SYBASE
An SAP Company

高可用性システムにおける Sybase フェールオーバの使用

Adaptive Server® Enterprise

15.7

ドキュメント ID：DC31661-01-1570-01
改訂：2011 年 9 月

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

目次

第 1 章 フェールオーバとフェールバック 1
フェールオーバの概要 1
フェールオーバ時のクライアント接続 3
フェールオーバにおけるユーザのログイン 3
フェールバックの概要 4
フェールバックの実行 5
高可用性ノードにおけるクラスタのロック 6

第 2 章 高可用性の概要 9
アクティブ／アクティブとアクティブ／パッシブの相違点 11
フェールオーバの稼働条件 12
リソースの条件 13
Sybase のフェールオーバと高可用性の機能 14
単一のシステムとしてアクセス可能なシステム 16
Sybase フェールオーバに関する注意事項 16
モニタリング・テーブル・スクリプトのインストール 16
ディスク・ミラーリングの使用 17
installevent スクリプトの実行 17
SYB_HACMP サーバ・エントリの作成 17
ユーザ定義データ型の定義 18
Adaptive Server と 2 フェーズ・コミット・トランザクション 18

第 3 章 非対称型と対称型の設定 19
非対称型と対称型の設定 19
非対称型コンパニオンの設定 19
対称型コンパニオンの設定 21
高可用性システムでの監査 22
監査オプションの設定 23

第 4 章 フェールオーバのモード 25
モードの概要 25
コンパニオン・サーバの各モードの詳細 26
コンパニオンのモードの確認 28
ドメイン 29

第 5 章 プロキシ・データベース、ユーザ・データベース、プロキシ・システム・テーブル 31
プロキシ・データベース 31
プロキシ・データベースの作成 32
プロキシ・データベースのサイズ 33
プロキシ・データベースでのコマンドとシステム・プロシージャ 34
手動でのプロキシ・データベースの更新 35
master のプロキシ・システム・テーブル 36

第 6 章 do_advisory の実行 37
do_advisory オプションの概要 37
do_advisory オプションの実行 41
定属性 42

第 7 章 HP システムでフェールオーバを使用できるように Adaptive Server を設定する 45
ハードウェアとオペレーティング・システムの稼働条件 45
Adaptive Server の高可用性を実現するための準備 46
Adaptive Server のインストール 46
両方の Adaptive Server のエントリを interfaces ファイルに追加する ... 46
$SYBASE の値の設定 47
sybha 実行プログラムの設定 47
マスタ・デバイス以外の新しいデフォルト・デバイスの作成 49
sysservers へのローカル・サーバの追加 49
sysservers へのセカンダリ・コンパニオンの追加 49
installhasvss の実行 50
システム管理者への ha_role の割り当て 50
設定パラメータの確認 50
HP システムをフェールオーバ用に設定する 51
パッケージ設定ファイルの作成 52
ASE_HA.sh スクリプトの編集 53
設定の確認と分配 61
フェールオーバ用コンパニオン・サーバの設定 63
do_advisory オプションを指定して sp_companion を実行する 63
非対称型コンパニオン設定の作成 64
対称型設定の作成 65
Sybase フェールオーバの管理 65
プライマリ・コンパニオンへのフェールバックとノーマル・コンパニオン・モードの再開 66
コンパニオン・モードのサスペンド 66
コンパニオン・モードの削除 67

HP システムでの Sybase フェールオーバのトラブルシューティング 68
エラー・メッセージ 18750 68
失敗した prepare_failback からのリカバリ 69
エラー・ログのロケーション 70
Adaptive Server のアップグレード 71

第 8 章 IBM AIX HACMP システムでフェールオーバを使用できるように Adaptive Server を設定する 75
ハードウェアとオペレーティング・システムの稼働条件 75
IBM AIX HACMP システムでフェールオーバを使用するための稼働条件 76
高可用性で動作するように Adaptive Server を準備する 77
Adaptive Server のインストール 77
両方の Adaptive Server のエントリを interfaces ファイルに追加する 78
$SYBASE の値の設定 78
sybha 実行プログラム 79
設定パラメータの確認 80
マスタ・ログへのスレッショルドの追加 81
マスタ・デバイス以外の新しいデフォルト・デバイスの作成 81
sysservers へのローカル・サーバの追加 82
sysservers へのセカンダリ・コンパニオンの追加 82
installhasvss スクリプトの実行 82
システム管理者への ha_role の割り当て 83
IBM AIX サブシステムを Sybase フェールオーバ用に設定する 83
ASE_HA.sh スクリプトの修正 83
HACMP でのリソース・グループの設定 87
フェールオーバ用コンパニオン・サーバの設定 90
do_advisory オプションを指定して sp_companion を実行する 90
非対称型コンパニオン設定の作成 91
対称型設定の作成 92
プライマリ・コンパニオンをモニタ対象リソースとして起動する 93
Sybase フェールオーバの管理 93
プライマリ・ノードへのフェールバック 94
コンパニオン・モードのサスペンド 95
ノーマル・コンパニオン・モードの再開 96
コンパニオン・モードの削除 97
HACMP for AIX でのフェールオーバのトラブルシューティング 98
エラー・メッセージ 18750 98
失敗した prepare_failback からのリカバリ 99
エラー・ログのロケーション 99
Adaptive Server のアップグレード 100

第 9 章 Sun Cluster 3.0 および 3.1 のアクティブ／アクティブ設定 103
ハードウェアとオペレーティング・システムの稼働条件 103
Sun Cluster のアクティブ／アクティブ設定 104
アクティブ／アクティブ設定のための Adaptive Server の準備 106
Adaptive Server のインストール 106
両方の Adaptive Server のエントリを interfaces ファイルに追加する 106
両方のコンパニオンの $SYBASE を同じ値にする 107
sybha の実行 108
新しいデフォルト・デバイスの作成 109
sysservers へのローカル・サーバの追加 109
sysservers へのセカンダリ・コンパニオンの追加 110
システム管理者への ha_role の割り当て 110
installhasvss スクリプトの実行 110
設定パラメータの確認 111
マスタ・ログへのスレッショルドの追加 111
フォールト・モニタ用のユーザとログインの追加 111
Sun Cluster サブシステムの設定 112
syscadm スクリプトの使用 113
Adaptive Server リソースの拡張プロパティ 117
Adaptive Server リソース・グループの設定 119
SUNW.HAStoragePlus の使用 121
フェールオーバ用コンパニオン・サーバの設定 123
Adaptive Server 内の高可用性サービス・ライブラリ 123
do_advisory を指定して sp_companion を実行する 124
非対称型コンパニオン設定の作成 125
対称型設定の作成 127
Sybase フェールオーバの管理 128
プライマリ・コンパニオンへのフェールバック 128
ノーマル・コンパニオン・モードのサスペンド 129
ノーマル・コンパニオン・モードの再開 129
コンパニオン・モードの削除 130
Sun Cluster での高可用性の確認 130
リソース・グループの手動設定 131
プライマリ・コンパニオン・リソース・グループ 131
セカンダリ・コンパニオン・リソース・グループ 134
Adaptive Server のアップグレード 136
トラブルシューティング 139
失敗した prepare_failback からのリカバリ 139
セカンダリ・コンパニオンでのセカンダリ・フェールオーバからのリカバリ 140
セカンダリ・コンパニオンのフェールオーバ防止 140
リソースとリソース・グループの状態の変更 141
エラー・ログのロケーション 141

第 10 章 Sun Cluster 3.0 および 3.1 のアクティブ／パッシブ設定 143
ハードウェアとオペレーティング・システムの稼働条件 144
Sun Cluster のアクティブ／パッシブ設定 144
アクティブ／パッシブ設定でのフェールバック 147
アクティブ／パッシブ設定のクライアント 147
アクティブ／パッシブ設定のための Adaptive Server の準備 147
Adaptive Server のインストール 147
Adaptive Server への環境の引き渡し 149
クラスタでの SySam License Manager の実行 149
Adaptive Server のエントリを interfaces ファイルに追加する 150
設定パラメータの確認 152
マスタ・ログへのスレッショルドの追加 152
フォールト・モニタ用のユーザとログインの追加 152
Sun Cluster サブシステムの設定 153
syscadm スクリプトの使用 154
Adaptive Server リソース・グループの設定 158
SUNW.HAStoragePlus の使用 160
アクティブ／パッシブ設定の確認 161
リソース・グループの手動設定 163
Adaptive Server のアップグレード 165
エラー・ログのロケーション 167

第 11 章 Veritas 5.0 以降でフェールオーバを使用できるように Adaptive Server を設定する 169
ハードウェアとオペレーティング・システムの稼働条件 170
高可用性で動作するように Adaptive Server を準備する 171
Adaptive Server のインストール 172
両方の Adaptive Server のエントリを interfaces ファイルに追加する 172
sybha 実行プログラム 173
新しいデフォルト・デバイスの作成 174
sysservers へのローカル・サーバの追加 174
ha_role の割り当て 175
高可用性ストアド・プロシージャのインストール 175
設定パラメータの確認 176
マスタ・ログへのスレッショルドの追加 176
Veritas サブシステムを Sybase フェールオーバ用に設定する 177
HAase エージェントのインストール 177
Adaptive Server ログイン・ファイルの作成 177
HAase リソース・タイプのインポート 178
HAase エージェントの起動 178
HAase リソースの追加 179
各サービス・グループの HAase リソースのインスタンスの設定 180

フェールオーバ用コンパニオン・サーバの設定 181
高可用性モニタ用のユーザとログインの追加 181
do_advisory オプションを指定して sp_companion を実行する 182
高可用性エージェントの確認 182
非対称型コンパニオン設定の作成 183
対称型設定の設定 184
Sybase フェールオーバの管理 185
フェールオーバ中 185
プライマリ・コンパニオンへのフェールバック 185
ノーマル・コンパニオン・モードのサスペンド 186
ノーマル・コンパニオン・モードの再開 186
コンパニオン・モードの削除 187
リソース・タイプ Sybase のエージェントからのアップグレード 187
Adaptive Server のアップグレード 188
Veritas クラスタでのフェールオーバのトラブルシューティング 191
失敗した prepare_failback からのリカバリ 192
ログのロケーション 192

付録 A　フェールオーバ設定での Open Client の機能 193
CTLIB アプリケーションの変更 193

付録 B　2 次ポイント障害のトラブルシューティング 195
dbcc ha_admin を使ったトラブルシューティング 195
installmaster の再インストール 196
installhasvss の再実行 196
フェールオーバ・コマンドのロールバックのための
dbcc ha_admin の使用 197
@@hacmpservername の使用 197
エラー・メッセージ 198

付録 C　コマンド、システム・プロシージャ、データベースの変更 199
コマンドの変更 199
システム・プロシージャの変更 201
テーブル・ロックを保持するシステム・プロシージャ 201
変更を同期するシステム・プロシージャ 202
システム・プロシージャのその他の変更 203
高可用性システムの dbcc オプション 205

用語解説 207

索引 209

第 1 章

フェールオーバとフェールバック

この章では、フェールオーバとフェールバックの特性について説明します。

トピック名	ページ
フェールオーバの概要	1
フェールバックの概要	4
高可用性ノードにおけるクラスタのロック	6

フェールオーバの概要

コンピュータ・システムに障害が発生すると、データベース、メタデータ、ユーザ接続はセカンダリ・サーバに移動され、ユーザはそのままデータにアクセスできるようになります。これをフェールオーバと呼びます。

Adaptive Server® では、フェールオーバを実現するように設定された高可用性クラスタを構成します。フェールオーバには、次の一連の 3 つのステップがあります。

1 システム・フェールオーバ – プライマリ・ノードがセカンダリ・ノードにフェールオーバします。

2 コンパニオン・フェールオーバ – プライマリ・コンパニオンがセカンダリ・ノードにフェールオーバします。

3 接続フェールオーバ – フェールオーバ・プロパティを持つ接続がセカンダリ・コンパニオンにフェールオーバします。

ステップ 2 と 3 について、次に説明します。ステップ 1 については、使用している高可用性システムのマニュアルを参照してください。

フェールオーバ時は、セカンダリのサーバがオペレーティング・システムの高可用性システムを介してプライマリの障害を検出し、フェールオーバ機能を開始します。この機能では、次のことが実行されます。

1 disk reinit を実行して、マスタ・デバイスのパス名をローカル・ドライブに再マップします。disk reinit は、マスタ・デバイスの内容を変更しません。

2 master データベースをマウントしてリカバリし、オンラインにします。

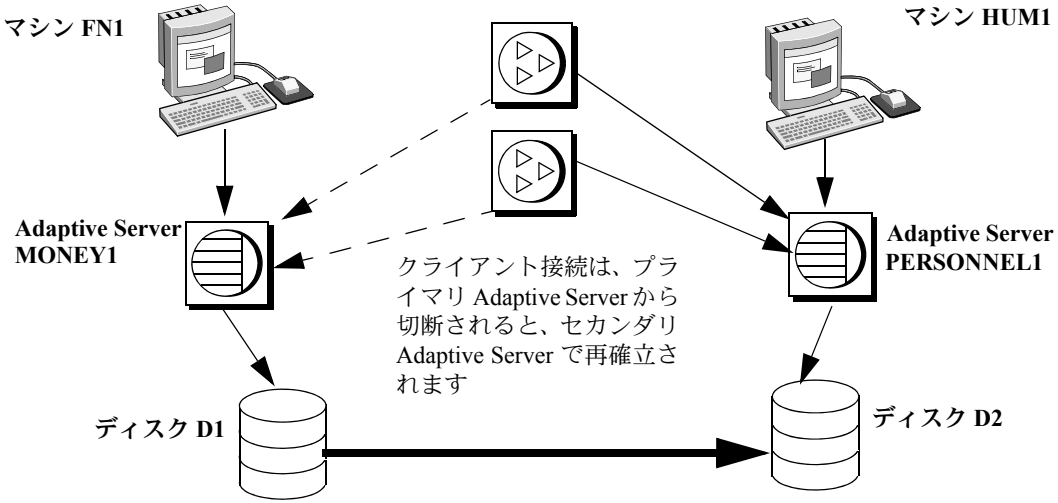
3 プライマリ・コンパニオンの sysdevices にリストされている各デバイスをセカンダリ・コンパニオンの sysdevices にマップして、ディスク上で disk reinit を実行します。

4 プライマリ・コンパニオンのすべてのデータベースをセカンダリ・コンパニオンにマウントします。セカンダリ・コンパニオンは、ログからリカバリを実行してからすべてのデータベースをオンラインにします。tempdb および model はマウントされませんプロキシ・データベースは、comp_dbid_dbname という名前でマウントされます。

   セカンダリ・コンパニオンがマウントする各データベースの名前には、_companion というサフィックスが追加されます。たとえば、master データベースは master_companion になり、sybsystemprocs は sybsystemprocs_companion になります。セカンダリの Adaptive Server はこのサフィックスを追加して、システム上に現在あるデータベースをユニークに識別できるようにします。_companion サフィックスが追加されないユーザ・データベースについては、その名前自体がユニークであることになります。

フェールオーバ・プロパティのあるユーザ接続と CS_FAILOVER プロパティを使用しているクライアントは保持され、セカンダリ・コンパニオン上で接続が再設定されます。コミットされていないトランザクションは、再送信する必要があります。

図 1-1: フェールオーバ処理



master データベースとシステム・データベースは、セカンダリ Adaptive Server にマイグレートされ、suffix_companion が追加されます。プロキシ・データベースは、名前が変更され、停止されて、ユーザ・データベースで置き換えられます。

セカンダリ・コンパニオンが高可用性システムからフェールオーバのメッセージを受信すると、プライマリ・コンパニオンに接続したクライアントでは新しいトランザクションは開始されません。フェールオーバ時に完了しなかったトランザクションは、ロールバックされます。フェールオーバが終了した後に、クライアントやユーザは、ロールバックされたトランザクションを再送信する必要があります。

フェールオーバ時のクライアント接続

フェールオーバ・プロパティのあるクライアントは、フェールオーバ時に自動的に再接続します。このためには、*interfaces* ファイルに *hafailover* とラベル付けされた行を追加して、クライアントがセカンダリ・コンパニオンに接続するのに必要な接続情報を提供する必要があります。この行を追加するには、ファイル・エディタか dsedit ユーティリティを使用します。

