータを消去します。
- *$SYBASE_UA/nodes/<node_hostname>/plugins* の *cluster* ディレクトリを削除します。フォルダを確実に削除するか、ディレクトリ構造の外側に移動します。

注意 **snmp* および **sysam* フォルダは削除しないでください。

これらのいずれかの手順を実行した後で、UAF エージェントを再開します。手動によるプラグインの再実装が必要になる可能性があります。

注意 上記のいずれかの手順で、ノードの 1 つからクラスタの *plugin* ディレクトリを削除した場合、deploy plugin を使用して、このディレクトリを作成し直してください。クラスタでは、クラスタの各ノードに *plugin* ディレクトリを含める必要があります。

ディスクのデータが使用可能ではありません。データベース作成に影響を与える問題があります

ディスク・ラベルは各ディスクのブロック 0 に保存されます。デバイス作成に使用されるロー・データ・スライスがブロック 0 で開始される場合、ディスク・ラベルが上書きされ、ディスクのデータにアクセスできなくなる可能性があります。

ブロック 0 で始まるロー・ディスク・スライスは作成しないでください。

デバイスへのアクセス許可が、I/O フェンシングを有効にした後に拒否されます

Solaris システムでは、Cluster Edition を開始する UNIX ユーザは、データベースやクォーラムに使用されるロー・デバイス (*/dev/raw/raw#* または */dev/rdsk/c#t#d#s#*) にアクセスするために、継承可能な SYS_DEVICES パーミッションを持つ必要があります。

SYS_DEVICES パーミッションは、I/O フェンシングで使用される SCSI-3 PGR コマンドを実行する機能を Adaptive Server に付与します。UNIX ユーザには、一時的または永続的な SYS_DEVICES パーミッションを付与できます。

- 一時的な SYS_DEVICES パーミッションを付与するには、SYS_DEVICES パーミッションを現在のユーザのシェル・プロセスの継承可能なパーミッション・セットに追加します。

  ```
  ppriv -s l+sys_devices $$
  ```

- 永続的な SYS_DEVICES パーミッションを付与するには、usermod UNIX コマンドを使用して SYS_DEVICES を任意のユーザの継承可能なパーミッション・セットに追加します。

  ```
  usermod -K defaultpriv=basic,sys_devices user_login
  ```

sybcluster が interfaces ファイルを見つけることができません

sybcluster は、存在しないディレクトリの interfaces ファイルを見つけることができないことを示すエラー・メッセージを報告する場合があります。

```
ERROR - The interfaces file /remote/ase_cluster/sdclinux/UAF-2_5/bin/interfaces could not
be found.
```

このエラーは一般的に、1 つまたは複数の UAF エージェントがクォーラム・デバイスにアクセスできないときに発生します。次のことが原因でこのエラーが発生する場合があります。

- クォーラム・デバイスが設定された場所に存在しない。
- クォーラム・デバイスのパーミッションで、UAF エージェントにデータの読み書きが許可されていない。

この問題を解決するために、次のことを確認します。

1. クラスタの各ノードで UAF エージェントを開始する前に、*SYBASE.sh* または *SYBASE.csh* ファイルを読み込んでいること。
2. クォーラム・デバイスが存在し、UAF エージェントを開始するために使用されるログインに、クォーラム・デバイスに対する読み書きパーミッションがあること。

IBM エラー

この項では、IBM AIX オペレーティング・システムで Cluster Edition を実行するときに発生するエラーについて説明します。

非同期 I/O が有効ではありません

Cluster Edition を開始しようとしたときに、非同期 I/O が IBM AIX で有効ではない場合、dataserver バイナリがこのエラー・メッセージを発行し、Cluster Edition は開始されません。

```
exec(): 0509-036 Cannot load program dataserver because of the following errors:
0509-130 Symbol resolution failed for /usr/ccs/lib/libc.a[aio_64.o] because:
0509-136   Symbol kaio_rdwr64 (number 1) is not exported from dependent module
/unix.
0509-136   Symbol listio64 (number 2) is not exported from dependent module
/unix.
0509-136   Symbol acancel64 (number 3) is not exported from dependent module /unix.
0509-136   Symbol iosuspend64 (number 4) is not exported from dependent module /unix.
0509-136   Symbol aio_nwait (number 5) is not exported from dependent module /unix.
0509-136   Symbol aio_nwait64 (number 6) is not exported from dependent module /unix.
0509-136   Symbol aio_nwait_timeout (number 7) is not exported from dependent module
/unix.
0509-136   Symbol aio_nwait_timeout64 (number 8) is not exported from dependent module
/unix.
0509-026 System error: Error 0
0509-192 Examine .loader section symbols with the 'dump -Tv' command.
```

非同期 I/O を有効にする方法については、IBM AIX オペレーティング・システムのマニュアルを参照してください。

デバイスのパーミッションが間違っています

IBM AIX のロー・デバイスを管理するパーミッションがない場合、ユーザが Cluster Edition の開始を試みたときにクラスタが起動しないと、オペレーティング・システムから次のメッセージが発行されます。

```
dopen: open '/dev/device_name', Not owner
```

- 正しいパーミッション (PV_ROOT、PV_SU_、または PV_KER_RAS のいずれか) を次のように付与します。

  ```
  setsecattr -p iprivs=+PV_KER_RAS $$
  ```

  これらのパーミッションは、起動時に Cluster Edition プロセスによって継承されます。

- デバイスを管理するプロセスを実行する非ネットワーク (NIS) ユーザ・パーミッションを、次の手順で付与します。

  a　ユーザを作成します。

  ```
  mkuser sybase
  ```

  b　ロールを作成します。

  ```
  mkrole authorizations=aix.device.manage.change
  role_disk_access
  ```

  c　ユーザにロールを割り当てます。

  ```
  chuser roles=role_disk_access
  default_roles=role_disk_access sybase
  ```

別のマシンがデバイスを使用しています

一度に 1 つのマシンのみが使用できるデバイスが別のマシンのプロセスによって使用されており、そのため Cluster Edition にはそのデバイスにアクセスするパーミッションがない場合、オペレーティング・システムから次のメッセージが発行されます。

```
The IBM AIX SDC dataserver may fail to run with one of the following errors:
Quorum library error 1: Failed to open quorum device '/dev/disk_name'.  OS error 16,
'Device busy'
```

または

```
dopen: open '/dev/disk_name', Device busy
```

このデバイスは、複数のサーバの同時アクセスを許可する必要があります。データベース・デバイスには、予約キーがクラスタ設定ファイルで定義されたインスタンス ID である、共有された予約が必要です (例、ID = 1)。

クラスタの各マシンのクォーラム・デバイスのデバイス・アクセス制限を変更します。

```
chdev -l device_name -a reserve_policy=no_reserve
```

たとえば、ディスク・デバイスが */dev/hdisk1* という名前の場合、次のようになります。

```
chdev -l /dev/hdisk1 -a reserve_policy=no_reserve
```

データベース・デバイスのデバイス・アクセス制限を変更するために、クラスタの各インスタンスの各データベースに対して次のコマンドを実行します。

```
chdev -l device_name -a PR_key_value=instance_ID -a reserve_policy=PR_shared
```

たとえば、インスタンス 1 で */dev/rhdisk2* のデバイス・アクセス制限を変更するには、次のようにします。

```
chdev -l hdisk2 -a PR_key_value=1 -a reserve_policy=PR_shared
```

chdev の実行エラー

このエラーは、chdev コマンドの実行時に表示されます。

```
Method error (/usr/lib/methods/chgdisk):
          0514-047 Cannot access a device.
```

デバイスは現在使用中です。デバイスにアクセスするすべてのプロセスを停止します。

第 14 章

Adaptive Server プラグインによるクラスタの管理

Sybase Central 用 Adaptive Server プラグインを使用すると、クラスタの管理タスクを実行できます。たとえば、クラスタの作成、インスタンスの追加、クラスタまたはインスタンスの開始と停止、論理クラスタの作成または修正、負荷の管理などです。

Adaptive Server プラグインは、コマンド・ライン・メソッドの代わりに使用されます。

Sybase Central と Adaptive Server プラグインの使用の詳細については、『システム管理ガイド』の「第 4 章 Adaptive Server Plug-in for Sybase Central の概要」を参照してください。

共有ディスク・クラスタの管理

Adaptive Server プラグインを使用すると、Sybase Central 内から共有ディスク・クラスタを管理できます。

Adaptive Server プラグインですべてのクラスタ管理機能を使用できるようにするためには、クラスタ・エージェント Adaptive Server プラグインによって実行される Unified Agent が必要です。「Adaptive Server プラグインでの Unified Agent 機能の有効化」(221 ページ) を参照してください。

クラスタへの接続

Sybase Central を起動すると、メイン・ウィンドウが開き、Adaptive Server プラグインと、以前に接続したクラスタとインスタンスのアイコンのリストが表示されます。クラスタが稼働中の場合、クラスタ名の横に緑色の三角形が表示されます。

Adaptive Server プラグインが管理するクラスタが稼働中でない場合、赤い四角形がサーバ・アイコン内に表示されます。三角形も四角形も表示されない場合は、「クラスタの起動」(226 ページ) に記載されている Adaptive Server プラグインおよび Univied Agent の設定に関する指示を参照してください。

リスト内の稼働中のクラスタに接続する場合、クラスタ名を右クリックし、[選択] を選択するのが最も早い方法です。Adaptive Server プラグインでは、以前の接続データを使用して接続が行われます。クラスタがツリー・ビューに表示されない場合は、Server Discovery を使用して探すか、クラスタのホストとポート、ログイン名、およびパスワード情報を指定します。いずれの方法も Sybase Central ウィンドウの上部付近のツールバーの [接続] アイコンをクリックして開始します。必要な接続情報についてわかっている場合、[接続] ウィンドウの該当のフィールドにこれを入力します。クラスタまたはクラスタ・ノードのホストとポート番号がわからない場合は、ログイン名とパスワードを入力して、[検索] ボタンをクリックします。Unified Agent がクラスタを検索し、利用可能なクラスタを一覧表示します。リストに検索しているクラスタが含まれていない場合は、「サーバ検出設定の変更」(221 ページ) を参照してください。

❖ クラスタへの接続

1 [ツール] – [接続] を選択します。

登録済みの複数の Sybase Central プラグインを実行している場合は、Adaptive Server プラグインを選択します。

2 インスタンスとの接続に使用するログイン名を入力します。

3 ログインのパスワードを入力します。

4 [サーバ名] リストからクラスタ名を選択するか、クラスタ・ノードのホストとポートを入力します ([サーバ名] リストは、Linux、Solaris、IBM AIX、HP-UX の interfaces ファイルのエントリ、および Windows の *sql.ini* から生成されます)。

5 (オプション) クラスタ内のインスタンスのホストとポートを指定します。

6 [OK] をクリックします。

❖ ショートカット

- クラスタ・アイコンを右クリックし、[接続] を選択します。

クラスタに接続すると、クラスタ・アイコンは灰色から青に変化します。

ツールバーを使用したクラスタからの切断

1　切断するクラスタのアイコンを選択します。

2　[ファイル] － [切断] を選択します。

ショートカット

- クラスタのショートカット・アイコンを右クリックし、[切断] を選択します。
- 切断するクラスタを選択します。ツールバーから [切断] を選択します。

Adaptive Server プラグインでの Unified Agent 機能の有効化

1　[ツール] － [Adaptive Server Enterprise] － [設定] を選択します。[設定] タブで [Enable Unified Agent features]、[Check Server Status]、[エージェントのポート番号] の順に選択します。デフォルトの UAF ポートは 9999 です。異なるポートで UAF エージェントを確認する必要がある場合、この値をここに入力します。

UAF ポート番号は変更できます。デフォルトは 9999 です。

2　[OK] をクリックします。UAF によってクラスタが監視されている場合、クラスタが稼働中でないと、赤い四角形がクラスタ上に表示されます。クラスタが稼働中の場合、クラスタ アイコン上に緑色の三角形が表示されます。

サーバ検出設定の変更

単一検出方式を選択する必要はありません。サーバ検出では指定のすべての検出方式が検索されます。

1　クラスタ名を右クリックします。

2　[接続] を選択します。

3　[接続] ダイアログ ボックスで [設定] を選択します。

4　[Server Discovery] タブを選択します。

5　検出方式を選択します。

- JINI － 変更への高い適応能力を備えたネットワーク中心型サービスの開発を可能にするオープン・アーキテクチャ。JINI は標準のルックアップ・サービスで検出を実行します。

  『Unified Agent および Agent Management Console バージョン 2.0 Windows および UNIX 版』の「第 2 章 Unified Agent とAgent Management Console のインストールと設定」を参照してください。

- UDP (User Datagram Protocol) − アプリケーション・プログラムが、最小限のプロトコル・メカニズムで他のプログラムにメッセージを送信するためのプロシージャを提供するネットワーク・プロトコル。

注意 UDP のみが使用される場合、Sybase Central が稼働している同じサブネット上のサーバのみ検出されます。

6 [追加] をクリックします。

7 前の手順で JINI を選択した場合は、次の手順を実行します。

- JINI サーバのホストを選択します。
- デフォルトのホストとポート番号を選択するか、新たに入力します。

8 [OK] をクリックします。

9 検出フィルタを追加または編集するには、[フィルタ] をクリックします。Server Discovery は、検索に選択したフィルタのみ使用してフィルタを指定します。

a [追加] をクリックします。

b [フィルタを有効にする] を選択します。

c フィルタするターゲット、ホスト、名前、OS、プラットフォーム、ポート、リリース・タイプ、ステータス、バージョン、ビルド日付を選択します

d 含む、含まない、存在する、存在しない、～で始まる、～で終わる、のいずれかの条件を選択します。

e フィルタする条件文字列を入力します。

f [OK] をクリックします。

10 システムで現在稼働中のクラスタを検出するように、Adaptive Server プラグインを設定します。次のオプションを使用します。

- 削除 − リストから検出サービスを削除します。
- 編集 − 現在の検出サービスの設定を編集します。
- 上 − 選択した検出サービスをリストの上に移動します。
- [下へ] − 選択された検出サービスをリスト内で下に移動する。

11 LDAP サーバを使用する場合は、[LDAP] ウィンドウ枠を選択します。

a LDAP サーバ名を選択します。

b ゲージを使用して、検索タイムアウト時間を設定します。

c ノードにログインするユーザ名とパスワードを入力します。

d　[クラスタ・ノードの選択] ボックスでログインするノードを選択します。

e　[OK] をクリックします。

クラスタ・プロパティの表示

クラスタ・プロパティを表示するには、クラスタ名を右クリックし、[プロパティ] を選択します。Adaptive Server プラグインによって、[サーバのプロパティ] ダイアログ・ボックスが表示されます。このダイアログ・ボックスには、[一般]、[設定]、[ログ領域]、[Job Scheduler サーバ]、[エージェント]、[サーバ・ログ]、[クラスタ]、[ローカリゼーション] タブが含まれます。

[一般] プロパティ・タブ

[一般] タブには、Cluster Edition に関する以下の情報が表示されます。

- タイプ － Adaptive Server のエディション。
- バージョン － ソフトウェアのバージョン。
- リリース・タイプ － リリースがソフトウェアのベータであるか、または運用バージョンであるか。
- プラットフォーム － ノードを稼働中のマシンのプラットフォーム。
- ホスト名 － ノードを稼働中のマシン名。
- オペレーティング・システム － ノードで稼働中のオペレーティング・システム。
- ビルド・オプション － Adaptive Server の現在稼働中のバージョンに特有のオプション。
- ビルド日付 － dataserver バイナリが構築された日付。
- エディション － 現時点で稼働中の Adaptive Server のエディション。
- ライセンス － Adaptive Server ライセンスの現在のステータス。詳細については、[詳細] を選択します。
- 文字セット － 現時点で設定されているデフォルトの文字セット。
- 言語 － 現時点で設定されている言語。
- ソート順 － 現時点で設定されているソート順。
- ステータス － サーバのステータス (稼働中かダウン状態か)。
- ASE ログ・ファイル － ログ・ファイルの位置。

[クラスタ] タブ

[クラスタ] タブには、次の情報を含むクラスタについての一般情報があります。

- 最大インスタンス数 ― このクラスタで使用できるインスタンスの最大数
- インストール・モード ― 共有またはプライベート
- メンバシップ・モード ― vcs またはネイティブ
- クォーラム・デバイス ― クォーラム・デバイスのロケーション
- マスタ・デバイス ― マスタ・デバイスのロケーション

[クラスタ] タブには、ノード名とポート番号のある使用可能なインスタンスのリストがあります。

[設定] タブ

Cluster Creation ウィザードでは、各インスタンスは、同じサーバ設定ファイル (*server_name.cfg*) を使用して設定を決定します。デフォルトで、クラスタ内のインスタンスはすべて、クラスタ設定用に *cluster_name.cfg* ファイルを使用します。ただし、インスタンスを設定するときに異なる設定ファイルを指定することで、異なるインスタンスについて異なる設定値を設定できます。

クラスタとインスタンスについて [設定] パネルを使用できます。

現在の構成設定を表示するには、クラスタまたはインスタンス名を右クリックし、[設定] を選択します。[プロパティ] を選択してから、[設定] パネルを選択することもできます。

設定パラメータの詳細については、『システム管理ガイド』を参照してください。設定を最適化するときに考慮すべき設定上の問題については、『パフォーマンス&チューニング・ガイド：基本』を参照してください。設定パラメータを設定できるユーザは、次の規則に従って決められます。

- システム・セキュリティ担当者 (“sso_role”) の役割を割り当てられたログインでは、次のパラメータをリセットできます。
  - allow updates
  - audit queue size
- default character set id パラメータは、クラスタ・インストールの間に自動的に設定され、Sybase Central の内部からはリセットできません。
- ログインを割り当てられたシステム管理 (“sso_role”) 役割では、他のすべてのパラメータをリセットできます。

再起動が必要なパラメータ

設定パラメータには、ユーザが値をリセットするとただちに有効になる動的パラメータと、ユーザがクラスタを再起動するまで変化しない静的パラメータがあります。Adaptive Server プラグインでは、ユーザがパラメータ名を選択すると、パラメータを有効にするために再起動が必要かどうかが示されます。

インスタンス固有の設定パラメータの削除

クラスタ・インスタンス用の [設定] パネルには [削除] ボタンがあります。これはインスタンスレベルの設定値についてのみ有効になり、クラスタ設定のプロパティのタブには表示されません。パラメータを削除するということは、クラスタ全体の値とは別に、インスタンス固有の設定としてそのパラメータを削除することを意味します。

値を変更し、[適用] を選択するまで、[削除] ボタンは無効です。この設定パラメータを次に選択したときに [削除] が有効になるので、その設定パラメータを選択して削除できます。

設定パラメータの設定

Cluster Edition には、グローバル設定パラメータとインスタンス設定パラメータがあります。グローバル設定パラメータがクラスタ全体に影響するのに対し、インスタンス設定パラメータは設定されているインスタンスにのみ影響を及ぼします。グローバル設定パラメータを設定するには、クラスタの [設定] タブを開き、クラスタの名前を選択してから、リセットする設定パラメータを選択します。

デフォルトで、インスタンスはグローバル設定値を使用します。ただし、インスタンス設定で上書きした場合は別です。

❖ **インスタンス用の設定パラメータの設定**

1　設定するインスタンス名を右クリックします。

2　[設定] を選択します (または [ファイル] － [プロパティ] を選択してから、[設定] タブをクリックします)。

3　表示する機能グループを選択するか、または [すべて選択] を選択します。

4　更新するパラメータを選択します。選択したパラメータの簡単な説明については、[説明] ボックスを参照してください。

5　テーブルの [値] カラムに新しい値を入力します。

6　[OK] をクリックします (または、複数の設定値を変更する場合は、[適用] をクリックします)。

- パラメータがただちに効力を持つ場合は、[値] カラムにリストされます。
- パラメータを有効にするために Adaptive Server を再起動する必要がある場合は、[保留中の値] カラムにリストされます。

[ログ領域] タブ

[ログ領域] パネルには、クラスタの現在のログ要領に関する次の情報が表示されます。

- データベース (インスタンス) − ログ領域の名前。ログがインスタンスに特有の場合、インスタンス名はカッコ内に表示されます。カッコがない場合、ログはクラスタについてのものです。
- 総領域 (MB) − 利用できるログ容量の合計 (メガバイト単位)。
- 使用済み領域 (MB) − 現在利用されているログ容量の合計 (メガバイト単位)。
- 空き領域 (MB) − ログに利用できる空き容量の総量 (メガバイト単位)。
- 使用済み − 現時点で使用されている合計容量のパーセント。

[Job Scheduler サーバ] タブ

Job Scheduler サーバとして指定したサーバには、Job Scheduler をインストールしておく必要があります。Job Scheduler のインストールの詳細については、Job Scheduler の『ユーザーズ・ガイド』を参照してください。

[ローカリゼーション] タブ

[ローカリゼーション] タブには、デフォルト言語、文字セット、ソート順の現在の値が表示されます。デフォルト値を変更し、言語を追加または削除できます。

クラスタの起動

Adaptive Server プラグインには、クラスタを起動するよう有効にされた Unified Agent 機能が必要です。エージェントが管理するクラスタは赤い四角形 (クラスタが稼働中でない場合) または緑の三角形 (クラスタが稼働中の場合) が、クラスタ名ごとにサーバ・アイコンに表示されます。稼働中でないクラスタを起動するには、次の手順に従います。

1 ツリー・ビューの左のウィンドウ枠で、クラスタ・ショートカット・アイコンを右クリックし、[起動] をクリックします。

2 クラスタを起動する管理アクセスを使用して、Unified Agent ログインとパスワードを入力します。

3 エージェントがサーバを起動するとき、メッセージ・ログ・ウィンドウが開きます。クラスタ起動プロセスが完了すると、[OK] ボタンが有効になります。

4 クラスタ・アイコン上の赤い四角形が緑の三角形になり、クラスタが稼働中であることを示します。

クラスタの停止

接続されていないクラスタを停止するには、次の手順に従います。

1　停止するクラスタのアイコンを右クリックします。

2　[停止] を選択します。

3　[はい] をクリックして停止を確定します。

接続されているクラスタを停止するには、次の手順に従います。

1　ツリー・ビューの左のウィンドウ枠で、クラスタ・アイコンを右クリックし、[停止] を選択します。

2　いずれかのチェックボックスをオンにします。

- プロセスが終了した後でクラスタを停止し、停止に 1、5、または 10 分以上かかった場合には通知を受け取る。
- クラスタをただちに停止する。

3　[はい] をクリックしてクラスタを停止します。

クラスタの削除

クラスタを削除すると、ユーザは、クラスタを作成するために実行したすべての手順を取り消すことができます。クラスタの削除は、サーバ・グループからクラスタを削除するのとは異なります。削除できるクラスタは停止しているクラスタのみです。

注意　いったんクラスタを削除すると、そのクラスタは完全に削除され、再起動できなくなります。

クラスタを削除するには、次の手順に従います。

- 削除するクラスタの名前を右クリックします。
- リストから [クラスタの削除] を選択します。
- 管理エージェント用のログインを入力します (通常、uafadmin)。
- [OK] を選択します。

サーバ・グループの削除

サーバ・グループを削除すると、Adaptive Server プラグインからサーバ・グループのクラスタ・エントリが削除されます。クラスタは影響を受けません。

デフォルト・グループからクラスタを削除するには、次の手順に従います。

1 クラスタ名を右クリックし、[デフォルト] から [削除] を選択します。

2 [削除の確認] ダイアログ・ボックスでクラスタ名を選択し、[はい] を選択して、削除を確定します。

クラスタのステータスの表示

クラスタが Unified Agent によって管理されている場合、接続されているかどうかにかかわらずエージェントはステータスを通知しますが、接続されているクラスタについてはより詳細な情報を表示します。

ツリー・ビューで、[サーバ・インスタンス] フォルダをクリックし、右ウィンドウ枠にインスタンスを表示します。ステータスの詳細には以下が含まれます。

- インスタンス − クラスタ内のインスタンスの名前。
- ID − クラスタ内のインスタンスの数的順序。
- ステータス − クラスタの現在のステータス。オンラインまたはオフライン。
- アドレス − クラスタのインスタンスのアドレス。
- 起動時刻 − クラスタが起動した時刻。
- アクティブな接続 − 接続の数。
- オンラインのエンジン − エンジンの数。

クラスタ・インスタンスの管理

この機能によって、クラスタ内のインスタンスの管理が可能になります。

クラスタへのインスタンスの追加

インスタンスを追加する前に、max instances パラメータがより多くのインスタンスに対応できるようにしておく必要があります。また、クラスタにサポートされるエージェントがインスタンスを作成するノードで稼働されている必要があります。Unified Agent (UA) のホスト名とポート番号を入手しておくことも必要です。

1 左側のウィンドウ枠の [サーバ・インスタンス] フォルダを開き、右側のウィンドウ枠にサーバ・インスタンスとオプションを表示します。

2　[クラスタ・サーバ・インスタンスの追加] を選択し、クラスタ・サーバ・インスタンスの追加ウィザードを開きます。

3　ウィザードの手順に従い、クラスタにインスタンスを追加します。

4　[完了] をクリックします。[サーバ・インスタンス] ビューの下に、新しいインスタンスがリストされます。

5　インスタンスを起動します。

クラスタからのインスタンスの削除

クラスタからインスタンスを削除する前に、インスタンスを停止する必要があります。インスタンスを削除すると、バインドおよびグループ・メンバシップなど、そのインスタンスのすべてのテンポラリ・データベース定義が削除されます。

注意　クラスタの最後に残ったインスタンスは削除できません。

1　右のウィンドウ枠で、削除するインスタンスを右クリックし、[削除] を選択します。

2　[はい] をクリックします。インスタンスがクラスタから削除されます。

インスタンスのプロパティの表示

インスタンスのプロパティのダイアログには、Sybase リリース・ディレクトリ (*$SYBASE*)、*$SYBASE_ASE*、およびインタフェース・ディレクトリに関する情報があります。