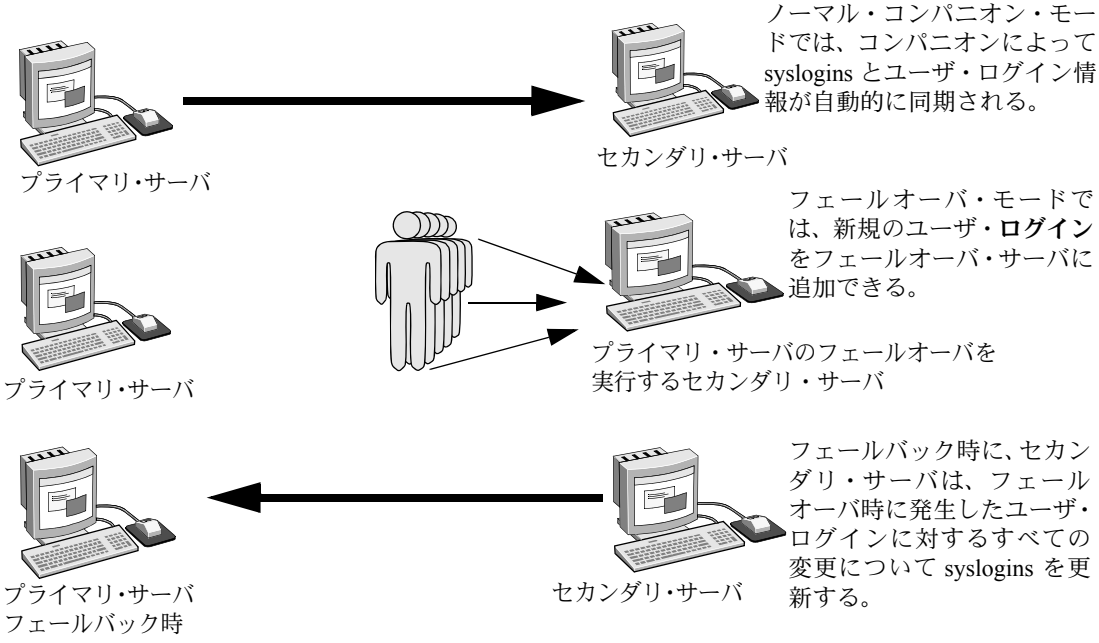
この情報を *interfaces* ファイルまたは *sql.ini* に追加する方法の詳細については、該当するプラットフォーム固有の設定に関する章を参照してください。

クライアント・アプリケーションは、フェールオーバによって送信できなかったクエリを再送する必要があります。クライアント・アプリケーションの詳細については、「付録 A フェールオーバ設定での Open Client の機能」を参照してください。

フェールオーバにおけるユーザのログイン

ノーマル・コンパニオン・モードの間、コンパニオンはユーザのログイン、アクセス、セキュリティ情報などの変更を自動的に同期します。フェールオーバ時に追加されたログインは、プライマリ・コンパニオンがフェールバックで更新されるときに、プライマリ・コンパニオンに自動的に追加されます。コンパニオンが正常にフェールオーバされたら、コミットされなかったトランザクションを再送信して、セッション・レベルで設定されたオプションを再確立する必要があります。

図 1-2: プライマリ・サーバとセカンダリ・サーバ間の syslogins の同期



すべてのユーザの役割と権限は、フェールオーバ後に保持されます。

フェールバックの概要

プライマリ・コンパニオンまたはマシンのオペレーションが再開できる状態になったら、ha_role を持つユーザがフェールバックを実行して、サーバをノーマル・コンパニオン・モードに戻します。フェールバックを実行するとフェールオーバされたコンパニオンのデータベースが一時的にシャットダウンされるので、フェールバックはアプリケーションの負荷が軽いときに実行する必要があります。Adaptive Server の負荷が重いときに実行すると、フェールバックは正常に完了しますが、処理が遅くなり、セカンダリ・コンパニオンのパフォーマンスが低下します。適切なタイミングでフェールバックを実行すると、クライアントが再接続するまでの待機時間をかなり短縮できます。

フェールバックの実行

フェールバックには 4 つのステップがあります。

1　フェールバックの準備を行います。

注意　AIX の IBM HACMP は、プライマリ・ノードがノーマル・コンパニオン・モードを再開できる状態になったら、自動的にフェールバックを実行します。詳細については、「第 8 章 IBM AIX HACMP システムでフェールオーバを使用できるように Adaptive Server を設定する」を参照してください。

---

prepare_failback をセカンダリ・コンパニオンから発行すると、データベース・デバイスとデータベースが解放されます。

```
sp_companion server_name 'prepare_failback'
```

server_name は、セカンダリ・コンパニオンの名前です。セカンダリ・コンパニオンは、フェールバック時に次のようなメッセージを発行します。

```
Step:Access across the servers verified
Step:Primary databases are shutdown in secondary
Step:Primary databases dropped from current secondary
Step:Primary devices released from current secondary
Step:Prepare failback for primary server  complete
(return status = 0)
```

個々のプラットフォームのサブシステムに従って、デバイスをプライマリ・ノードに戻します。

2　高可用性システムでは、プライマリ・コンパニオンは自動的に再起動されます。

3　do_advisory オプションを指定して sp_companion を実行し、フェールバックの成功を妨げる属性設定がないことを確認します。詳細については、「第 6 章 do_advisory の実行」を参照してください。

4　フェールバックが完了したら、プライマリ・コンパニオン (最初に障害が発生したコンパニオン) で sp_companion を発行して、ノーマル・コンパニオン・モードに戻します。sp_companion resume の詳細については、使用しているプラットフォームの該当する章を参照してください。

注意　sp_companion resume を発行しないと、フェールオーバ・プロパティが設定されたクライアントを接続できません。sp_companion prepare_failback を発行してからクライアントを再接続しようとすると、sp_companion resume を発行するまでクライアントはハングします。

高可用性ノードにおけるクラスタのロック

高可用性クラスタのコンパニオン・サーバでは、ユーザ情報を同期させる必要があります。コンパニオンの設定に影響するオペレーションをクラスタ・オペレーションといい、通常 sp_companion によって開始されます。コンパニオンは同期させる必要があるため、ノードの設定に影響するクラスタ・オペレーションを実行するクライアントは、並列ではなく逐次でのみ実行できます。つまり、クラスタ・オペレーションを実行できるのは、1 度に 1 クライアントだけです。

クライアントは、クラスタ全体のロックを取得してからクラスタ・オペレーションを実行します。このロックによって、他のクライアントはクラスタ・オペレーションを同時に実行できなくなります。クラスタのロックは、両方のコンパニオンが同期してから解放されます。クライアントがクラスタのロックを取得できない場合、クラスタ・オペレーションは正常に実行されません。オペレーションが逐次で実行されても、クライアントのキューはありません。正常に実行されなかったクラスタ・オペレーションは再送信する必要があります。

実行中のクラスタ・オペレーションで必要な場合には、クラスタのロックを取得することもできます。

通常、クラスタのロックを意識することはありません。クラスタのロックはデータベースで発生する他のトランザクションには影響せず、クラスタ・オペレーションだけに影響します。ただし、クラスタのロックを取得しているクライアント接続が、クラスタ・オペレーションの実行中に失われると (たとえば、完了前にクラスタ・オペレーションを終了した場合など)、ロックが残ります。この場合、次のクライントはクラスタのロックを取得できなくなります。

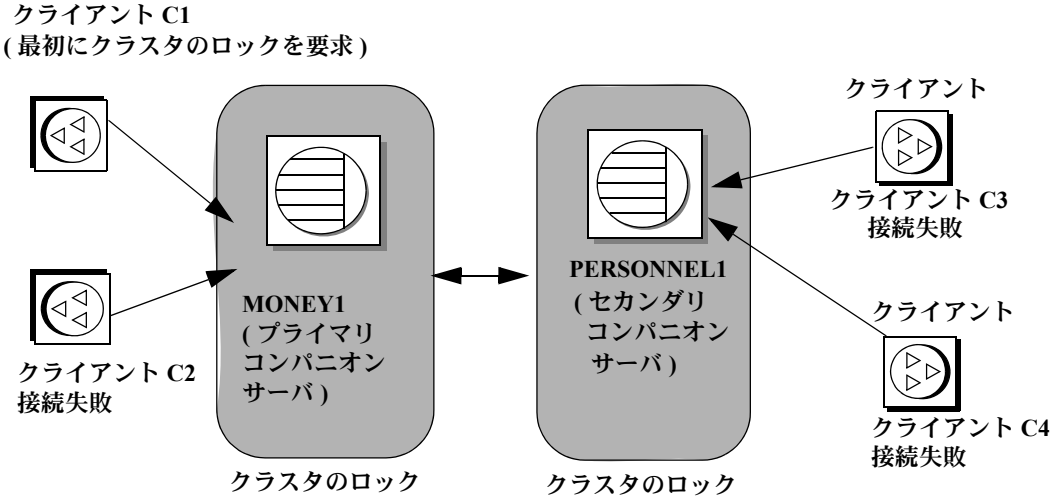
次のように dbcc ha_admin を発行すると、クラスタのロックを取得または解放できます。

```
dbcc ha_admin server_name clusterlock [acquire | release]
```

dbcc ha_admin の詳細については、「高可用性システムの dbcc オプション」(205 ページ) を参照してください。

図 1-3 は、4 クライアントが接続されている 2 つのコンパニオン・サーバを示しています。4 クライアントすべてがクラスタ・オペレーションを実行しようとしています。

図 1-3: クラスタ・オペレーションのために接続するクライアント



1　クライアント接続 C1 と C2 は、同時にクラスタ全体のロックを取得してクラスタ・オペレーションを実行しようとしています。

2　クライアント C1 はまず MONEY1 に接続して、クラスタ全体のロックを受信します。

3　クライアント C2 はクラスタ全体のロックを取得できないため、クラスタ・オペレーションを実行できません。

4　C1 がクラスタ・オペレーションを実行している間に、クライアント C3 と C4 は PERSONNEL1 からクラスタ全体のロックを取得しようとしています。

5　クラスタ全体のロックは C1 によって取得されているため、クライアント C3 と C4 はクラスタ全体のロックを取得できません。

6　クライアント C1 はクラスタ・オペレーションを完了してから、クラスタ全体のロックを解放します。

7　クライアント接続 C2、C3、C4 は、クラスタ全体のロックを取得できなかったことをシステム管理者に伝達します。クライアント C1 がクラスタ全体のロックを解放してから、システム管理者はクラスタ・オペレーションを行うクライアント接続を再送信できます。

第 2 章

高可用性の概要

トピック名	ページ
アクティブ／アクティブとアクティブ／パッシブの相違点	11
フェールオーバの稼働条件	12
リソースの条件	13
Sybase のフェールオーバと高可用性の機能	14
単一のシステムとしてアクセス可能なシステム	16
Sybase フェールオーバに関する注意事項	16

コンピュータ・システムを利用する業務が増えると、これらのシステムを常に使用できることが期待されます。高可用性とは、コンピュータ・システムやネットワークでハードウェアやソフトウェアの障害が発生しても、バックアップ・システムにフェールオーバされるようにシステムが設定されていることです。この設定によって、業務を通常どおりに続けることができます。問題が解決したら、システムはプライマリ・システムにフェールバックします。

Sybase® の高可用性フェールオーバによって、アクティブ／アクティブまたはアクティブ／パッシブなど特定の設定のネットワークにおいて、Adaptive Server Enterprise をサーバのクラスタで運用できます。このようなシステムによって、サーバのフェールオーバとフェールバックが可能になります。このマニュアルでは、Adaptive Server 高可用性システムの設定および実行方法について説明します。

「高可用性」クラスタには、2 つ以上のマシンが含まれます。これらのマシンは、1 つのマシン (またはアプリケーション) がダウンした場合にもう 1 つのマシンが両方のマシンの作業負荷を処理するように設定されています。各マシンを、高可用性クラスタの「ノード」といいます。一般的に、高可用性クラスタはシステムが常に稼働していなければならないような環境、たとえば、クライアントが絶えず接続する銀行のシステムなどで使用します。

「プライマリ・コンパニオン」またはマシンに障害が発生すると、データベース、メタデータ、ユーザ接続はセカンダリ・サーバに移動されるため、ユーザはそのままデータにアクセスできます。これを「フェールオーバ」と呼びます。

プライマリ・コンパニオンまたはマシンのオペレーションが再開できる状態になったら、ha_role を持つユーザがフェールバックを実行して、サーバをノーマル・コンパニオン・モードに戻します。

アクティブ／アクティブ設定

「アクティブ／アクティブ」設定は 2 つのノードで構成されます。「クラスタ」内のどちらのノードでも Adaptive Server が独立して負荷を管理しており、一方に障害が発生したときにもう一方がその負荷を引き受けることができます。

負荷を引き受ける Adaptive Server をセカンダリ・コンパニオン、障害の発生した Adaptive Server をプライマリ・コンパニオンといいます。2 つを合わせて、「コンパニオン・サーバ」といいます。障害が発生したときノード間で負荷が移動することを、フェールオーバといいます。プライマリ・コンパニオンが再度負荷を引き受けられる状態になると、負荷は元のノードに戻ります。これをフェールバックといいます。

フェールオーバが発生すると、プライマリ・コンパニオンに接続していたクライアントは、フェールオーバ・プロパティを使用して、自動的にセカンダリ・コンパニオンへのネットワーク接続を再確立します。

両方の Adaptive Server をフェールオーバ時に正常に管理できるように、オペレーティング・システムを設定します。高可用性に対応するようにシステムを設定する方法については、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。

注意 「アクティブ／アクティブ」設定でフェールオーバ用に設定された Adaptive Server を、shutdown コマンドを使用してシャットダウンできるのは、Adaptive Server のコンパニオン設定をサーバ・レベルとプラットフォーム・レベルの両方でサスペンドした場合のみです。詳細については、このマニュアル内の、該当するプラットフォーム固有の設定に関する章を参照してください。

アクティブ／パッシブ設定

「アクティブ／パッシブ」設定はマルチ・ノード設定です。この構成には、1 つの Adaptive Server、Adaptive Server が動作するプライマリ・ノード、必要に応じて Adaptive Server とそのリソースのホストとなる一連のセカンダリ・ノードが含まれます。

Adaptive Server がプライマリ・ノードで実行できなくなると、フェールオーバが発生し、Adaptive Server はセカンダリ・ノードに移動して再起動されます。プライマリ・ノードでリカバリが行われ、Adaptive Server と関連するリソースのホストとして正常に機能できるようになったら、Adaptive Server はプライマリ・ノードに戻ることができます。

フェールオーバまたはフェールバックでは、Adaptive Server がセカンダリ・ノードで再起動されると、Adaptive Server に接続していたクライアントがネットワーク接続を再確立し、コミットされていないトランザクションを再送信します。フェールオーバ・プロパティを使用するクライアント接続では、接続が自動的に再確立されます。

Sybase では、Sun Cluster 3.0 のアクティブ／パッシブ設定がサポートされています。他のクラスタ・プラットフォームについては、担当各社にお問い合わせください。Sun Cluster 3.0 でのアクティブ／パッシブ・モードの Adaptive Server の設定方法の詳細については、「第 10 章 Sun Cluster 3.0 および 3.1 のアクティブ／パッシブ設定」を参照してください。このマニュアルの他の章では、特に明記しないかぎりアクティブ／アクティブ設定を対象としています。

注意「アクティブ／パッシブ」設定でフェールオーバ用に設定された Adaptive Server は、プラットフォーム・レベルで Adaptive Server のモニタを無効にした後でのみ shutdown コマンドを使用してシャットダウンできます。

Adaptive Server は、次のクラスタ・プラットフォームについて Sybase の高可用性設定をサポートします。

- HPIA － MCSG 11.19
- IBM AIX － HACMP 6.2
- Sun Solaris － VCS 5.0、SunCluster 3.2
- Linux の場合：
  - RHEL 5.0 / VCS 5.0
  - RHEL 6.0 / VCS 5.1 SP1
  - SuSE Enterprise 11 / VCS 5.1

注意　Adaptive Server バージョン 15.7 以降では、Windows x86 32 ビットでの高可用性をサポートしません。Windows x86 32 ビットで現在高可用性を実行している場合は、カスタマ・サポート要員にご連絡ください。

アクティブ／アクティブとアクティブ／パッシブの相違点

表 2-1 は、アクティブ／アクティブ設定とアクティブ／パッシブ設定の相違点を示します。

表 2-1: アクティブ／アクティブとアクティブ／パッシブの相違点

アクティブ／アクティブ	アクティブ／パッシブ
設定 : 2 つの Adaptive Server がコンパニオン・サーバとして設定され、それぞれが独立した負荷を管理する。これら 2 つのコンパニオンは、一方がフェールオーバしないかぎり、それぞれプライマリ・ノードとセカンダリ・ノード上で独立したサーバとして稼働する。	設定 : 1 つの Adaptive Server がプライマリ・ノードまたはセカンダリ・ノードのいずれかで稼働する。Adaptive Server は、フェールオーバ前はプライマリ・ノードで、フェールオーバ後はセカンダリ・ノードで稼働する。

アクティブ／アクティブ	アクティブ／パッシブ
フェールオーバ : フェールオーバが発生すると、プライマリ・コンパニオンのデバイスやクライアント接続などをセカンダリ・コンパニオンが引き継ぐ。プライマリ・コンパニオンがフェールバックしてアクティビティを再開するまで、セカンダリ・コンパニオンが、フェールオーバされたクライアントと新規クライアントにサービスを提供する。	フェールオーバ : フェールオーバが発生すると、Adaptive Server と関連するリソースはセカンダリ・ノードに移動され、再起動される。
フェールバック : フェールバックとは、プライマリ・コンパニオンが、セカンダリ・コンパニオンからデバイスとクライアント接続を引き取ってサービスを再開する予定されたイベントである。	フェールバック : フェールバックとは、Adaptive Server とそのリソースをプライマリ・ノードに移す予定されたフェールオーバ、つまり移動である。フェールバックは必須ではないが、管理の目的で実行できる。
クライアント接続のフェールオーバ : フェールオーバ時に、クライアントはセカンダリ・コンパニオンに接続して、コミットされていないトランザクションを再送信する。フェールバック時に、クライアントはプライマリ・コンパニオンに接続してトランザクションを再送信する。フェールオーバ・プロパティを使用するクライアントでは、接続が自動的に再確立される。	クライアント接続のフェールオーバ : フェールオーバ時とフェールバック時に、クライアントは同じ Adaptive Server に接続して、コミットされていないトランザクションを再送信する。フェールオーバ・プロパティを使用するクライアントでは、接続が自動的に再確立される。