インスタンスの起動

インスタンス・アイコンを右クリックし、[起動] を選択します。[Start in Progress] というタイトルのバーが表示されます。

インスタンスが起動すると、ステータスは「オンライン」になります。

注意　インスタンスの起動に予想よりも時間がかかる場合、フォルダを手動で更新して、インスタンスのステータスを更新する必要が生じることがあります。

インスタンスの停止

1 [インスタンス] アイコンを右クリックし、[停止] を選択します。

2 以下のいずれかを選択します。

- すべてのプロセスが終了した後でインスタンスを停止し、停止に 1、5、または 10 分以上かかった場合には通知を受ける。
- インスタンスをただちにシャットダウンします。

3 [はい] をクリックします。停止が完了したら、インスタンスの新しいステータスが表示されます。

共有データベース・デバイスの作成

共有データベース・デバイスはすべてのクラスタ・インスタンスにアクセスできます。ツリー・ビューで Database Devices フォルダを選択します。

- [データベース・デバイスの追加] を選択し、データベース・デバイスの追加ウィザードを開始します。
- ウィザードの指示に従います。
- [完了] をクリックします。右ウィンドウ枠のデバイス・リストの下方にデバイスが表示されます。

複数のテンポラリ・データベースの管理

Cluster Edition には、グローバル・システム、ローカル・システム、ユーザ作成グローバル、ユーザ作成ローカル、といった 4 種類のテンポラリ・データベースがあります。

テンポラリ・データベースを表示するには、Temporary Databases フォルダから次のいずれかを選択します。

- グループ ビュー ― テンポラリ・データベース・グループの一覧を表示します。ローカル・テンポラリ・データベースのみがテンポラリ・データベース・グループに所属できます。
- リスト・ビュー ― グローバル・テンポラリ・データベースを表示します (ローカル・テンポラリ・データベースではありません)。

ローカル・テンポラリ・データベースの管理

ローカル・テンポラリ・データベースには、ローカル・インスタンスのみがアクセスできます。クラスタ内の他のサーバ・インスタンスはこのテンポラリ・データベースにアクセスできません。

テンポラリ・データベースを表示するには、次の手順に従います。

1　[サーバ・インスタンス] フォルダを選択します。

2　インスタンスの名前を選択します。

3　ローカル・テンポラリ・データベースを選択します。

テンポラリ・データベース名を右クリックして、ローカル・テンポラリ・データベースを設定および保持します。オプションを選択します。

- Interactive SQL のオープン － Interactive SQL を使用してセッションを開始します。
- 一貫性チェック － ウィザードの指示に従って、テンポラリ・データベース上でデータベース一貫性チェッカを実行します (dbcc)。
- チェックポイント － [はい] を選択して、このデータベースのチェックポイントを実行します。[プレビュー] を選択して、現在稼働中の SQL を表示します。
- 統計の表示 － ウィザードの指示に従って、テンポラリ・データベース上で optdiag を実行します。
- DDL の生成 － [データベース DDL の作成] を選択して、現在稼働中の DDL を表示します。[Exclude DDL] を選択すると、DDL から除外できるオブジェクトのリストが表示されます。除外するオブジェクトを確認し、[OK] を選択します。
- 削除 － [削除] を選択して、このテンポラリ・データベースを削除します。[はい] を選択して確定します。最後のシステム・テンポラリ・データベースは削除できないので、これが残された唯一のテンポラリ・データベースである場合、[削除] オプションは無効です。
- プロパティ － 以下に関する情報に該当するウィンドウ枠を選択します。
  - 一般 － データベースのタイプ、データベース所有者 (データベース所有者を変更するには [変更] を選択)、データベース作成日、トランザクション・ログが最後にダンプされた時刻、データベースにゲスト・ユーザがいるかどうか、データ・キャッシュのタイプ、デフォルトのデータベース位置、プロキシ・テーブルを再同期するかどうかを記述します。
  - デバイス － 現時点で設定可能なデータベース・デバイスをリストします。次を選択します。

- 追加 − デバイスを追加します。[デバイスのサイズ] ウィンドウで [データ] または [トランザクション・ログ] を選択し、デバイス・サイズを指定します。[OK] をクリックして確定します。
- 削除 − デバイスを削除します。
- 編集 − デバイスを設定します。[デバイス・サイズ] ウィンドウには、現時点で設定されている名前、サイズ、未使用部分、現在の割り付け、合計容量割り付けがリストされ、デバイスに領域を追加できます。
- ログの移動 − ログ・デバイスの新しい位置を指定します。
- プロパティ − ウィンドウ枠が 4 つあり、一般情報、ミラー・デバイスのステータス、このデバイスにあるデータベース、およびセグメント情報が表示されます。

- 使用法 − データベースが使用する容量の詳細を記述します。情報を表示する単位 (ページ、KB、MB、GB) を選択します。
- トランザクション・ログ − バインド、ログ I/O サイズ、ラストチャンス・スレッショルドを使用するセグメントを設定できます。
- オプション − このデータベースを設定できるデータベース・オプションをリストします。オプションをオンまたはオフにし、[OK] をクリックします。

- アクティブ・セッション − このデータベースに割り当てられたセクションについての SPID およびログイン情報を表示します。

システム・テンポラリ・データベース

クラスタまたはインスタンスの作成時に、システム・テンポラリ・データベースを作成できます。グローバルまたはローカルのシステム・テンポラリ・データベースを削除することはできません。ただし、インスタンスを削除すると、ローカル・システム・テンポラリ・データベースは自動的に削除されます。

ユーザ作成グローバル・テンポラリ・データベースの追加

Cluster Edition では、グローバル・テンポラリ・データベース (クラスタ全体ですべてのインスタンスに利用可能) およびローカル・テンポラリ・データベース (個別のインスタンスに利用可能) を作成できます。

1 [データベース] − [テンポラリ・データベース] − [リスト・ビュー] に移動します。

2 テンポラリ・データベースの追加ウィザードを選択します。

3　ウィザードに概説してある指示に従います。[次へ] をクリックすると次の指示に進み、[戻る] を選択すると前の指示に戻ります。

4　終了したら、[完了] をクリックします。テンポラリ・データベースがデータベース・リストの下方の右のウィンドウ枠に表示されます。

ユーザ作成ローカル・テンポラリ・データベースの追加

ユーザ作成ローカル・テンポラリ・データベースは、所有インスタンスによってのみアクセスできます。

1　ツリー・ビューの右のウィンドウ枠で、[サーバ・インスタンス] − [*Instance_name*] − [ローカル・テンポラリ・データベース] に移動します。

2　[ローカル・テンポラリ・データベースの追加] を選択します。

3　ウィザードの指示に従って、ローカル・テンポラリ・データベースを作成します。

テンポラリ・データベースのグループへの追加

データベース・グループを表示するには、Temporary Databases フォルダから [グループ・ビュー] を選択します。

グループの追加

ローカル・テンポラリ・デバイスを格納するグループを管理者が作成します。default グループがデフォルトで作成されます。効率的な管理のために、個別のテンポラリ・データベースにではなく、グループ上にバインドを作成することをおすすめします。

Group フォルダを表示するには、Temporary Databases フォルダから [グループ・ビュー] を選択します。

1　ツリー・ビューから、[データベース] − [テンポラリ・データベース] − [リスト・ビュー] − [グループ・ビュー] に移動します。

2　[テンポラリ・データベース・グループの追加] を選択します。

3　Adaptive Server プラグインは、Add Temporary Database Group ウィザードを開始します。ウィザードの指示に従って、ローカル・テンポラリ・データベース・グループを作成します。

グループのプロパティ

グループ名を右クリックし、[プロパティ] を選択します。Sybase Central では、[バインド] ウィンドウ枠と [データベース] ウィンドウ枠が表示されます

[データベース] ウィンドウ枠

[データベース] ウィンドウ枠には、グループ内の現在のテンポラリ・データベースがすべて表示されます。

テンポラリ・データベースをグループに追加するには、次の手順に従います。

1 [追加] を選択します。

2 [テンポラリ・データベースの追加] 画面で、追加するテンポラリ・データベースの名前を選択します。

3 [OK] をクリックします。

テンポラリ・データベースをグループから削除するには、次の手順に従います。

1 テンポラリ・データベース名を選択します。

2 [削除] を選択します。

3 [はい] をクリックします。

[バインド] ウィンドウ枠

[バインド] ウィンドウ枠には、現在のグループのアプリケーションとログインのバインドが表示されます。

新規アプリケーションをバインドするには、次の手順に従います。

1 [バインド・アプリケーション] を選択します。

2 アプリケーション名を入力し、[OK] をクリックします。

ログインをバインドするには、次の手順に従います。

1 [バインド・ログイン] を選択します。

2 [新しいログイン・バインド] 画面で、バインドするログインを選択します。

3 [OK] をクリックします。

現時点でバインドされているアプリケーションまたはログインを削除するには、次の手順に従います。

1 アプリケーションまたはログイン名を選択します。

2 [バインド解除] を選択します。

3 [はい] をクリックして操作を確定します。

アプリケーションまたはログインをテンポラリ・データベースまたはテンポラリ・データベース・グループにバインドできます。

現在のバインドを表示するには、次の手順に従います。

1 テンポラリ・データベースまたはテンポラリ・データベース・グループを右クリックします。

2　[バインド] タブを選択します。現在のバインドが一覧表示されます。

ログインまたはアプリケーションのバインドを解除するには、次の手順に従います。

1　一覧からログインまたはアプリケーションを選択します。

2　[バインド解除] をクリックします。

3　[はい] を選択して確定します。

すべてのログインおよびアプリケーションをバインド解除するには、次の手順に従います。

1　[すべてをバインド解除] をクリックします。

2　[はい] を選択して確定します。

負荷の管理

Workload Manager を使用して、論理クラスタ、負荷プロファイル、負荷スコア、ルートを表示、作成、操作できます。

負荷プロファイル

負荷プロファイルによって、論理クラスタの操作基準を定義できます。これらの基準は通常、「負荷スコア測定基準」と呼ばれます。ここには、各基準に関連付けられた値が、負荷プロファイルを使用する論理クラスタ内のインスタンスごとに「スコア」に集められます。論理クラスタ内部のさまざまなインスタンスについて負荷スコアを定期的に比較することで、負荷が意図に反して 1 つまたは複数のインスタンスに割り当てられたケースを検出したり、あるインスタンスが十分に活用されているかどうかを判定したりできます。

複数の論理クラスタに含まれているインスタンスは複数の負荷プロファイルの影響を受ける可能性があります。したがって、インスタンスを複数の論理クラスタと関連付けるとき、および負荷プロファイルを定義および適用するときは、注意を払ってください。

注意　Cluster Edition には、sybase_profile_oltp (OLTP 環境用) と sybase_profile_dss (DSS 環境用) の 2 つのシステム負荷プロファイルがあります。システム負荷プロファイルは修正または削除できません。ただし、これらを複製し、複製したものを修正して自分専用の負荷プロファイルを作成することはできます。

負荷プロファイル・ステータスは、以下を報告します。

- 名前 − 負荷プロファイル設定の名前。
- タイプ − 負荷プロファイル・タイプ (システムまたはユーザ)。
- 測定基準の重み − 負荷プロファイル内の測定基準ごとに割り当てられる相対的な重み。測定基準には以下のものがあります。
  - ユーザ接続 − 特定の負荷プロファイルに接続されたユーザの重みを表示します。
  - CPU ビジー − 現在ビジーな CPU の重みを表示します。
  - 実行キューの長さ − 実行キューの重みを表示します。
  - I/O 負荷 − I/O 負荷の重みを表示します。
  - エンジンの不足 − エンジンの結果の重みを表示します。
  - ユーザ − ユーザが測定用に選択した基準の重み。
- スレッショルド − 論理クラスタ内の 2 つのインスタンス間の負荷の差異設定 (パーセント)。インスタンス間の負荷の差がこの設定値を満たすと、以下が発生する可能性があります。
  - ログイン・リダイレクト − 接続時間負荷バランシング、および論理クラスタへのルーティング接続時間について使用されます。必要に応じてインスタンスはクライアントに、現在のログイン試行を停止して利用可能ネットワーク・アドレスのリストとして提示されたインスタンスに接続を試みるよう指令します。
  - 動的マイグレーション − ("ヒステリシス値" とも呼ばれる) 動的マイグレーション設定。
- 最小負荷スコア − ログイン・リダイレクトおよび動的マイグレーションをトリガするのに必要な負荷スコア。

負荷プロファイルの追加

1 [負荷管理] フォルダから [負荷プロファイル] を選択して、[負荷プロファイルの追加] を選択します。

2 プロファイルの名前を入力します。

3 [次へ] をクリックします。

4 負荷プロファイルの測定基準の重みを調整します。

負荷プロファイルが論理クラスタに関連付けられると、Workload Manager は論理クラスタ内のインスタンスごとに負荷スコアを計算します。これを計算するために使用するのは、各測定基準について指定した重み、インスタンスについての各測定基準の raw の値、負荷測定アルゴリズムです。「負荷ステータスの表示」(246 ページ) を参照してください。

サーバが測定する測定基準は次のとおりです。

- ユーザ接続 − 使用できるリソースに基づいた、新しい接続を受け入れるインスタンスの能力。
- CPU 利用率 − 追加の作業を処理するインスタンスの能力。
- 実行キューの長さ − システム上の実行可能なタスクの数。実行キューの長さは、処理バックログを測定するものであり、相対的な応答時間の良いインジケータです。
- I/O 負荷 − 未処理の非同期 I/O。
- エンジンの不足 − クラスタ内のインスタンス間のオンライン・エンジン数の違い。

  注意 エンジン不足は、クラスタ内のインスタンスが異なる数のエンジンを持っている場合にのみ測定可能です。エンジンの不足は、相対的な最大容量の測定基準を負荷スコアに追加します。

- ユーザ測定基準 − ユーザが提供する、ユーザ環境に固有のオプションの測定基準。

  指定した負荷スコアが合計 100 になることを確認します。100 にならない場合、Workload Manager は、そのスコアに加算して、合計 100 になる比例した値を作成します。

5 [次へ] をクリックします。

6 以下の値を入力します。

- 最小負荷スコア − 負荷スコアはパーセントでなく、作業を他のインスタンスにリダイレクトする前に Workload Manager で必要な最小スコアです。最小負荷スコアに意味があるのは、負荷プロファイルを使用して論理クラスタ内の他のインスタンスの負荷スコアと比較するときです。
- ログイン・リダイレクト (%) − 着信接続を最良に分散する方法を決定する負荷スレッショルド。
- 動的接続マイグレーション (%) − 既存の接続を分散するかどうかを決定する負荷スレッショルド。

負荷スレッショルドとは、現在のインスタンスと論理クラスタ内の最小負荷インスタンスの、それぞれの負荷の差異 (パーセント) です。その値が満たされてから、Cluster Edition はログインをリダイレクトするか、または既存の接続をマイグレートします。

注意 ログイン・リダイレクトと動的接続マイグレーションのパーセントは、独立したパーセントであり、合計 100 になる必要はありません。

7 [完了] を選択して負荷プロファイルを作成します。

負荷プロファイルの削除

負荷プロファイルを削除するには、次の手順に従います。

1 [負荷管理] − [負荷プロファイル] フォルダで、負荷プロファイル名を右クリックします。

2 [削除] を選択します。

注意 ユーザが作成した負荷プロファイルだけを削除できます。

負荷プロファイルの論理クラスタとの関連付け

1 [負荷管理] − [論理クラスタ] フォルダで、論理クラスタ名を右クリックします。

2 プロパティを選択します。

3 [負荷プロファイル] タブを選択します。

4 [変更] をクリックします。使用可能な負荷プロファイルのリストが表示されます。

5 (オプション) [プロファイルのプレビュー] を選択してウィンドウを表示します。このウィンドウで、プロファイルを選択し、そのプロファイルが論理クラスタ内のインスタンスの重み付けされた測定基準値にどのように影響するかを表示できます。

   a この論理クラスタと関連付ける負荷プロファイルを強調表示します。

   b [閉じる] を選択します。

   c [OK] を選択して [プロパティ] ダイアログ・ボックスを終了します。

6 負荷プロファイルを選択します。

7 [OK] を選択します。

8 [OK] または [適用] を選択して、新規負荷プロファイルを論理クラスタと関連付けします。

負荷プロファイルの [一般] タブ

負荷プロファイルのプロパティを表示するには、次の手順に従います。

1　Workload Management フォルダから、[負荷プロファイル] を選択します。

2　負荷プロファイル名を右クリックし、[プロパティ] を選択します。

[一般] タブに、負荷プロファイルの説明が表示されます。名前とタイプ (システム負荷プロファイルかユーザ負荷プロファイルか) などです。

「負荷プロファイル」(235 ページ) を参照してください。

[測定基準の重み] タブ

[測定基準の重み] タブには、測定基準に適用されている現在の重みの説明が表示されます。重みの合計が 100 にならない場合、Workload Manager は比例する調整値を使用して合計 100 になるようにします。

- ユーザ接続 － 使用できるリソースに基づいた、新しい接続を受け入れるインスタンスの能力。
- CPU 利用率 － 追加の作業を処理するインスタンスの能力。
- 実行キューの長さ － システム上の実行可能なタスクの数。実行キューの長さは、処理バックログを測定するものであり、相対的な応答時間の良いインジケータです。
- I/O 負荷 － 未処理の非同期 I/O。
- エンジンの不足 － クラスタ内のインスタンス間のオンライン・エンジン数の違い。

  注意　エンジン不足は、クラスタ内のインスタンスが異なる数のエンジンを持っている場合にのみ測定可能です。エンジンの不足は、相対的な最大容量の測定基準を負荷スコアに追加します。

- ユーザ測定基準 － ユーザが提供する、ユーザ環境に固有のオプションの測定基準。

  指定する負荷スコアが最高 100 になることを確認します。

[スレッショルド] タブ

[スレッショルド] タブを使用して、負荷プロファイルのスレッショルド設定を表示および変更します。以下の値を入力します。

- 最小負荷スコア － 負荷スコアはパーセントでなく、作業を他のインスタンスにリダイレクトする前に Workload Manager で必要な最小スコアです。最小負荷スコアに意味があるのは、負荷プロファイルを使用して論理クラスタ内の他のインスタンスの負荷スコアと比較するときです。

- ログイン・リダイレクト (%) − 着信接続を最良に分散する方法を決定する負荷スレッショルド。
- 動的接続マイグレーション (%) − 既存の接続を分散するかどうかを決定する負荷スレッショルド。

論理クラスタの管理

Logical Clusters フォルダには、定義済みの論理クラスタのリスト、および新規論理クラスタ追加用のウィザードがあります。右のウィンドウ枠には、次の詳細が表示されます

- クラスタ ID − 各論理クラスタの関連する ID 番号を表示します。
- 論理クラスタの現在のステータス − 論理クラスタの接続ステータス (オンラインまたはオフライン)。
- 接続 − 論理クラスタ内のアクティブな接続数。
- 基本インスタンス − 論理クラスタ内の基本インスタンスの数。
- アクティブな基本インスタンス − 現時点でアクティブな、論理クラスタ内の基本インスタンスの数を表示します。
- フェールオーバ・インスタンス − フェールオーバ用に設定されているインスタンス数。
- アクティブなフェールオーバ・インスタンス − アクティブなフェールオーバ・インスタンスの数を表示します。
- down-routing モード − 下方ルーティングの設定を表示します。以下のいずれかです。
  - system − ルート指定不可能な接続をシステム論理クラスタに送信します。system によって、システム論理クラスタがすべてのインスタンスについて必ずオンラインとなり、最高の可用性が確保されます。デフォルトの設定です。
  - open − ルート指定不可能な接続を、open 論理クラスタに送信します。open 論理クラスタに接続を送信できない場合、または接続がリダイレクションをサポートしない場合、インスタンスは open 論理クラスタの down-routing モードを適用します。
  - disconnect − ルート指定不可能な接続を切断します。この設定をした場合、ターゲット論理クラスタ内のインスタンスによってサービスできないクライアントを切断することによって、リソースを保持できます。
- フェールオーバ・モード − フェールオーバ・モード設定を表示します。オプションはインスタンスまたはグループです。

- 起動モード ― 起動モード設定を表示します。自動 (オプションは選択不可能) または手動。
- システム・ビュー ― システム・ビュー設定、インスタンス、またはクラスタを表示します。
- 役割 ― システム役割はシステム作成の論理クラスタのみにあります。デフォルトで、オープンな役割がシステム論理クラスタに割り当てられます。オープンな役割が新しい論理クラスタに割り当てられると、以前にその役割を所有していた論理クラスタからオープンな役割は削除されます。
- 負荷プロファイル ― クラスタに使用される負荷プロファイルの名前を表示します。

論理クラスタの追加

1 [論理クラスタの追加] を右クリックします。

2 [開く] を選択します。

3 クラスタ名を入力し、[次へ] をクリックします。

4 [追加] をクリックし、論理クラスタに加えるサーバ・インスタンスを選択します。[次へ] をクリックします。

5 論理クラスタのフェールオーバ・サーバ・インスタンスを追加します。[次へ] をクリックします。

6 ルート指定されたアプリケーション、ログイン、およびエイリアスを入力します。最初にルートを選択して、ルートを削除することもできます。

7 デフォルト以外の負荷プロファイルを選択するには、[変更] をクリックしてから、[次へ] をクリックします。

8 新規論理クラスタ用のオプションを設定します。[次へ] をクリックします。

9 Adaptive Server プラグインで要約が表示されます。変更を行うには、[戻る] ボタンを使用します。終了したら、[完了] をクリックします。

論理クラスタの削除

論理クラスタを削除する場合、その論理クラスタはオフラインである必要があります。

1 論理クラスタを右クリックし、[削除] を選択します。

2 [はい] をクリックします。

論理クラスタ・プロパティ

論理クラスタ名を右クリックし、[プロパティ] を選択し、論理クラスタの設定を表示します。

「負荷プロファイル」(235 ページ) を参照してください。

[一般] タブ

[一般] タブには以下のオプションが表示されます。

- システム ビュー － Cluster Edition が論理クラスタ・ユーザにクラスタ全体として表示されるか、または個別のインスタンスとして表示されるかを決定します。これは、一部のクエリとストアド・プロシージャに影響を及ぼします。[インスタンス] または [クラスタ] を選択します。
- 論理クラスタの自動起動 － このオプションを選択すると、クラスタの起動時にこの論理クラスタが起動します。
- フェールオーバ・モード － 論理クラスタが他のインスタンスまたはグループにフェールオーバするかどうかを決定します。
  - Instance (インスタンス) － 一度に 1 つのインスタンスにフェールオーバする論理クラスタ。
  - Group (グループ) － すべての基本インスタンスが失敗し、すべてのフェールオーバ・インスタンスがオンラインになるときにのみ基本インスタンスが置き換えられることを指定します。たとえば、SalesLC のフェールオーバ・モードは “group” です。基本インスタンス “ase1” が失敗すると、クラスタは基本インスタンス “ase2” 上で実行され続けます。オンラインになるフェールオーバ・インスタンスはありません。ただし、“ase1” と “ase2” の両方が失敗すると、クラスタはフェールオーバ・インスタンス “ase3” と “ase4” で実行されます。
  - Fail-to-any (どれにでもフェールオーバ) － 利用できるフェールオーバ・インスタンスがない場合でも、論理クラスタがフェールオーバできるようにします論理クラスタは、負荷に基づいて、クラスタ内の利用できる物理インスタンスのどれにでもフェールオーバします。

    Fail-to-any (どれにでもフェールオーバ) は、その論理クラスタ内部のフェールオーバ・インスタンスとして定義されていなくても、利用できるどのインスタンスにもフェールオーバするよう論理クラスタを構成します。

- down-routing モード － 下方ルーティングの設定を表示します。以下のいずれかです。
  - system － ルート指定不可能な接続をシステム論理クラスタに送信します。system によって、システム論理クラスタがすべてのインスタンスについて必ずオンラインとなり、最高の可用性が確保されます。デフォルトの設定です。
  - open － ルート指定不可能な接続を、open 論理クラスタに送信します。open 論理クラスタに接続を送信できない場合、または接続がリダイレクションをサポートしない場合、インスタンスは open 論理クラスタの down-routing モードを適用します。
  - disconnect － ルート指定不可能な接続を切断します。この設定をした場合、ターゲット論理クラスタ内のインスタンスによってサービスできないクライアントを切断することによって、リソースを保持できます。
- 論理クラスタの役割 － システム役割は「システム論理クラスタ」に自動的に設定されます。この設定は変更できません。

  デフォルトで、オープンな役割がシステム論理クラスタに割り当てられます。オープンな役割は付与できません。また、オープンな役割は別のクラスタがその役割を負ったときのみ削除されます。

[ベース・インスタンス] タブ

[ベース・インスタンス] タブには、現在設定されている論理クラスタ・インスタンスが一覧表示されます。

- 論理クラスタにインスタンスを追加するには、[追加] を選択します。
- 論理クラスタからインスタンスを削除するには、次の手順に従います。
  a　削除するインスタンスを強調表示します。このインスタンスはオフラインにしてから、選択する必要があります。
  b　[削除] を選択します。
  c　[はい] を選択して、削除を確定します。
- インスタンスをオフラインにするには、[オフライン] を選択します。インスタンスをただちにオフラインにするか、または次第にオフラインにして指定した時間の経過後に通知するかを指定します。
- オフライン・インスタンスをオンラインにするには、[オンライン] を選択します。
- このインスタンスから別のインスタンスにフェールオーバするには、[フェールオーバ] を選択します。

論理クラスタへのインスタンスの追加

論理クラスタにインスタンスを追加するには、次の手順に従います。

1 [追加] をクリックします。

2 追加するインスタンスを強調表示します。

3 [OK] をクリックします。これらの変更が効果を持つのは、[適用] または [OK] をクリックした後です。

[フェールオーバ・インスタンス] タブ

現在設定されているフェールオーバ・インスタンスについて、以下の情報を一覧表示します。

- 名前
- ID
- ステータス
- フェールオーバ・グループ

フェールオーバ・インスタンスの追加

フェールオーバ・インスタンスを追加するには、次の手順に従います。

1 [追加] をクリックします。

2 設定するフェールオーバ・グループを [論理クラスタにフェールオーバ・インスタンスを追加] ウィンドウから選択します。

フェールオーバ・グループによって、フェールオーバの場合にフェールオーバ・インスタンスがアクティブになる順序を指定できます。1 つのグループは、1 以上のインスタンスを持つことができます。

3 この一覧からフェールオーバ・インスタンスとして設定するインスタンスを強調表示し、[OK] をクリックします。

[負荷プロファイル] タブ

論理クラスタに関連付けられた負荷プロファイルの情報を表示します。

- 名前 − この論理クラスタと関連付けられた負荷プロファイルの名前。[変更] をクリックし、別の負荷プロファイルをこの論理クラスタと関連付けます。
- タイプ − 負荷プロファイル・タイプ (システムまたはユーザ) を表示します。
- 最小負荷スコア − ログイン・リダイレクトまたはマイグレーションをアクティブにする最小負荷スコア。

- 測定基準 − 負荷プロファイルに関するさまざまな統計が含まれます。以下が含まれます。
  - ユーザ接続 − 使用できるリソースに基づいた、新しい接続を受け入れるインスタンスの能力。
  - CPU 利用率 − 追加の作業を処理するインスタンスの能力。
  - 実行キューの長さ − システム上の実行可能なタスクの数。実行キューの長さは、処理バックログを測定するものであり、相対的な応答時間の良いインジケータです。
  - I/O 負荷 − 未処理の非同期 I/O。
  - エンジンの不足 − クラスタ内のインスタンス間のオンライン・エンジン数の違い。
  - ユーザ測定基準 − ユーザが提供する、ユーザ環境に固有のオプションの測定基準。
- 重み − 測定基準が負荷スコアにどのくらい重要か示します。これは、相対的な測定であり、100 という上限を基本にしています。重みが 0 の測定基準には価値がなく、負荷スコアの計算に使用されません。重みが 50 の測定基準は負荷スコアに半分の影響を及ぼします。5 つの測定基準すべてが重み 20 の場合、負荷スコアを計算するときの各測定基準の重要性は等しくなります。
- スレッショルド − 現時点で、この負荷プロファイルと関連付けられているスレッショルド。

[ルート] タブ

このタブには、この論理クラスタに割り当てられたアプリケーション、ログイン、エイリアス用のルートの記述があります。

アプリケーション・ルートの追加

1　[アプリケーション・ルートの追加] をクリックします。

2　アプリケーション名を入力し、[OK] をクリックします。

ログイン・ルートの追加

1　[ログイン・ルートの追加] をクリックします。

2　ルートのログイン (複数も可) を選択し、[OK] をクリックします。

エイリアス・ルートの追加

1 [エイリアス・ルートの追加] をクリックします。

2 エイリアス名を入力します。

3 [OK] をクリックします。

ルートの削除

1 ルートを選択します。

2 [ルートの削除] をクリックします。

3 [はい] を選択して、削除を確定します。

負荷ステータスの表示

Workloads フォルダには、[Weighted Scores] と [ベース測定基準値] の 2 つのタブに、負荷測定基準のクラスタ全体のビューが表示されます。

ツリー・ビューの Workload Management フォルダから [負荷] を選択します。負荷スコアは、重みスコアと基本スコアを報告します。

[Weighted Scores] タブ

各インスタンスが一連の負荷測定基準を追跡します。負荷スコアと測定基準はインスタンスと論理クラスタの組み合わせごとに計算され、論理クラスタの負荷プロファイルをインスタンスの負荷統計に適用することで決定されます。結果は、特定のインスタンス属性の相対的影響を表す一連の重み付けされたスコアと、全体的な負荷スコアです。

[Weighted Score] タブには、インスタンスと論理クラスタの組み合わせごとの、負荷スコアと重み付けされた測定基準値が表示されます。あるインスタンスが 2 つの論理クラスタと関連付けられている場合、[詳細] タブにはそのインスタンスについての 2 つのエントリがあります。

[Weighted Score] タブには以下の情報が含まれます。

- インスタンス − 負荷が表示されるインスタンスの名前。
- 論理クラスタ − インスタンスと関連付けられている論理クラスタの名前。
- 負荷プロファイル − 論理クラスタに割り当てられている負荷プロファイル。
- 負荷スコア − インスタンス上の全般的な負荷を表す計算済みの値。負荷を比較する手段として、インスタンス全体でこの単位のない数字を比較します。
- ユーザ接続 − 使用できるリソースに基づいた、新しい接続を受け入れるインスタンスの能力。
- CPU ビジー − エンジンがどのくらいビジーかの測定。sp_sysmon と同じ情報を提供します。追加作業を受け入れるためのインスタンスの容量を決定します。

- 実行キューの長さ − システム上の実行可能なタスクの数。実行キューの長さは、処理バックログを測定するものであり、相対的な応答時間の良いインジケータです。
- I/O 負荷 − 未処理の非同期 I/O を測定します。これは、インスタンス間の相対的な I/O 飽和状態を示します。
- エンジンの不足 − インスタンス間のオンライン・エンジンの相違を測定します。すべてのインスタンスに同じ数のエンジンがあるクラスタでは、エンジン不足はありません。ただし、たとえば “instance1” に 4 つのエンジンがあり、“instance2” に 2 つのエンジンがある 2 インスタンス・クラスタの場合、“instance1” にはエンジン不足がないものの、“instance2” には 50% の不足があります。“instance2” には “instance1” の半分のエンジンしかないからです。
- ユーザ − 負荷プロファイルでユーザが指定した測定基準の重み付けされた値です。

注意 各インスタンスは複数の論理クラスタに含まれる可能性があるので、各インスタンスには、所属する論理クラスタごとに 1 セットの測定基準値があります。

[ベース測定基準値] タブ

[ベース測定基準値] タブには、クラスタ内のインスタンスごとのすべての負荷値が表示されます。各インスタンスには 1 セットの値しかないので、そのインスタンスがいくつの論理クラスタに属しているかとは無関係に、1 セットの値がインスタンスごとに表示されます。

- インスタンス − インスタンスの名前。
- % User Connections − 使用中の設定済みユーザ接続のパーセント。
- % CPU busy − インスタンスが作業を実行していてビジーであった時間のパーセント。1 分間、システム上のすべてのエンジンから取られた移動平均です。
- % Run queue length − システムで実行可能なタスクの基本パーセント。実行キューの長さは、処理バックログを測定するものであり、相対的な応答時間の良いインジケータです。
- % IO load − 未処理非同期 I/O のパーセント。
- % Engine deficit − クラスタ内のインスタンス間のオンライン・エンジン数の基本パーセントの相違。
- % User − 負荷プロファイルに指定したユーザ測定基準用に提供するパーセント値。

インスタンスの負荷ステータスの表示

負荷ステータスは、インスタンスと関連付けられた負荷プロファイルを基礎にしています。ステータスは、各測定基準領域に関して、インスタンスがどのように重い負担をかけられるかを示す raw 値を表示します。

負荷ステータスを表示するには、次の手順に従います。

1 [サーバ・インスタンス] フォルダをクリックして、右ウィンドウ枠にあるインスタンスのリストを表示します。

2 インスタンス名を右クリックし、[負荷ステータス] を選択します。負荷ステータスは、各インスタンスが測定基準領域ごとに実行している作業の総量を示す raw 値を表示します。

- ユーザ接続 − 使用できるリソースに基づいた、新しい接続を受け入れるインスタンスの能力。
- CPU ビジー − エンジンがどのくらいビジーかの測定。sp_sysmon と同じ情報を提供します。追加作業を受け入れるためのインスタンスの容量を決定します。1 分間の移動平均として測定されます。
- 実行キューの長さ − システム上の実行可能なタスクの数。実行キューの長さは、処理バックログを測定するものであり、相対的な応答時間の良いインジケータです。1 分間の移動平均として測定されます。
- I/O 負荷 − 未処理の非同期 I/O を測定します。これは、インスタンス間の相対的な I/O 飽和状態を示します。1 分間の移動平均として測定されます。
- エンジンの不足 − インスタンス間のオンライン・エンジンの相違を測定します。すべてのインスタンスに同じ数のエンジンがあるクラスタでは、エンジン不足はありません。ただし、たとえば “instance1” に 4 つのエンジンがあり、“instance2” に 2 つのエンジンがある 2 インスタンス・クラスタの場合、“instance1” にはエンジン不足がないものの、“instance2” には 50% の不足があります。“instance2” には “instance1” の半分のエンジンしかないからです。
- ユーザ − workload_metric 関数でユーザが指定した測定基準の重み付けされた値です。

注意 各インスタンスは複数の論理クラスタに含まれる可能性があるので、各インスタンスには、所属する論理クラスタごとに 1 セットの測定基準値があります。

ルートの管理

ルートによって、クライアント接続を特定の論理クラスタに向けることができます。

ルート・プロパティ

現在のルートを表示するには、次の手順に従います。

1　[負荷管理] － [論理クラスタ] フォルダで、論理クラスタ名を右クリックします。

2　プロパティを選択します。

3　[ルート] タブを選択します。

[Routes General] タブは以下を報告します。

- ルートの名前。
- ルート・タイプ － ルートのタイプ (アプリケーション、ログイン、またはエイリアス) を表示します。
- 論理クラスタ － このルートが関連付けられている論理クラスタの名前。

  このルートを別の論理クラスタに関連付けるには、別の論理クラスタ名を選択し、[OK] をクリックします。

ルートの作成

ルートの追加ウィザードを使用してルートを作成します。

ウィザードを開始するには、次の手順に従います。

- Workload Management フォルダから、[ルート] を選択します。
- [ルートの追加] をダブルクリックします。
- ウィザードの指示に従って、ルートを追加します。

第 15 章

sybcluster を使用したクラスタの管理

sybcluster は、コマンド・ラインからクラスタの管理タスクを実行できる対話型のユーティリティです。

sybcluster を使用すると、クラスタの作成、インスタンスの追加、クラスタまたはインスタンスの起動と停止、ステータス情報の表示などができます。また、「第 14 章 Adaptive Server プラグインによるクラスタの管理」での説明のとおり、これらのタスクは、Adaptive Server プラグインを使用して実行することもできます。

sybcluster を使用してテンポラリ・データベースまたは論理クラスタを管理することはできません。

- テンポラリ・データベースを管理するには、Adaptive Server プラグインを使用するか、「第 8 章 テンポラリ・データベースの使用」の手順に従ってください。
- 論理クラスタを管理するには、Adaptive Server プラグインを使用するか、「第 6 章 負荷の管理」で説明されている sp_cluster logical ストアド・プロシージャおよび sp_cluster profile ストアド・プロシージャを使用します。

sybcluster コマンドの完全な構文および使用法については、『ユーティリティ・ガイド』を参照してください。

sybcluster の使用

sybcluster コマンドには、クラスタに接続する前に使用可能であるものと、クラスタに接続した後のみに使用可能になるものがあります (表 15-1 を参照)。

sybcluster プロンプトには、sybcluster がクラスタに接続されているかどうかを示す情報が表示されます。また、クラスタとインスタンスのいずれか、または両方が設定されている場合は、デフォルトのクラスタとインスタンスを示す情報が表示されます。

- sybcluster がクラスタに接続されていない場合、プロンプトには次のように表示されます。

  ```
  >
  ```

- クラスタに接続後は、表示が以下のように変わります。

  ```
  cluster_name>
  ```

  次に例を示します。

  ```
  mycluster>
  ```

- デフォルトのインスタンスを宣言する場合、プロンプトには次のように表示されます。

  ```
  cluster_name instance_name>
  ```

  次に例を示します。

  ```
  mycluster ase1>
  ```

表 15-1: sybcluster の対話型コマンド

コマンド名	説明	アクティブな期間 (クラスタへの接続前または接続後)
add backupserver	Backup Server が現在設定されていないノードに 1 つまたは複数の Backup Server を設定します。	接続後
add instance	クラスタに新しいインスタンスを 1 つ追加します。	接続後
connect	既存のクラスタに接続します。	接続前
create backupserver	Backup Server を作成します。	接続後
create cluster	新しいクラスタを作成します。	接続前
create xpserver	XP Server を作成します。	接続後
deploy plugin	クラスタの 1 つのインスタンスの設定情報を Unified Agent に追加します。	接続前
diagnose { cluster | instance }	一連のチェックを実行して、クラスタまたはインスタンスが適切に動作していることを確認します。	接続後
disconnect	現在のクラスタのすべての接続を閉じ、クラスタを未接続の状態に戻します。	接続後
drop backupserver	Backup Server を削除します。	接続後
drop cluster	クラスタから各インスタンスを削除し、クラスタ設定ファイルからクラスタ定義を削除します。	接続後

コマンド名	説明	アクティブな期間 (クラスタへの接続前または接続後)
drop xpserver	XP Server を削除します。	接続後
drop instance	クラスタからインスタンスを削除します。	接続後
exit	sybcluster を終了します。	接続前または接続後
help	現在使用できる sybcluster の対話型コマンドのリストを表示します。	接続前または接続後
localize	デフォルトの言語、文字セット、ソート順の現在の値を表示します。デフォルト値の変更および言語の追加と削除ができます。	接続後
quit	sybcluster を終了します。	接続前または接続後
set backupserver	1 つまたは複数のノード上にある Backup Server の受信ポート番号を変更します。	接続後
set cluster	クラスタにプロパティを設定します。	接続後
set instance	インスタンスのプロパティを設定します。	接続後
set xpserver	1 つまたは複数のノード上にある XP Server の受信ポート番号を変更します。	接続後
show agents	使用できる UAF エージェントに関する情報を表示します。	接続前
show backupserver config	Backup Server が設定されているノード名と関連付けられている受信ポート番号を表示します。	接続後
show cluster	クラスタの設定、ログ、ステータスの値を表示します。	接続後
show instance	インスタンスについての情報を表示します。	接続後
show membership mode	クラスタの現在のメンバシップ・モードを表示します。メンバシップ・モードは、クラスタが Veritas Cluster Server の統合をサポートしているかどうかを示します。	接続後
show session	現在のエージェントと検出の情報を表示します。	接続後
show xpserver config	XP Server が設定されているインスタンス名とノード名、および関連付けられている受信ポート番号を表示します。	接続後
shutdown cluster	各インスタンスに対して、Transact-SQL shutdown コマンドを実行してクラスタを停止します。	接続後
shutdown instance	Transact-SQL shutdown コマンドを実行してインスタンスを停止します。	接続後
start cluster	クラスタ内のすべてのインスタンスを起動します。	接続後
start instance	クラスタ内の 1 つのインスタンスを起動します。	接続後
upgrade server	Adaptive Server を Adaptive Server Cluster Edition にアップグレードします。	接続前
use	デフォルトのインスタンスを設定します。	接続後

sybcluster と Unified Agent Framework

sybcluster は、Unified Agent Framework (UAF) を使用して、分散 Sybase リソースを管理するためのランタイム・サービスという形式のリモート管理機能を提供します。この機能には、共通のサービス・セットが装備されています。また、sybcluster や Adaptive Server プラグインなどの Sybase プロセスを有効にしてエージェントにプラグインし、サーバ・リソースを管理したりさまざまな操作を実行したりできます。

Unified Agent サーバは、UDP を使用してサブネット上でブロードキャストを実行したり、Jini や LDAP などのルックアップ・サーバに登録したりできます。

UAFの詳細については、『Unified Agent および Agent Management Console ユーザーズ・ガイド』を参照してください。

sybcluster の起動

sybcluster をコマンド・ラインから起動します。以下を入力するのが最も簡単な方法です。

```
sybcluster -U uafadmin -P
```

“uafadmin” はデフォルトのユーザ名で、デフォルトのパスワードは null またはブランクです。認証済みの Adaptive Server データベースのユーザ名およびオペレーティング・システムのユーザ名を使用することもできます。「ユーザの認証」(255 ページ) を参照してください。

また、sybcluster を起動すると同時に、クラスタを指定し、デフォルトのインスタンスを識別し、そのクラスタ上の 1 つまたは複数の Unified Agent に接続することもできます。次に例を示します。

```
sybcluster -U uafadmin -P -C mycluster -I ase1
-F "blade1:9999,blade2:9999,blade3:9999"
```

この例では、-F オプションによって、クラスタ内の各 Unified Agent のノードと受信ポートが識別されています。このコマンドを頻繁に使用する場合は、簡単なエイリアスを作成できます。次に例を示します。

```
sybcluster_mycluster
```

クラスタの作成

sybcluster を使用してクラスタを作成する手順については、使用しているプラットフォーム用の『インストール・ガイド』を参照してください。

クラスタへの接続

sybcluster の起動時に、必要な情報を指定してクラスタに接続できます。また、sybcluster を起動してから connect コマンドを使用してクラスタに接続することもできます。クラスタに接続してから、クラスタを起動する必要があります。

クラスタのインストールおよび設定時に、Unified Agent のクラスタ内の各ノードへのインストールおよび設定も行います。sybcluster の起動時には、認証済みのユーザ名とパスワードを入力する必要があります。また、直接接続方法または検出方法を使用して、1 つまたは複数のクラスタのノード上の Unified Agent を識別する必要があります。sybcluster は、Unified Agent にプラグインし、コマンド、制御、および検出の各操作を実行できます。

ユーザの認証

クラスタへの接続には、任意の UAF の認証済みユーザ名とパスワードを使用できます。デフォルトでは、ユーザ名は “uafadmin”、パスワードは null またはブランク文字列です。次に例を示します。

```
sybcluster -U uafadmin -P -C mycluster
```

ユーザ名とパスワードの設定

デフォルトのユーザ名とパスワードを変更し、新しいパスワードを暗号化することをおすすめします。UAF 用のユーザ名とパスワードは、クラスタ内の各ノード上にある *csi.properties* ファイルに格納されます。

❖ ユーザ名の設定

- 新しいユーザ名を、各 *$SYBASE/UAF-2_5/nodes/<node_name>/conf/csi.properties* ファイルの Simple Login Module セクションにある “username” プロパティに入力します。次に例を示します。

```
# Simple Login Module
...
CSI.loginModule.2.options.username=newusername
CSI.loginModule.2.options.password=
CSI.loginModule.2.options.encrypted=false
CSI.loginModule.2.options.roles=uaAgentAdmin,uaPluginAdmin
```

注意　クラスタ内の各ノードの *csi.properties* ファイルを必ず編集してください。

❖ **パスワードの暗号化と設定**

1 *$SYBASE/UAF-2_5UAF-2_5/bin* にある passencrypt を実行すると、暗号化パスワードが生成されます。

2 各 *$SYBASE/UAF-2_5/nodes/<node_name>/conf/csi.properties* ファイルの Simple Login Module セクションで次の操作を行います。

1 暗号化された値を password プロパティにペーストします。

2 encrypted プロパティを "true" に設定します。次に例を示します。

```
# Simple Login Module
...
CSI.loginModule.2.options.username=newusername
CSI.loginModule.2.options.password=REVTe1NZVUFGfWNvbS5zdW4uY3J5cHRvLnByb3ZpZ
GVyLlN1bkpDRXtTWVVBRn1nTUJacVh5R3pnN09RSDJDN1NPUXhBPT0=
CSI.loginModule.2.options.encrypted=true
CSI.loginModule.2.options.roles=uaAgentAdmin,uaPluginAdmin
```

注意 クラスタ内の各ノードの *csi.properties* ファイルを必ず編集してください。

❖ **新しいユーザ名とパスワードのアクティブ化**

- 新しいユーザ名およびパスワードを有効にするには、クラスタ内の各ノード上の Unified Agent を終了して再起動します。

Unified Agent の識別

Unified Agent は、sybcluster の起動時、またはその後の connect to 対話型コマンドの使用時のいずれかで、直接接続方法または検出方法を使用して識別できます。

注意 クラスタへの接続中にそのクラスタが実行中の場合は、通常、そのクラスタの 1 つのノード上で 1 つの Unified Agent のみを識別する必要があります。ただし、クラスタまたは XP Server を起動する場合は、クラスタ内の各ノードでエージェントを識別する必要があります。

sybcluster では、追加の情報が必要になるたびに、その情報を要求するプロンプトが表示されます。

クラスタの Unified Agent の仕様がわからない場合、sybcluster show agents コマンドを使用して、サブネット上で使用可能な Unified Agent とクラスタを検出します。

直接接続方法の使用

クラスタの管理を行う Unified Agent のクラスタ・ノードとポート番号を指定すると、1 つまたは複数のエージェントに直接接続できます。エージェント指定の指定には次のものを使用できます。

- クラスタの 1 つまたは複数のノード名およびオプションのポート番号 − 正確なアドレスが指定できます。ポート番号を入力しない場合、sybcluster は、デフォルト値である 9999 を使用します。たとえば、“mycluster” の “blade1”、“blade2”、および “blade3” 上のエージェントを指定するには、次のように入力します。

  ```
  sybcluster -U uafadmin -P -C mycluster
  -F "blade1:1234,blade2:2345,blade3:3456"
  ```

  sybcluster の起動後に connect を使用してノードとポート番号を指定するには、次のように入力します。

  ```
  connect to mycluster -U uafadmin -P
  -F "blade1:1234,blade2:2345,blade3:3456"
  ```

- ノードのドメイン − ドメイン名を使用して正確なアドレスを指定できます。次に例を示します。

  ```
  sybcluster -U uafadmin -P -C mycluster
  -F "blade1.mydomain.com"
  ```

検出方法の使用

sybcluster では、エージェントと検出順序を特定する次の 3 つの検出方法がサポートされています。

- UDP (User Datagram Protocol) − sybcluster が、同じサブネット応答上に存在する要求およびエージェントをブロードキャストするブロードキャスト検出方法。たとえば、UDP を使用して “mycluster” 内のノードのロケーションを検索するには、次のように入力します。

  ```
  sybcluster -U uafadmin -P -C mycluster
  -d "udp()"
  ```

  sybcluster の起動後に connect を使用して検出を実行するには、次のように入力します。

  ```
  connect to mycluster login uafadmin password " " discovery
  "udp(),
  jini(myjinihost1:5678;myjinihost2:1234)"
  ```

- Jini サーバ技術 − 検索機能が装備されています。Jini サーバを使用してエージェントを登録すると、それぞれのエージェントの各ノードのロケーションおよびステータス情報が Jini サーバに格納されます。“mycluster” 内のノードのエージェントのロケーションを検索するには、次のように入力します。

  ```
  sybcluster -U uafadmin -P -C mycluster
  -d "jini(myjiniserver:4564)"
  ```

sybcluster の起動後に connect を使用して検出を実行するには、次のように入力します。

```
connect to mycluster discovery "jini(myjiniserver:4564)"
```

- LDAP 技術 – 検索機能が装備されています。LDAP サーバを使用してエージェントを登録すると、それぞれのエージェントの各ノードのロケーションおよびステータス情報が LDAP サーバに格納されます。

“mycluster” の Unified Agent のロケーションを、3 つの検出方法のすべてを使用して検索するには、次のように入力します。

```
sybcluster -U uafadmin -P -C mycluster
-d "udp(),jini(myjiniserver:4123),
ldap(myldapserver:6123)
```

クラスタの起動

クラスタを起動する前に、ターゲット・クラスタにある各ノード上の Unified Agent に接続する必要があります。「クラスタへの接続」(255 ページ) を参照してください。接続後にクラスタを起動するには、次のように入力します。

```
start cluster
```

sybcluster によって、クラスタ内のすべてのインスタンスが起動され、クラスタの説明が表示されます。

クラスタの管理

この項では、クラスタとその環境を管理するために使用するタスクの実行方法を説明します。

クラスタの作成

sybcluster と Adaptive Server プラグインのいずれかを使用してクラスタを作成できます。いずれの方法でも、必要となる情報を要求するプロンプトが表示され、クラスタが自動的に作成されます。また、すべてのタスクを手動で実行してクラスタを作成することもできます。

これらの方法でクラスタを作成するための手順については、『インストール・ガイド』を参照してください。

クラスタの確認

diagnose cluster では、クラスタが正常に動作していることを確認するための一連のチェックが実行されます。次のように入力します。

```
diagnose cluster
```

diagnose cluster では、クラスタに関する情報が表示され、次のチェックが実行されます。

- Unified Agent がクラスタのすべてのノード上で実行されているかどうか。
- クラスタ内のノード数がクラスタのインスタンスの最大数を超過していないかどうか。
- クォーラム・デバイスが存在するかどうか。存在しない場合、ディレクトリに書き込みパーミッションが設定されているかどうか。
- すべてのノードに対して interfaces ファイルが存在するかどうか。また、ノード名およびポート番号の競合がないかどうか。
- プライマリ・プロトコルとセカンダリ・プロトコルの指定が重複していないかどうか。
- 各ノード上の Sybase ホーム・ディレクトリが共有されているかどうか。

使用可能な Unified Agent に関する情報の表示

show agents を使用すると、サブネット上に設定済みのすべての Unified Agent を識別したり、検索を絞り込んで特定の Unified Agent に関する情報を表示したりできます。

たとえば、すべての Unified Agent を識別するには、次のように入力します。

```
show agents
```

sybcluster によって、各 Unified Agent の直接接続アドレス、そのノード、クラスタ名、およびその他の関連情報が表示されます。

検出を制限したり目的のエージェントを指定したりすることで、特定の Unified Agent に関する情報を表示できます。たとえば、“mycluster” の “blade2” にある Unified Agent に関する情報を表示するには、次のように入力します。

```
show agents agent "blade2:9999"
```

クラスタ情報の表示

この項では、sybcluster を使用して、クラスタ、クラスタ内のインスタンス、およびクラスタ環境に関する情報を表示する方法を説明します。完全な構文および使用方法については、『ユーティリティ・ガイド』を参照してください。

同じタスクの実行に、Adaptive Server プラグイン使用することもできます。「第 14 章 Adaptive Server プラグインによるクラスタの管理」を参照してください。

- 設定情報、プライマリ・プロトコルおよびセカンダリ・プロトコルの値、使用されているトレース・フラグ、クォーラム・デバイスおよびマスタ・デバイスのアドレスを表示するには、次のように入力します。

  ```
  show cluster config
  ```

  フォーマットされたクラスタの設定情報を表示するには、次のように入力します。

  ```
  show cluster config template
  ```