フェールオーバの稼働条件

Adaptive Server でフェールオーバを使用するには、ASE_HA ライセンス・オプションを購入する必要があります。ASE_HA ライセンスの有効化については、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。

高可用性システムの 2 つの Adaptive Server は、同じような、互換性のある設定にする必要があります。どちらも次の条件を満たす必要があります。

- Adaptive Server 15.0 以降が動作していること
- Open Client™ の最新バージョンが動作していること
- リリース・レベルが同じであること
- 構成に互換性があること
- コンポーネント統合サービス (CIS) が動作していること
- 高可用性システム (Sun Cluster など) が動作していること
- 並列処理または非並列処理に対応した設定になっていること

リソースの条件

Adaptive Server が高可用性システムでコンパニオンとして設定されていると、単独で機能する Adaptive Server とは異なるリソース条件があります。このような違いがあるのは、フェールオーバ時に「セカンダリ・コンパニオン」がすべての処理を実行するためです。コンパニオンを非対称型で設定していても、このことに変わりはありません。結果的に、高可用性システムの Adaptive Server のリソース条件は、単一のサーバの場合よりも厳しくなります。詳細については、「単一のシステムとしてアクセス可能なシステム」(16 ページ) を参照してください。

リソース条件の一部を次に示します。Adaptive Server をクラスタ・コンパニオンとして設定する場合にこれらの条件を考慮する必要があります (サイト自体にも、考慮する必要がある一連のリソース条件があります)。

- ログイン、役割、データベース － 1 つの Adaptive Server に対する総数と同じ数のログイン、役割、データベースの数をクラスタに設定します。
- number of user connections － システムで必要なユーザ接続の総数に対応するように、各コンパニオンを設定します。
- number of open databases － システムで必要なオープン・データベースの総数に対応するように、各コンパニオンを設定します。
- srids － システムで必要な srids の総数に対応するように、各コンパニオンを設定します。
- number of devices － 個々に使用するデバイスの数ではなく、クラスタで使用するデバイスの総数に対応するように、各 Adaptive Server を設定します。つまり、1 つのコンパニオンで 14 デバイスを使用していて、もう 1 つが 23 デバイスを使用している場合、number of devices として 37 を各 Adaptive Server に設定します。
- sp_configure オプションの number of open databases － フェールオーバ用に設定された Adaptive Server では、オープンしているデータベース数は 2 つ減らされます。これは、フェールオーバを正常に実行するためです。つまり、number of open databases を 10 に設定している場合は、8 つのデータベースだけを開くことができます。
- sp_configure オプションの number of user connections － フェールオーバ用に設定された Adaptive Server では、ユーザ接続の数は 2 減らされます。これは、フェールオーバを正常に実行するためです。つまり、number of user connections が 50 の場合、48 のユーザ接続だけを使用できます。

コンパニオン・サーバに接続するクライアント・アプリケーションでは、そのライブラリをフェールオーバ・ソフトウェアに含まれるライブラリに再リンクする必要があります。Open Client をフェールオーバとともに使用する方法の詳細については、「CTLIB アプリケーションの変更」(193 ページ) を参照してください。

Sybase のフェールオーバと高可用性の機能

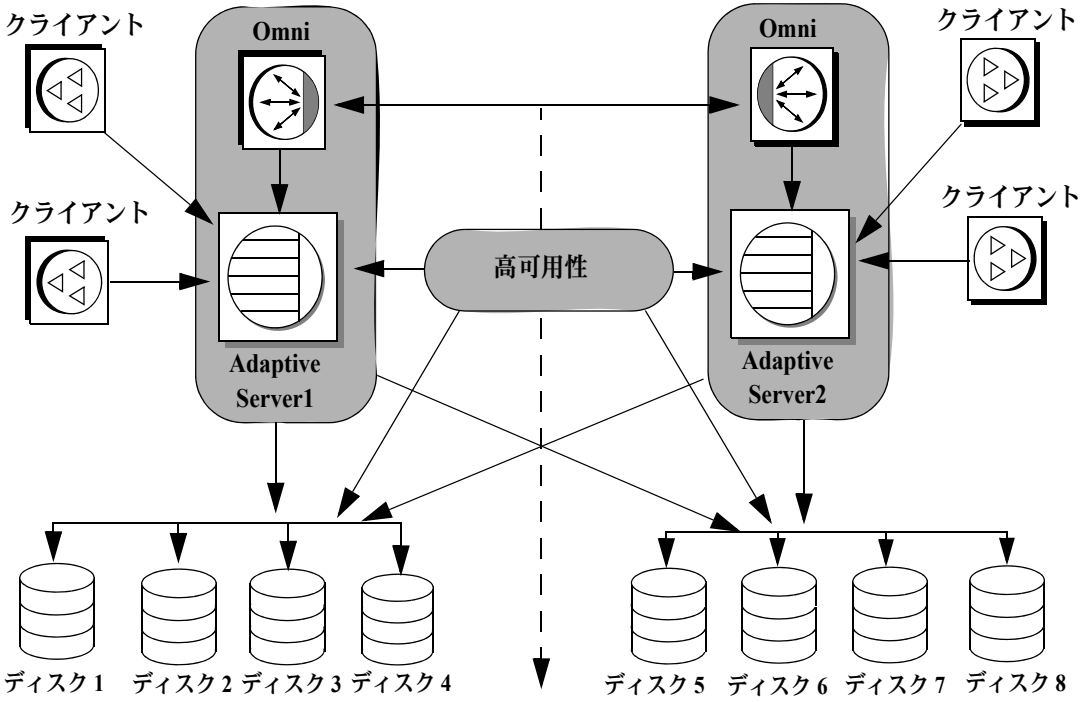
高可用性システムには、ハードウェアとソフトウェアの両方が含まれます。Sybase フェールオーバは、コンパニオン・サーバがクラスタでのシングル・ポイント障害に耐性を持つことができるようにするソフトウェアです。

Sybase フェールオーバを使用するシステムには、2 台のマシンが含まれます。各マシンは、高可用性「クラスタ」の「ノード」です。各 Adaptive Server は、「プライマリ・コンパニオン」と「セカンダリ・コンパニオン」のどちらかとなります。各コンパニオンは、稼働中に処理を実行します。プライマリ・コンパニオンに障害やダウンが発生した場合、セカンダリ・コンパニオンがその負荷を引き受けます。プライマリ・コンパニオンは、定期保守、システム障害、停電などによってダウンすることがあります。セカンダリ・サーバがもう一方のサーバの負荷を引き受ける場合、これを「フェールオーバ」といいます。障害が発生したサーバが再び正常に稼働するようになった後に、そのサーバに負荷を戻すことを、「フェールバック」といいます。

図 2-1 は、2 つの Adaptive Server からなる一般的な構成を示します。

オペレーティング・システムには、Sun の Sun Cluster などの高可用性システムが組み込まれています。高可用性システムは、システムの一部で障害が発生したり、保守のためにシステムがシャットダウンしたことを検出して、クラスタにその情報を伝達します。Adaptive Server がダウンした場合、高可用性システムは、負荷を引き受けるようにもう一方のマシンに通知します。障害の発生した Adaptive Server に接続していたクライアントは、もう一方の Adaptive Server に自動的に再接続されます。

図 2-1: Sybase フェールオーバを使用した高可用性システム



高可用性クラスタ

図 2-1 のマシンは、他のマシンのディスクを各マシンで読み込みできるように設定されています。ただし、同時読み込みはできません (フェールオーバで使用するすべてのディスクは共有ディスクに設定してください)。

たとえば、プライマリ・コンパニオンである Adaptive Server 1 で障害が発生した場合、セカンダリ・コンパニオンである Adaptive Server 2 は、Adaptive Server 1 がオンラインに戻れるようになるまでそのディスク (1 ～ 4) を読み込んで、ディスク上のデータベースすべてを管理します。Adaptive Server 1 に接続し、フェールオーバ・プロパティを使用しているすべてのクライアントは、自動的に Adaptive Server 2 に接続されます。

単一のシステムとしてアクセス可能なシステム

クラスタ・システムの優れた特長の 1 つは、2 つの Adaptive Server からなるシステムにログインしていることをユーザが意識しなくてもよいことです。ユーザは、単一のシステムにログインしてクラスタ上のすべてのデータベースにアクセスしているかのような印象を受けます。また、アプリケーションにとってもシステムは単一として存在します。ユーザやアプリケーションは、どちらかのコンパニオンにログインして、クラスタ上のすべてのデータベースにアクセスします。

しかし、システム管理者はシステムを 2 つの別個の Adaptive Server からなるものとして扱う必要があります。両方の Adaptive Server を個別にインストールして設定しますが、その構成は同じである必要はありません。個々の Adaptive Server とクラスタには、システムの保守が必要です。

Sybase フェールオーバに関する注意事項

Sybase のフェールオーバを設定する場合、この項で説明する Adaptive Server の機能については注意が必要です。

モニタリング・テーブル・スクリプトのインストール

モニタリング・テーブルを高可用性設定に追加する場合、プライマリ・コンパニオンとセカンダリ・コンパニオンのパフォーマンスをモニタするには、次のいずれかを両方のサーバの interfaces エントリに追加する必要があります。

```
loopback
master tcp ether localhost port_number
query tcp ether localhost port_number
```

または、

```
loopback
master tcp ether servername port_number
query tcp ether servername port_number
```

port_number は、プライマリ・コンパニオン上で開いている任意のポートです。

ディスク・ミラーリングの使用

Sybase フェールオーバと高可用性システムを使用すると、接続していたサーバで障害が発生している間もデータにアクセスできるようになります。しかし、どちらのシステムでもディスク障害が発生する可能性はあります。ディスク障害によってデータが消失しないようにするには、ディスク・ミラーリングや RAID などのデータ保護技術を Sybase フェールオーバとともに使用してください。

Sybase のディスク・ミラーリングは、Adaptive Server コンパニオン・クラスタではサポートされていません。また、sp_companion を発行して Adaptive Server をコンパニオンとして設定するときは使用できません。サードパーティ製のミラーリング・システムを使用して、ディスク・デバイスを保護してください。

installevent スクリプトの実行

フェールオーバに必要なストアド・プロシージャは、*installmaster* スクリプトには含まれていません。Adaptive Server のフェールオーバの設定に必要なストアド・プロシージャをインストールする *installhasvss* スクリプトを実行します。*installhasvss* は、*$SYBASE/$SYBASE_ASE/scripts* ディレクトリにあります。

注意　*installmaster* スクリプトを再実行する際は、*installhasvss* を再実行する必要があります (「installmaster の再インストール」および「installhasvss の再実行」(196 ページ) を参照してください)。

異なるバージョンの Adaptive Server の *installhasvss* スクリプトは使用しないでください。

詳細については、該当するプラットフォームの設定に関する章を参照してください。

SYB_HACMP サーバ・エントリの作成

installhasvss スクリプトによって、サーバの sysservers に SYB_HACMP という名前のエントリが作成されます。Adaptive Server をコンパニオンとして設定する前は、SYB_HACMP サーバ・エントリはローカル・サーバを指しています。SYB_HACMP sysservers エントリによって、プライマリ・コンパニオンとセカンダリ・コンパニオンが *interfaces* ファイル内のそれぞれのエントリを使用して通信できます。SYB_HACMP サーバ・エントリをコンパニオン・サーバとともに使用して、クエリやストアド・プロシージャを作成しないでください。

SYB_HACMP サーバ・エントリを削除しないでください。誤ってこのエントリを削除した場合は、*installmaster* と *installhasvss* を再実行する必要があります。

ユーザ定義データ型の定義

高可用性システムの Adaptive Server をプライマリ・コンパニオン・サーバとセカンダリ・コンパニオン・サーバに設定した後は、Java データ型またはユーザ定義データ型を含むテーブルを更新しても、更新した内容は同期されません。たとえば、プライマリ・コンパニオン上の pubs2 データベース内のテーブルに Java オブジェクトがカラム・データとして格納されている場合、このカラムを更新しても、プロキシ・テーブルにはその更新が反映されません。ユーザ定義データ型を格納しているカラムは、手動で更新する必要があります。

また、プライマリ・コンパニオン上の pubs2 データベースにユーザ定義データ型を使用するテーブルが格納されている場合、このテーブルを変更しても、セカンダリ・コンパニオンの pubs2 プロキシ・テーブルにはその更新は反映されません。

Adaptive Server と 2 フェーズ・コミット・トランザクション

Adaptive Server が Sybase フェールオーバを使用してコンパニオン・サーバとして設定されていると、Sybase 2 フェーズ・コミット (SYB2PC) トランザクションはサポートされません。

第 3 章

非対称型と対称型の設定

この章では、高可用性システムで Adaptive Server を非対称型と対称型に設定する方法について説明します。

トピック名	ページ
非対称型と対称型の設定	19
高可用性システムでの監査	22

非対称型と対称型の設定

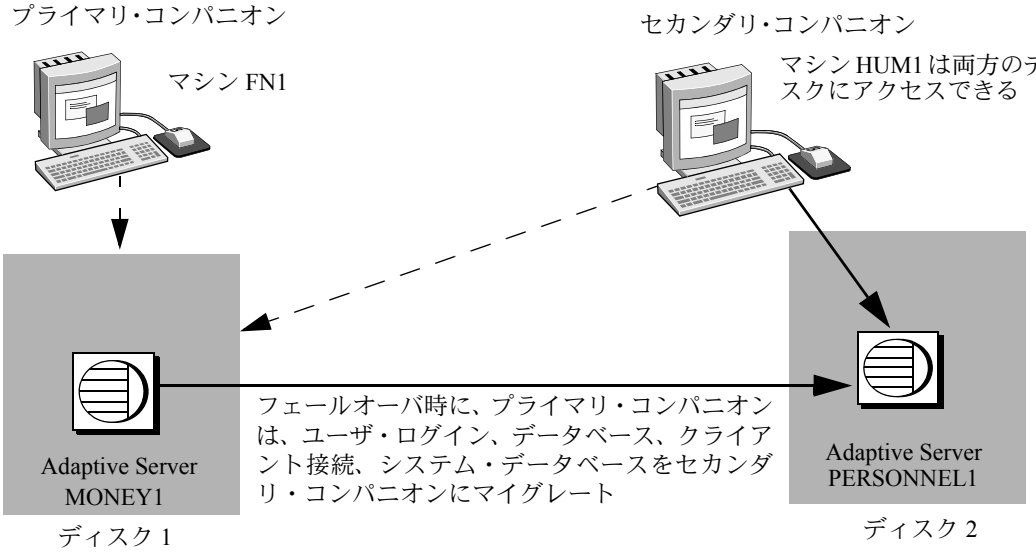
コンパニオン・サーバは、非対称型と対称型のいずれかに設定できます。コンパニオンを対称型に設定するには、その前に非対称型に設定しておく必要があります。

非対称型コンパニオンの設定

非対称型は、別々のマシンで動作している 2 つの Adaptive Server で構成されています。プライマリの Adaptive Server は日常的な業務で動作しますが、セカンダリの Adaptive Server はシステム障害や定期保守の間に動作を引き継ぐように設定されています。セカンダリ・コンパニオンは独立した Adaptive Server なので、単独でアプリケーションを動作させることができます。フェールオーバ用に設定するには、セカンダリ・コンパニオンが新規にインストールされた Adaptive Server である必要があり、またユーザ・ログインやユーザ・データベースを含ませることはできません。設定が完了したら、セカンダリ・コンパニオンにユーザ・ログインやデータベースを追加できます。

Adaptive Server をインストールしてフェールオーバ用に設定すると、Adaptive Server は**シングルサーバ・モード**になります。sp_companion を使用して、シングルサーバ・モードから非対称型設定のコンパニオン・サーバに変更します。sp_companion 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。

図 3-1: 高可用性システムにおける非対称型設定



この設定では、MONEY1 がプライマリ・コンパニオンで、セカンダリ・コンパニオンである PERSONNEL1 にフェールオーバしています。いずれのディスクも、デュアルポート SCSI の付いた FN1 のマシンに接続している HUM1, のマシンに表示されます。非対称型設定なので、PERSONNEL1 が MONEY1 にフェールオーバすることはできません。ディスク 1 は共有ディスクでなくてはなりませんが、ディスク 2 はローカル・ディスクでもかまいません。

Adaptive Server を非対称型に設定する方法の詳細については、該当するプラットフォーム固有の設定を解説した章を参照してください。

非対称型に設定した Adaptive Server のパフォーマンス

ノーマル・コンパニオン・モードでは、ユーザ情報を更新するシステム・プロシージャ (sp_addlogin や sp_addrole など) およびコマンド (create database など) のパフォーマンスが若干低下します。これは、プライマリ・コンパニオンはローカルでコマンドを実行した後にその情報をセカンダリ・コンパニオンと同期する必要があるためです。たとえば “joe” というユーザをプライマリ・コンパニオンに追加する場合、プライマリ・コンパニオンとセカンダリ・コンパニオンの両方で、この新しいユーザを含むように syslogins を更新する必要があります。