- クラスタ内の各インスタンスのステータスおよびハートビート情報を表示するには、次のように入力します。

  ```
  show cluster status
  ```

  ステータス値は次のとおりです。

  - Up (起動)
  - Down (停止)
  - Undefined (未定義)
  - Invalid (無効)
  - Start (開始)
  - Init (初期化)
  - Quiesce (静止)

- すべてのログ情報を表示するには、次のように入力します。

  ```
  show cluster log
  ```

  次のように指定すると、出力を制限できます。

  - エラーの重大度によって制限する場合の例

    ```
    show cluster log minseverity 5
    ```

  - ログ・エントリの日付範囲によって制限する場合の例

    ```
    show cluster log startdate 03:31:08
    enddate 04:30:08
    ```

  - エラー・ログからの表示行数 (最新行から逆順にカウント) によって制限する場合の例

    ```
    show cluster log last 25
    ```

- クラスタへのすべての UAF および JDBC 接続を表示するには、次のように入力します。

  ```
  show cluster connection
  ```

- クラスタに関する一般情報およびエージェント接続に関する詳細情報を表示するには、次のように入力します。

  ```
  show session
  ```

- Symantec Veritas Cluster Server (VCS) は、クラスタ環境内のアプリケーション・サービスを管理します。サイトで VCS がサポートされている場合、クラスタ上で VCS を有効にできます。詳細については、Veritas のマニュアルを参照してください。クラスタで VCS が有効であるかどうかを検出するには、メンバシップ・モードを表示します。次のように入力します。

  ```
  show membership mode
  ```

  メンバシップ・モードの値は、次のとおりです。

  - `vcs` ― クラスタは、VCS 統合をサポートしています。
  - `native` ― クラスタは、VCS をサポートしていません。

  「メンバシップ・モードの変更」(263 ページ) を参照してください。

クラスタ設定値の変更

set cluster やその他のコマンドを使用して、クラスタの特定の設定値を変更できます。クラスタのステータスを確認するには、show cluster status を実行します。

クラスタの停止時には、次の変更ができます。

- インスタンスの最大数
- アクティブなトレース・フラグ
- プライマリ・プロトコルまたはセカンダリ・プロトコル
- メンバシップ・モード

クラスタの実行時には、次の変更ができます。

- Unified Agent がクラスタへのログインに使用するログイン名またはパスワード
- クラスタのデフォルトの言語、文字セット、ソート順

たとえば、“mycluster” のインスタンスの最大数を 4 に変更するには、次のように入力します。

```
set cluster maxinst 4
```

クラスタのプライマリ・プロトコルを “udp” にリセットするには、次のように入力します。

```
set cluster primary protocol udp
```

ユーザ名またはパスワードの変更

sybcluster は、クラスタを実行するために設定されている Unified Agent に接続するクライアント・プログラムです。sybcluster の起動時には、sybcluster が Unified Agent にログインするためのログインとパスワードを指定します Unified Agent のログインまたはパスワードの値を変更するには、Agent Management Console Sybase Central プラグインを使用します。パスワードの暗号化については、「ユーザ名とパスワードの設定」(255 ページ) を参照してください。

操作によっては、Unified Agent がクラスタにログインする必要がある場合もあります。sybcluster または Sybase Central が shutdown コマンドを実行する場合、または Unified Agent がクラスタ・ハートビート・タスクを実行してクラスタのステータスを判断する場合がこれにあたります。これらのタスクでは、Unified Agent は sa_role を使用したログインを使用する必要があります。デフォルトでは、Unified Agent は “sa” ログインをパスワードなしで使用します。このパスワードを変更するには、sybcluster set cluster login を使用します。

たとえば、“sa” ログインのパスワードを “newpassword” に変更するには、次のように入力します。

```
set cluster login sa password newpassword
```

このコマンドを実行するには、クラスタが実行中である必要があります。

完全な構文および使用方法については、『ユーティリティ・ガイド』を参照してください。

ローカライゼーション値の変更

sybcluster localize コマンドを使用して、言語、文字セット、およびソート順の現在値を表示します。現在のデフォルト値の表示後に、これらの各値を受け入れるか、変更するかをたずねるプロンプトが sybcluster localize によって表示されます。たとえば、現在値を変更せずに表示するには、次のように入力します。

```
localize

Current default locale properties are:
Default Language - portuguese
Default Charset - mac
Default SortOrder - Binary ordering, for use with the Macintosh
charcter set(mac).

Options for default Language are:
1. spanish
```

```
2. portuguese
3. german
4. us_english
5. thai
6. french
7. japanese
8. chinese
9. korean
10. polish
Enter the number representing the language to be set as
defaults: [2]

Options for default charsets are:
1. gb18030
2. eucgb
3. uttf8
Enter the number representing the charset to be set as default:
[1]

Options for sort orders are:
1. Binary ordering, for the EUC GB2312-80 character set
(eucgb).
Enter the number representing the sort order to be set as
default [1]

Do you want to install any language? [Y] n
Do you want to remove any language? [N ]
The cluster mycluster was successfully localized with default
language portuguese, charset gb18030, sortorder bin_eucgb.
```

クラスタ全体での一貫性を保つために、ローカライゼーション値のいずれかを変更した場合は、クラスタを停止して再起動してください。

メンバシップ・モードの変更

サイトで VCS がサポートされている場合、クラスタの作成時に VCS を有効にできます。qrmutil コマンドまたは Veritas ユーティリティを使用して、メンバシップ・モードを変更します。詳細については、『ユーティリティ・ガイド』と Veritas のマニュアルを参照してください。

クラスタからの切断

現在のクラスタへのすべての接続を切断するには、次のように入力します。

```
disconnect
```

クラスタの停止

クラスタを安全モードで停止すると、Adaptive Server がクラスタの設定ファイル内で指定された順番で各インスタンスを停止する前に、トランザクションを完了できます。

```
shutdown cluster
```

トランザクションの終了を待たずに、クラスタをただちに停止するには、次のように入力します。

```
shutdown cluster nowait
```

注意 クラスタが Veritas Cluster Server (VCS) モードで実行されている場合にサーバおよびクラスタを停止または起動する際には、必ず VCS の停止および起動のメカニズムを使用してください。sybcluster shutdown コマンドは使用しないでください。

クラスタの削除

クラスタの削除を実行する前に、クラスタの状態が Down で、Unified Agent が実行されていることを確認してください。確認したら、次のように入力します。

```
drop cluster
```

Adaptive Server によって、クラスタとインスタンスのエントリの interfaces ファイルからの削除、クラスタの設定ファイルの削除、クォーラム・ディスクの未使用な状態へのマーク付け、ログ・ファイルの削除、クラスタの Unified Agent プラグインの停止と削除が行われます。削除は、ユーザが確認する必要があります。

インスタンスの管理

この項では、クラスタ内のインスタンスを管理するために使用するタスクの実行方法を説明します。

インスタンスについての情報の表示

show cluster と同様に、show instance では、インスタンスの設定、ステータス、およびログ情報が表示されます。

- ホスト・ノード名、プライマリおよびセカンダリ・ネットワーク情報、およびログ・ファイルへのパスを含む設定情報を表示するには、次のように入力します。

  ```
  show instance instance_name config
  ```

- ステータス情報を表示するには、次のように入力します。

  ```
  show instance instance_name status
  ```

  このコマンドでは、指定されたインスタンスのステータス情報が次のように表示されます。

  - Up (起動)
  - Down (停止)
  - Undefined (未定義)
  - Invalid (無効)
  - Start (開始)
  - Init (初期化)
  - Quiesce (静止)

- すべてのログ情報を表示するには、次のように入力します。

  ```
  show instance instance_name log
  ```

  次のように指定すると、出力を制限できます。

  - エラーの重大度によって制限する場合の例

    ```
    show instance instance_name log minseverity 5
    ```

  - ログ・エントリの日付範囲によって制限する場合の例

    ```
    show instance instance_name log startdate 03:31:08
    enddate 04:30:08
    ```

  - エラー・ログからの表示行数 (最新行から逆順にカウント) によって制限する場合の例

    ```
    show instance instance_name log last 25
    ```

インスタンスの追加

クラスタにインスタンスを追加するには、sybcluster のプロンプトで必要な値を対話形式で指定する方法、または入力ファイルを使用する方法のいずれかの方法を使用できます。

注意 add instance は、新しいインスタンス用のローカル・システム・テンポラリ・データベースを作成します。デバイス上に十分な領域があることを確認してください。

インスタンスを対話形式で追加する場合、次の情報を要求するプロンプトが Adaptive Server によって表示されます。

- インスタンス名 (コマンド文で指定されていない場合)
- インスタンスをホストするノード名
- ノード上の Unified Agent のポート番号
- クエリ・ポート番号
- ノードのプライマリ・アドレスとセカンダリ・アドレス
- プライマリ・ポートとセカンダリ・ポートの指定

入力ファイルを使用してインスタンスを追加する場合、ファイルに含める必要がある定義は新しいインスタンスの定義のみですが、そのファイルの形式がクラスタ入力ファイルの形式を反映していることを確認してください (詳細は、プラットフォームの『インストール・ガイド』を参照)。補助サーバが定義されている場合、新しいインスタンス用に補助サーバを設定するために、ポートおよびその他の必要となる情報を含めてください。

入力ファイルを使用してインスタンスを追加するには、たとえば、次のように入力します。

```
add instance new_instance file /$SYBASE/myfile
```

インスタンスを対話形式で追加するには、次のように入力します。

```
add instance new_instance
```

インスタンスの確認

diagnose instance では、インスタンスが適切に設定されていることを確認するための一連のチェックが実行されます。たとえば、“ase1” の設定を確認するには、次のように入力します。

```
diagnose instance ase1
```

diagnose instance によって、次のようにインスタンスの設定情報が表示および確認されます。

- クエリ・ポート
- JDBC 接続が使用可能であること
- インスタンスがパブリック・ネットワーク上で使用可能であること
- ポート数の最小値と最大値
- プライマリ・プロトコル・ポートとセカンダリ・プロトコル・ポートの範囲

デフォルトのインスタンスの変更

use コマンドを使用すると、sybcluster コマンド・ラインで指定されたデフォルトのインスタンスを設定または変更できます。デフォルトのインスタンスの設定後は、対話型コマンドのコマンド・ライン内でインスタンスを指定する必要はありません。次のように入力します。

```
use ase1
```

対話型コマンド内にインスタンス名を含めると、デフォルトのインスタンスをオーバーライドできます。ただし、この場合、デフォルトの指定は変更されません。

デフォルトの指定を削除するには、インスタンス名を省略します。次のように入力します。

```
use
```

インスタンスのプロパティの変更

set instance を使用すると、インスタンスの特定のプロパティを変更できます。set instance を使用する場合、インスタンスは停止状態にある必要があります。show cluster status を実行してステータスを確認してください。

変更できるインスタンスのプロパティは、次のとおりです。

- ログ・パス
- インスタンスの起動に使用する引数
- インスタンスのプライマリ・アドレスまたはセカンダリ・アドレス

- インスタンスが使用するプライマリ・ポート範囲またはセカンダリ・ポート範囲

たとえば、プライマリ・ポート範囲を 6123 から 6126 に設定し直すには、次のように入力します。

```
set instance primary port 6123 6126
```

インスタンスの停止

インスタンスを安全モードで停止すると、トランザクションを完了できます。たとえば、“ase1” を停止するには、次のように入力します。

```
shutdown instance ase1
```

トランザクションの終了を待たずに、インスタンスをただちに停止するには、次のように入力します。

```
shutdown instance ase1 nowait
```

クラスタ内の最後のインスタンスを停止すると、クラスタのステータスが Down に変化します。クラスタ・コーディネータをホストしているノード上でインスタンスを停止した場合は、別のノードがコーディネータをホストします。

インスタンスの削除

インスタンスの削除を実行する前に、インスタンスが停止状態、クラスタが実行状態であることを確認してください。確認したら、次のように入力します。

```
drop instance instance_name
```

Adaptive Server は、interfaces ファイルとクォーラム・デバイスからそのインスタンスのエントリを削除し、トポロジの変化についてクラスタに通知を行います。削除をそれぞれ確認します。

注意 drop instance は、クラスタ内の最後のインスタンスの削除には使用できません。この場合、drop cluster を使用します。

クラスタの手動作成後の sybcluster の有効化

クラスタの作成には、通常の場合、sybcluster または Adaptive Server プラグインを使用します。この場合、Adaptive Server は、sybcluster または Adaptive Server プラグインが各ノード上の Unified Agent に接続するために必要な設定情報を自動的に追加します。クラスタを手動で設定する場合 (『インストール・ガイド』を参照)、sybcluster または Adaptive Server プラグインを使用してクラスタの管理を行う前に、Unified Agent に設定情報を追加する必要があります。

まず、プラグイン設定情報を配備する必要があります。

1　クラスタ上で Unified Agent を起動します (まだ起動されていない場合)。使用しているプラットフォーム用の『インストール・ガイド』の「第 3 章　サーバのインストールとクラスタの起動」を参照してください。

2　プラグインを配備します。たとえば、デフォルトのクラスタ “mycluster” にあるクラスタ・エージェントのプラグイン情報を配備するには、次のように入力します。

```
deploy plugin agent "blade1,blade2,blade3"
```

Unified Agent は、「Unified Agent の識別」(256 ページ) で説明されている直接接続方法または検出方法のいずれかを使用して指定できます。

エージェントを指定すると、Adaptive Server によって、次のパスを指定するためのプロンプトが表示されます。

- クォーラム・デバイス
- 環境シェル・スクリプト
- ASE ホーム・ディレクトリ

注意　sybcluster または Adaptive Server プラグインを使用すると、クラスタ内の単一のノードにプラグインを配備した後のクラスタを管理できます。ただし、クラスタを起動するには、クラスタ内のすべてのノードにプラグインを配備する必要があります。

---

deploy plugin を使用して、既存のプラグインの値を更新することもできます。

補助サーバの作成と管理

sybcluster を使用して、次の補助サーバに対して、ポート番号の作成、削除、設定、および現在のポート番号の表示ができます。

- Backup Server
- XP Server

補助サーバの作成

sybcluster を使用して、Backup Server および XP Server を作成する手順については、『インストール・ガイド』を参照してください。補助サーバの作成には、Adaptive Server プラグインを使用することもできます。

構文および使用方法については、『ユーティリティ・ガイド』を参照してください。

複数の Backup Server を作成する場合、クラスタ内のすべてのインスタンスに Backup Server が存在する必要があります。1 つの Backup Server を 1 つまたは複数のノードに対して作成できます。

複数の XP Server を作成する場合、これらのサーバは、クラスタ内のすべてのインスタンスに対して作成する必要があります。

補助サーバの削除

drop backupserver、または drop xpserver は、クラスタから補助サーバを削除するときに使用します。サーバの削除の前に、確認のプロンプトが表示されます。

クラスタに複数の Backup Server を設定した場合は、すべての Backup Server を削除する必要があります。1 つの Backup Server を設定した場合は、1 つまたはすべての Backup Server を削除できます。ただし、drop xpserver を使用すると、すべての XP Server がクラスタから削除されます。

“mycluster” からすべての XP Server を削除するには、次のように入力します。

```
drop xpserver

Are you sure you want to drop the XP Servers from cluster "mycluster"? (Y or N): [N] y
The XP Servers have been dropped for all instances.
```

“mycluster” の “blade2” から Backup Server を削除するには、次のように入力します。

```
drop backupserver

Do you want to drop the Backup Server from:
        1. Selected nodes
        2. Cluster
Enter choice: 1
Do you want to drop Backup Server from node "blade1"? [N] n
Do you want to drop Backup Server from node "blade2"? [N] y
Do you want to drop Backup Server from node "blade3"? [N] n
The Backup Server has been dropped.
```

受信ポート情報の表示

Backup Server または XP Server 用の現在の受信ポート番号を表示するには、次のコマンドを使用します。

- show backupserver config
- show xpserver config

たとえば、“mycluster” の Backup Server の受信ポートに関する情報を表示するには、次のように入力します。

```
show backupserver config

Backup Server is configured on the following nodes:
    1. blade1:5001
    3. blade3: 5003
```

受信ポート情報の変更

補助サーバの受信ポートを変更するには、次のコマンドを使用します。

- set backupserver
- set xpserver config

たとえば、“blade3” 上のインスタンス “ase3” の Backup Server の受信ポート番号を変更するには、次のように入力します。

```
set backupserver

Enter the Backup Server port number for instance "blade1": [6011] <CR>
Enter the Backup Server port number for instance "blade2": [6012] <CR>
Enter the Backup Server port number for instance "blade3": [6013] 6666
```

サーバのアップグレード

Adaptive Server を最新バージョンの Cluster Edition にアップグレードするための手順については、使用しているプラットフォーム用の『インストール・ガイド』を参照してください。

第 2 部

一般的な設定に関する問題

ここでは、Cluster Edition の一般的な設定に関する問題を示します。

第 16 章

オペレーティング・システムの設定

この章では、Cluster Edition をインストールまたはアップグレードした後で調整できるオペレーティング・システムの設定について説明します。特に明記されていないかぎり、この章の情報はサポートされている UNIX プラットフォームすべてに適用できます。

stty 設定の使用

stty tostop オプションを設定すると、バックグラウンド・サーバがターミナルに書き込もうとした時にそのサーバが停止してしまいます。このエラーを回避するには、次のコマンドを実行してから Cluster Edition を起動します。

```
stty -tostop
```

Cluster Edition の出力をすべてファイルにリダイレクトする場合、stty の設定を変更する必要はありません。

正しいパーミッションのリストア

Sybase ソフトウェアのファイルとディレクトリには、インストール時に正しいアクセス・パーミッションが設定されます。パーミッションが正しくないことに気づいた場合、*$SYBASE/$SYBASE_ASE/install* ディレクトリに保存されているスクリプト setperm_all を使用して正しいパーミッションをリストアできます。

ファイル記述子とユーザ接続

Cluster Edition によって使用されるユーザ接続の数は、オペレーティング・システムで Cluster Edition が使用できるファイル記述子の数を超えることはできません。Cluster Edition のユーザ接続を設定する場合、システム管理者は 1 つのプロセスあたりで使用可能なファイル記述子の数を考慮に入れる必要があります。オープン可能なファイル記述子のほとんどはユーザ接続で使用できます。Cluster Edition によってファイルとデバイスのオープンに使用されるものはごく一部です。

Linux の場合

1 プロセスあたりのファイル記述子の数は 10,000 に制限されています。ulimit を使用してファイル記述子の数を設定できます。

Sun Solaris の場合

Sun Solaris では、ファイル記述子に対してソフト制限値とハード制限値の両方を設定できます。ソフト制限値はハード制限値を上限としてユーザが増やせますが、ハード制限値を増やせるのは “root” パーミッションを持ったユーザだけです。ソフト制限値によって、Cluster Edition エンジンでオープン可能なファイル記述子の数が決まります。制限値は 10,000 です。

オープン可能なファイル記述子のほとんどはユーザ接続で使用できます。Cluster Edition エンジンによってファイルとデバイスのオープンに使用されるものはごく一部です。

ユーザ接続の詳細については、『システム管理ガイド』を参照してください。

HP-UX の場合

カーネル・パラメータの maxfiles と maxfiles_lim が、任意の 1 プロセスで使用可能なファイル記述子の数を制御します。HP-UX でのファイル記述子の最大数は、32 ビット・システムで 10,000、64 ビット・システムで 60,000 です。

現在のファイル記述子の値を表示するには、次のように Korn シェルまたは Bourne シェルの ulimit コマンドを使用します。

```
ulimit -n
```

現在のソフト制限値とハード制限値の表示

現在のソフト制限値を表示するには、C シェルの場合、次のように入力します。

```
limit descriptors
```

Bourne シェルの場合、次のように入力します。

```
ulimit -n
```

現在のハード制限値を表示するには、C シェルの場合次のように入力します。

```
limit -h descriptors
```

Bourne シェルの場合、次のように入力します。

```
ulimit -Hn
```

ソフト制限値を増やす方法

ソフト制限値を増やすには、C シェルの場合次のように入力します。

```
limit descriptors n
```

Bourne シェルの場合、次のように入力します。

```
ulimit -Sn new_value
```

ここで *n* はソフト制限値の現在値で、*new_value* は増加後のソフト制限値を示します。

注意　上記のコマンドを *RUN_server_name* ファイル内で使用して、ハード制限値とソフト制限値を増やすことができます。*RUN_server_name* ファイルは Bourne シェル・スクリプトなので、*RUN_server_name* ファイル内では必ず Bourne シェル用のコマンドを使用してください。

ハード制限値を増やす方法

ハード制限値を増やすには、「サンプル・プログラム」(278 ページ) の例で示すようなプログラムを使用します。

❖ サンプル・プログラムを設定してハード制限値を増やす

1　ASCII エディタを使用して、*file_name.c* (*file_name* にはファイルの名前を指定する) を作成します。「サンプル・プログラム」(278 ページ) の例に示すテキストを入力します。

2 次のように入力してファイルをコンパイルします。

```
cc file_name.c -o program_name
```

ここで *file_name* は作成したソース・ファイルの名前、*program_name* はプログラムに付ける名前です。

3 プログラムのパーミッションと所有権を変更して、“root” 権限で実行されるようにします。

```
chmod 755 program_name
chown root program_name
```

ここで *program_name* は、コンパイルしたプログラムの名前です。

4 オペレーティング・システムのプロンプトで次のコマンドを入力することにより、“root” ユーザはこのプログラムを使用してユーザ接続の数を増やして Cluster Edition を起動できます。

```
# program_name dataserver -d master_device_name
```

ここで *program_name* はコンパイルしたプログラムの名前、*master_device_name* は Cluster Edition のマスタ・デバイスへのフル・パスです。オペレーティング・システムのプロンプトでコマンドを入力する代わりに、Cluster Edition の *RUN_server_name* ファイル内で dataserver コマンドラインの先頭に *program_name* を挿入することもできます。

サンプル・プログラム

注意 これはサンプル・スクリプトです。必要に応じて変更してください。

次の例は、ハード制限値を増やす場合に使用できるソース・コードを示します。

```
#include <sys/time.h>
 #include <sys/resource.h>
 #include <sys/types.h>
 /*
 ** define MAX_CONNECTIONS to a number less than
 ** 10000. The number defined will then become the maximum
 ** number of connections allowed by an Adaptive Server.
 */
 #define MAX_CONNECTIONS 9999
 extern int errno;

 main(argc,argv)
 char **argv;
 {
     struct rlimit rlp;
     uid_t uid;
```

```
    rlp.rlim_cur = MAX_CONNECTIONS;
    rlp.rlim_max = MAX_CONNECTIONS;
  /* set the number of open file desriptors to
    MAX_CONNECTIONS */
    if (setrlimit (RLIMIT_NOFILE,&rlp) == -1)
    {
       perror("setrlimit");
       exit(1);
    }

   /* reset the user id to disable superuser
    privileges */
    uid = getuid();
    setuid(uid);
   /* run the program indicated as arguments to
    this program */
    execv(*++argv, argv);
}
```

ユーザ接続の詳細については、『システム管理ガイド』を参照してください。

クライアント接続のタイムアウト時間の調整

Cluster Edition では TCP/IP プロトコルの KEEPALIVE オプションを使用して、アクティブではなくなったクライアントを検出します。クライアントへの接続が所定の時間 (タイムアウト時間) 非アクティブであった場合、オペレーティング・システムは KEEPALIVE パケットを一定間隔で送信します。これらのパケットに対してクライアント・マシンから応答がない場合、オペレーティング・システムはクライアントが応答しなくなったことを Cluster Edition に通知します。次に、Cluster Edition はクライアントの接続を終了します。

KEEPALIVE のデフォルトのタイムアウト時間は、2 時間 (7,200,000 ミリ秒) です。現在のタイムアウト時間の値を表示するには、後の項で説明する、それぞれのプラットフォーム用のコマンドを使用します。

Sun Solaris の場合

タイムアウトの値を表示するには、次のコマンドを入力します。

```
/usr/sbin/ndd -get /dev/tcp tcp_keepalive_interval
```

タイムアウト時間を 15 分 (900,000 ミリ秒) に短縮するには、次のコマンドを入力します。

```
/usr/sbin/ndd -set /dev/tcp tcp_keepalive_interval 900000
```

Linux の場合

タイムアウトの値を表示するには、次のコマンドを入力します。

```
/sbin/sysctl -e net.ipv4.tcp_keepalive_time
```

タイムアウト時間を 15 分 (900 秒) に短縮するには、次のコマンドを入力します。

```
/sbin/sysctl -w net.ipv4tcp_keepalive_time=900
```

HP-UX の場合

現在のタイムアウト時間の値を表示するには、次のコマンドを入力します。

```
/usr/contrib/bin/nettune -l
```

tcp-keepstart パラメータは、接続がこれ以降は確立されないかどうかをシステムがチェックするまで、アイドル状態の接続をアクティブなまま保持する時間 (秒単位) を指定します。

タイムアウト時間を変更するには、nettune -s コマンドを使用します。

ハードウェア・エラーのチェック

データベースの破壊につながる可能性のある問題を示すハードウェア・エラー・メッセージには、次のような種類があります。

- ディスクの読み込みエラー、書き込みエラー、またはリトライ・エラー
- タイムアウト
- システム障害
- メモリに関するあらゆる種類の問題

Sun Solaris の場合

/var/adm/messages ファイルを定期的にチェックします。この項の最初で説明した種類のハードウェア・エラーが検出された場合は、Sun Microsystems の診断ツール sundiag を使用してメモリとディスクをチェックします。オペレーティング・システム用のマニュアルを参照してください。

Linux の場合

/var/log/messages ファイルを定期的にチェックします。オペレーティング・システム用のマニュアルを参照してください。

HP-UX の場合

/var/adm/syslog/syslog.log ファイルを定期的にチェックします。このファイルは直接表示できますが、HP-UX の dmesg コマンドを使用する方法もあります。オペレーティング・システム用のマニュアルを参照してください。