フェールオーバ後のパフォーマンスは、セカンダリ・コンパニオンの設定によって異なります。セカンダリ・サーバがプライマリ・サーバと同じように設定されている場合、フェールオーバの前後でパフォーマンスはほとんど同じです。しかし、セカンダリ・サーバのパフォーマンスがプライマリ・サーバほど高くない (たとえばメモリが少ない、CPU 速度が遅いなど) 場合、フェールオーバ後のパフォーマンスは低下します。また、フェールオーバ後は、プライマリ・コンパニオンおよびすべてのアプリケーションが動作するため、セカンダリ・コンパニオンのパフォーマンスが低下する場合があります。

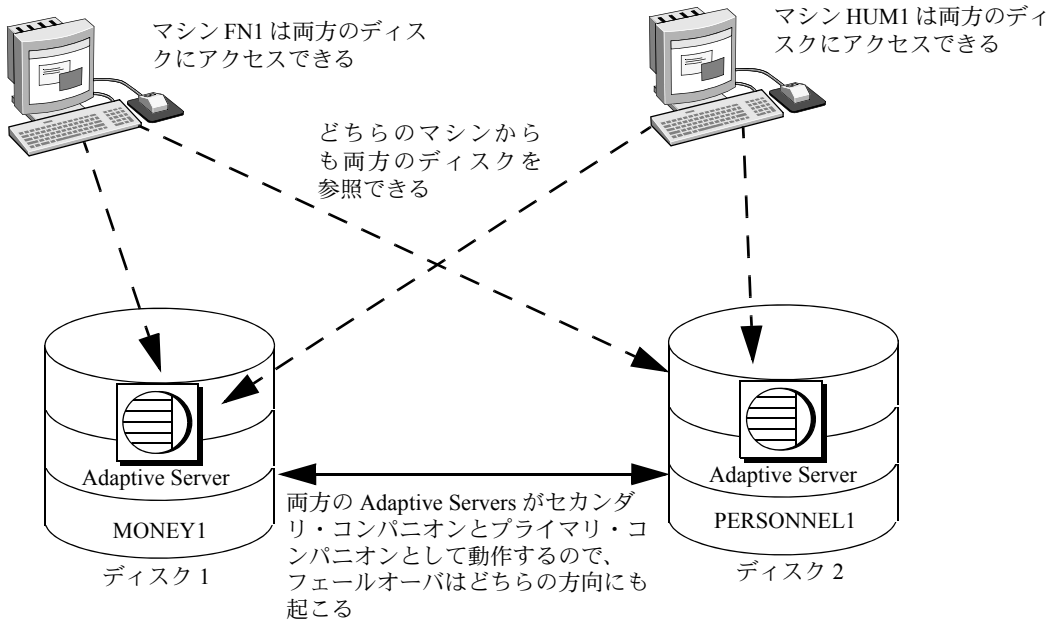
対称型コンパニオンの設定

対称型の設定も非対称型の設定と同様に、別々のマシンで動作している完全な機能を持つ 2 つの Adaptive Server で構成されており、それぞれがシステム・デバイス、システム・データベース、ユーザ・データベースおよびユーザ・ログインを持っています。しかしフェールオーバが発生すると、一方の Adaptive Server がもう一方の Adaptive Server のプライマリ・コンパニオンあるいはセカンダリ・コンパニオンとして動作します。

2 つの Adaptive Server を対称型のコンパニオンに設定する前に、まずそれらを非対称型のコンパニオンとして設定してください。

図 3-2 は、財務部のマシン (Adaptive Server MONEY1 が動作する FN1) と人事部のマシン (Adaptive Server PERSONNEL1 が動作する HUM1) の間での対称型設定を示しています。

図 3-2: 高可用性システムにおける対称型設定



定期保守やシステム障害時には、MONEY1 が PERSONNEL1 に、または PERSONNEL1 が MONEY1 にフェールオーバします。ディスク 1 とディスク 2 の両方が共有ディスクです。

Adaptive Server を対称型に設定する方法の詳細については、該当するプラットフォーム固有の設定について説明した章を参照してください。

対称型に設定した Adaptive Server のパフォーマンス

ノーマル・コンパニオン・モードでは、対称型に設定した両方の Adaptive Server をシステム・リソース容量いっぱいまで動作させないでください。たとえば、ユーザ接続、データ・キャッシュ、リモート・サーバ接続などには、各マシンの設定可能なリソースの 60% を動作させます。これによりセカンダリ・コンパニオンは、フェールオーバした Adaptive Server と自身の Adaptive Server の両方を、適切なレベルのパフォーマンスで動作させることができます。両方の Adaptive Server でシステム・リソースを最大限に使用していると、フェールオーバは正常に行われますが、パフォーマンスが低下する可能性があります。

高可用性システムでの監査

監査用コンパニオンの設定方法は、フェールオーバを使用しないサーバの設定方法と同じです。詳細については、「監査オプションの設定」(23 ページ) を参照してください。

ユーザ情報やセキュリティ情報 (たとえば sp_addlogin や sp_addrole など) の更新はすべて、両方のシステムにおいてトランザクション形式で実行されます。これにより、両方のコンパニオンのユーザ・データとセキュリティ・データが同じになります。

次の監査パラメータでは、両方のコンパニオンを同じ設定にする必要があります。これらのパラメータは、定属性として、または do_advisory で明示的にリストされている場合にチェックされます。

- allow procedure grouping
- unified login required
- secure default login
- systemwide password expiration
- use security services
- check password for digit
- minimum password length
- maximum failed logins

- auditing − このパラメータのオン／オフは、コンパニオンに対して動的に同期されない。このパラメータをローカルで変更した場合は、リモート・コンパニオンを手動で変更する必要がある。

監査オプションの設定

コンパニオン・サーバの監査オプション (グローバル、データベース全体およびログイン単位) は、ノード単位で設定できます。つまり、各コンパニオンに独自の監査設定ができます。グローバル・オプションは、コンパニオン間では同期していません。

フェールオーバ中は、データベース全体のオプションは現在の設定どおりに監査されます。

フェールオーバ後は次のようになります。

- 監査により、グローバル・オプションが適用され、データベース全体のオプションは引き続きフェールオーバ前と同じように動作する。
- ユーザは、引き続きデータベース全体のオプションを設定できる。
- ローカル・ドメインの監査オプションは、ローカル・ログインとリモート・ログインの両方に使用される。

sybsecurity と Sybase フェールオーバ

sybsecurity データベースは、監査インストール中に、*installsecurity* によって作成されます。Sybase フェールオーバの初期設定時にいずれかのコンパニオンに sybsecurity データベースが存在する場合、このデータベースは両方のコンパニオンに存在する必要があります。

監査証跡と Sybase フェールオーバ

監査証跡は、sybsecurity データベースの監査テーブルに記録されます。フェールオーバ中は、障害が起こったサーバの sybsecurity は、sybsecurity_companion としてセカンダリ・コンパニオン上にマウントされます。ただし、監査証跡は常に現在のサーバの監査テーブルに配置されます。したがって、フェールオーバ後の新しい監査証跡はすべてセカンダリ・コンパニオンの監査テーブルに配置されます。すべての設定、または一方のコンパニオンで行われた監査に関連する個々のレコードの変更は、自動的にもう一方のコンパニオンに適用されるわけではありません。たとえば、プライマリ・コンパニオンで監査設定パラメータを変更しても、その変更はセカンダリ・コンパニオンには適用されません。ユーザがプライマリ・コンパニオン上で監査レコードを必要とするデータベースの変更を行っても、その監査レコードはセカンダリ・コンパニオン上には作成されません。

フェールバック時は、監査証跡はフェールオーバ先のドメインから失敗したドメインへは転送されません。

第 4 章

フェールオーバのモード

この章では、Sybase フェールオーバが処理中に切り替えるモードについて説明します。

トピック名	ページ
モードの概要	25
ドメイン	29

モードの概要

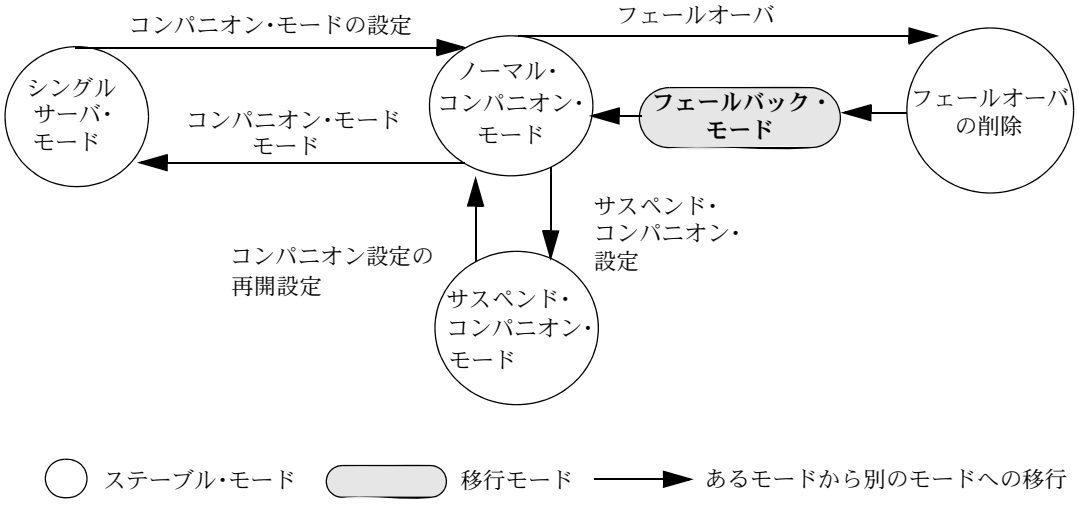
高可用性には、Adaptive Server が実行される一連のモードがあります。モードには、ステーブルと移行の 2 種類があります。「ステーブル・モード」は、日々の操作のような、長期間 Adaptive Server が存続できるシステム・ステータスです。

ステーブル・モードには、次のものがあります。

- シングルサーバ・モード
- ノーマル・コンパニオン・モード
- **フェールオーバ・モード**
- サスペンド・コンパニオン・モード

フェールバック・モードは、Adaptive Server がフェールオーバ・モードからノーマル・コンパニオン・モードに移行するときに発生する**移行**モードです。通常は非常に短時間です。プライマリ・コンパニオンがモードを変更するときの処理を、図 4-1 に示します。

図 4-1: 高可用性を実現する処理モード



2 つの Adaptive Server をコンパニオンとして設定するには、どちらともシングルサーバ・モードにしてから行います。シングルサーバ・モードは、*installhasvss* を実行したあとに新しくインストールされる Adaptive Server のデフォルト・モードです。Adaptive Server をコンパニオンとして設定すると、次の 3 つのステーブル・モードのうちのいずれかになります。

- ノーマル・コンパニオン・モード
- フェールオーバ・モード
- サスペンド・コンパニオン・モード

コンパニオン・サーバの各モードの詳細

この項では、各モードについて詳しく説明します。

シングルサーバ・モード

このモードでは、Adaptive Server はスタンドアロン・サーバとして動作します。インストールした直後の Adaptive Server は、デフォルトでシングルサーバ・モードとなっています。

ノーマル・コンパニオン・モード

両方のコンパニオンが実行中で、フェールオーバ用に設定されている場合、どちらもノーマル・コンパニオン・モードで動作します。これは Adaptive Server が日々の操作を行うモードです。**非対称型**のシステムでは、プライマリ・コンパニオンがセカンダリ・コンパニオンにフェールオーバできます。**対称型**のシステムでは、どちらのコンパニオンも、もう一方のコンパニオンにフェールオーバできます。

サスペンド・モード

サスペンド・モードでは、サーバはどちらともシングルサーバとして動作します。サスペンド・モードは、Adaptive Server とその関連リソースをフェールオーバせずに開始したり停止したりできるので、システムの保守を実行するのに便利です。

コンパニオンがフェールオーバできなくても、コンパニオンが動作しているノードはフェールオーバできるので、ノードのフェールオーバをサスペンドするには、プラットフォーム固有の手順を実行してください。サスペンド・モードになっているコンパニオンの停止も、プラットフォーム固有のタスクを実行してから行います。詳細については、該当するプラットフォームの章を参照してください。

サスペンド・モードでは、多くのユーティリティとコマンドの使用が厳しく制限されています。詳細については、「付録 C コマンド、システム・プロシージャ、データベースの変更」を参照してください。

注意　コンパニオン・モードのサスペンドは、セカンダリ・コンパニオンからのみ行ってください。

フェールバック・モード

Adaptive Server がフェールバック・モードに入らなければ、セカンダリ・コンパニオンのフェールオーバ・モードからプライマリ・コンパニオンのノーマル・コンパニオン・モードに移行することはできません。

フェールバックは予定されたイベントです。つまり、システムが通常のオペレーションを再開できるとシステム管理者が判断したときにのみ行われます。sp_companion prepare_failback を使用して、フェールバックを開始し、フェールオーバした Adaptive Server を本来のノードにマイグレートします。詳細については、「フェールバックの実行」(5 ページ) を参照してください。

サスペンド・モードからのノーマル・コンパニオン・モードの再開

ノーマル・コンパニオン・モードを再開するには、次のように入力します。

```
sp_companion "primary_server_name", resume
```

フェールオーバ・モードの削除

コンパニオン・モードを永久に無効にするには、次のように入力します。

```
sp_companion ìserver_nameî, 'dropí
```

このコマンドが完了すると、2 つの Adaptive Server はコンパニオン・サーバではなくなり、シングルサーバ・モードで動作します。

注意　drop 処理は、元に戻すことができません。一度コンパニオン・サーバをシングルサーバ・モードに戻したら、全ユーザ・データベースをダンプ、削除、再ロードして、コンパニオンとして再設定してください。

削除するコンパニオンが対称型の設定の場合、クラスタは自動的にコンパニオン間に非対称型の設定を行います。

コンパニオンのモードの確認

オプションを指定しないで sp_companion を発行すると、次のようにコンパニオンの現在のモードを表示できます。

```
sp_companion
Server 'MONEY1' is alive and cluster configured.
Server 'MONEY1' is configured for HA services.
Server 'MONEY1' is currently in 'Symmetric normal' mode.
```

MONEY1 は対称型のフェールオーバに設定され、ノーマル・コンパニオン・モードで動作しています。*@@cmpstate* グローバル変数を使用してモードを確認することもできます。isql プロンプトで、次のように入力します。

```
select @@cmpstate
```

注意 *@@cmpstate* グローバル変数は、非高可用性環境では使用されず、-2 という値が返されます。

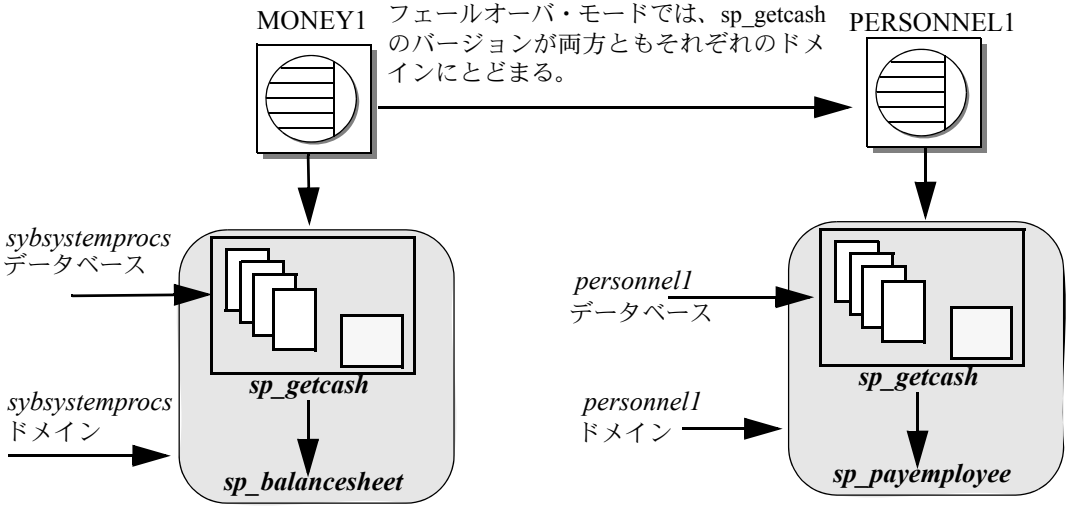
表 4-1: @@cmpstate が返す値

@@cmpstate	コンパニオン・モード
0	シングルサーバ
1	予約済み
2	セカンダリのノーマル
3	セカンダリのサスペンド
4	セカンダリのフェールオーバ
5	セカンダリのフェールバック
6	予約済み
7	プライマリのノーマル
8	プライマリのサスペンド
9	プライマリのフェールバック
10	予約済み
11	対称型ノーマル
12	対称型フェールオーバ
13	対称型サスペンド
14	対称型フェールバック
15	予約済み