オペレーティング・システム・リソースの使用状況のモニタリング

『システム管理ガイド』では、負荷とシステムの設定に対応して最適な数のサーバ・エンジンを管理する方法が説明されています。最適な数を決めるには、システムと CPU の使用率をモニタします。

Sun Solaris と Linux では、パフォーマンスをモニタするために次のツールが用意されています。

- iostat コマンド － 端末とハード・ディスクの入出力の量、および CPU 時間の使用状況をレポートする。
- vmstat コマンド － 仮想メモリの使用状況をモニタする。
- netstat コマンド － ネットワーク・ステータスをモニタする。
- ps コマンド － 個々のプロセスの累積 CPU 時間と CPU 使用率の正確なスナップショットを表示する。このコマンドは、データサーバ、エンジン、プロセスの負荷を確認するときに役立つ。
- time コマンド － 実行が完了するまでに使用されたさまざまなユーザ・リソース、システム・リソース、リアルタイム・リソースを確認する場合に役立つ。

これらのツールの詳細については、オペレーティング・システムのマニュアルを参照してください。

C シェル管理スクリプトのサンプル

dbcc チェックを実行してデータベースのバックアップを行うことで、Cluster Edition データベースの整合性とリカバリ性を維持します。次の C シェル・スクリプトの例では、この作業を行うために複数の isql スクリプトが呼び出されます。

```
#!/bin/csh -f
 if ( -e dbcc_mail.out) then
   rm dbcc_mail.out
 endif
 foreach i (*.dbcc)
 isql -Usa -Ppassword < $i > dbcc_out
```

```
if ( 'grep -c 'Msg 25[0-9][0-9]' dbcc_out' ) then
   echo "There are errors in" $i >> dbcc_mail.out
   cat dbcc_out >> dbcc_mail.out
 else
   echo "Backing up " $i:r >> dbcc_mail.out
   isql -Usa -Ppassword < $i:r.backup
 endif
 end
 mail -s "Backup Report" jjones < dbcc_mail.out
```

スクリプトの最初のセット (各データベースに 1 つあり、ファイル名に *.dbcc* が付く) は、各データベースに対して dbcc checkalloc と dbcc checkdb を実行し、*dbcc_out* と呼ばれる出力ファイルにメッセージを送信します。

たとえば、スクリプト master.dbcc は、dbcc を実行して master データベースをチェックします。

```
dbcc checkalloc (master)
go
dbcc checkdb (master)
go
```

次に、C シェル・スクリプトは grep コマンドを実行して、dbcc 出力にある 2500 番台のエラー・メッセージを検索します。grep コマンドの結果は、*dbcc_mail.out* と呼ばれる出力ファイルに送られます。

次に、このスクリプトは、2500 番台のエラーが発生しなかった各データベースについて isql バックアップ・スクリプトを呼び出し、"Backing up *database_name*" という行を *dbcc_mail.out* に追加します。たとえば、スクリプト master.backup は master データベースをバックアップします。

```
use master
go
dump database master to master_dump
go
```

適切な dump transaction コマンドをスクリプトに追加できます。

2500 番台のエラー・メッセージがある場合、スクリプトはデータベースをバックアップしません。スクリプトの最後で、*dbcc_mail.out* がシステム管理者 "jjones" にメールで送信されます。これによって、システム管理者は重大な dbcc エラーと正常なバックアップの記録を得ることができます。

前述のサンプルのシェル・スクリプトと isql スクリプトは、インストール環境での必要性に合わせてカスタマイズできます。

このスクリプトを自動的に実行させるには、*crontab* ファイルを編集して、次のようなエントリを追加します。

```
00 02 * * * /usr/u/sybase/dbcc_ck 2>&1
```

この例では、C シェル・スクリプト dbcc_ck が、毎朝午前 2 時に実行されます。

第 17 章

Cluster Edition のローカライゼーションのカスタマイズ

この章では、言語、文字セット、ソート順の設定を含めた各言語に対応したインストールのための、Sybase のローカライゼーション・サポートについて説明します。詳細については、『システム管理ガイド』を参照してください。

ローカライゼーション・サポートの概要

ローカライゼーションとは、アプリケーションをある特定の言語または地域の稼働条件に適応させることです。これには、その国の言語に翻訳されたシステム・メッセージや、その国で使用している日付、時刻、通貨の正しいフォーマットの提供も含まれます。Cluster Edition では、世界各国の顧客や異機種間環境で使用する顧客向けに、ローカライゼーションをサポートしています。

サポートする内容は次のとおりです。

- データ処理のサポート － Cluster Edition には、さまざまな言語で使用する文字を処理するための、文字セットとソート順を定義したファイルが付属しています。

  Sybase は、次の地域の主要な言語をサポートします。

  - 西欧
  - 東欧
  - 中東
  - ラテン・アメリカ
  - アジア

- システム・メッセージの変換 − Cluster Edition には、次の言語のモジュールがあります。
  - ブラジル系ポルトガル語
  - 中国語 (簡体字)
  - フランス語
  - ドイツ語
  - 日本語
  - 韓国語
  - ポーランド語
  - スペイン語
  - タイ語

言語モジュール

Cluster Edition は、ローカライズされたソフトウェア・メッセージを別の言語モジュールに格納します。

言語モジュールをインストールすると、適切な場所に格納された新しい言語をサポートするメッセージ、文字セット、ソート順を定義したファイルを、インストール・プログラムがロードします。

Cluster Edition と Backup Server をインストールすると、デフォルトでは英語のシステム・メッセージがインストールされます。

サーバのデフォルトの文字セット

デフォルトの文字セットとは、データがコード化されていて、Cluster Edition データベースに格納されている文字セットです。

デフォルトの言語と文字セットの変更

警告！ 新しいサーバの文字セットやソート順の変更をすべて行ってから、ユーザ・データベースの作成または Sybase が提供するデータベースの変更を行ってください。データやデータ構造が Cluster Edition に追加されてから文字セットやソート順を変更すると、さらに処理が必要な場合があります。データを追加した後の文字セットやソート順の変更については、『システム管理ガイド』を参照してください。

sybcluster および Adaptive Server プラグインは、次のデフォルトを使用してインスタンスを作成します。

- us_english 言語
- iso_1 文字セット (HP-UX プラットフォームでは Roman8 を使用)
- バイナリ・ソート順

サーバのデフォルトの文字セットの変更

Cluster Edition のデフォルトとして、任意の文字セットを選択できます。これにはプラットフォームのデフォルトと異なる文字セットも含まれます。次のガイドラインを考慮して、新しいデフォルトの文字セットを選択します。

- 変換エラーやオーバヘッドを防止するため、クライアントが使用している文字セットに基づいてデフォルトの文字セットを決定する。

  たとえば、ほとんどのクライアントが ISO 8859-1 を使用している場合、ISO 8859-1 を指定するとデータ変換の量を最小化することができます。
- 使用しているサーバが異機種言語環境で稼働している場合は、必要とされるすべての文字セットで動作する文字セットを選択します。通常、これは Unicode (UTF-8) になります。

警告！ 新しいインスタンスのデフォルトの文字セットとソート順の変更をすべて行ってから、ユーザ・データベースの作成または Sybase が提供するデータベースの変更を行ってください。データやデータ構造体がインスタンスに追加されてから文字セットやソート順を変更すると、不正な動作が発生する場合があります。データを追加した後の文字セットやソート順の変更については、『システム管理ガイド』を参照してください。

サポートされている文字セット

Cluster Edition では、次の言語、スクリプト、文字セットがサポートされています。

- アラビア語 － 表 17-1 (286 ページ) を参照。
- バルト語 － 表 17-2 (286 ページ) を参照。
- 中国語 (簡体字) － 表 17-3 (286 ページ) を参照。
- 中国語 (繁体字) － 表 17-4 (287 ページ) を参照。
- キリル語 － 表 17-5 (287 ページ) を参照。
- 東欧言語 － 表 17-6 (287 ページ) を参照。
- ギリシャ語 － 表 17-7 (287 ページ) を参照。

- ヘブライ語 − 表 17-8 (288 ページ) を参照。
- 日本語 − 表 17-9 (288 ページ) を参照。
- 韓国語 − 表 17-10 (288 ページ) を参照。
- タイ語 − 表 17-11 (288 ページ) を参照。
- トルコ語 − 表 17-12 (288 ページ) を参照。
- Unicode (650 を超える言語をサポート) − 表 17-13 (289 ページ) を参照。
- ベトナム語 − 表 17-14 (289 ページ) を参照。
- 西欧言語 − 表 17-15 (289 ページ) を参照。

表には、各文字セットの定義が示されます。

「文字セット変換」(289 ページ) を参照してください。

表 17-1 に、アラビア語の文字セットを示します。

表 17-1: アラビア語の文字セット

文字セット	説明
cp864	PC アラビア語
cp1256	Microsoft Windows アラビア語
iso88596	ISO 8859-6 ラテン語／アラビア語

表 17-2 に、バルト語の文字セットを示します。

表 17-2: バルト語の文字セット

文字セット	説明
cp1257	Microsoft Windows バルト語

表 17-3 に、中国語 (簡体字) の文字セットを示します。

表 17-3: 中国語 (簡体字) の文字セット

文字セット	説明
eucgb	EUC GB コード化 = 中国語 (簡体字) の文字セット
cp936	Microsoft 中国語 (簡体字) の文字セット
gb18030	PRC 18030 標準

表 17-4 に、中国語 (繁体字) の文字セットを示します。

表 17-4: 中国語 (繁体字) の文字セット

文字セット	説明
cp950	PC (Microsoft) 中国語 (繁体字)
euccns	EUC CNS コード化 = 中国語 (繁体字) の文字セット
big5	Big 5 中国語 (繁体字)
big5hk	Big 5 HKSCS の拡張付き

表 17-5 に、キリル語の文字セットを示します。

表 17-5: キリル語の文字セット

文字セット	説明
cp855	IBM PC キリル語
cp866	PC ロシア語
cp1251	Microsoft Windows 3.1 キリル語
iso88595	ISO 8859-5 ラテン語／キリル語
koi8	KOI-8 キリル語
mac_cyr	Macintosh キリル語

表 17-6 に、東欧言語の文字セットを示します。

表 17-6: 東欧言語の文字セット

文字セット	説明
cp852	PC 東欧言語
cp1250	Microsoft Windows 3.1 東欧言語
iso88592	ISO 8859-2 Latin-2

表 17-7 に、ギリシャ語の文字セットを示します。

表 17-7: ギリシャ語の文字セット

文字セット	説明
cp869	IBM PC ギリシャ語
cp1253	MS Windows ギリシャ語
greek8	HP GREEK8
iso88597	ISO 8859-7 ラテン語／ギリシャ語
macgrk2	Macintosh ギリシャ語

表 17-8 に、ヘブライ語の文字セットを示します。

表 17-8: ヘブライ語の文字セット

文字セット	説明
cp1255	Microsoft Windows ヘブライ語
iso88598	ISO 8859-8 ヘブライ語

表 17-9 に、日本語の文字セットを示します。

表 17-9: 日本語文字セット

文字セット	説明
cp932	IBM J-DBCS:CP897 + CP301 (シフト JIS)
eucjis	EUC JIS コード化
sjis	シフト JIS (拡張なし)
deckanji	DEC Kanji

表 17-10 に、韓国語の文字セットを示します。

表 17-10: 韓国語の文字セット

文字セット	説明
eucksc	EUC KSC 韓国語コード化 = CP949
cp949	Ms Windows 韓国語

表 17-11 に、タイ語の文字セットを示します。

表 17-11: タイ語のクライアント文字セット

文字セット	説明
tis620	TIS-620 標準タイ語
cp874	Microsoft Windows タイ語

表 17-12 に、トルコ語の文字セットを示します。

表 17-12: トルコ語の文字セット

文字セット	説明
cp857	IBM PC トルコ語
cp1254	Microsoft Windows トルコ語
iso88599	ISO 8859-9 Latin-5 トルコ語
turkish8	HP TURKISH8
macturk	Macintosh トルコ語

表 17-13 に、Unicode の文字セットを示します。

表 17-13: Unicode の文字セット

文字セット	説明
utf8	Unicode UTF-8 コード化

表 17-14 に、ベトナム語の文字セットを示します。

表 17-14: ベトナム語の文字セット

文字セット	説明
cp1258	Microsoft Windows ベトナム語

表 17-15 に、西欧言語の文字セットを示します。

表 17-15: 西欧言語の文字セット

文字セット	説明
ascii8	US ASCII、8 ビット・データ、ISO 646
cp437	IBM CP437 － U.S. コード・セット
cp850	IBM CP850 － ヨーロッパ・コード・セット
cp860	PC ポルトガル語
cp858	cp850 ヨーロッパ言語のサポート付き
cp1252	Microsoft Windows US (ANSI)
iso_1	ISO 8859-1 Latin-1
roman8	HP ROMAN8
iso15	ISO 8859-15 Latin-1 ヨーロッパ言語のサポート付き
roman9	HP ROMAN8 ヨーロッパ言語のサポート付き
mac	Macintosh Roman
mac_euro	ヨーロッパ言語のサポート付き Macintosh Roman

文字セット変換

Backup Server は、クライアントの言語と Cluster Edition の文字セットを使用して、Cluster Edition にメッセージを渡します。Cluster Edition はこのメッセージを変換し、クライアントの言語と文字セットでメッセージを発行します。次の必要条件を考慮して、文字セットを選択します。

- 異機種間環境では、Cluster Edition と Backup Server は、稼働するプラットフォームや使用する文字セットが異なる複数のクライアントとの通信を必要とする場合があります。データの整合性を維持するため、サーバは異なる文字セット間でコード変換を行います。

- Unicode 変換は、すべてのネイティブな文字セットについて行われます。2 つのネイティブな文字セット間の Unicode 変換では、Unicode が中間文字セットとして使用されます。たとえば、サーバのデフォルトの文字セット (CP 437) とクライアントの文字セット (CP 860) との間で変換する場合、最初に CP 437 が Unicode に変換され、次に Unicode が CP 860 に変換されます。デフォルトでは、文字セット変換には Unicode 変換が使用されます。
- Adaptive Server 直接変換は、同じ言語グループの 2 つのネイティブな文字セット間での変換をサポートします。たとえば、CP 437 と CP 850 は両方ともグループ 1 の言語グループに属しているため、Adaptive Server はこれらの間での変換をサポートします。直接変換を使用するには、クライアントが使用しているすべての文字セットについて、文字セット定義ファイルをサーバにインストールする必要があります。直接変換を有効にするには、sp_configure 'enable unicode conversion', 0 を実行して、Unicode 変換を無効にする必要があります。

Cluster Edition または Backup Server のいずれかが、クライアントの言語と文字セットをサポートしていないと、サーバは警告メッセージを表示します。Backup Server の文字セットが Cluster Edition の文字セットと互換性がない場合にも、エラーが発生します。

実行可能な変換の詳細については、『システム管理ガイド』を参照してください。

クライアント／サーバ間の変換

Cluster Edition がクライアントの言語または文字セットをサポートしていない場合、クライアントはそのサーバと接続できますが、文字変換は実行されません。

ローカライズされたクライアント・アプリケーションを Cluster Edition に接続する場合、サーバは、そのクライアントの言語と文字セットをサポートしているかどうかをチェックします。

- Cluster Edition がその言語をサポートしていれば、すべての文字セット変換が自動的に実行され、クライアントの言語と文字セットを使用してメッセージを表示します。
- Cluster Edition がその言語をサポートしていなければ、ユーザのデフォルト言語または Cluster Edition のデフォルト言語を使用します。
- Cluster Edition がその文字セットをサポートしていなければ、クライアントに対して警告を表示し、変換機能をオフにして、言語をアメリカ英語に設定します。

ソート順

それぞれの文字セットには、1 つ以上のソート順 (照合順) が定義されています。ソート順は、ソート順定義ファイル (*.srt* ファイル) にあるか、Unicode ソート順の場合はシステムにインストールされています。ソート順定義ファイルは、文字セット定義ファイルに付属しており、同じディレクトリに格納されています。

サイトでの必要に応じて、データのソート順を選択できます。ただし、サーバは一度に 1 つのソート順だけをサポートすることを考慮して、すべてのクライアントで動作するソート順を選択します。

警告！ 新しいサーバのデフォルトの文字セットとソート順の変更をすべて行ってから、ユーザ・データベースの作成または Sybase が提供するデータベースの変更を行ってください。データやデータ構造体が Cluster Edition に追加されてから文字セットやソート順を変更すると、不正な動作が発生する場合があります。データを追加した後の文字セットやソート順の変更については、『システム管理ガイド』を参照してください。

利用できるソート順

Cluster Edition が文字データの整列、比較やインデックスの付与に使用する照合順は、ソート順によって決定されます。それぞれの文字セットには、1 つ以上のソート順が定義されています。

ソート順は、文字セット定義ファイルに付属しているソート順定義ファイル (*.srt* ファイル) の中にあります。

注意 利用できるソート順は、Cluster Edition にインストールされている文字セットによって異なります。

使用している言語の *.srt* ファイルを調べれば、その文字セットで利用できるソート順がわかります。ソート順は、次のロケーションに格納されています。

$SYBASE/charsets/<charset_name>/.srt*

利用可能な Unicode ソート順を表示するには、sp_helpsort を実行します。「ローカライゼーションのディレクトリ」(295 ページ) を参照してください。

表 17-16 に、利用できるソート順を示します。

表 17-16: Cluster Edition で利用できるソート順

ソート順の名前	説明
バイナリ順	文字セットの数値バイトの値に従って、すべてのデータをソートする。バイナリ順では、すべて ASCII の大文字をソートしてから小文字をソートする。アクセント付き文字または表意文字 (マルチバイト文字) は、それぞれの標準の順序 (任意の場合もあり) でソートされる。 すべての文字セットでは、デフォルトとしてバイナリ順を使用する。バイナリ順では要求に合わない場合は、インストール時、またはインストールした後に charset ユーティリティを使用して、他のソート順を指定できる。
辞書順 (大文字と小文字、およびアクセント記号を区別する)	大文字と小文字を区別する。アクセント付き文字も含めて、大文字をソートしてから小文字をソートする。各種のアクセント付き文字を認識し、対応するアクセントなしの文字の後にソートする。
辞書順 (大文字と小文字を区別しない、アクセント記号を区別する)	大文字と小文字を区別しない辞書のソート順。大文字と小文字は等しく処理され、ソート結果では両者が混在する。
辞書順 (大文字と小文字、およびアクセント記号を区別しない)	大文字と小文字を区別しない辞書のソート順。アクセント記号は無視される。
辞書順 (大文字と小文字を区別しない、優先度を付けた順位)	状況によって照合の優先度が異なる、大文字と小文字を区別しない辞書のソート順。大文字で書かれた語が小文字で書かれた同一語として扱われる。 大文字と小文字が区別されるのは、order by 句を使用する場合のみ。order by 句を使用すると、大文字がソートされてから小文字がソートされる。 **注意** インストール環境で、order by 句を使用したときに大文字をソートしてから小文字をソートし、それ以外は区別しないで処理する必要がある場合を除いては、このソート順を選択しないでください。このソート順を使用すると、order by 句で指定されたカラムがテーブルのクラスタード・インデックスのキーと一致した場合に、大きなテーブルでのパフォーマンスが低下する場合があります。
代替辞書順 (大文字と小文字を区別する)	大文字と小文字を区別する代替辞書ソート順。小文字をソートしてから大文字をソートする。 いくつかの西欧言語で使用する。
代替辞書順 (大文字と小文字を区別しない、アクセントを区別しない)	大文字と小文字、およびアクセントを区別しない代替辞書のソート順。 いくつかの西欧言語で使用する。
代替辞書順 (大文字と小文字を区別しない、大文字を優先する)	大文字と小文字を区別しないが、大文字が優先される代替辞書のソート順。 いくつかの西欧言語で使用する。
スペイン語辞書順 (大文字と小文字を区別する)	大文字と小文字を区別するスペイン語辞書のソート順。 スペイン語およびほとんどのラテン・アメリカの言語用ロケールで使用する。
スペイン語辞書順 (大文字と小文字を区別しない)	大文字と小文字を区別しないスペイン語辞書のソート順。 スペイン語およびほとんどのラテン・アメリカの言語用ロケールで使用する。
スペイン語辞書順 (大文字と小文字を区別しない、アクセントを区別しない)	大文字と小文字、およびアクセントを区別しないスペイン語辞書のソート順。 スペイン語およびほとんどのラテン・アメリカの言語用ロケールで使用する。

ソート順の名前	説明
スカンジナビア語辞書順 (大文字と小文字を区別する)	大文字と小文字を区別する辞書のソート順。 スカンジナビア語で使用する。
スカンジナビア語辞書順 (大文字と小文字を区別しない、大文字を優先する)	大文字と小文字、およびアクセントを区別しないが、大文字が優先度される辞書のソート順。 スカンジナビア語で使用する。

利用できるソート順を確認するには、charset ユーティリティを使用して、使用する文字セットのソート順を表示します。UTF-8 の Unicode ソート順の詳細については、『システム管理ガイド』の「文字セット、ソート順、言語の設定」を参照してください。

表 17-17 は、使用可能なデフォルトの Unicode ソート順を示します。

表 17-17: デフォルトの Unicode ソート順

名前	ID	説明
defaultml	20	デフォルト Unicode マルチ言語順
thaidict	21	タイ語辞書順
iso14651	22	ISO14651 標準順
utf8bin	24	UTF-8 バイナリと一致する UTF-16 順
binary	25	バイナリ・ソート
altnoacc	39	代替 (アクセント記号を区別しない)
altdict	45	代替 (辞書順)
altnocsp	46	代替 (大文字と小文字を区別しない、優先度を付けた順位)
scandict	47	スカンジナビア語 (辞書順)
scannocp	48	スカンジナビア語 (大文字と小文字を区別しない、優先度を付けた順位)
bin_utf8	50	UTF-8 バイナリ・ソート順
dict	51	汎用 (辞書順)
nocase	52	汎用 (大文字と小文字を区別しない、辞書順)
nocasep	53	汎用 (大文字と小文字を区別しない、優先度を付けた順位)
noaccent	54	汎用 (アクセント記号を区別しない、辞書順)
espdict	55	スペイン語 (辞書順)
espnocs	56	スペイン語 (大文字と小文字を区別しない、辞書順)
espnoac	57	スペイン語 (アクセント記号を区別しない、辞書順)
rusnocs	59	ロシア語 (大文字と小文字を区別しない、辞書順)
cyrnocs	64	キリル語 (大文字と小文字を区別しない、辞書順)
elldict	65	ギリシャ語 (辞書順)
hundict	69	ハンガリー語 (辞書順)
hunnoac	70	ハンガリー語 (アクセント記号を区別しない、辞書順)
hunnocs	71	ハンガリー語 (大文字と小文字を区別しない、辞書順)
turknoac	73	トルコ語 (アクセント記号を区別しない、辞書順)
cp932bin	129	CP932 バイナリ・ソート順

名前	ID	説明
dynix	130	GB ピンイン・ソート順
gb2312bn	137	GB2312 バイナリ・ソート順
cyrdict	140	キリル語辞書ソート順
turdict	155	トルコ語辞書ソート順
euckscbn	161	EUCKSC バイナリ・ソート順
gbpinyin	163	GB ピンイン・ソート順
rusdict	165	ロシア語辞書ソート順
sjisbin	179	SJIS バイナリ・ソート順
eucjisbn	192	EUCJIS バイナリ・ソート順
big5bin	194	BIG5 バイナリ・ソート順

言語モジュール

Cluster Edition のエラー・メッセージをアメリカ英語 (us_english) 以外の言語で表示する場合は、適切な言語モジュールをインストールしてください。

新しい言語モジュールをインストールすると、その新しい言語をサポートする言語を定義したファイルが、Sybase インストール・ディレクトリに自動的にロードされます。「ローカライゼーションのディレクトリ」(295 ページ) を参照してください。

新しい言語モジュールのインストール

サーバのフル・インストールでは、すべての言語コンポーネントが自動的にインストールされます。フル・インストールを選択しなかった場合は、手動で追加の言語モジュールをインストールする必要があります。

新しい言語モジュールをインストールするには、次の手順に従います。

1 配布メディアから言語モジュール・ソフトウェアをロードします。このソフトウェアは、Cluster Edition をロードしたディレクトリと同じディレクトリにロードしてください。

2 言語を再設定し、必要に応じて Cluster Edition の文字セットやソート順も再設定します。「ローカライゼーション設定の変更」(299 ページ) を参照してください。

メッセージ言語

メッセージ言語として、Cluster Edition にはデフォルトでアメリカ英語がインストールされています。言語モジュールには次の規則が適用されます。

- Cluster Edition のインストールまたは再設定中に、デフォルト言語としてアメリカ英語以外の言語を指定できますが、指定する言語の言語モジュールがインストールされていることが必要です。
- クライアントがアメリカ英語以外の Cluster Edition メッセージを必要とする場合は、その言語用の言語モジュールのロードが必要です。ロード完了後、クライアントが使用する言語用に Cluster Edition を設定できます。
- Cluster Edition がクライアントの言語をサポートしていない場合は、クライアントは Adaptive Server のデフォルト言語でメッセージを受け取ります。

  たとえば、クライアントの言語がラテン系言語の場合、スペイン語の言語モジュールをインストールして、Cluster Edition のデフォルト言語としてスペイン語を指定すると、クライアントはスペイン語でメッセージを受け取ります。

ローカライゼーション

デフォルトでは、Cluster Edition と Backup Server の設定には英語のロケール設定ローカライゼーションを使用します。この設定には次のファイルが含まれます。

- 西欧の文字セットを定義したファイル
- 西欧の文字セットで使用される、ソート順を定義したファイル
- アメリカ英語のシステム・メッセージ・ファイル

インストール・プロセス中、あるいは再設定中に、デフォルト以外の言語、文字セット、ソート順を指定できます。

ローカライゼーションのディレクトリ

Sybase のローカライゼーション設定には、次のディレクトリがあります。

- *locales*
- *charsets*

次の表に、ローカライゼーション・ファイルの構造を示します。表に示されているのは、ローカライゼーション・ファイルの一部です。

$SYBASE/charsets	*charset_name* *charset_name...* *Unicode*	**.srt* ファイル *charset.loc* **.uct* ファイル
$SYBASE/$SYBASE_ASE/locales	*language_name* *language_name...*	*charset_name* *charset_name...*
%SYBASE/locales	*language.dat* *message*	*language_name* *language_name...*

ディレクトリについて

$SYBASE/locales および *$SYBASE/SYBASE_ASE/locales* ディレクトリには、利用できる各言語に対応するサブディレクトリがあります。各言語のサブディレクトリには、その言語で利用できるそれぞれの文字セットのサブディレクトリがあります。

- これらのサブディレクトリにある *.loc* ファイルを使用して、Cluster Edition と Backup Server は、指定された文字セットでコード化された指定の言語でエラーをレポートします。

  各サブディレクトリには、各種の *.loc* ファイルが用意されています。これらのファイルの大部分には、指定の製品やユーティリティで使用する、変換済みのエラー・メッセージが格納されています。

- 各サブディレクトリ内の *common.loc* ファイルには、現地の日付や時刻、通貨のフォーマットなど、ローカライズされた情報が含まれており、すべての製品で使用されます。

- *locales.dat* ファイルには、プラットフォーム固有のロケール名を Sybase の言語と文字セットの組み合わせに関連付けるエントリが含まれます。

charsets ディレクトリについて

$SYBASE/charsets/charset_name 内の各ファイルには、文字セットの定義と、その文字セットで利用できるすべてのソート順の定義など、特定の文字セットに関連する情報があります。

locales.dat ファイルについて

locales.dat ファイルは、次のように編集できます。

- プラットフォームのデフォルトの言語または文字セットを変更する
- プラットフォームのロケール名と Sybase の言語や文字セット名の間の新しい関連付けを行う

locales.dat ファイルのエントリで使用するフォーマット

locales.dat ファイル内の各エントリにより、プラットフォーム固有のロケール定義が Sybase の言語と文字セットの組み合わせにリンクされます。各エントリのフォーマットは次のとおりです。

```
locale = platform_locale, syb_language, syb_charset
```

各パラメータの意味は、次のとおりです。

- *platform_locale* は、ロケールのプラットフォーム固有のキーワード。受け入れられる値の詳細については、オペレーティング・システムのマニュアルを参照してください。

  ロケールがそのサイトのデフォルトとして定義されている場合、*platform_locale* は “default” になります。

- *syb_language* は、*$SYBASE/locales/language_name* 内で使用される言語ディレクトリの名前。
- *syb_charset* は、文字セットの変換方法を決定して、*$SYBASE/locales/language_name/charset_name* 内でクライアントのメッセージ・ファイルのディレクトリの場所を識別する文字セットの名前。

たとえば、次のエントリでは、デフォルトのロケールで言語に us_english を使用し、文字セットに iso_1 を使用するように指定します。

```
locale = default, us_english, iso_1
```

クライアント・アプリケーションの *locales.dat* ファイルの使用方法

クライアント・アプリケーションは、*locales.dat* ファイルを使用して、使用する言語と文字セットを識別します。接続のプロセスは次の手順に従います。

1　クライアント・アプリケーションは、起動すると、オペレーティング・システムのロケール設定をチェックしてから *locales.dat* ファイルをチェックし、その設定が Cluster Edition に適切なものであるかどうかを確認します。フランス語のロケール・エントリの例を次に示します。

```
locale = fr_FR, french, iso_1
```

2　クライアントが Cluster Edition と接続すると、言語と文字セットについての情報がログイン・レコードにある Cluster Edition に渡されます。

3 Cluster Edition は、次を使用します。

- iso_1 などの文字セットについての情報。この情報をもとに、クライアントの文字セットを識別し、文字データをこの文字セットに変換できるかどうかを確認します。
- 言語 (前述の例ではフランス語) と文字セットについての情報。この情報をもとに、クライアントの言語を使用したメッセージがあるかどうかを確認します。

注意 Cluster Edition ソフトウェアが使用するいくつかのロケール・エントリは、すでに *locales.dat* ファイルで定義されています。定義されたエントリが要求に合わない場合は、修正したり、新しいロケール・エントリを追加したりできます。

---

locales.dat ファイルの編集

編集を開始する前に元のファイルをコピーし、編集後のファイルで問題が発生する場合に備えます。

locales.dat ファイルの編集は、次の手順に従います。

1 *locales.dat* ファイルのコピーを、テキスト・エディタで開きます。

2 次のような角カッコで囲まれたセクションを探します。

- Sun Solaris では、*[sun_svr4]* です。
- HP では、*[hp ux]* です。
- IBM では、*[aix]* です。

3 使用する言語 (*syb_language*) と文字セット (*syb_charset*) の組み合わせを指定したエントリが、このセクションにあるかどうかを確認します。

注意 *platform_locale* の値を、オペレーティング・システムで必要な値と一致させてください。システムの設定ファイルで行われているロケール定義が Sybase のロケール定義と一致しないと、アプリケーションは正しく実行されません。

たとえば、Open Client のメッセージをフランス語で表示する場合、Cluster Edition が文字セット ROMAN8 を使用しているときは、使用しているプラットフォームの *locales.dat* エントリを調べて次のエントリを探します。

```
locale = fr_FR, french, roman8
```

4 必要なエントリを追加するか、既存のエントリを修正します。

5 変更があった場合はその内容を保存し、テキスト・エディタを終了します。

ローカライゼーション設定の変更

デフォルトでは、Cluster Edition と Backup Server の設定には英語のロケール設定ローカライゼーションを使用します。この設定には次のファイルが含まれます。

- 西欧の文字セットを定義したファイル
- 西欧の文字セットで使用される、ソート順を定義したファイル
- us_english のシステム・メッセージ・ファイル

インストール・プロセス中と再設定中に、デフォルト以外の言語、文字セット、ソート順を指定できます。

Cluster Edition のローカライゼーション

各言語は、モジュールごとに約 2MB のデータベース領域を使用します。必要に応じて、alter database コマンドを使用して master データベースのサイズを増やし、それから他の言語を追加します。

注意　複数の言語を Cluster Edition にインストールしたいが、master データベースのサイズが複数言語の管理に十分でない場合は、トランザクション・ログが満杯になる可能性があります。master データベースは、マスタ・デバイス上でのみ拡張できます。詳細については、『システム管理ガイド』を参照してください。

1　Sybase 環境変数を設定していない場合は、source コマンドを使用して、*SYBASE.csh* または *SYBASE.sh* のいずれかを実行します。

2　langinstall ユーティリティを使用して、Cluster Edition のローカライゼーションを設定します。

```
$SYBASE/$SYBASE_ASE/bin/langinstall
```

以下は、langinstall の構文です。

```
langinstall [-S server_name] [-U user_name] [-P password]
[-R release_number] [-I path_to_interfaces] [-v] language character_set
```

たとえば、iso_1 デフォルト文字セットを使用してフランス語をインストールする場合は、次のようになります。

```
langinstall -Usa -P -Sserver_name french iso_1
```

Backup Server のローカライゼーション

Backup Server の言語と文字セットは、*RUN_<backup_server_name>* ファイルを修正することにより変更することができます。backupserver コマンドの引数の詳細については、『ユーティリティ・ガイド』を参照してください。

Cluster Edition へのその他の文字セットの設定

使用している言語の文字セットとソート順で Cluster Edition を設定するには、次の手順に従います。システム・メッセージは、デフォルト言語の英語で表示されます。

1 charset ユーティリティを使用して、デフォルトの文字セットとソート順をロードします。

charset を使用するには、あらかじめサーバを起動し、システム管理者権限を持っている必要があります。ソート順の *file name* を次のように使用します。

```
$SYBASE/$SYBASE_ASE/bin/charset -Usa -Ppassword
-Sserver_name sort_order_file character_set
```

sort_order_file をソート順のファイル名と置き換えます。表 17-18 (301 ページ) を参照してください。*character_set* を使用する文字セットの Sybase 名と置き換えます。表 17-19 (303 ページ) を参照してください。

2 charset ユーティリティを使用して、追加する文字セットをロードします。「charset ユーティリティ」(304 ページ) を参照してください。

Cluster Edition の組み込み文字セット変換を使用する場合は、クライアントのプラットフォームで使用しているすべての文字セットの定義ファイルをロードする必要があります。Unilib 文字セット変換を使用する場合は、ロードの必要はありません。

3 isql を使用してサーバに “sa” としてログインし、master データベースを選択します。

```
1> use master
2> go
```

4 ソート順の ID を使って、サーバに新しい文字セットとソート順を設定します。

```
1> sp_configure "default sortorder_id",
2> sort_order_id, "character_set"
3> go
```

sort_order_id を使用するソート順の ID と置き換えます。表 17-18 (301 ページ) を参照してください。*character_set* を使用する文字セットの Sybase 名と置き換えます。表 17-19 (303 ページ) を参照してください。

5　クラスタを停止します。sybcluster、Adaptive Server プラグインを使用できます。クラスタを手動で設定した場合は、コマンド・ライン・オプションを使用できます。

6　sybcluster または Adaptive Server プラグインを使用して、クラスタでインスタンスを再起動できます。

注意　クラスタを手動で設定した場合、UNIX システムで通常のプロセスを使用して、インスタンスを再起動します。これは通常、次の dataserver コマンドを呼び出すということです。

```
$SYBASE/$SYBASE_ASE/BIN/dataserver
 --quorum_dev=quorum_path --instance=instance_name
```

7　インスタンスを起動し、すべてのシステム・インデックスを再構築してから停止します。インスタンスを再起動して、安定した状態になるまで待ちます。

8　クラスタ・ログ・ファイルをチェックして、charset および sortorder の変更が正常に完了したかを確認します。

ソート順

表 17-18 に、利用できるソート順を示します。使用している言語が一覧になく、その言語固有のソート順がない場合は、バイナリ・ソート順を使用します。

表 17-18: 利用できるソート順

言語またはスクリプト	ソート順	ファイル名	ID
すべての言語	バイナリ順	*binary.srt*	50
中央ヨーロッパ・チェコ語、スロバキア語 これらのソート順は、 CP 852、CP 1250、および ISO 8859-2 のみで使用される。	辞書順 (大文字と小文字、およびアクセント記号を区別する)	czedit.srt	80
	辞書順 (大文字と小文字を区別しない、アクセント記号を区別する)	*czeocs.srt*	82
	辞書順 (大文字と小文字、およびアクセント記号を区別しない)	*czenoac.srt*	81
キリル語	辞書順 (大文字と小文字、およびアクセント記号を区別する)	*cyrdict.srt*	63
	辞書順 (大文字と小文字、およびアクセント記号を区別する)	*cyrnocs.srt*	64

言語またはスクリプト	ソート順	ファイル名	ID
英語 フランス語 ドイツ語 これらのソート順はすべての西欧の文字セットで使用される。	辞書順 (大文字と小文字、およびアクセント記号を区別する)	dictiona.srt	51
	辞書順 (大文字と小文字を区別しない、アクセント記号を区別する)	*nocase.srt*	52
	辞書順 (大文字と小文字を区別しない、アクセント記号を区別する、優先度を付けた順位)	*nocasepr.srt*	53
	辞書順 (大文字と小文字、およびアクセント記号を区別しない)	*noaccent.srt*	54
英語 フランス語 ドイツ語 これらのソート順は CP 850 のみで使用される。	代替辞書順 (大文字と小文字を区別する)	*altdict.srt*	45
	代替辞書順 (大文字と小文字を区別し、アクセント記号を区別しない)	*altnoacc.srt*	39
	代替辞書順 (大文字と小文字を区別する、優先度を付けた順位)	*altnocsp.srt*	46
ギリシャ語 このソート順は ISO 8859-7 のみで使用される。	辞書順 (大文字と小文字、およびアクセント記号を区別する)	*elldict.srt*	65
ハンガリー語 これらのソート順は ISO 8859-2 のみで使用される。	辞書順 (大文字と小文字、およびアクセント記号を区別する)	*hundict.srt*	69
	辞書順 (大文字と小文字を区別しない、アクセント記号を区別する)	*hunnoac.srt*	70
	辞書順 (大文字と小文字、およびアクセント記号を区別しない)	*hunnocs.srt*	71
ロシア語 このソート順は CP 855 を除いたすべてのキリル語の文字セットで使用される。	辞書順 (大文字と小文字、およびアクセント記号を区別する)	*rusdict.srt*	58
	辞書順 (大文字と小文字を区別しない、アクセント記号を区別する)	*rusnocs.srt*	59
スカンジナビア語 これらのソート順は CP 850 および CP858 のみで使用される。	辞書順 (大文字と小文字、およびアクセント記号を区別する)	*scandict.srt*	47
	辞書順 (大文字と小文字を区別しない、優先度を付けた順位)	*scannocp.srt*	48
スペイン語	辞書順 (大文字と小文字、およびアクセント記号を区別する)	*espdict.srt*	55
	辞書順 (大文字と小文字を区別しない、アクセント記号を区別する)	*espnocs.srt*	56
	辞書順 (大文字と小文字、およびアクセント記号を区別しない)	*espnoac.srt*	57
タイ語	辞書順	*dictionary.srt*	51
トルコ語 これらのソート順は ISO 8859-9 のみで使用される。	辞書順 (大文字と小文字、およびアクセント記号を区別する)	*turdict.srt*	72
	辞書順 (大文字と小文字、およびアクセント記号を区別しない)	*turnoac.srt*	73
	辞書順 (大文字と小文字を区別しない、アクセント記号を区別する)	*turnocs.srt*	74

文字セット

表 17-19 に、利用できる文字セットとその Sybase 名を示します。

表 17-19: 文字セットの Sybase 名

文字セット	Sybase 名
ASCII 8	acsii_8
Big 5	big5
Big 5HK	big5hk
CP 437	cp437
CP 850	cp850
CP 852	cp852
CP 855	cp855
CP 857	cp857
CP 858	cp858
CP 860	cp860
CP 864	cp864
CP 866	cp866
CP 869	cp869
CP 874	cp874
CP 932	cp932
CP 936	cp936
CP 949	cp 949
CP 950	cp950
CP 1250	cp1250
CP 1251	cp1251
CP 1252	cp1252
CP 1253	cp1253
CP 1254	cp1254
CP 1255	cp1255
CP 1256	cp1256
CP 1257	cp1257
CP 1258	cp1258
DEC Kanji	deckanji
EUC-CNS	euccns
EUC-GB	eucgb
EUC-JIS	eucjis
EUC-KSC	eucksc
GB 18030	gb18030
GREEK8	greek8
ISO 8859-1	iso_1
ISO 8859-2	iso88592

文字セット	Sybase 名
ISO 8859-5	iso88595
ISO 8859-6	iso88596
ISO 8859-7	iso88597
ISO 8859-8	iso88598
ISO 8859-9	iso88599
ISO 8859-15	iso15
Koi8	koi8
MAC	mac
MAC_CYR	mac_cyr
MAC_EE	mac_ee
MAC_EURO	mac_euro
MACGRK2	macgrk2
MACTURK	macturk
ROMAN8	roman8
ROMAN9	roman9
Shift-JIS	sjis
TIS 620	tis620
TURKISH8	turkish8
UTF-8	utf8

charset ユーティリティ

charset ユーティリティを使用して、Cluster Edition に文字セットとソート順をロードします。charset ユーティリティを使用してデフォルトの文字セットとソート順をロードできるのは、インストール時だけです。

Cluster Edition のデフォルトの文字セットとソート順を変更する場合は、『システム管理ガイド』を参照してください。

構文

```
charset
[ -U username ]
[ -P password ]
[ -S server ]
[ -I interfaces ]
[ -v version ]
sort_order
[ charset ]
```

表 17-20: charsets のキーワードとオプション

キーワードとオプション	説明
-U	“sa” としてオペレーティング・システムにログインしていない場合は、コマンド・ラインに “-Usa” または “/username = sa” を指定する必要があります。
-P	コマンド・ラインに “sa” パスワードを指定する。指定しない場合は、“sa” パスワードの入力が要求される。
-S	サーバの名前を指定する。指定しない場合は、charset が DSQUERY 環境変数を使ってサーバ名を特定する。DSQUERY 環境変数がない場合は、charset は “SYBASE” という名前のサーバへ接続しようとする。
-I	使用する interfaces ファイルを指定する。指定しない場合は、charset は SYBASE ディレクトリの interfaces ファイルを使用する。
-v	Sybase バージョン文字列を印刷してから、終了させる。他のオプションを同時に指定しない。
sort_order	charset を使ってデフォルトの文字セットとソート順をロードする場合、Cluster Edition が使用するソート順のファイル名を指定する *sort_order* パラメータは必須。追加の文字セットをロードする場合は、*charset.loc* を使用して文字セット・ファイルの名前を特定する。
charset	Cluster Edition が使用する文字セットのディレクトリを指定する。

第 18 章

Cluster Edition へのオプション機能の追加

この章では、Cluster Edition にオプションの機能を追加する方法について説明します。

使用しているシステムに Sybase 製品をインストールしたら、設定や管理について、その製品のマニュアルを参照してください。

監査の追加

監査は、データベース管理システムのセキュリティの重要な機能です。セキュリティ関連のシステム・アクティビティは監査証跡に記録されます。監査証跡はシステムへの侵入やリソースの誤用を発見するのに使用します。システム・セキュリティ担当者は、監査証跡を詳細に調べることで、データベース内のオブジェクトへのアクセスのパターンを調べ、特定ユーザの作業を監視できます。監査レコードはユーザごとに追跡できるため、監査システムはユーザによるシステムの誤用に対する抑止になります。

システム・セキュリティ担当者は、監査システムを管理し、監査の開始と停止、監査オプションの設定、監査データの処理を実行できる唯一のユーザです。

監査システムのデバイスとデータベース

監査システムはいくつかのコンポーネントで構成されています。主要なコンポーネントは次のとおりです。

- sybsecurity デバイスと sybsecurity データベース。監査情報を保存する。
- 監査証跡。設定時間に決定する監査デバイスとテーブルで構成される。
- syslogs トランザクション・ログ・デバイス。トランザクション・ログを格納します。

sybsecurity デバイスとデータベース

sybsecurity デバイスは sybsecurity データベースを保存します。sybsecurity データベースは、監査設定プロセスの一部として作成されます。sybsecurity データベースは、model データベース内のすべてのシステム・テーブルと、サーバワイドな監査オプションを追跡するためのシステム・テーブルおよび監査証跡用のシステム・テーブルを格納します。

監査証跡のためのテーブルとデバイス

Cluster Edition は、sysaudits_01 から sysaudits_08 までのシステム・テーブルに監査証跡を保存します。「現在の監査テーブル」は常に 1 つしかありません。Cluster Edition は、現在の監査テーブルにすべての監査データを書き込みます。システム・セキュリティ担当者は sp_configure を使用して、どの監査テーブルを現在のものにするかを設定したり、変更したりできます。

Cluster Edition を監査用に設定する場合、使用しているインストール環境に合わせて監査テーブルの数を決定します。指定できるシステム・テーブルは最高 8 つです (sysaudits_01 から sysaudits_08 まで)。監査証跡には最低 2 つまたは 3 つのシステム・テーブルを使用し、各システム・テーブルはマスタ・デバイスとは独立した独自のデバイスに保存します。こうしておけばスレッショルド・プロシージャを使用して現在の監査テーブルを自動的に保管し、いっぱいになったら新しい空のテーブルに切り替えて次の監査レコードを保管できます。

syslogs システム・テーブルのデバイス

監査用に設定するとき、トランザクション・ログを含む syslogs システム・テーブル用に別のデバイスを指定します。すべてのデータベースにある syslogs テーブルには、そのデータベース内で実行されたトランザクションのログが記録されています。

Cluster Edition を使用した auditinit の実行

ノンクラスタ・バージョンの Adaptive Server と共に auditinit を使用する場合、サーバがまだ起動していなければ auditinit はサーバを起動します。ただし Cluster Edition では、auditinit ユーティリティを使用する前にインスタンスが実行中であることが必要です。auditinit は、インスタンスがまだ開始されていない場合にそれを開始しません。開始されていない auditinit を使用して Cluster Edition にログインしようとすると、auditinit では次の警告メッセージが表示されます。

```
Can not login to server because it is not running.
Please manually start the server and retry
```

監査デバイスのインストール前の作業

sybsecurity、syslogs、sysaudits の各テーブル・デバイス用に用意するロー・デバイスのロケーションを決定します。この情報はあとで必要になります。

Sybase では次のことをおすすめします。

- システムを必要最低限の数の監査デバイスで設定する。これには最低 3 つのデバイスが必要です。sp_addaudittable を使用すれば、あとでデバイスを追加できます。詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。
- 監査テーブルとデバイスを 1 対 1 の割合でインストールする。同一デバイスを共有するテーブルは、スレッショルド上限値も共有することになります。これらのテーブルは、同じデバイス上にあるので、そのデバイスがいっぱいになってしまうと、続けて使用することができません。
- デバイスごとに 1 つずつ監査テーブルをインストールする。これによって、監査レコードを失うことなくスムーズに実行する監査システムを設定できます。2 つの監査テーブルを使用すると、1 つがいっぱいになったら、もう 1 つの方に切り替えることができます。さらに、3 つの監査テーブルを使用すると、たとえ 1 つのデバイスが壊れても、システム・セキュリティ担当者が新しいスレッショルド・プロシージャをインストールして、直るまで壊れたデバイスを省略するようにデバイスのローテーションを変更できます。
- デバイスをテーブルより大きくする。監査テーブルとデバイスを 3 つしか使用しない場合、テーブルとデバイスのサイズは同じでかまいません。監査テーブルとデバイスを追加して (最高 8 つまで) 監査能力を強化できるからです。テーブルやデバイスが上限値 (6 から 8) 近くで作業する場合は、デバイスをテーブルより十分に大きくします。このようにすると、監査機能を強化したいが、追加できるデバイスがほとんどないかゼロの場合、後でテーブルのサイズをデバイスのサイズに合わせて増やすことができます。

監査のインストール

❖ 監査のための Cluster Edition の設定

1. Sybase 環境変数を設定していない場合は、source コマンドを使用して、*SYBASE.csh* または *SYBASE.sh* のいずれかを実行します。
2. UNIX プロンプト (*$SYBASE/$SYBASE_ASE/install* にある auditinit ユーティリティ) で auditinit を開始します。

   ```
   $SYBASE/$SYBASE_ASE/install/auditinit
   ```

   auditinit によって、次のメニューが表示されます。

   ```
   AUDITINIT
   1. Release directory:  /usr/u/sybase
   2. Configure a Server product
   ```

3 [Configure a Server Product] を選択します。

4 Adaptive Server を選択します。

5 [Configure an Exsisting Sybase Server] を選択します。

6 設定するサーバを選択します。

7 選択したサーバに SA パスワードを入力します。

8 [Sybase Server Configuration] 画面で [Configure Auditing] を選択します。

auditinit のメニューに従って作業を進めるときに表示されるデフォルト値は変更できます。各メニューを終了するときは、[Ctrl + A] を押してデフォルト値や変更した値を受け入れて、次のメニューに移ります。

```
CONFIGURE AUDITING
1. Configure auditing:  no
2. Add a device for audit table(s)
3. Add a device for the audit database transaction log
4. Delete a device entry
5. Change a device entry

List of devices for the audit tables:
Logical name    Physical name    Segment name    Table name    Size

Device for the audit datbase transaction log:
Logical name    Physical name    Segment name    Table name    Size
```

9 [Configure Auditing] 画面で [Configure Auditing:] を選択します。

auditint によって [Configure Auditing] メニューが再表示され、[Configure Auditing:] に “yes” と表示されます。

10 Cluster Edition を再起動して、変更を有効にします。

❖ 監査テーブルのデバイスの作成

1 [Configure Auditing] 画面で [Add a Device for Audit Table(s)] を選択します。

auditinit によって、次のメニューが表示されます。

```
ADD/CHANGE A NEW DEVICE FOR AUDITING
1. sybsecurity physical device name:
2. Logical name of the device:
3. Size of the device (Meg):
4. Device size for auditing:
```

2 [Sybsecurity Physical Device Name:] を選択します。

監査テーブルのデバイスを作成するには、次の手順に従います。

1　「監査デバイスのインストール前の作業」(309 ページ) でロケーションを決定した物理デバイス (ロー・パーティション) の「フル・パス」を入力します。*path_to_partition* は、デバイスのロー・パーティションへのパスです。

```
Enter the physical name of the device to use for the audit
database (default is " "):

/dev/path_to_partition
```

オペレーティング・システム・ファイルを指定すると、次の警告が表示されます。

```
WARNING: '/secret1/sybase_dr/install/aud1.dat' is a
regular file which is not recommended for a Server device.
```

2　[Return] キーを押して警告を確認します。

auditinit によって [Add/Change a New Device for Auditing] メニューが再表示され、デバイスの物理名が表示されます。

```
ADD/CHANGE A NEW DEVICE FOR AUDITING
1. sybsecurity physical device
name:  /secret1/sybase_dr/install/aud1.dat
2. Logical name of the device:
3. Size of the device:
4. Device size for auditing:
```

3　このメニューの残りの項目に進みます。

注意　[Size of the Device:] の値は、[Device Size for Auditing:] の値以上になるようにしてください。[Device Size for Auditing:] は、デバイスのサイズと同じにします。Sybase の監査ガイドラインに従う場合は、[Device Size for Auditing:] に表示された値を変更する必要はありません。

4　[Ctrl + A] を押して設定を受け入れます。auditinit は [Configure Auditing] メニューに戻り、作成したデバイスを表示します。

```
CONFIGURE AUDITING
 1.  Configure auditing:  yes
 2.  Add a device for audit table(s)
 3.  Add a device for the audit database transaction log
 4.  Delete a device entry
 5.  Change a device entry

List of devices for the audit tables:
Logical name    Physical name     Segment name          Table name     Size

6.Audit_01' secret1/sybase_dr/install/aud1.datí sysaudits_01 5
```