ドメイン

プライマリとセカンダリのコンパニオンは、同じ名前のストアド・プロシージャ、ユーザ、デバイスを持つことができます。フェールオーバ用に設定された Adaptive Server は「ドメイン」を使用して、これらのオブジェクトが属するデータベースを決定します。たとえば、Adaptive Server、MONEY1 と PERSONNEL1 の両方に、sp_getcash という名前のストアド・プロシージャがあるとします (図 4-2 参照)。

図 4-2: フェールオーバ中のドメイン



MONEY1 では、sp_getcash は、sybsystemprocs ドメインに定義されています。この sp_getcash は sp_balancesheet という名前のセカンダリ・ストアド・プロシージャを発行します。PERSONNEL1 では、sp_getcash は、personnel ドメインに定義されています。この sp_getcash は sp_payemployee という名前のセカンダリ・ストアド・プロシージャを発行します。フェールオーバの間、MONEY1 の sybsystemprocs は PERSONNEL1 に sybsystemprocs_companion としてマイグレートしますが、ドメイン自体は変更せず、このドメインに定義されたオブジェクトも変更しません。フェールオーバの間、MONEY1 の sybsystemprocs にある sp_getcash を発行するユーザも、正しいセカンダリ・ストアド・プロシージャの sp_balancesheet を発行していることになります。

ドメインの概念はユーザにとって透過的です。

master データベースに格納されたシステム・プロシージャは、ドメインでは管理しません。システム・プロシージャが、master データベースに格納されたオブジェクトに依存しないようにしてください。

第 5 章

プロキシ・データベース、ユーザ・データベース、プロキシ・システム・テーブル

トピック名	ページ
プロキシ・データベース	31
master のプロキシ・システム・テーブル	36

プロキシ・データベースとプロキシ・テーブルの詳細については、『コンポーネント統合サービス・ユーザーズ・ガイド』を参照してください。

プロキシ・データベース

Adaptive Server を非対称型設定のコンパニオンとして設定した場合、デフォルトではプロキシ・データベースは作成されません。フェールオーバを sp_companion の with_proxydb オプションを使用するように設定した場合にのみ、リモート・サーバに作成されます。この章では、sp_companion を with_proxydb オプションを指定して使用したと仮定して、説明を進めます。**プロキシ・データベース**は必要に応じて動的に作成されます。『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。

対称型設定に対しては with_proxydb を使用しないでください。

コンパニオン・サーバのデータベースは、プライマリ・データベースかプロキシ・データベースのどちらかになります。プライマリ・データベースとは、データが物理的に存在する場所のことです。各プロキシ・データベースは、プライマリ・データベース 1 つと対応しており、名前はプライマリ・データベースと同じで、プライマリ・データベースのすべてのオブジェクトに対してプロキシ・エントリを持っています。ただし、データは入っていません。

フェールオーバ用にコンパニオン設定をして、プロキシ・データベースを作成すると、どちらのコンパニオンからでもユーザ・データベースを参照できます。つまり、どちらのコンパニオンからでも、プライマリ・データベースでトランザクションを実行できます。たとえば、PERSONNEL1 というプライマリ・コンパニオンに salary というデータベースがある場合、セカンダリ・コンパニオンである MONEY1 には salary というプロキシ・データベースが含まれます。MONEY1 と PERSONNEL1 のどちらからでも、salary に対して挿入、更新、削除処理ができます。また、各コンパニオンの sysdatabases に、salary データベースがリストされます。たとえば、次のクエリは、PERSONNEL1 と MONEY1 で同じ結果を生成します。

```
1> select name from sysdatabases
 name
 ------------------------------
 master
 model
 salary
 sybsystemdb
 sybsystemprocs
 tempdb
```

注意 プライマリ・コンパニオンがフェールオーバするとき、セカンダリ・サーバ上のプロキシ・データベースへの現在の接続はすべて終了され、切断されます。フェールバック時、セカンダリ・コンパニオンは逆にプライマリ・データベースをマウントし、プロキシ・データベースを再作成します。

プロキシ・データベースの作成

Adaptive Server はコンポーネント統合サービス (CIS) を使用して、プロキシ・データベースを作成します。プライマリとセカンダリのどちらの Adaptive Server でも、CIS を実行してから Sybase フェールオーバを設定してください。

CIS は、プライマリ・データベースに存在するカラムのデフォルトをプロキシ・テーブルにインポートしません。

CIS が実行中かどうかを確認するには、次のように入力します。

```
sp_configure "enable cis"

 Parameter Name      Default    Memory Used  Config Value  Run Value
 ------------------  ---------  -----------  ------------  ---------
 enable cis               1            0            1              1
```

Run Value の 1 は、CIS が実行中であることを示します。

CIS 用に Adaptive Server を設定する方法の詳細については、『コンポーネント統合サービス・ユーザーズ・ガイド』を参照してください。

CIS は次の手順でプロキシ・データベースを作成します。

1　サイズまたはデータベース・デバイスが指定されなかった場合は、すべてのプロキシ・テーブルを格納するために必要なデータベースのサイズを推測します。

2　すべてのプロキシ・テーブルを作成します。プロキシ・テーブルは、プライマリ・コンパニオン・データベースで見つかったテーブルおよびビューのプレースホルダとして機能します。

3　プライマリ・コンパニオンからカラム名、サイズ、インデックスなどのメタデータをインポートします。

4　プロキシ・テーブルのすべてのパーミッションを public にします。

5　プロキシ・データベースに、ユーザ guest を追加します。

6　プロキシ・データベースであることを示すように、データベースのステータスを設定します。ステータスは、sysdatabases の status3 カラムに示されます。sp_helpdb には、データベースがプロキシなのかプライマリなのかの情報が含まれています。

プロキシ・データベースの作成されるタイミング

sp_companion...with_proxydb オプションを使用してコンパニオンを設定すると、次のようになります。

- コンパニオンの設定をしたときに、プライマリ・コンパニオンのすべてのユーザ・データベースについてプロキシ・データベースが作成されます。
- create database を使ってプライマリ・コンパニオンにユーザ・データベースを新しく作成すると、そのプロキシ・データベースが作成されます。
- フェールオーバ時、セカンダリ・コンパニオンはプライマリ・データベースをマウントし、プロキシ・データベースを削除します。フェールバック時、セカンダリ・コンパニオンは逆にプライマリ・データベースをマウントし、プロキシ・データベースを再作成します。

プロキシ・データベースのサイズ

Adaptive Server はプロキシ・データベースを作成するときに、プライマリ・データベース内のテーブルとビューの数を確認し、プロキシ・データベースに同数のプロキシ・テーブルを作成するのに必要な領域を計算します。プロキシ・テーブルごとに、8 ページ (1 エクステント) が必要です。プロキシ・テーブルの各インデックスも 8 ページを必要とします。また、Adaptive Server は、テーブルの拡大に対応するため、領域の 10 % または 500 ページのどちらか大きい方をデータベースに追加します。

このため、プロキシ・データベースのサイズは、プライマリ・データベース内のテーブルとビューの数によって変わります。プロキシ・データベースにはデフォルトのサイズはありません。最小サイズは、少なくとも model データベースのサイズとなります。

プロキシ・データベースでのコマンドとシステム・プロシージャ

コマンドとシステム・プロシージャには、プロキシ・データベースで発行した場合に異なる動作をするものがあります。

プロキシ・データベースでのコマンドの相違点

ほとんどのコマンドはプライマリとプロキシのどちらのデータベースからでも発行でき、プライマリ・データベースだけが更新されます。次のコマンドはプロキシ・データベースからは発行できません。

- create または drop procedure
- create または drop view
- create または drop trigger
- create または drop rule
- create または drop default

データベース・コマンド dump と load は、プライマリ・コンパニオンから実行してください。コマンドをプロキシ・データベースから発行すると、プロキシ・データベースだけが更新され、プライマリ・コンパニオンは更新されません。

プロキシ・データベースでのシステム・プロシージャの相違点

システム・プロシージャは、タスクを常にローカルに実行します。つまり、プロキシ・データベースでシステム・プロシージャを発行すると、その変更はプライマリ・データベースには反映されません。逆の場合も同じです。

プロキシ・データベースでのユーザ定義ストアド・プロシージャの発行

ユーザ定義ストアド・プロシージャは、常にプライマリ・データベース内でタスクを実行します。たとえば、user_created_proc は、pubs2 プライマリ・データベースから発行しても、pubs2 プロキシ・データベースから発行しても、pubs2 プライマリ・データベースで実行されます。

プロキシ・データベースから発行されるシステム・プロシージャは、次の条件に基づいて処理されます。

- ユーザ定義ストアド・プロシージャを高可用性システムのプロキシ・データベースで実行するという要求が、リモート・プロシージャ・コール (RPC) 要求に変換され、元のデータベースを所有するサーバに送信されます。
- システム・プロシージャの場合は、探索規則 (現在のデータベース、sybsystemprocs、master の順でプロシージャを検索するなど) が呼び出されます。プロシージャが検出されない場合は、要求がリモート・プロシージャ・コール (RPC) に変換され、元のデータベースを所有するサーバへ転送されます (ユーザ定義ストアド・プロシージャの場合と同じです)。
- CIS はまずローカル・サーバ内で、そのシステム・プロシージャを探します。見つかった場合、ローカルのストアド・プロシージャとして実行されます。
- システム・プロシージャがローカルで見つからない場合、RPC としてプライマリ・コンパニオンへ転送されます。
- ユーザ定義ストアド・プロシージャの場合、RPC としてプライマリ・コンパニオンへ転送されます。

この動作が適用されるのは、「システム」プロキシ・データベース、つまり、高可用性設定で自動作成されるプロキシ・データベースのみです。ユーザ・プロキシ・データベースではこのような動作はありません。

コンパニオンとして設定したサーバから発行されたシステム・プロシージャは、シングルサーバの場合と同じ規則に従って処理されます。システム・プロシージャがどのように処理されるかについては、『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_dboption がプロキシ・データベースを更新しない場合

sp_dboption を使ってプライマリ・データベースのデータベース・オプションを変更しても、その変更はセカンダリ・コンパニオンのプロキシ・データベースに自動的には転送されません。プロキシ・データベースにも、sp_dboption を設定する必要があります。

たとえば、sp_dboption を使用して、プライマリ・コンパニオン上に select into bulkcopy/pllsort があるように pubs2 データベースを変更しても、セカンダリ・コンパニオン上の pubs2 プロキシ・データベースには変更は設定されません。

手動でのプロキシ・データベースの更新

alter database に for proxy_update オプションを指定すると、プロキシ・データベースをそのプライマリ・データベースと手動で同期できます。

このコマンドは、master データベースから発行する必要があります。

```
alter database <dbname>
    [existing options]
    [for proxy_update]
```

for proxy_update は、プロキシ・データベースに自動的にマイグレートされていないプライマリ・データベースへの変更を同期させるのに便利です。たとえば、sp_rename を使用してプライマリ・データベースの名前を変更しても、プロキシ・データベースの名前は自動的には変更されません。しかし、データベースの名前を変更した後 alter database... for proxy_update を発行すれば、プロキシ・データベースは新しいデータベース名で再構築されます。

alter database pubs2 for proxy_update のように、他のオプションを指定しないで for proxy_update を入力すると、データベースのサイズは拡張されません。その代わり、プロキシ・テーブルはプロキシ・データベースから削除され、プライマリ・コンパニオン・データベースのメタデータを使用して再作成されます。

alter database を使用してデータベースのサイズを拡張する場合は、サイズの拡張後にプロキシ・テーブルの更新が行われます。

for proxy_update はクラスタ環境にあるプライマリ・コンパニオンだけでなく、すべての外部データ・ソースでサポートされています。また、手動で更新できるようにするためにデータベースを for proxy_update 句で作成する必要もありません。create database コマンドか sp_defaultloc のどちらかを使用してデフォルトの記憶領域のロケーションを指定すると、プライマリ・コンパニオンのメタデータを、リモートの記憶領域のロケーションのメタデータと同期させることができます。

『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。

master のプロキシ・システム・テーブル

プロキシ・システム・テーブルによって、セカンダリ・コンパニオンはプライマリ・コンパニオンのシステム・テーブルにアクセスできます。sysobjects のプロキシ・システム・テーブルには 1 エクステントが割り付けられています。プロキシ・システム・テーブルは削除できません。プロキシ・システム・テーブルは次の命名構文を使用しています。

```
rmt_ha_system_table_name
```

第 6 章

do_advisory の実行

この章では、do_advisory オプションを指定して sp_companion を実行する方法について説明します。

トピック名	ページ
do_advisory オプションの概要	37
定属性	42

do_advisory オプションの概要

フェールオーバ・モードからノーマル・コンパニオン・モードへの変更などのクラスタ・オペレーションを実行する場合、クラスタ・オペレーションの実行を妨げる属性がコンパニオンのどちらかに設定されていることがあります。たとえば、セカンダリ・コンパニオンに設定されているスタックのサイズが小さすぎて、フェールオーバ・モード時に両方のコンパニオンを動作させる余裕がなかったり、コンパニオンに設定されている言語が異なっていたりすることがあります。

このような問題を防ぐために、sp_companion には do_advisory オプションがあります。このオプションによって、各コンパニオンの数百個の属性設定がチェックされます。クラスタ・オペレーションが正常に実行できないような設定に対しては、警告が発行されます。両方のコンパニオン属性に同じ値を設定する必要はありません。しかし、多くの属性では、コンパニオン間で互換性のある値を設定する必要があります。sp_companion...do_advisory を実行しても属性は変更されませんが、発生する可能性のある問題について警告が発行されます。

sp_companion...do_advisory は自動的にはトリガされません。定期的に実行して、コンパニオン間に互換性の問題がないことを確認してください。

do_advisory によって、調べる属性を指定できます。すべての属性を指定することも、属性のサブセットを指定することもできます。

サブセットは、「グループ属性」、「基本属性」、「定属性」から選択できます。グループ・サブセットには、すべてのログイン属性や領域属性などサーバのあらゆる設定が含まれます。基本サブセットには、ユーザ・ログインや CIS の設定など、グループ・サブセット内の特定の設定が含まれます。クラスタ・オペレーションの実行を妨げる可能性のある、指定したサブセット内の属性だけが do_advisory によって報告されます。

定属性とは、グループ属性または基本属性が指定されているか否かに関わらず、sp_companion を実行するたびにチェックされる設定パラメータです。正常なクラスタ・オペレーションの実行を妨げるような設定が定属性にあることを、sp_companion が検出した場合、このコマンドの実行は失敗します。詳細については、「定属性」(42 ページ) を参照してください。

各グループを構成するサーバ設定について、次に説明します。

- アプリケーション・グループ － ローカル・コンパニオンで実行するアプリケーションの設定がリモート・コンパニオンと互換性があるかを確認します。アプリケーション・グループには、次のものがあります。
  - Charsets － セカンダリ・コンパニオンに設定されている文字セットに、プライマリ・コンパニオンに設定されているすべての文字セットが含まれているかを確認します。
  - Java Archives － プライマリ・コンパニオンの Java アーカイブに、セカンダリ・コンパニオンと同じ名前とクラス定義が設定されていることを確認します。プライマリ・コンパニオンの Java アーカイブに属しているクラス定義は、セカンダリ・コンパニオンの同じ Java アーカイブにも属している必要があります。

    注意 アーカイブは自動的に同期されません。Java について 1 つのコンパニオンを設定したら、手動でもう 1 つのコンパニオンを設定してください。

  - Languages － セカンダリ・コンパニオンに設定されている言語に、プライマリ・コンパニオンに設定されているすべての言語が含まれているかを確認します。
  - Remote servers － プライマリ・コンパニオンのアプリケーションが使用しているリモート・サーバ・エントリが、セカンダリ・コンパニオンでも同じであることを確認します (エントリがセカンダリ・コンパニオンに存在する場合)。これにより、コンパニオンによって使用されるサーバ名と関連するサーバ ID がユニークであり、クラスタ内で一貫性が保たれていることを確認できます。

    すべてのデフォルトのサーバ・エントリ (SYB_BACKUP、ローカル・サーバ名、コンパニオン・サーバ名、SYB_HACMP、ローカル XP Server、コンパニオン XP Server など) は、自動的に同期されます。
  - Sort order － セカンダリ・コンパニオンに設定されているソート順に、プライマリ・コンパニオンに設定されているすべてのソート順が含まれているかを確認します。
  - Time ranges － プライマリ・コンパニオンによって定義され使用されている時間範囲定義が、セカンダリ・コンパニオンでも同じであることを確認します (定義がセカンダリ・コンパニオンに存在する場合)。

- User types − プライマリ・コンパニオンのアプリケーションが使用している master データベース内にあるすべてのユーザ定義データ型の定義が、セカンダリ・コンパニオンでも同じように定義されていることを確認します (定義がセカンダリ・コンパニオンに存在する場合)。