5 複数の監査デバイスを追加するには、手順 1 ～ 6 を繰り返します。

追加できるデバイスは最高 8 つまでです。Sybase では 3 つ以上の監査テーブルを追加することをおすすめします。

デバイスを追加すると、auditinit は [Configure Auditing] メニューに戻って、作成したデバイスをすべて表示します。

```
CONFIGURE AUDITING
1. Configure auditing:  yes
2. Add a device for audit table(s)
3. Add a device for the audit database transaction log
4. Delete a device entry
5. Change a device entry

List of devices for the audit tables:
Logical name    Physical name    Segment name         Table name     Size

6. Audit_01'   /secret1/sybase_dr/install/aud1.datí sysaudits_01   5
7. Audit_02'   /secret1/sybase_dr/install/aud2.dat' sysaudits_02   5
```

❖ 監査データベース・トランザクション・ログのデバイスの作成

1 [Configure Auditing] - [Add a Device for the Audit Database Transaction Log] を選択します。

auditinit は [Add/Change a New Device for Auditing] メニューを表示します。

```
ADD/CHANGE A NEW DEVICE FOR AUDITING
1.  sybsecurity physical device name:
2.  Logical name of the device:
3.  Size of the new device (Meg):
4.  Device size for auditing:
```

2 [Sybsecurity Physical Device Name:] を選択します。

auditinit は物理名を入力するようプロンプトを表示し、デフォルトがある場合はそれを表示します。

```
Enter the physical name of the device to use for the
sybsecurity database (default is''):
 /dev/path_to_partition
```

path_to_partition は、デバイスのロー・パーティションへのパスです。

3 物理デバイスのフル・パス名を入力します。

オペレーティング・システム・ファイル名を指定すると、次の警告が表示されます。

```
WARNING: '/secret1/sybase_dr/install/audlog' is a regular
file, which is not recommended for a Server device.
```

4　[Return] キーを押して警告を確認します。

auditinit によって [Add/Change a New Device for Auditing] メニューが再表示され、デバイスの物理名に選択した値が表示されます。

```
ADD/CHANGE A NEW DEVICE FOR AUDITING
1.sybsecurity physical device name:
      /secret1/sybase_dr/install/auditlog.dat
2.Logical name of the device:
3.Size of the device:
4.Device size for auditing:
```

5　このメニューの残りの項目に進みます。次のことに注意してください。

- Sybase では、トランザクション・ログのサイズを最低 2MB にすることをおすすめします。
- auditinit は、[Add/Change a New Device for Auditing] メニューの [Size of the Device:] と [Device Size for Auditing:] の両方にサイズを表示します。
- [Device Size for Auditing:] のデフォルト値は、監査タスクのログにデバイス全体を使用するという前提に基づいて、デバイス・サイズと等しくなっています。デバイスのサブセットだけを使用したい場合は、[Size of the Device:] の値を編集できます。

6　[Ctrl+A] を押して、[Add/Change a New Device for Auditing] で表示された設定を受け入れます。

auditinit は [Configure Auditing] メニューに戻り、作成したデバイスを表示します。

```
CONFIGURE AUDITING
1. Configure auditing:  yes
2. Add a device for audit table(s)
3. Add a device for the audit database transaction log
4. Delete a device entry
5. Change a device entry

List of devices for the audit tables:
Logical name    Physical name    Segment name         Table
name     Size

6. Audit_01'   /secret1/sybase_  dr/install/aud1.datí sysaudits_01   5
7. Audit_02'   /secret1/sybase_  dr/install/aud2.dat' sysaudits_02   5
8. auditlog    /secret1/.../auditlog.dat logsegment   syslogs        2
```

7　監査設定を実行する準備ができたら、[Ctrl + A] を押します。auditinit は [Sybase Server Configuration] 画面に戻ります。

8　[Ctrl + A] を再度押します。auditinit プロンプトが表示されます。

```
Execute the Sybase Server Configuration now?
```

9 “y” (yes) を入力します。

auditinit は監査をインストールするタスクを開始します。インストールが正常に完了すると、次のメッセージが表示されます。

```
Running task: install auditing capabilities.
.....................Done
Auditing capability installed.
Task succeeded: install auditing capabilities.
Configuration completed successfully.
Press <return> to continue.
```

監査の有効化

監査をインストールしても、システム・セキュリティ担当者が sq_configure を使用して有効にしないと監査は開始されません。『システム管理ガイド 第 1 巻』を参照してください。

❖ デバイス・エントリの削除

1 [Configure Auditing] - [Delete a Device Entry] を選択します。

2 削除するデバイスの番号を入力します。

3 [Return] キーを押します。

❖ デバイス・エントリを変更する

1 [Configure Auditing] - [Change a Device Entry] を選択します。

2 変更するデバイスの番号を入力します。

auditinit は、[Add/Change a New Device for Auditing] メニューに、選択したデバイスの情報を表示します。

```
ADD/CHANGE A NEW DEVICE FOR AUDITING
1. sybsecurity physical device name:
      /secret1/sybase_dr/install/audlog
2. Logical name of the device: aud.log
3. size of the new device (Meg): 5
4. Device size for auditing:5
```

3 変更する残りのエントリをそれぞれ選択します。

4 [Ctrl + A] を押して新しいエントリを保存します。

Transact-SQL 構文のオンライン・ヘルプのインストール

この項では、Transact-SQL 構文のオンライン・ヘルプをインストールする方法を説明します。

オンライン構文ヘルプ：*sp_syntax*

$SYBASE/$SYBASE_ASE/scripts ディレクトリには、構文ヘルプ・データベース sybsyntax をインストールするためのスクリプトが格納されています。このデータは sp_syntax を使用して検索できます。sp_syntax の詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。

scripts ディレクトリには、どの Sybase 製品がサーバに含まれているかによって、表 18-1 に示す sp_syntax スクリプトが 1 つ以上入っています。

表 18-1: sp_syntax インストール・スクリプト

スクリプト	製品
ins_syn_cl	Open Client Client-Library™
ins_syn_esql	Embedded SQL™
ins_syn_os	Open Server
ins_syn_sql	Transact-SQL

ins_syn_sql スクリプトは、すべての Cluster Edition インストール環境に含まれます。このスクリプトには Transact-SQL、システム・プロシージャ、Sybase ユーティリティの構文情報が含まれています。このスクリプトを実行すると、sybsyntax データベースの SQL に関する部分がインストールされます。

これらのスクリプトは、使用しているサーバでの Sybase 情報の必要度に応じてインストールできます。最初に実行するスクリプトでは、sybsyntax データベースと必要なテーブルおよびインデックスを作成します。それ以降に実行するスクリプトはすべて、データベースにある既存の情報に追加されます。以前に実行されたスクリプトを実行すると、以前にインストールした情報ローがデータベース内のテーブルから削除されて再インストールされます。

警告！ *ins_syn_cl* スクリプトと *ins_syn_os* スクリプトは矛盾します。両方のスクリプトを実行すると、エラーが発生します。

sybsyntax データベースのデフォルト・デバイス

sybsyntax データベースは、データベース・デバイス上に 3MB の領域を必要とします。デフォルトでは、デフォルトのデータベース・デバイスとして指定されたデバイス上に、sybsyntax インストール・スクリプトが sybsyntax データベースをインストールします。

デフォルト・ディスクとしてインストールされているマスタ・デバイスのステータスを変更したり、別のデフォルト・デバイスを指定したりするときに、sp_diskdefault を使用していない場合は、スクリプトによってマスタ・デバイス上に sybsyntax がインストールされます。この設定はおすすめできません。本来は master データベースの今後の拡張に使用できるよう残して置くべき貴重な領域を sybsyntax が使用してしまうためです。

マスタ・デバイスに sybsyntax をインストールしないようにするには、次のいずれかの方法に従います。

- sp_diskdefault を使用して、デフォルトのデバイスをマスタ・デバイス以外に指定します。sp_diskdefault の詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。
- 実行する sybsyntax インストール・スクリプトを修正して、異なるデバイスを指定します。次の項を参照してください。

sybsyntax のインストール

実行する sybsyntax インストール・スクリプトごとに、次の手順を実行します。

1 sybsyntax データベースを格納するデバイスのタイプ (ロー・パーティション、論理ボリューム、オペレーティング・システム・ファイルなど) とロケーションを決定します。この情報はあとで必要になります。

2 オリジナル・スクリプトのコピーを作成します。編集したスクリプトに問題が発生したときは、このコピーを使用します。

3 必要に応じて、テキスト・エディタを使用してスクリプトを編集し、デフォルト・デバイスを、マスタ・デバイスから手順 1 で作成したデバイスに変更します。「sybsyntax データベースのデフォルト・デバイス」(316 ページ) を参照してください。

   - デフォルト・デバイスを指定する次のセクションをコメントにします。

```
/* create the database, if it does not exist */
 if not exists (select name from sysdatabases
 where name = "sybsyntax")
 begin
     /* create the sybsyntax table if it doesn't exist */
     /* is the space left on the default database
     devices > size of model? */
     if (select sum (high-low +1) from sysdevices where status
     & 1 = 1) - (select sum(size) from sysusages, sysdevices
```

```
            where vstart >= sysdevices.low
            and vstart <= sysdevices.high
            and sysdevices.status &1 = 1) >
            (select sum(sysusages.size) from sysusages
            where dbid = 3)
        begin
            create database sybsyntax
        end
        else
        begin
            print "There is not enough room on the default
            devices to create the sybsyntax database."
        return
        end
    end
```

- このセクション全体をコメントにしたあとで、次のような行をスクリプトに追加します。

```
create database sybsyntax on device_name
```

device_name は、sybsyntax をインストールするデバイス名です。

4　次のようなコマンドでスクリプトを実行します。

```
isql -Usa -Ppassword -Sservername < $SYBASE/$SYBASE_ASE/scripts/ins_syn_sql
```

上記で、*sa* はシステム管理者のユーザ ID、*password* はシステム管理者のパスワード、*servername* はデータベースをインストールする Cluster Edition 名です。

DSQUERY 環境変数を *servername* に設定している場合は、サーバ名を $DSQUERY に置き換えることができます。

5　インストールした sybsyntax データベースが正しく動作していることを確認するには、isql を使用して、データベースをインストールしたサーバにログインし、sp_syntax を実行します。次に例を示します。

```
isql -Usa -Ppassword -Sservername

1> sp_syntax "select"
2> go
```

Cluster Edition は、“select” という単語またはその単語の一部を含むコマンド・リストを表示します。

第 19 章

エラー・メッセージのロギングとイベントのロギング

この章では、Cluster Edition のエラー・ロギング機能の使用方法を説明します。

Cluster Edition のエラー・ロギング

Cluster Edition は起動するたびに、Adaptive Server エラー・ログと呼ばれるローカルのエラー・ログ・ファイルに情報を書き込みます。

$SYBASE/$SYBASE_ASE/install/instance_name.log

このファイルの役割と特性は、次のとおりです。

- 起動のたびごとに、起動の成功／失敗に関する情報を格納する。
- オペレーション中にサーバによって生成されたエラー・メッセージと情報メッセージを記録する。
- サーバ・プロセスを停止するまで開いたままになる。
- Cluster Edition からの起動メッセージを格納する。

Cluster Edition では、次のようなフォーマットで、エラー・ログ・ヘッダの先頭にインスタンス ID が記載されます。

```
instance id: engine number : family id: process id: date time
```

注意 エラー・ログのサイズを減らして使用可能なディスク領域を増やす場合は、インスタンスを停止してから、ロギングされたメッセージを削除してください。インスタンスを停止するまで、ログ・ファイルはメモリ領域を解放できません。

エラー・ロギングの有効化と無効化

Cluster Edition エラー・ログへの記録は常に有効になっています。ただし、特定のユーザ定義メッセージを作成したり変更したりする場合は、そのメッセージをログから省略するように設定できます。「ユーザ定義メッセージのロギング」(321 ページ) を参照してください。

エラー・ログのパスの設定

Cluster Edition を新たに設定するときに、インストール・プログラムによって、Sybase インストール・ディレクトリにエラー・ログのロケーションが設定されます。Backup Server には独自のエラー・ログがあります。

各サーバのエラー・ログのデフォルト・ロケーションは次のとおりです。

- Cluster Edition: *$SYBASE/$SYBASE_ASE/install/instance_name.log*
- Backup Server: *$SYBASE/$SYBASE_ASE/install/instance_name_back.log*

起動時に、Cluster Edition のエラー・ログ・ファイルの名前とロケーションをコマンド・ラインからリセットできます。-e 起動パラメータと dataserver コマンドの値を使用して、Cluster Edition を起動します。ただし、sybcluster を使用してクラスタを管理している場合は、sybcluster 'set instance logpath' パラメータを使用して各インスタンスのエラー・ログ・ファイルのロケーションを変更してください。

注意 複数のインスタンスが、同一のエラー・ログを共有することはできません。

Cluster Edition のエラー・ログのパス設定

注意 Cluster Edition インストーラは、*RUN_server* ファイルを作成しません。ただし、*RUN_server* ファイルを作成する場合、エラー・ログの新しいロケーションを追加できます。

エラー・ログへのパスを指定しないと、Cluster Edition は以下に従ってエラー・ログを追加します。

- クラスタ入力ファイル・エラー・ログのロケーション。この情報は、クォーラム・デバイスに格納され、dataserver コマンドによって使用されます。
- エラー・ログのデフォルトのロケーション。エラー・ログのパスを指定しないと、*instance_name.log* という名前のログ・ファイルがデータサーバ用の現在の作業ディレクトリ内に作成されます。

qrmutil --errorlog パラメータまたは sybcluster 'set instance errorlog' を使用すると、クォーラム・デバイスに格納されているログのロケーションを変更できます。

シェル・スクリプトを使用してクラスタ・インスタンスを起動する場合は、dataserver -e パラメータの値を変更します。これによって、エラー・ログのロケーションに関する他の設定が上書きされます。

メッセージの管理

イベント・ロギングが有効になっているときは、次の方法でその機能を管理できます。

- sp_addmessage または sp_altermessage を使用して、特定のユーザ定義メッセージが Cluster Edition エラー・ログにログインされているかどうか制御します。

  sp_addmessage と sp_altermessage の完全な構文については、『リファレンス・マニュアル：プロシージャ』を参照してください。

- 設定パラメータを使用して、監査イベントのログを取るかどうかを指定します。監査イベントは、ユーザが Cluster Edition にログインするときの、成功 log audit logon success、または失敗 log audit logon failure に関連しています。

ユーザ定義メッセージのロギング

Cluster Edition エラー・ログにユーザ定義メッセージのログを取るかどうかを指定できます。Cluster Edition では、次のメッセージに対してこの設定ができます。

- 新しいメッセージ (sp_addmessage)
- 既存のメッセージ (sp_altermessage)

これらのコマンドとそのパラメータの詳細については、『リファレンス・マニュアル：プロシージャ』の「sp_addmessage」と「sp_altermessage」を参照してください。

新しいメッセージ

新しいユーザ定義メッセージを sysusermessages に追加するときは、sp_addmessage に with_log オプションを指定します。このパラメータは、メッセージが表示されるたびにそのメッセージのログを取るよう Cluster Edition を設定します。

既存のメッセージ

既存のユーザ定義メッセージを変更するには、sp_altermessage に with_log オプションを指定します。このパラメータで、次のようにそのメッセージの報告ステータスを切り替えます。

- TRUE − ロギングを有効にする
- FALSE − ロギングを無効にする

監査イベントのロギング

デフォルトでは、Cluster Edition は監査イベントのログを取りません。ただし、sp_configure パラメータを使用して Cluster Edition のエラー・ログに、ログインなど監査イベントのログを取るように指定できます。

使用可能なパラメータと値は次のとおりです。

- log audit logon success を 1 に − Cluster Edition ログインのロギングを有効にする。

  ```
  sp_configure "log audit logon success", 1
  ```

- log audit logon failure を 1 に − Cluster Edition ログインの不成功のロギングを有効にする。

  ```
  sp_configure "log audit logon failure", 1
  ```

- どちらかのパラメータを 0 に − そのメッセージ・タイプのロギングを無効にする。

  ```
  sp_configure "log audit logon success", 0
  sp_configure "log audit logon failure", 0
  ```

sp_configure の詳細については、『システム管理ガイド 第 1 巻』を参照してください。

第 20 章

ネットワークを介する通信の設定

クラスタの各インスタンスはネットワークを介して、他の Adaptive Server、Open Server アプリケーション、クライアント・ソフトウェアと通信できます。リモート・プロシージャ・コールを介して、クライアントは、1 つまたは複数のサーバと、サーバは他のサーバと通信できます。

ディレクトリ・サービスには、サーバのネットワーク・ロケーションについての情報が入っています。ディレクトリ・サービスには、Adaptive Server、Backup Server、ネットワーク上にある他のサーバ製品のエントリがすべて入っています。

Sybase のクライアント／サーバ環境では、クライアントがネットワーク上のサーバのロケーションを知っていて、サーバがクライアントの言語または文字セットをサポートしている場合、クライアントはインスタンスに接続できます。クライアントが接続を開始する場合、クライアントはディレクトリ・サービスの中から接続先サーバのネットワーク・ロケーションを検索します。

ディレクトリ・サービスには、Backup Server と XP Server を含むすべてのサーバの名前とアドレスがリストされています。クライアント・プログラムを使用して特定のサーバと接続する場合、クライアント・プログラムはディレクトリ・サービスでサーバ名を検索し、そのサーバに接続します。

サーバには、ネットワーク情報も必要です。サーバは起動時に interfaces ファイルの内容を調べて、クライアントの接続要求の受信先を決定します。さらに、各インスタンスは、他のインスタンスにリモート・プロシージャ・コールを実行する場合、クライアントとしても機能できます。

表 20-1: interfaces ファイルのタスクとトピックの参照先

interfaces ファイルの種類	タスクまたはトピック	参照箇所
UNIX のサーバまたはクライアント	複数の Cluster Edition インストール環境用のエントリの追加	Adaptive Server の設定ガイド
	複数のインストール環境で使用するマスタ interfaces ファイルの作成	「マスタ interfaces ファイルの作成」(330 ページ)
	複数のネットワーク用の設定	「複数のネットワークで使用する interfaces ファイルの設定」(332 ページ)
	リファレンス情報	「interfaces ファイルのフォーマットについて」(328 ページ)
PC クライアント	クライアントの設定	使用しているプラットフォームの『ASE インストール・ガイド』
	上級タスクについてのリファレンス情報と指示	使用している PC クライアント・プラットフォームの『Open Client/Server プログラマーズ・ガイド補足』または該当する Open Client のマニュアル
リストされていないクライアント・プラットフォーム	上級タスクの設定、リファレンス情報、指示	使用している PC クライアント・プラットフォームの『Open Client/Server プログラマーズ・ガイド補足』または該当する Open Client のマニュアル

Cluster Edition で使用するディレクトリ・サービス・エントリの決定方法

各インスタンスは、ディレクトリ・サービスを使用してクライアントから受信するアドレスを決定します。インスタンスの起動時に、次の手順が実行されます。

1 コマンド・ラインで -s オプションに指定されたサーバ名を調べます。コマンド・ラインでサーバ名が指定されていない場合は、次の処理が行われます。

2 DSLISTEN 環境変数の値を確認して、自身の名前を決定します。DSLISTEN 環境変数が設定されていない場合、サーバ名は SYBASE と見なされます。

3 上記の手順で見つけた名前と一致するエントリ名をディレクトリ・サービス内で検索します。

4 検出したディレクトリ・サービス・エントリに指定されているネットワーク情報を、クライアント接続の受信に使用します。

クライアントのディレクトリ・サービスの使用方法

クライアントがサーバに接続するとき、次の処理を実行します。

- プログラムを通して、または DSQUERY 環境変数を参照して、サーバの名前を決定する。アプリケーション・ユーザが DSQUERY を設定していない場合、サーバ名のランタイム値はデフォルトの SYBASE になる。
- ディレクトリ・サービス内で、サーバの名前と一致するエントリ名を検索する。
- ディレクトリ・サービス・エントリで指定されているネットワーク情報を使用して、サーバに接続する。クライアントが 1 回で接続できない場合、ディレクトリ・サービスで示されている遅延間隔とリトライ回数に従って接続を試行し続ける。一致するエントリが見つからない場合、クライアントの標準エラー・ファイルにエラー・メッセージが書き込まれる。複数のネットワークをサポートしている場合、クライアントはそのサーバの 2 番目のネットワーク・アドレス・エントリの情報を使用して接続を試行する。

クライアント接続については、Open Client のマニュアルで詳細に説明しています。使用しているクライアント・プラットフォームの『Open/Client プログラマーズ・ガイド補足』または適切な Open/Client のマニュアルを参照してください。

ディレクトリ・サービス・エントリの作成

sybcluster ユーティリティは、クラスタの作成時にクラスタおよび各インスタンス用のディレクトリ・サービスのエントリを作成します。また、次の Sybase ユーティリティを使用してディレクトリ・サービスのネットワーク情報を編集することもできます。

- dsedit − GUI ユーティリティ
- dscp − UNIX コマンド・ライン・ユーティリティ

これらのユーティリティの使用方法の詳細については、『ユーティリティ・ガイド』を参照してください。

サポートされているディレクトリ・ドライバ

UNIX 用に次の 3 つのディレクトリ・ドライバがサポートされています。

- interfaces ドライバ
- ライトウェイト・ディレクトリ・サービス・ドライバ
- DCE (Distributed Computing Environment) で提供される CDS (Cell Directory Service)

この章の残りの部分では、*interfaces* ファイルについて説明し、サポートする UNIX プラットフォームごとに固有の設定情報を記述します。LDAP ドライバと Cell Directory Service の詳細、interfaces ファイルと LDAP ディレクトリ・サービスの比較については、使用しているプラットフォームの『Open Client/Server 設定ガイド』を参照してください。

interfaces ファイルの内容

interfaces ファイルには、ネットワーク上で動作しているすべてのサーバについてのネットワーク情報が入っています。これらのサーバには、インスタンス、Backup Server、XP Server に加えて Replication Server や Open Server などのサーバ・アプリケーションが含まれます。

interfaces ファイル内のネットワーク情報は、サーバ名、ホスト・マシンのネットワーク名またはネットワーク・アドレス、サーバがクエリを受信するポート、オブジェクトまたはソケット番号 (ネットワーク・プロトコルによって異なる) で構成されています。「interfaces ファイルのフォーマットについて」(328 ページ) を参照してください。

interfaces ファイルの各エントリには、次の 2 種類の行を指定できます。

- master 行。サーバ・アプリケーションによって、ネットワークを介してクエリを受信するときに使用されます。この情報をリスナ・サービスと呼びます。
- query 行。クライアント・アプリケーションによって、ネットワークを介してサーバに接続するときに使用されます。この情報をクエリ・サービスと呼びます。

サーバはクライアントが接続要求に使用するのと同一のポートから接続要求を受信するため、サーバの master 行と query 行に指定されているネットワーク情報は同一です。

サーバの *interfaces* ファイルには、master 行が必要です。サーバが他のサーバに対しクライアントとして機能する場合、これらのサーバには query 行が必要です。

クライアントの interfaces ファイルには master 行は必要ありません。クライアントの interfaces ファイルは query 行だけで正しく機能します。

サイトに複数のインストール環境がある場合

Adaptive Server または Cluster Edition のインストール環境が複数ある場合、ネットワーク上で動作しているすべてのサーバについての情報を、各サーバの interfaces ファイルに格納してください。

すべてのサーバ製品を同じプラットフォームで実行している場合、マスタ *interfaces* ファイルを 1 つ作成して、これを各マシンにコピーできます。「マスタ interfaces ファイルの作成」(330 ページ) を参照してください。

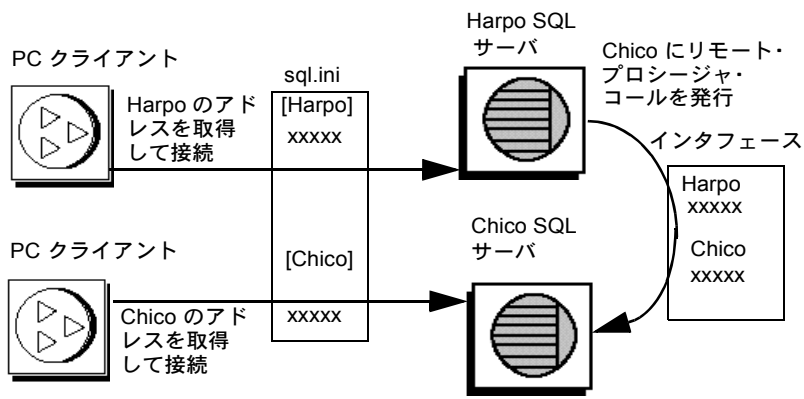
ホスト・マシンが複数のネットワークをサポートしている場合は、「複数のネットワークで使用する interfaces ファイルの設定」(332 ページ) を参照してください。

異機種間環境と同機種間環境

Cluster Edition とクライアントは、同一のプラットフォームまたは異なるプラットフォームで実行できます。

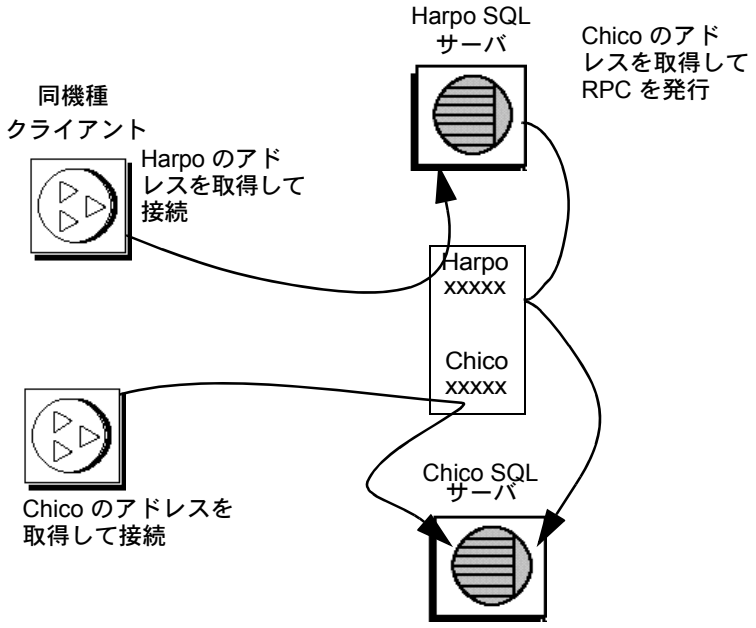
プラットフォームが異なる場合、各プラットフォームが *interfaces* ファイルに対して異なるフォーマットと異なる設定を要求する場合があります。図 20-1 は、クライアント PC が interfaces ファイル (*sql.ini*) 内のネットワーク情報を使用して UNIX 上で実行されているインスタンスに接続する方法と、インスタンスが *interfaces* ファイルを使用してリモート・プロシージャ・コールの実行中に他のサーバに接続する方法を示します。