- 設定グループ − 設定ファイルに定義されている設定パラメータ間の互換性をチェックします (設定ファイルは、*$SYBASE/server_name.cfg* にあります)。Adaptive Server をコンパニオンとして設定しても、設定オプションは自動的に同期されません。設定グループには、次の基本属性があります。
  - CIS − CIS が正常にクラスタ・オペレーション対応に設定されていることを確認します。
  - DTM − コンパニオン間で、分散トランザクション・マネージャのパラメータに互換性があることを確認します。
  - Disk i/o − コンパニオン間で、ディスク設定 (disk i/o structures、allow sql server async i/o など) に互換性があることを確認します。
  - ESP − コンパニオン間で、拡張ストアド・プロシージャに互換性があることを確認します。
  - Errorlog − コンパニオン間で、エラー・ログ情報 (event logging、event log computer name など) に互換性があることを確認します。
  - General config − 設定ファイル内のすべての一般設定パラメータが、クラスタ・オペレーション対応に設定されていることを確認します。
  - Java − 両方のコンパニオンで、Java の実行が可能か不可能かを確認します。
  - Languages − 両方のコンパニオンに同じ言語、文字セット、ソート順が設定されていることを確認します。
  - Network − コンパニオン間で、ネットワーク関連パラメータ (allow remote access、default network packet size など) に互換性があることを確認します。
  - Parallel − コンパニオン間で、並列設定パラメータ (max parallel degree、memory per worker process など) に互換性があることを確認します。
  - Q Diag − コンパニオン間で、Q 診断属性 (autostart collector、sql text pipe active など) に互換性があることを確認します。
  - Security − コンパニオンのセキュリティ設定 (auditing、allow procedure grouping など) に互換性があることを確認します。
- データベース・グループ − コンパニオン間で、データベース属性に互換性があることを確認します。データベース・グループには、次のものがあります。

- Unique dbid − プライマリ・コンパニオンのデータベース ID がセカンダリ・コンパニオンで使用されていないことを確認します。

  注意 ユーザ・データベース ID がセカンダリ・コンパニオンのシステム・データベース ID と重複している場合、セカンダリ・コンパニオンのシステム・データベースをいったん削除して再作成する必要があります。

- デバイス・グループ − コンパニオン間で、デバイス属性に互換性があることを確認します。デバイス・グループには、次のものがあります。
  - Devnames − プライマリ・コンパニオンの論理デバイス名がセカンダリ・コンパニオンで使用されていないことを確認します。
- ログイン・グループ − プライマリ・コンパニオンとセカンダリ・コンパニオン間で、ログインとパーミッションに一貫性があることを確認します。
  - プライマリ・コンパニオンで定義されているすべてのユーザ情報 (ログインやパーミッションなど) は、セカンダリ・コンパニオンでも定義され、使用可能な状態であり、互換性がある必要があります (ユーザ情報がセカンダリ・コンパニオンに存在する場合)。プライマリ・コンパニオンのログインについて調べて、ログイン名がユニークであることと suids がセカンダリ・コンパニオンにも設定されていることを確認します。また、ログイン・グループは、リモート・ログイン、外部ログイン、および master 内のエイリアスとユーザ名がコンパニオン間で互換性があることも確認します。do_advisory によって、検出された問題が自動的に修正されます。
  - probe、qcollector、qrepository など、デフォルトのログインの非互換性は自動的に修正されます。
- 役割グループ − プライマリ・コンパニオンとセカンダリ・コンパニオン間で、すべてのユーザ定義の役割、ログインの役割、サーバ全体のパーミッションに互換性があることを確認します。
- 領域グループ − フェールオーバ時にプライマリ・コンパニオンのデータベースを格納できる十分な領域がセカンダリ・コンパニオンにあることを確認します。
  - Master Space − コンパニオン・サーバの初期設定中や sp_companion...resume の実行中にメタデータを同期させるのに必要な容量を計算します。
  - Proxydb Space − プロキシ・データベースの作成 (非対称型の設定で、コンパニオン・サーバを with_proxydb で設定する場合) に必要な容量を見積もります。

do_advisory オプションの実行

sp_companion do_advisory の構文は、次のようになります。

```
sp_companion server_name, do_advisory [, all | help |
    group_attribute_name | base_attribute_name]
```

各パラメータの意味は次のとおりです。

- *server_name* は、リモートの Adaptive Server の名前です。
- all は、グループ属性と基本属性の両方の情報が含まれることを示します。
- help は、sp_companion do_advisory 構文およびグループ属性と基本属性のリストを印刷します。
- *group_attribute_name* は、sp_companion を使用してレポートするグループ属性の名前です。
- *base_attribute_name* は、sp_companion do_advisory を使用してレポートする基本属性の名前です。

sp_companion do_advisory の出力には、次のものが含まれます。

- Attribute name － sp_companion do_advisory で調べる属性の名前です。
- Attribute type － 属性のタイプです。たとえば、CIS、disk i/o、General Config のタイプが表示されることがあります (これらは、*server_name.cfg* ファイルで設定されている設定パラメータです)。
- Local value － sp_companion do_advisory を入力したコンパニオンの属性の値です。
- Remote value － リモート・コンパニオンの属性の値です。
- Advisory － 2 つのコンパニオンの属性を比較した後に、sp_companion do_advisory によって結果が Advisory カラムに出力されます。カラムの値は、次のようになります。
  - 0 － 属性はクラスタ・オペレーションに影響しません。
  - 1 － 属性は最適に設定されていませんが、クラスタ・オペレーションを妨げることはありません。
  - 2 － クラスタ・オペレーションを実行するには、属性を変更する必要があります。

次の例では、Adaptive Server である MONEY1 と PERSONNEL1 の間の属性をチェックします。

```
sp_companion "MONEY1", do_advisory, 'all'
go

Attribute Name   Attrib Type  Local Value  Remote Value  Advisory
--------------   -----------  -----------  ------------  ------
cis connect time  CIS         1            0             2
cis rpc handling  CIS         1            0             2
```

```
max cis remote se CIS         10           25           2

(1 row affected)
(return status = 0)
```

この例では、属性 cis connect、cis rpc handling、max cis remote servers のいずれも、Advisory カラムに値 2 が表示されています。これは、これらの属性が、MONEY1 と PERSONNEL1 の間の正常なコンパニオン設定を妨げることを示します。これら 3 つの属性の Local Value カラムの値は、Remote Value カラムの値と異なります。コンパニオンは、同じ値または互換性のある値を持つ必要があります。

定属性

configure オプションまたは resume オプションとともに sp_companion を発行すると、sp_companion によって特定の属性グループがチェックされ、コンパニオンに互換性のある値が設定されていることが確認されます。これらの属性を、定属性といいます。コンパニオンの 1 つに他のコンパニオンと互換性のない定属性の値がある場合、sp_companion の実行は失敗します。

定属性が原因で sp_companion が正常に完了しなかったというメッセージが発行された場合は、sp_companion... do_advisory を実行して、問題のある属性のリストを表示してください。do_advisory によって、次の設定パラメータが定属性としてチェックされます。

- enable cis
- cis packet size
- max cis remote connections
- max cis remote servers
- number of devices
- esp execution stack size
- start mail session
- xp_cmdshell context
- default character set id
- default language id
- default sortorder id
- disable character set conversions
- enable repagent thread
- allow backward scans

- allow netsted triggers
- allow resource limits
- parition groups
- size of auto identity columns
- SQL perform integration
- cfg read committed with lock
- enable Java
- enable DTM
- number of DTX participants
- strict dtm enforcement
- allow remote access
- default network packetsize
- max network packetsize
- max parallel degree
- number of remote logins
- number of remote sites
- max parallel degree

do_advisory によって、次のデータベース属性もチェックされます。

- Charsets
- Java archives
- Languages
- Remote servers
- Sort order
- Time ranges
- User types
- Unique dbid
- Devnames
- Logins
- Roles

第 7 章

HP システムでフェールオーバを使用できるように Adaptive Server を設定する

この章では、HP システム上の Adaptive Server をフェールオーバ用に設定するための情報について説明します。

トピック名	ページ
ハードウェアとオペレーティング・システムの稼働条件	45
Adaptive Server の高可用性を実現するための準備	46
HP システムをフェールオーバ用に設定する	51
フェールオーバ用コンパニオン・サーバの設定	63
Sybase フェールオーバの管理	65
HP システムでの Sybase フェールオーバのトラブルシューティング	68
Adaptive Server のアップグレード	71

ハードウェアとオペレーティング・システムの稼働条件

高可用性を実現するには、次のハードウェアとシステム・コンポーネントが必要です。

- CPU やメモリなどのリソースに関して同様に設定されている、2 台の同種のネットワーク・システム
- 高可用性システム・パッケージと関連ハードウェア
- 両方のノードからアクセス可能なデバイス
- さまざまなクラスタ・ノードに割り当てるデバイス・パス名をユニークに保つための論理ボリューム・マネージャ (LVM)
- メディア障害に対処するためのサードパーティのミラーリング

  プラットフォームに固有の高可用性ソフトウェアのインストールについては、ご使用のハードウェアとソフトウェアのマニュアルを参照してください。

Adaptive Server の高可用性を実現するための準備

この項では、Adaptive Server の高可用性設定を行うための準備について説明します。

Adaptive Server のインストール

プライマリ・サーバとセカンダリ・サーバをインストールします。各ノードの同じロケーションにインストールする必要があります。プライマリ・コンパニオンは、新しくインストールしても、既存のデータベースやユーザなどを使用して以前のバージョンからアップグレードしてもかまいません。セカンダリ・コンパニオンには、ユーザ・ログインやユーザ・データベースを使用しない、新しくインストールした Adaptive Server を使用します。これは、すべてのユーザ・ログインやデータベース名をクラスタ内でユニークにするためです。

フェールオーバ用の設定を完了したら、セカンダリ・コンパニオンにユーザ・ログインやデータベースを追加できます。

ローカル・ディスクにインストールする場合は、すべてのデータベースがマルチホスト・ディスク上に作成されていることを確認してください。

Adaptive Server のインストールと設定については、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。

両方の Adaptive Server のエントリを *interfaces* ファイルに追加する

プライマリ・コンパニオンとセカンダリ・コンパニオンの *interfaces* ファイルには、この両方のコンパニオンのエントリが必要です。両方のコンパニオンの *interfaces* ファイルには、以下の出力のホスト名文字列と一致するホスト名 (IP アドレスではなく) が含まれている必要があります。

```
/usr/sbin/cmviewcl -p package_name
```

interfaces ファイル内のサーバ・エントリには、sysservers に指定されているネットワーク名を使用してください。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』を参照してください。

クライアント接続用のエントリを *interfaces* ファイルに追加する

フェールオーバしたコンパニオンにクライアントが再接続できるようにするには、*interfaces* ファイルに行を追加します。デフォルトでは、クライアントはサーバ・エントリの query 行にリストされているポートに接続します。サーバのフェールオーバが原因でこのポートが使用できない場合、クライアントはサーバ・エントリの *hafailover* 行で指定されているサーバに接続します。次に示すのは、MONEY1 という名前のプライマリ・コンパニオン用と PERSONNEL1 という名前のセカンダリ・コンパニオン用の *interfaces* ファイルの例です。

```
MONEY1
    master tcp ether FN1 4100
    query tcp ether FN1 4100
    hafailover PERSONNEL1
```

interfaces ファイルにエントリを追加するには、dsedit を使用します。*interfaces* エントリがすでに存在する場合は、フェールオーバに使用できるように必要に応じて変更します。

dsedit 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。

$SYBASE の値の設定

ローカル・ファイル・システムに *$SYBASE* をインストールした場合、*$SYBASE* は両方のコンパニオンで同じディレクトリ名を指すようにします。

- 各コンパニオン上の *$SYBASE* リリース・ディレクトリが同じディレクトリに作成されていることを確認します。
- 各コンパニオンの *$SYBASE* リリース・ディレクトリが異なるロケーションにある場合は、両方のコンパニオンに同じパスのディレクトリを作成します。このディレクトリは、実際の *$SYBASE* リリース・ディレクトリへのシンボリック・リンクとして機能します。

  たとえば、プライマリ・コンパニオン MONEY1 とセカンダリ・コンパニオン PERSONNEL1 が、それぞれ */usr/u/sybase1* と */usr/u/sybase2* をリリース・ディレクトリとして使用している場合、各コンパニオンの *$SYBASE* が同じパスを指すようにします。

  MONEY1 と PERSONNEL1 には、対応する *$SYBASE* リリース・ディレクトリへのシンボリック・リンクとして確立する */sybase* があります。MONEY1 上では、*/sybase* は */usr/u/sybase1* へのリンクであり、PERSONNEL1 上では、*/sybase* は */usr/u/sybase2* へのリンクです。

$SYBASE をローカル・ファイル・システム上にインストールした場合は、両ノードの *$SYBASE/$SYBASE_ASE/install* に、それぞれのコンパニオンの RUN_<*SERVERNAME*> ファイルもコピーしてください。

sybha 実行プログラムの設定

sybha 実行プログラムによって、Adaptive Server High Availability Basis Services ライブラリは、各プラットフォームの高可用性クラスタ・サブシステムと対話できるようになります。*sybha* は、所有権とパーミッションを変更することにより実行可能になります。*$SYBASE/$SYBASE_ASE/install* の *sybhauser* という名前のファイルも編集する必要があります。このファイルには、そのクラスタに対してシステム管理者権限を持つユーザのリストがあります。クラスタに対するシステム管理者権限を持つユーザの数を制限することを強くおすすめします。

“root” 権限で、次の作業を行います。

1 *sybhagrp* という新しいグループを追加します。このグループを */etc/group* ファイルに追加することも、または NIS マップに追加することもできます。このグループに sybase ユーザを追加します (これは *$SYBASE* ディレクトリを所有するユーザです)。サーバの起動時に sybase ユーザがデータ・サーバを実行します。複数のサーバを実行し、各サーバの *$SYBASE* ディレクトリを異なるユーザが所有する場合は、これらのユーザをすべてこのグループに追加してください。

2 *$SYBASE/$SYBASE_ASE/bin* ディレクトリに移動します。

```
cd $SYBASE/$SYBASE_ASE/bin
```

3 *sybha* の所有権を “root” に変更します。

```
chown root sybha
```

4 *sybha* プログラムのグループを *sybhagrp* に変更します。

```
chgrp sybhagrp sybha
```

5 *sybha* のファイル・パーミッションを 4550 に修正します。

```
chmod 4550 sybha
```

6 *$SYBASE/$SYBASE_ASE/install* ディレクトリに変更します。

```
cd $SYBASE/$SYBASE_ASE/install
```

7 sybase ユーザを *sybhauser* ファイルに追加します。追加するユーザのログインは、Adaptive Server のログインではなく、UNIX の形式のログイン ID である必要があります。次に例を示します。

```
sybase
coffeecup
spooner
venting
howe
```

8 *sybhauser* の所有権を “root” に変更します。

```
chown root sybhauser
```

9 *sybhauser* のファイル・パーミッションを修正します。

```
chmod 600 sybhauser
```

マスタ・デバイス以外の新しいデフォルト・デバイスの作成

新しくインストールした Adaptive Server では、マスタ・デバイスがデフォルト・デバイスです。つまり、データベース (フェールオーバによって使用されるプロキシ・データベースも含む) を作成すると、そのデータベースは自動的にマスタ・デバイス上に作成されます。しかし、ユーザ・データベースをマスタ・デバイスに追加すると、システム障害時に、マスタ・デバイスをリストアすることが困難になります。そのため、マスタ・デバイスには極力余分なユーザ・データベースを置かないようにします。それには、disk init コマンドを使用し、新しいデバイスを作成します。sp_diskdefault を使用して新しいデバイスをデフォルトに指定してから、Adaptive Server をフェールオーバ用のコンパニオンとして設定します。たとえば、money_default1 という名前の新しいデフォルト・デバイスを MONEY1 Adaptive Server に追加するには、次のように入力します。

```
sp_diskdefault money1_default1, defaulton
```

次のコマンドを発行して、明確にデフォルトとしての設定を無効にしないかぎり、マスタ・デバイスはデフォルト・デバイスのままです。

```
sp_diskdefault master, defaultoff
```

disk init と sp_diskdefault 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。

sysservers へのローカル・サーバの追加

sp_addserver を使用して、ローカル・サーバを sysservers にローカル・サーバとして追加します。*interfaces* ファイルで指定されているネットワーク名を使用します。たとえば、MONEY1 というコンパニオンが、*interfaces* ファイルで指定されている MONEY1 というネットワーク名を使う場合は、次のように入力します。

```
sp_addserver MONEY1, local, MONEY1
```

この変更を有効にするには、Adaptive Server の再起動が必要です。

sysservers へのセカンダリ・コンパニオンの追加

セカンダリ・コンパニオンを sysservers にリモート・サーバとして追加します。

```
sp_addserver server_name
```

デフォルトでは、Adaptive Server は、srvid が 1000 のサーバを追加します。この変更は、Adaptive Server を再起動しなくても有効になります。

installhasvss の実行

注意 *installhasvss* を実行する前に、「両方の Adaptive Server のエントリを interfaces ファイルに追加する」(46 ページ) に記載されている作業を行ってください。この作業を行わずに *installhasvss* を実行した場合は、*installmaster* を再実行して、すべてのシステム・ストアド・プロシージャを再インストールする必要があります。

高可用性を有効にし、Adaptive Server を再起動します。

```
sp_configure "enable HA", 1
```

installhasvss スクリプトにより、次の作業が行われます。

- フェールオーバに必要なストアド・プロシージャ (sp_companion など) のインストール。
- SYB_HACMP サーバの sysservers へのインストール。

installhasvss を実行するには、システム管理者権限が必要です。

installhasvss は、*$SYBASE/$SYBASE_ASE/scripts* ディレクトリにあります。*installhasvss* スクリプトを実行するには、次のように入力します。

```
$SYBASE/$SYBASE_OCS/bin/isql -Usa -Ppassword -Sservername <
$SYBASE/$SYBASE_ASE/scripts/installhasvss
```

installhasvss は、ストアド・プロシージャや SYB_HACMP サーバを作成するときにメッセージを表示します。

システム管理者への *ha_role* の割り当て

sp_companion を実行するために、ha_role パーミッションを両方の Adaptive Server に割り当てます。ha_role を割り当てるには、isql から次のコマンドを発行します。

```
sp_role "grant", ha_role, sa
```

一度ログアウトしてから、再度 Adaptive Server にログインして、変更内容を有効にしてください。

設定パラメータの確認

次の設定パラメータを有効にしてから、Adaptive Server をフェールオーバ用に設定します。

- enable cis – コンポーネント統合サービス (CIS) を有効にします。この設定パラメータは、デフォルトで有効です。

- enable xact coordination − 分散トランザクション管理 (DTM) を有効にします。この設定パラメータは、デフォルトで有効です。
- enable HA − 高可用性システムで、Adaptive Server をコンパニオンとして機能させます。enable HA は、デフォルトでは無効です。この設定は静的であるため、Adaptive Server を再起動して変更内容を有効にする必要があります。また、このパラメータを有効にすると、高可用性システムで Adaptive Server を起動したというメッセージがエラー・ログに書き込まれます。

設定パラメータの有効化の詳細については、『システム管理ガイド』を参照してください。

HP システムをフェールオーバ用に設定する

この項では、HP MC/ServiceGuard 高可用性システムを Sybase のフェールオーバ用に設定する手順について説明します。この項では、次のことを前提としています。

- 作業者は HP MC/ServiceGuard に精通している。
- 2 つのノードからなるクラスタ・ハードウェアを MC/ServiceGuard 用に設定している。
- HPUX 11.11 で動作する HP MC/ServiceGuard のバージョン 11.15 を両方のノードにインストールしている。
- クラスタ・システムをインストールし、設定している。
- クラスタ内のすべてのデータベース・デバイスが共有ディスク・デバイスに所属するようにボリューム・グループを設定している。
- すべての共有ボリューム・グループがクラスタ構成に含まれている。

MC/ServiceGuard のインストール、設定、管理の詳細については、HP のマニュアル『Managing MC/ServiceGuard』を参照してください。

パッケージ設定ファイルの作成

パッケージ設定プロセスでは、Adaptive Server を定義するほか、クラスタ内のノードでパッケージを起動した時に実行される関連のリソースが定義されます。パッケージの設定には、パッケージが動作するクラスタ・ノードの優先順位リストが含まれ、そのパッケージがサポートするフェールオーバの種類も定義されています。パッケージは、各コンパニオン・サーバに定義します。

注意 HP MC/ServiceGuard package の名前は、*interfaces* ファイルに指定した Adaptive Server の名前と同じにする必要があります。

たとえば、プライマリ・コンパニオン MONEY1 には MONEY1 という名前のパッケージ、セカンダリ・コンパニオン PERSONNEL1 には PERSONNEL1 という名前の別のパッケージを作成します。

注意 専用のパッケージ設定ファイルを作成しカスタマイズするには、SAM のコマンドまたは MC/ServiceGuard のコマンドを使用します。SAM を使った手順については、HP MC/ServiceGuard のマニュアルを参照してください。このマニュアルでは、MC/ServiceGuard のコマンドを使った手順を説明しています。

“root” 権限で、プライマリ・コンパニオンおよびセカンダリ・コンパニオンの両方に対し次の手順を行います。

1 プライマリ・ノードの */etc/cmcluster* ディレクトリ内にサブディレクトリを作成し、プライマリ・コンパニオンのパッケージ情報を保存します。たとえば、プライマリ・コンパニオン MONEY1 のディレクトリは、次のコマンドで作成します。

```
mkdir /etc/cmcluster/MONEY1
```

2 このディレクトリのパーミッションを変更し、“root” のみがアクセスできるようにします。

```
chmod 700 /etc/cmcluster/MONEY1
```

3 セカンダリ・ノード上に同じサブディレクトリを作成します。たとえば、プライマリ・コンパニオン MONEY1 用の FN1 というマシンにこのディレクトリを作成するには、次のコマンドを実行します。

```
rsh FN1 "mkdir /etc/cmcluster/MONEY1"
```

4 このディレクトリのパーミッションを変更し、“root” のみがアクセスできるようにします。

```
rsh FN1 chmod 700 /etc/cmcluster/MONEY1
```

5 cmmakepkg コマンドを使用して、プライマリ・コンパニオン用のパッケージ設定テンプレートを生成します。

```
/usr/sbin/cmmakepkg -p /etc/cmcluster/subdirectory_name/companion_name.ascii
```

- *subdirectory_name* は、手順 1 で作成したサブディレクトリの名前です。
- *companion_name* は、パッケージを設定するコンパニオンの名前です。

たとえば、プライマリ・コンパニオン MONEY1 にパッケージ設定テンプレートを作るには、次のように入力します。

```
/usr/sbin/cmmakepkg -p /etc/cmcluster/MONEY1/MONEY1.ascii
```

6　手順 5 で作成したパッケージ設定テンプレート・ファイルを編集し、パッケージごとにパッケージ名、ノードの優先順位リスト、制御スクリプトのロケーション、フェールオーバ・パラメータなどを指定します。

下記は、*MONEY1.ascii* 設定ファイルの編集例です (編集内容は使用するマシン名とパラメータによって異なります)。

```
PACKAGE_NAME                    MONEY1
FAILOVER_POLICY                 CONFIGURED_NODE
FAILBACK_POLICY                 MANUAL
NODE_NAME                       FN1
NODE_NAME                       HUM1
RUN_SCRIPT                      /etc/cmcluster/MONEY1/MONEY1.cntl
HALT_SCRIPT                     /etc/cmcluster/MONEY1/MONEY1.cntl
SERVICE_NAME                    MONEY1
SERVICE_FAIL_FAST_ENABLED       NO
SERVICE_HALT_TIMEOUT            300
```

この設定ファイルを、手順 3 で作成したセカンダリ・ノード上のサブディレクトリにコピーします。たとえば、rcp を使用して *MONEY1.ascii* ファイルをコピーするには、次のように入力します。

```
rcp /etc/cmcluster/MONEY1/MONEY1.ascii HUM1:/etc/cmcluster/MONEY1/MONEY1.ascii
```

ASE_HA.sh スクリプトの編集

ASE_HA.sh テンプレート・スクリプトでは、フェールオーバ用 Adaptive Server の起動、終了、監視を行うように高可用性システムを設定します。*ASE_HA.sh* テンプレート・スクリプトは *$SYBASE/$SYBASE_ASE/install* ディレクトリにあります。前の項の手順 1 で作成したパッケージ・サブディレクトリにこのスクリプトをコピーしておき、固有のクラスタ環境に必要な環境変数を追加します。プライマリ・コンパニオンとセカンダリ・コンパニオンは、どちらもこのスクリプトのコピーを必要とします。“root” 権限で、次の手順を行います。

1　高可用性システムで実行する Adaptive Server アプリケーションを設定するために、スクリプトをすでに使っている場合は、そのスクリプトのバックアップ・ファイルを作成します。たとえば、*SYBASE1.sh* という名前のスクリプトを使っている場合、このスクリプトを *SYBASE1.sh.backup* というファイルにコピーします。

2 プライマリ・ノードで、*/etc/cmcluster* の下にあるパッケージ・サブディレクトリに移動します。たとえばプライマリ・コンパニオン MONEY1 を設定する場合、次のように入力します。

```
cd /etc/cmcluster/MONEY1
```

3 *ASE_HA.sh* テンプレート・スクリプトを *$SYBASE/$SYBASE_ASE/install* ディレクトリからプライマリ・コンパニオンのパッケージ・サブディレクトリにコピーします。パッケージ・テンプレート名は、下記の構文規則に従って決めます。

```
<package_name>.sh
```

package_name は、設定しようとしているコンパニオン・サーバの名前です。たとえば、*ASE_HA.sh* ファイルを MONEY1 のサブディレクトリにコピーする場合は、次のように入力します。

```
cp ASE_HA.sh /etc/cmcluster/MONEY1/MONEY1.sh
```

4 使用する環境に合わせて *server_name.sh* ファイルを編集します。サイトに合わせて、"__FILL_IN__" を含む行や、その他編集が必要な行の値を編集してください。下記は個々の行リストについての説明です。

- *ASE_12_0* − Adaptive Server のバージョンを指定します。これによって、$SYBASE ディレクトリ構造が *$SYBASE/$SYBASE_ASE/bin* (12.0 以降) になっているか、または *$SYBASE/bin* (12.0 以前) になっているかを示します。

  次のように設定します。

  - 両方のサーバが Sybase Adaptive Server バージョン 12.0 以降を使用している場合は yes を設定します。
  - Adaptive Server の以前のバージョンを使用している場合は no を設定します。

- *ASE_HAFAILOVER* − Sybase のフェールオーバを使用しているかどうかを指定します。次のように設定します。
  - Sybase のフェールオーバを使用している場合は yes を設定します。
  - モード 0 のフェールオーバを使用している場合は no を設定します。
- *BASIC_FAILOVER* − yes または no に設定します。
  - yes − フェールオーバが機能するモードでサーバが稼働している場合は、HP MC/ServiceGuard 高可用性システムに備わっているフェールオーバ・メカニズムが使用されます。

フェールオーバが発生すると、スクリプトは、まず両方のコンパニオンがフェールオーバを実行できるモードで動作しているかどうかを確認します。コンパニオンで Sybase の フェールオーバが機能しない場合 (つまり、シングルサーバ・モードで実行されている場合)、スクリプトはプライマリ・コンパニオンをセカンダリ・ノードで起動しようとします。

- no − モード 0 フェールオーバを復元しません。

  つまり、*BASIC_FAILOVER* に no を設定すると、ノード・レベルでもコンパニオン・レベルでもフェールオーバは発生しません。

- *PACKAGE_NAME* − MC/ServiceGuard パッケージ設定スクリプトで指定したパッケージの名前。

注意 PACKAGE_NAME の値は、コンパニオン名と同じにします。

- *MONITOR_INTERVAL* − Adaptive Server プロセスの動作を確認する待ち時間 (秒単位)。この間、このスクリプトは待機します。
- *SHUTDOWN_TIMEOUT* − コンパニオン・サーバのアボート処理を始めてから、SYBASE Adaptive Server プロセスを強制終了するまでの最大待ち時間 (秒単位)。*SHUTDOWN_TIMEOUT* は、停止スクリプトの完了を妨げるサスペンドされたコンパニオン・サーバを保護します。*SHUTDOWN_TIMEOUT* の値は、パッケージ設定ファイルで設定された *timeout* 変数よりも小さく設定してください。
- *RECOVERY_TIMEOUT* − 高可用性システムの最大待ち時間 (秒単位)。この時間を超過すると、コンパニオンが起動に失敗したとみなされます。この値には、ロードされているコンパニオンが再起動するために十分な時間を設定してください。また、RECOVERY_TIMEOUT はサブシステムの、フェールオーバからフェールバックまでの処理完了の最大待ち時間としても使用されます。
- *SYBASE* − Sybase 製品がインストールされているロケーション。この値は、プライマリ・ホストの場合は PRIM_SYBASE に、セカンダリ・ホストの場合は SEC_SYBASE に自動的に設定されます。
- *SYBASE_ASE* − Sybase Adaptive Server 製品のインストール・ディレクトリ。
- *SYBASE_OCS* − Sybase オープン・クライアント製品のインストール・ディレクトリ。
- *SYBUSER* − Adaptive Server セッションを開始したユーザの名前。
- *HALOGIN* − sa_role と ha_role を持つユーザのログイン。これは、プライマリ・コンパニオンとセカンダリ・コンパニオンの両方で同一にします。

- *HAPWD* − HALOGIN のパスワード。これは、プライマリ・コンパニオンとセカンダリ・コンパニオンの両方で同一にします。

  注意 Adaptive Server をコンパニオン・サーバとして設定している場合 (つまり sp_companion を実行している場合)、HA_LOGIN と HA_PWD で同一の名前とパスワードを使用してください。

- *PRIM_SYBASE* − Adaptive Server 製品がインストールされているプライマリ・ノード上のディレクトリへのパス。ローカル・デバイスを使用している場合は、ロケーションは両ノードで同じにします。共有デバイスを使用している場合は、ノードごとに異なるロケーションに設定します。
- *PRIM_ASE_HOME* − プライマリ・ノード上にインストールされている Adaptive Server 製品のディレクトリへのパス。デフォルトは、*$SYBASE/$SYBASE_ASE* です。
- *PRIM_SERVER* − プライマリ・コンパニオンの名前。
- *PRIM_HOSTNAME* − プライマリ・ノードの名前。
- *PRIM_CONSOLE_LOG* − 現在のプライマリ・コンパニオン・セッションのエラー・ログへのフル・パス。十分な空き領域があり SYBUSER によって書き込みが可能であれば、どのファイルでもかまいません。デフォルトは、*$SYBASE/$SYBASE_ASE/install/server_name.cs_log* です。
- *PRIM_RUNSCRIPT* − プライマリ・コンパニオンを起動するために使用する RUNSERVER ファイルの名前。デフォルトは、*$SYBASE/$SYBASE_ASE/install/RUN_server_name* です。
- *SEC_SYBASE* − セカンダリ・ノードにインストールされた Adaptive Server 製品のディレクトリ。ローカル・デバイスを使用している場合は、ロケーションは両ノードで同じにします。共有デバイスを使用している場合は、異なるロケーションに設定します。
- *SEC_ASE_HOME* − Adaptive Server 製品がインストールされているセカンダリ・ノード上のディレクトリへのパス。デフォルトは、*$SYBASE/$SYBASE_ASE* です。
- *SEC_SERVER* − セカンダリ・コンパニオンの名前。
- *SEC_HOSTNAME* − セカンダリ・ノードの名前。
- *SEC_CONSOLE_LOG* − 現在のセカンダリ・コンパニオン・セッションのエラー・ログへのフル・パス。十分な空き領域があり SYBUSER によって書き込みが可能であれば、どのファイルでもかまいません。デフォルトは、*$SYBASE/$SYBASE_ASE/install/server_name.cs_log* です。
- *ISQL* − isql バイナリへのパス。デフォルトは、*$SYBASE/$SYBASE_OCS/bin/isql* です。

表 7-1 は、ホスト FN1 で動作するプライマリ・コンパニオン MONEY1 と、ホスト HUM1 で動作するセカンダリ・コンパニオン PERSONNEL1 の *MONEY1.sh* 内の *ASE_HA.sh* の設定を示しています。どちらもローカル・ファイル・システムを使用します。フェールオーバ時に PERSONNEL1 が動作していない場合や、コンパニオン・モードでない場合は、MONEY1 は HUM1 で再起動します。

表 7-1: ASE_HA.sh スクリプトでの MONEY1 の設定

変数	設定
ASE_12_0	yes
ASE_HAFAILOVER	yes
BASIC_FAILOVER	yes
PACKAGE_NAME	MONEY1
MONITOR_INTERVAL	5
SHUTDOWN_TIMEOUT	60
RECOVERY_TIMEOUT	300
SYBASE_ASE	ASE-15_0
SYBASE_OCS	OCS-15_0
HALOGIN	“SA”
HAPASSWD	“Odd2Think”
PRIM_ASE_HOME	デフォルトはディレクトリ *$SYBASE/$SYBASE_ASE*
PRIM_SYBASE	*/opt/sybase*
PRIM_SERVER	MONEY1
PRIM_HOSTNAME	FN1
PRIM_CONSOLE_LOG	*$PRIM_SYBASE/$SYBASE_ASE/install/MONEY1.cs_log*
PRIM_RUNSCRIPT	RUNSERVER ファイルの名前 ― デフォルトは *$SYBASE/$SYBASE_ASE/install/RUN_<servername>*
SYBASE	プライマリの場合は PRIM_SYBASE、セカンダリ・ホスト上にある場合は SEC_SYBASE。自動的に設定
SEC_ASE_HOME	デフォルトは *$SYBASE/$SYBASE_ASE*
SEC_SYBASE	*/opt/sybase*
SEC_SERVER	PERSONNEL1
SEC_HOSTNAME	HUM1
ISQL	デフォルトは *$SYBASE/$SYBASE_OCS/bin/isql*
SEC_CONSOLE_LOG	*$PRIM_SYBASE/$SYBASE_ASE/install/PERSONNEL1.cs_log*

5　ファイルのパーミッションを 700 に変更すると、“root” だけが読み込み、書き込み、実行をできるようになります。たとえば、*MONEY1.sh* のパーミッションを変更するには、次のように入力します。

```
chmod 700 MONEY1.sh
```

6 スクリプトをセカンダリ・ノードに分配します。たとえば、ファイルをセカンダリ・ノード HUM1 に分配するには、次のように入力します。

```
rcp /etc/cmcluster/MONEY1/MONEY1.sh
HUM1:/etc/cmcluster/MONEY1/MONEY1.sh
```