図 20-1: 異機種間環境でのネットワーク接続の確立



クライアントとサーバの両方が UNIX 上で実行されている場合、同じ interfaces ファイルが両方に対して有効です。図 20-2 は、同機種間環境で実行されているクライアントとインスタンスが 1 つの interfaces ファイルのコピーを使用して接続を確立する方法を示します。2 つのインスタンスが同一のオペレーティング・システム上で実行されているため、同じ interfaces ファイルまたは同じファイルの完全に同一なコピーを使用できます。

図 20-2: 同機種間環境でのネットワーク接続の確立



interfaces ファイルのフォーマットについて

次の規則が、interfaces ファイルのエントリのフォーマットに適用されます。

- 各インスタンスに対してエントリは 1 つしかないが、エントリ内には複数の行がある場合がある。
- *servername* 行に続く各行は、スペースまたはタブ文字で開始する。
- 行の各要素は 1 つのスペースで区切る。
- 各エントリは空白行で区切る。
- 行頭にシャープ記号 (#) を追加し、行末に改行を追加して、interfaces ファイルにコメントを追加できる。

interfaces ファイルのエントリのフォーマットには TLI と TCP の 2 つがあります。

TLI スタイルのエントリを次に示します。

```
servername retry_attempts delay_interval<newline>
 <tab>service_type api protocol device address filter<newline>
 <tab>ha_failover servername<newline>
```

TCP スタイルのエントリを次に示します。

```
servername retry_attempts delay_interval<newline>
 <tab>service_type protocol network machine port filter<newline>
 <tab>ha_failover servername<newline>
```

interfaces ファイルのエントリの要素

表 20-2 は、interfaces ファイルのエントリの要素を示します。

表 20-2: interfaces ファイルの要素

コンポーネント	値
servername	インスタンスまたは Backup Server の名前。サーバ名の要件は、次のとおりです。 • 名前の長さは 30 文字以内 • 最初の文字は英字 (ASCII 文字の a-z、A-Z) • 2 文字目以降は、英字、数字、またはアンダースコア (_)
retry_attempts (オプション)	最初に失敗した後で、クライアントがサーバへの接続を試行する回数。デフォルトは 0 秒。
delay_interval (オプション)	何秒おきに接続を試行するか。デフォルトは 0 秒。
service_type	エントリによって定義されるサービスのタイプ。次のいずれかにする。 • master • query
api	ネットワークで使用できるアプリケーション・プログラミング・インタフェース。サポートされている値は tli。
protocol	ネットワーク・プロトコルの名前。使用できるプロトコル・タイプは次のとおり。 • TCP/IP、“tcp” と表す。
network	ネットワークの名前。Cluster Edition では現在使用されていません。sybcluster によって、プレースホルダとして “ether” が書き込まれます。
host	サーバのホスト・マシンのネットワーク名またはネットワーク・アドレス。 • TCP/IP の場合、ホスト名またはインターネット・アドレスを使用する。エントリの最大サイズは 32 バイト。 マシンのホスト名を調べるには、そのマシンにログインし、次のように入力します。 `/bin/hostname`
machine	サーバのホスト・マシンのネットワーク名またはネットワーク・アドレス。 ホスト名またはインターネット・アドレスを使用できる。エントリの最大サイズは 32 バイト。 マシンのホスト名を調べるには、そのマシンにログインし、次のように入力する。 `/bin/hostname`

コンポーネント	値
device	ネットワーク・デバイスの終了ポイント。 TCP ネットワークの場合、ネットワーク・ソフトウェアを提供しているベンダによって異なる。デバイス名については、ベンダ固有のマニュアルを確認する。TCP プロトコル・スイートのさまざまなプロトコルに対応した複数のストリーム・デバイスをネットワークが提供している場合がある。TCP ストリーム・デバイスを選択する。一般的な TCP ストリーム・デバイスは */dev/tcp* です。
TLI プロトコルのエントリ用の *address*	アドレスは、次の要素から構成されます。 • TLI のアドレス・プレフィクスは “¥x”。 • ネットワーク・タイプは常に 0002。 • ポート番号は、4 桁の 16 進数に変換。1025 から 65535 の間のユニークな数にします。ネットワーク上の各マシンで */etc/services* ファイルを確認して、使用中のポート番号を調べます。*/etc/services* の新しいセクションに “Sybase specific services” とラベルを付けて、インスタンスのポート番号を入力します。このエントリを作成しなくてもオペレーティング・システムは正常に動作しますが、ファイルにポート番号があれば他のユーザがその番号を使用することを回避できます。 • 8 桁の 16 進数に変換したホスト・マシンの IP ネットワーク・ノード・アドレス。 • 末尾の 16 桁の 0。オプション。
port	1025 から 65535 の間のユニークなポート番号にします。ネットワーク上の各マシンの */etc/services* ファイルを確認して、使用中のポート番号を調べます。*/etc/services* の新しいセクションに “Sybase specific services” とラベルを付けて、インスタンスのポート番号を入力します。このエントリを作成しなくてもオペレーティング・システムは正常に動作しますが、ファイルにポート番号があれば他のユーザがその番号を使用することを回避できます。
ha_failover	ディレクトリ・サービスまたは interfaces ファイルに作成される高可用性のエントリ。
filter	Cluster Edition は、ディレクトリ・サービスの master 行と query 行に追加するフィルタとして SSL (Secure Socket Layers) プロトコルをサポートします。SSL は機密情報を安全に送信するための標準です。SSL の詳細については、「クラスタ環境での SSL の使用」(39 ページ) を参照してください。

マスタ *interfaces* ファイルの作成

マスタ *interfaces* ファイルには、ネットワーク上にあるすべての Sybase サーバ用のエントリが格納されています。ネットワークに接続しているすべてのサーバとクライアントで、このファイルを使用できます。マスタ *interfaces* ファイルのコピーを配布すれば、ネットワーク上にあるすべての Sybase 製品が互いに対話できるようになります。

すべてのインスタンスのエントリが格納されている interfaces ファイル (マスタ・ファイル) のコピーを配布すれば、ネットワーク上の同機種間環境で簡単に interfaces ファイル間の一貫性を管理できます。

ファイルの 1 つのバージョンに対してすべての変更を行い、次に更新したマスタ・ファイルをすべての適当な UNIX Sybase ディレクトリにコピーできます。

次の 2 とおりの方法のいずれかでマスタ・ファイルを作成できます。

- dsedit または dscp を使用する。
- テキスト・エディタを使用する。

dsedit または dscp を使用してマスタ interfaces ファイルを作成する

dsedit ユーティリティまたは dscp ユーティリティを使用してマスタ interfaces ファイルを作成して、すべてのサーバに配布できます。Sybase 製品の使用経験があまりないユーザの場合、テキスト・エディタを使用するよりも dsedit ユーティリティまたは dscp ユーティリティを使用する方が簡単です。dsedit または dscp を使用すると、interfaces ファイルのフォーマットの一貫性が保証されます。

dsedit または dscp を使用してマスタ interfaces ファイルを作成するには、次の手順に従います。

1　最も完全で最新の情報が格納されている interfaces ファイルを選択します。

2　最新の Sybase インストール環境で dsedit または dscp セッションを開始して、この interfaces ファイルを編集します。

3　このファイルにリストされていないインスタンスまたは Backup Server のエントリを追加します。

これらのユーティリティの使用方法の詳細については、『ユーティリティ・ガイド』を参照してください。

テキスト・エディタを使用したマスタ interfaces ファイルの作成

個々の interfaces ファイルから 1 つのマスタ *interfaces* ファイルを構成するには、次の手順に従います。

1　個々の interfaces ファイルを連結します。

2　そのファイルのコピーを作成します。

3　ASCII テキスト・エディタを使用して、連結したファイルのコピーを修正します。

注意　*interfaces* ファイルを手動で編集する場合、各エントリの 2 行目以降の各行が <タブ> 文字で始まっていることを確認してください。

編集したファイル内で、次の要素が一意で正しく指定されている必要があります。

- *servername* − *interfaces* ファイル内の各サーバ・エントリを一意にする必要があります。各クラスタおよびクラスタ内の各インスタンスのサーバ名のエントリは、interfaces ファイル内で一意であることが必要です。
- ホスト・マシンのネットワーク名またはネットワーク・アドレスとインスタンスのポート番号またはオブジェクト番号との組み合わせ。
- 元の *interfaces* ファイルがネットワーク上にマシンが 1 つしかないときに作成された場合、そのファイルのエントリのマシン名 (アドレス) の場所に “loghost” という語が指定されている場合があります。*loghost* がある場合、それをマシン名と置き換えます。

複数のネットワークで使用する interfaces ファイルの設定

プラットフォームによっては、Cluster Edition は複数のネットワークに対応できます。この場合、インスタンスは複数のネットワーク・インタフェースを介してクライアントからの接続要求を受信できます。各ネットワーク・インタフェース用のエントリを interfaces ファイルに追加します。

複数のネットワーク・ハンドラ用にサーバを設定する

複数のネットワーク・リスナを設定するには、次の手順に従います。

1 オペレーティング・システムのホスト・データベース内で、各ネットワーク・インタフェースに対して一意のホスト名を定義します。

2 テキスト・エディタを使用して interfaces ファイルを開き、サーバが受信する追加の各インタフェースにインスタンスの “master” 行のコピーを追加します。

3 各行にユニークなホスト名を指定し、各ネットワーク・インタフェースのネットワーク・ハンドラを設定します。

4 インタフェース内のポート番号は同一である必要はありませんが、同一にすることもできます。これらポート番号の名前と数値の範囲は、プライマリ・ネットワーク・インタフェースと同じ規則に従います。

複数のネットワーク・ハンドラ用 interfaces ファイルの例

次の例は、2 つのネットワーク・インタフェースを持つインスタンスの interfaces ファイルを示します。サーバのホスト・マシンは、コーポレート・ネットワークでは serv_corpnet と呼ばれ、エンジニアリング・ネットワークでは serv_engnet と呼ばれます。

```
# PRODUCTION server with two network listeners
 PRODUCTION<tab>3<tab>3<newline>
 <tab>master tcp ether SERV_CORPNET 4559
 <tab>query tcp ether SERV_CORPNET 4559
 <tab>master tcp ether SERV_ENGNET 5479
```

インスタンスを再起動すると、サーバの DSLISTEN の値に対応するエントリの各 master 行に対してネットワーク・ハンドラ・プロセスが生成されます。各インタフェースで確立される接続は、ピアとして同等に処理されます。

クライアント接続の設定

インスタンスのクライアントが interfaces ファイルをスキャンしてサーバ名を検索する場合、クライアントはサーバのエントリ内で最初に検出した “query” エントリを使用します。このため、複数のネットワーク接続を使用するクライアントの設定は、サーバ・ポートの設定よりも複雑になります。次の 2 つの方法があります。

- すべてのクライアントで DSQUERY 名を使用する。違うマシンの interfaces ファイルには、違うネットワーク名が格納されている。
- クライアントに対して異なる DSQUERY 名を使用する。すべてのマシンで interfaces ファイルは同じだが、interfaces ファイルには複数の DSQUERY 名が含まれている。

単一のネットワーク独立 DSQUERY 名の使用

クライアントに同一の DSQUERY 名を付けることが重要な場合、クライアントのネットワーク・アドレスに必要な変更を interfaces ファイル内に加えることができます。各ネットワークのクライアント・ファイル・サーバに別々の Sybase インストール・ディレクトリおよび別個の interfaces ファイルをインストールすれば、ユーザが正しいネットワーク・アドレスに接続できます。クライアントが使用する DSQUERY 名を変更する代わりに、すべてのネットワークのすべてのクライアントで単一の DSQUERY 名を使用して、各ネットワークの interfaces ファイルを必要に応じて変更します。

この方法は、次を前提にしています。

- Sybase インストール環境のクライアントから各ネットワーク上で読み取り可能なファイルを、システム管理者が完全に制御している。
- 異なるネットワーク上にある Sybase インストール環境の間で、(少なくとも) interfaces ファイルは共有もコピーもされていません。

“engineering” ネットワーク上の interfaces ファイルは、次の例のようになります。

```
PRODUCTION<tab>3<tab>3<newline>
<tab>query tcp ether SERV_ENGNET 5470
<tab>master tcp ether SERV_CORPNET 4559
<tab>master tcp ether SERV_ENGNET 5479
```

“corporate” ネットワーク上の interfaces ファイルは、次の例のようになります。

```
PRODUCTION<tab>3<tab>3<newline>
<tab>query tcp ether SERV_CORPNET 4559
<tab>master tcp ether SERV_CORPNET 4559
<tab>master tcp ether SERV_ENGNET 5479
```

各ファイル内の “query” 行は使用するネットワークによって異なります。

両方のファイル内に完全な “master” 行があります。インスタンスだけが “master” 行を使用するため、これが可能になります。サーバのホスト・マシンから両方のネットワークを認識できると仮定した場合 (両方のホスト名が交換可能な場合)、インスタンスの起動時にどちらの interfaces ファイルが使用されていても問題はありません。

異なる DSQUERY 名の使用

各ネットワーク・リスナで異なる DSQUERY 名を使用するには、次の手順に従います。

1 追加するサーバ名を選択します。

元のサーバ名とネットワーク名を連結できます。たとえば、サーバの名前が PRODUCTION の場合、PRODUCTION_network1 と PRODUCTION_network2 という名前を選択できます。

2 次のいずれかを実行します。

- PC クライアントの場合、sqledit を使用して各ネットワークに 1 つずつ *sql.ini* ファイルのエントリをサーバに対して複数作成します。次の例では、PRODUCTION_network1 と PRODUCTION_network2 に 1 つずつエントリを作成します。詳細については、クライアント・プラットフォームの Open Client のマニュアルを参照してください。

- UNIX クライアントの場合、ASCII テキスト・エディタで *interfaces* ファイルを編集できます。サーバの *interfaces* ファイルから、各ネットワークのサーバ名の行と “master” 行をクライアントの interfaces ファイルにコピーします。各エントリに適切なサーバ名を追加して、"master" を "query" に変更する。

  各ネットワーク上のクライアントは、クライアントが動作しているネットワークに対応した DSQUERY の値を使用します。次の例では、PRODUCTION_network1 または PRODUCTION_network2 を使用できます。

```
# Client entry for PRODUCTION on network1
PRODUCTION_network1<tab>3<tab>3<newline>
<tab>query tcp ether serv_corpnet 4559
# Client entry for PRODUCTION on network2
PRODUCTION_network2<tab>3<tab>3<newline>
<tab>query tcp ether serv_engnet 5479
```

クエリ・ポート・バックアップの設定

複数のネットワーク・インタフェースは、ネットワークで障害が発生した場合のバックアップとしても使用できます。クライアントが 2 つのネットワークを介して 1 つのサーバに接続している場合、最初のネットワークが停止したときは 2 番目のネットワークを介してクライアントは接続を確立できます。

クエリ・ポート・バックアップ用にインスタンスを設定するには、次の手順に従います。

1. interfaces ファイルのサーバ・エントリ内に、複数の “master” 行と “query” 行をインストールします。
2. インスタンスは両方のポートで接続を受信します。インスタンスに接続するためにホスト名とポート番号を検索するクライアントは、接続が確立されるまで各 “query” 行のポートに対して順番に接続を試行します。

   次に、通常の接続が失敗した場合だけに使用するバックアップ・ネットワークの設定方法の例を示します。プライマリ・ネットワークは「コーポレート・ネットワーク」、バックアップは「エンジニアリング・ネットワーク」です。

```
# PRODUCTION server with two network listeners
PRODUCTION<tab>3<tab>3<newline>
<tab>master tcp ether SERV_CORPNET 4559
<tab>master tcp ether SERV_ENGNET 5479
<tab>query tcp ether SERV_CORPNET 4559
<tab>query tcp ether SERV_ENGNET 5479
```

3 Open Client のマニュアルの説明に従って、PC クライアントの interfaces ファイルを適切な複数の “query” エントリで設定します。同機種間環境のクライアント interfaces ファイルの場合、インスタンスの interfaces ファイルのエントリすべてをクライアントの interfaces ファイルにコピーできます。

4 コーポレート・ネットワークが無効の場合、またはホスト・マシンのコーポレート・ネットワーク・インタフェースに障害が発生したり、ネットワーク障害によってシャット・ダウンされた場合、セカンダリ・ポートに接続されます。

IPv6 のサポート

Cluster Edition では、IPv6 技術がサポートされます。

IPv6 について

IPv6 アドレス指定の用語

- リンクローカル・アドレス － 1 つのリンク経由だけで使用できる IPv6 アドレス。
- サイトローカル・アドレス － 1 つのサイト内だけで使用できる IPv6 アドレス。
- グローバル・アドレス － グローバルなインターネットにわたって使用できる IPv6 アドレス。

IPv6 アプリケーションのタイプ

- IPv6-unaware (非認識) － IPv6 アドレスを処理できないアプリケーション。
- IPv6-aware (認識) － IPv4 アドレスを持たないノードと通信できるアプリケーション。API が実際のアドレスの内容とフォーマットを隠す場合など、これはアプリケーションに対して透過的になることがあります。
- IPv6-enabled (有効化) － IPv6-aware (認識) の特徴を持ち、さらに IPv6 の一部の機能を利用できるアプリケーション。
- IPv6-required (要求) － IPv6 の機能を必要とし、IPv4 経由では動作しないアプリケーション。

IPv6 のインフラストラクチャ

デュアル・スタック・インフラストラクチャは IPv4 と IPv6 の両方を実装します。これは、IPv6 対応のサーバとして Cluster Edition を使用するために推奨されるインフラストラクチャの実装です。

Sybase アプリケーションは IPv6 に対応します。Cluster Edition と Open Client/Server コンポーネントを IPv6 対応にするためのすべてのコードは、IETF 設計プリミティブを使用して実行されます。「Creating or converting for IPv6-aware applications」を参照してください。次に、プラットフォームのランタイムでの動作条件と指定の製品およびそのリリース・バージョンの一覧を示します。

表 20-3: IPv6 のサポート

プラットフォーム	Cluster Edition IPv6 への対応	Open Client/Server の IPv6 への対応
Sun Solaris 8 32 ビット版および 64 ビット版	12.5.3a および 15.0	12.5 および 15.0
HP-UX 11i(v1) 32 ビット版および 64 ビット版	12.5.3a および 15.0	12.5 および 15.0
Microsoft Server 2003	12.5.3a および 15.0	12.5 および 15.0
Linux RHEL 3.0	15.0	12.5 および 15.0

XP Server、Backup Server、Replication Server、および Open Switch のように Open Client/Server ベースの多くの Sybase 製品は、ネットワーク・ソケット処理に対して IPv6 を認識するレイヤ構成の Open Client トランスポート制御層 (CTlib->NETlib) により、自動的に IPv6 認識になります。重要なことは、DBlib ベースの Open Client 製品は IPv6 に対応しないということです。

Cluster Edition の IPv6 対応は、サーバ内の一部のコンポーネントが IPv6 に対応しないサード・パーティのコンポーネントであるために、複雑な問題になります。この問題の Cluster Edition に対する影響を理解するために、先に示したプラットフォームとリリースの一覧について、IPv6 に対応する Cluster Edition のすべての機能メカニズムを次に示します。

- 接続ハンドラ
- RPC メカニズム
- Job Scheduler Task／Agent セッション接続
- ネットワーク・ホスト API
- sybsendmsg に対する UDP メッセージのサポート
- コンポーネント統合サービス接続
- ホスト／名前解決
- XML URL 接続ハンドラ
- クライアント・アドレス・データの監査

次の Cluster Edition の機能メカニズムは Ipv6 に対応しません。これらの Cluster Edition のメカニズムは Ipv6 非対応です。

- Java サポート
- ライセンス管理サーバ
- LDAP ドライバ
- クラスタ内のさまざまなインスタンス間のプライベート相互接続

Cluster Edition の IPv6 対応としての起動

デフォルトでは、Cluster Edition は IPv6 非対応です。Ipv6 対応にするために、トレース・フラグ 7841 を使用して Cluster Edition を起動する必要があります。これにより、Cluster Edition は Ipv6 の可用性を決定し、Ipv6 対応になります。

IPv6 をサポートするためのプラットフォームとネットワークのインフラストラクチャの正しい設定は、ネットワークまたは IT の担当者に確認してください。

2 つ目のトレース•フラグである 7815 は、アドレス接続要求やホスト／名前の検索を取得および記録する Cluster Edition を起動する場合に設定できます。

IPv6 Cluster Edition のトレース・フラグは次のとおりです。

- T7841 － Cluster Edition を IPv6 対応にします。
- T7815 － すべての Cluster Edition の IPv4 および IPv6 のクライアント・アドレス接続要求をレポートします。

IPv6 対応の Cluster Edition を起動する前に、インフラストラクチャが正しく設定されていることを確認します。オペレーティング・システムが正しく設定されていれば、IPv6 **接続ハンドラ**を設定して有効にできます。IPv6 **接続ハンドラ**の設定と有効化には、補足の DCL エントリを追加する必要があります。1 つの Cluster Edition の設定では、通常は DCL に最高 32 個の**接続ハンドラ**を割り当てることができます。

たとえば、サイト・ローカルの設定時に、次のネームサーバ設定の 2 つのドメインを使用するとします。

```
sybase.com - being responsible for all IPv4 networking applications
v6.sybase.com - being responsible for all IPv6 networking applications
```

ポート 17100 に対するホスト “revival” で、“SYBASE” という名前でクラスタを起動する Cluster Edition の DCL エントリは、次のようになります。

```
SYBASE
master tcp ether revival.sybase.com 17100
query tcp ether revival.sybase.com 17100
master tcp ether revival.v6.sybase.com 17100
query tcp ether revival.v6.sybase.com 17100
```

この例で、Cluster Edition が IPv6 対応で起動すると、2 つの**接続ハンドラ**が作成されます。1 つは受信する IPv4 クライアント接続要求をポート 17100 で待機し、もう 1 つは受信する IPv6 クライアント接続要求をポート 17100 で待機します。

トラブルシューティング

この項では、サーバの起動不能の原因となるいくつかの一般的な状況について、その解決方法を説明します。

サーバが起動しない

サーバが起動せずに次のメッセージが表示されるときは、interfaces ファイル内で指定したポート番号が使用中の場合があります。

```
00:00000:00002:2003/09/22 12:37:23.63 kernel  network name SERV_CORPNET, type ether, port
4559, filter NONE
00:00000:00002:2003/09/22 12:37:23.65 kernel  ninit: bind, Address already in use
00:00000:00002:2003/09/22 12:37:23.68 server  Error: 1602, Severity: 18, State: 2
00:00000:00002:2003/09/22 12:37:23.68 server  Unable to initialize network 0
00:00000:00002:2003/09/22 12:37:23.68 kernel  ninit: All master network
listeners have failed.  Shutting down.
00:00000:00002:2003/09/22 12:37:23.68 kernel  ueshutdown: exiting
00:00000:00016:2003/09/22 16:11:35.46 server  SQL Server shutdown by request.
```

❖ ポートの割り当てを調べる

1　interfaces ファイルを調べて、サーバに割り当てたポート番号を確認します。

2　次のように入力して、別のプロセスが同じポート番号を使用していないか確認します。

```
netstat -a
```

netstat の出力にそのポート番号がローカル・アドレスとして表示された場合、このポートはサーバに使用できません。別のプロセスがすでにこのポートを使用しています。

3　サーバ・ポートが使用中か検証するには、サーバを手動で起動します。

割り当てられたポート番号がすでに使用中の場合、サーバは起動しません。

手動でサーバを起動する方法の詳細については、使用しているプラットフォームの『インストール・ガイド』と『ASE ユーティリティ・ガイド』を参照してください。

❖ **終了したはずのサーバ・プロセスがポート番号を開放しない場合**

1 次のいずれかを実行します。

- オペレーティング・システムの kill コマンドを使用して、プロセスを終了させる。
- interfaces ファイルを修正して、サーバに別のポート番号を使用します。

2 サーバを手動で起動して、ポート番号が使用できるか確認します。

手動でサーバを起動する方法の詳細については、使用しているプラットフォームの『インストール・ガイド』と『ASE ユーティリティ・ガイド』を参照してください。

ESP 実行時のエラー

ESP (拡張ストアド・プロシージャ) を実行しようとしたときに、次のようなエラーが表示される場合があります。

```
00:00000:00008:1997/09/10 12:52:53.03 kernel  XP Server failed to start. Try bringing up
XP Server manually. Check SQL Server documentation for more information on how to bring
XP Server up.
```

ポート番号が別のプロセスで使用されている可能性があるため、XP Server を起動できません。前の項で説明した netstat コマンドを使用して、XP Server 用に指定したポート番号が使用中か確認します。

同じポート番号を使用しているプロセスがない場合、先ほど試みた ESP を実行します。XP Server は、自動で起動します。

同じポート番号を使用するプロセスが見つかった場合は、次のいずれかを実行します。

- XP Server で新しいポート番号を使用するように interfaces ファイルを変更する。
- XP Server に割り当てられたポート番号を使用するプロセスを停止する。

Cluster Edition を再起動し、前に試行した ESP を実行します。XP Server は、自動で起動します。

索引

記号

A

B

C

D

H

I

S

T

U

V

W

X

あ

い

え

お

か

き

く

け

こ

さ

し

す

せ

そ

た

ち

つ

て

と

な

ね

の

は

ひ

ふ

へ

ほ

ま

め

も

や

ゆ

ら

り

る

れ

ろ