7 上記の手順をセカンダリ・コンパニオンでも行います。

セカンダリ・コンパニオンのパッケージ・スクリプトでは、PRIM_SERVER、PRIM_HOSTNAME、PRIM_SYBASE、SEC_SERVER、SEC_HOSTNAME、SEC_SYBASE の値が使用されます。この値は、プライマリ・コンパニオンのパッケージ・スクリプトの値と反対です。表 7-2 は、*PERSONNEL1.sh* の値を示します。

表 7-2: ASE_HA.sh スクリプトでの PERSONNEL1 の設定

変数	設定
ASE_12_0	yes
ASE_HAFAILOVER	yes
BASIC_FAILOVER	yes
PACKAGE_NAME	PERSONNEL1
MONITOR_INTERVAL	5
SHUTDOWN_TIMEOUT	60
RECOVERY_TIMEOUT	300
SYBASE_ASE	ASE-15_0
SYBASE_OCS	OCS-15_0
HALOGIN	“SA”
HAPASSWD	“Odd2Think”
PRIM_SYBASE	*/opt/sybase*
PRIM_SERVER	PERSONNEL1
PRIM_HOSTNAME	HUM1
PRIM_CONSOLE_LOG	*$PRIM_SYBASE/$SYBASE_ASE/install/MONEY1.cs_log*
PRIM_RUNSCRIPT	RUNSERVER ファイルの名前 ― デフォルトは *$SYBASE/$SYBASE_ASE/install/RUN_<servername>*
SEC_SYBASE	*/opt/sybase*
SEC_SERVER	MONEY1
EC_HOSTNAME	FN1
SEC_CONSOLE_LOG	*$PRIM_SYBASE/$SYBASE_ASE/install/PERSONNEL1.cs_log*

パッケージ制御スクリプトの作成

パッケージ制御スクリプトには、次の処理をするのに必要な情報が含まれています。

- パッケージ内のコンパニオン・サーバの実行。
- コンパニオン・サーバのモニタ。
- エラーへの応答
- パッケージの停止。

セキュリティ上の理由から、制御スクリプトは、パスに *cmcluster* が含まれるディレクトリに置く必要があります。

それぞれのパッケージに、別々の制御スクリプトが必要です。*/etc/cmcluster* 内のパッケージ・サブディレクトリにある制御スクリプトは、パッケージ設定ファイルに記述されているのと同じ名前が付いています。制御スクリプトは、実行プログラムでなければなりません。

“root” 権限で、次の作業を行います。

1　cmmakepkg ユーティリティを使用して、先ほど作成したディレクトリに、プライマリ・コンパニオン用にパッケージ制御スクリプトのテンプレートを生成します。cmmakepkg ユーティリティの構文は次のとおりです。

```
/usr/sbin/cmmakepkg -s
/etc/cmcluster/package_name/companion_name.cntl
```

各パラメータの意味は、次のとおりです。

- *package_name* − 作成したディレクトリの名前
- *companion_name* − 設定しているコンパニオンの名前

2　パッケージ制御スクリプトを編集して、クラスタ環境を反映させます。

a　このコンパニオン・サーバ・パッケージで使用されるボリューム・グループを定義します。

```
VG[0]=""
```

たとえば、プライマリ・コンパニオンがボリューム・グループ *ha_vg1* を使用する場合は、次のように入力します。

```
VG[0]="ha_vg1"
```

b　共有ファイル・システムを使用している場合は、スクリプトの FILESYSTEMS セクションに次の行を記述して、論理ボリュームとファイル・システムを定義します。

```
LV[0]="";FS[0]="", FS_MOUNT_OPT[0]="-Fvxfs -o rw, suid, log, mincache, dync,
blkclear, detainlog, largefiles"
```

たとえば、プライマリ・コンパニオンが、論理ボリューム *ha_lv1* 上の *ha_fs1* ファイル・システムにデータを持っている場合は、次のように指定します。

```
LV[0]="ha_lv1";FS[0]="/ha_fs1", FS_MOUNT_OPT[0]=""
```

c customer_defined_halt_cmds 関数内に、コンパニオン・サービスを停止するコマンドを入力します。このコマンドには、*ASE_HA.sh* ファイルのロケーションが含まれています (「ASE_HA.sh スクリプトの編集」(53 ページ) 参照)。編集前は、次のようになっています。

```
function customer_defined_halt_cmds
{
# ADD customer defined run commands.
: # do nothing instruction, because a function must contain some command.

test_return 52
}
```

関数を編集して halt コマンドを入力します。たとえば、コンパニオン MONEY1 に halt コマンドを入力するには、次のように指定します。

```
function customer_defined_halt_cmds
{
# ADD customer defined run commands.
: # do nothing instruction, because a function must contain some command.

/etc/cmcluster/MONEY1/MONEY1.sh halt
test_return 52
}
```

d *companion_name.cntl* の START OF CUSTOMER DEFINED FUNCTIONS セクションに移動し、コンパニオン・サービスを開始するコマンドを入力します。このコマンドは、customer_defined_run_cmds 関数の中に入力してください。このコマンドには、*ASE_HA.sh* ファイルのロケーションが含まれています (「ASE_HA.sh スクリプトの編集」(53 ページ) 参照)。編集前は、次のようになっています。

```
function customer_defined_run_cmds
{
# ADD customer defined run commands.
: # do nothing instruction, because a function must contain some command.

test_return 51
}
```

関数を編集して start コマンドを入力します。たとえば、コンパニオン MONEY1 に start コマンドを含めるには、次のように指定します。

```
function customer_defined_run_cmds
{
# ADD customer defined run commands.
: # do nothing instruction, because a function must contain some command.
```

```
/etc/cmcluster/MONEY1/MONEY1.sh start
test_return 51
}
```

e　スクリプトの SERVICE NAMES AND COMMANDS セクションに、サーバ・プロセスをサービスとしてモニタするスクリプトを定義します。

```
SERVICE_NAME[0]=""
SERVICE_CMD[0]=""
SERVICE_RESTART[0]=""
```

たとえば、プライマリ・コンパニオン MONEY1 をモニタするように設定するには、次のように入力します。

```
SERVICE_NAME[0]="MONEY1"
SERVICE_CMD[0]="/etc/cmcluster/MONEY1/MONEY1.sh monitor"
SERVICE_RESTART[0]="-R"
```

f　スクリプトをクラスタ内の各ノードに分配します。たとえば、スクリプトをセカンダリ・ノード HUM1 に分配するには、次のように指定します。

```
# rcp /etc/cmcluster/MONEY1/MONEY1.cntl
HUM1:/etc/cmcluster/MONEY1/MONEY1.cntl
```

g　この作業をセカンダリ・コンパニオンでも行います。

設定の確認と分配

1　cmquerycl ユーティリティを使用して、クラスタ設定情報ファイルを作成します。

```
cmquerycl -n primary_node_name -n secondary_node_name -v -C
/etc/cmcluster/cmclconfig.ascii
```

primary_node_name − プライマリ・ノードのホスト名。

secondary_node_name − セカンダリ・ノードのホスト名。

cmclconfig.ascii ファイルで、max_configured_packages を 2 に変更します。

2　cmcheckconf ユーティリティを使用して、パッケージ設定ファイルが正しいことを確認します。cmcheckconf は、次の構文を使用します。

```
cmcheckconf -C /etc/cmcluster/cmclconfig.ascii -P
/etc/cmcluster/package_name/primary_companion_name.ascii -P
/etc/cmcluster/secondary_package_name/secondary_companion_name.ascii
```

各パラメータの意味は次のとおりです。

- *package_name* は作成したディレクトリの名前です。
- *primary_companion_name* は、設定しているコンパニオンの名前です。

- *secondary_companion_name* は、セカンダリ・コンパニオンの名前です。

たとえば、MONEY1 のパッケージ設定ファイルを確認するには、次のように指定します。

```
cmcheckconf -C /etc/cmcluster/cmclconfig.ascii -P
/etc/cmcluster/MONEY1/MONEY1.ascii
-P /etc/cmcluster/PERSONNEL1/PERSONNEL1.ascii
```

3 バイナリ・クラスタの設定ファイルを分配するには、次の操作を行います。

a vgchange コマンドを発行して、クラスタ・ロック・ボリューム・グループをアクティブにすると、ロック・ディスクを初期化できます。

```
/usr/sbin/vgchange -a y /dev/vglock
```

b cmapplyconf ユーティリティを使用してバイナリ設定ファイルを生成し、各ノードに分配します。

```
/usr/sbin/cmapplyconf -v -C
/etc/cmcluster/cmclconf.ascii -P
/etc/cmcluster/primary_package_name/primary_companion
_name.ascii
-P
/etc/cmcluster/secondary_package_name/secondary_compa
nion_name.ascii
```

primary_package_name は作成したディレクトリ名、*primary_companion_name* は設定するコンパニオン名です。*secondary_package_name* は作成したセカンダリ・ディレクトリ名、*secondary_companion_name* はセカンダリ・コンパニオン名です。たとえば、MONEY1 のバイナリ設定ファイルを生成するには、次のように指定します。

```
# cmapplyconf -v -C /etc/cmcluster/cmclconf.ascii -P
 /etc/cmcluster/MONEY1/MONEY1.ascii
 -P /etc/cmcluster/PERSONNEL1/PERSONNEL1.ascii
```

c クラスタ・ロック・ボリューム・グループを非アクティブ化するには、次の操作を行います。

```
/etc/sbin/vgchange -a n /dev/vglock
```

注意 ロック・ディスクが初期化できるように、クラスタ・ロック・ボリューム・グループは、cmapplyconf コマンドを発行するノード上でのみアクティブ化できます。クラスタを設定する場合は、クラスタ・ロック・ボリューム・グループを設定ノードでのみアクティブ化し、他のすべてのノードでは非アクティブ化する必要があります。設定ノードのクラスタ・ロック・ボリューム・グループの非アクティブ化は、cmapplyconf を実行してから行ってください。

クラスタやパッケージの設定ファイルに変更を加える場合は、常に cmcheckconf と cmapplyconf を実行します。

プライマリ・コンパニオンとセカンダリ・コンパニオンの起動

コンパニオン・クラスタを、そのパッケージも含めて起動します。1 つのコンパニオン・ノードで、“root” として次の構文を使用します。

```
cmruncl -v
```

コンパニオン・クラスタの情報を表示するには、次のように入力します。

```
cmviewcl -v
```

個々のノードにパッケージを追加するには、次のように入力します。

/usr/sbin/cmrunpkg -n *node_name primary_companion _name*

次に例を示します。

```
/usr/cmrunpkg -n FN1 MONEY1
```

フェールオーバ用コンパニオン・サーバの設定

この項では、高可用性システムで Adaptive Server をプライマリ・コンパニオンとセカンダリ・コンパニオンとして設定する手順を示します。

do_advisory オプションを指定して *sp_companion* を実行する

十分なリソースを持つセカンダリ・コンパニオンを設定して、フェールオーバ中でもサーバ 2 台分の作業を実行できるようにします。セカンダリ・コンパニオンは、正常なクラスタ・オペレーションを妨げる属性を持っている場合があります。たとえば、プライマリ・コンパニオンとセカンダリ・コンパニオンのユーザ・ログイン数がともに 250 に設定されていると、フェールオーバ中、セカンダリ・コンパニオンには、発生する可能性のあるユーザ・ログインの半分を処理するリソースしかないことになります。したがって、MONEY1 と PERSONNEL1 の両方でユーザ・ログイン数を 500 に設定します。

sp_companion do_advisory は、クラスタ・オペレーション (Adaptive Server をセカンダリ・コンパニオンとして設定するなど) が正常に行われるようにするために、プライマリ・コンパニオンとセカンダリ・コンパニオンの両方の設定オプションを確認します。また、sp_companion do_advisory は、変更する必要のある設定オプションを通知します。

sp_companion do_advisory オプションの詳細については、「第 6 章 do_advisory の実行」 を参照してください。

非対称型コンパニオン設定の作成

プライマリ・コンパニオンを非対称型で設定するには、次のコマンドをセカンダリ・コンパニオンから発行します。

sp_companion "*primary_server_name*", configure, NULL, *login_name*, *password*

- *primary_server_name* − *interfaces* ファイルのエントリと sysservers に定義されているプライマリ Adaptive Server の名前。
- *login_name* − このクラスタ・オペレーションを行っているユーザの名前。ユーザは ha_role を持っている必要があります。
- *password* − このクラスタ・オペレーションを行っているユーザのパスワード。

注意 上記コマンドは、セカンダリ・コンパニオンからのみ実行します。

この例では、MONEY1 という名前の Adaptive Server をプライマリ・コンパニオンとして設定します。セカンダリ・サーバ PERSONNEL1 から次のコマンドを発行します。

```
sp_companion "MONEY1", configure, NULL, sa, Odd2Think
Server 'PERSONNEL1' is alive and cluster configured.
Step: Access verified from Server:'PERSONNEL1' to Server:'MONEY1'
Server 'MONEY1' is alive and cluster configured.
Step: Access verified from Server:'MONEY1' to Server:'PERSONNEL1'
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
.....
Step: Companion servers configuration check succeeded
Step: Server handshake succeeded
Step: Master device accessible from companion
Step: Added the servers 'PERSONNEL1' and 'MONEY1' for cluster config
Step: Server configuration initialization succeeded
Step: Synchronizing server logins from companion server
Step: Synchronizing remoteserver from companion server
Step: Synchronizing roles from companion server
Step: Synchronizing server-wide privs from companion server
Step: User information syncup succeeded
Step: Server configured in normal companion mode
```

sp_companion 設定を行っている間にユーザ・データベースが作成されると、次のようなメッセージが表示されます。

```
Step: Created proxy database ‘pubs2’
Step: Proxy status for database has been set. Please Checkpoint the database 'pubs2'
Step: Server configured in normal companion mode
Starting companion watch thread
```

対称型設定の作成

非対称型フェールオーバを使用できるようにコンパニオンを設定した後、対称型設定に設定できます。対称型設定では、両方のサーバがプライマリ・コンパニオンとしても、セカンダリ・コンパニオンとしても機能します。対称型設定については、図 3-2 (21 ページ) を参照してください。

sp_companion をセカンダリ・コンパニオンから発行して、対称型設定に設定します。sp_companion の構文については、前述の「非対称型コンパニオン設定の作成」を参照してください。

次は、「非対称型コンパニオン設定の作成」(64 ページ) で説明した、MONEY1 という Adaptive Server をセカンダリ・コンパニオンとして、PERSONNEL1 という Adaptive Server に追加する例です。MONEY1 サーバから次のコマンドを発行します。

```
sp_companion 'PERSONNEL1', configure, sa, Think2Odd
Server 'MONEY1' is alive and cluster configured.
Step: Access verified from Server:'MONEY1' to Server:'PERSONNEL1'
Server 'PERSONNEL1' is alive and cluster configured.
Step: Access verified from Server:'PERSONNEL1' to Server:'MONEY1'
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
.....
Step: Companion servers configuration check succeeded
Step: Server handshake succeeded
Step: Master device accessible from companion
Step: Added the servers 'MONEY1' and 'PERSONNEL1' for cluster config
Step: Server configuration initialization succeeded
Step: Synchronizing server logins from companion server
Step: Synchronizing remoteserver from companion server
Step: Synchronizing roles from companion server
Step: Synchronizing server-wide privs from companion server
Step: User information syncup succeeded
Step: Server configured in normal companion mode
```

Sybase フェールオーバの管理

この項では、Sybase のフェールオーバの使い方を説明します。

プライマリ・コンパニオンへのフェールバックとノーマル・コンパニオン・モードの再開

フェールバックは、プライマリ・コンパニオンの共有ディスクをセカンダリ・ノードからプライマリ・ノードに戻し、プライマリ・コンパニオンをプライマリ•ノードで起動します。フェールバックは計画的に実行されるイベントです。プライマリ・コンパニオンにフェールバックするには、次の手順に従います。

1 セカンダリ・コンパニオンから sp_companion を発行して、フェールオーバ・モードであることを確認します。

注意 高可用性システムでは、プライマリ・コンパニオンは自動的に再起動されます。

2 セカンダリ・コンパニオンから次のコマンドを発行します。*primary_companion_name* は、プライマリ・コンパニオン・サーバの名前です。

```
sp_companion primary_companion_name, prepare_failback
```

3 プライマリ・コンパニオンから、次のコマンドを発行します。*secondary_companion_name* は、セカンダリ・コンパニオン・サーバの名前です。

```
sp_companion secondary_companion_name, resume
```

4 オプションを使用せずに、どちらか一方のコンパニオンから sp_companion を発行して、ノーマル・コンパニオン・モードになっていることを確認します。

注意 sp_companion resume を発行しないと、フェールオーバ・プロパティが設定されたクライアントを接続できません。sp_companion prepare_failback を発行してからクライアントを再接続しようとすると、クライアントは sp_companion resume を発行するまで応答を停止します。

コンパニオン・モードのサスペンド

サスペンド・モードでは、一時的にプライマリ・コンパニオンがセカンダリ・コンパニオンにフェールオーバできなくなります。コンパニオンをサスペンド・モードに切り替えると、コンパニオン間の同期が発生せず、プライマリ・コンパニオンはセカンダリ・コンパニオンにフェールオーバできません。しかし、サスペンド・モードは、設定パラメータの変更などの管理タスクを実行するのに役立ちます。ノーマル・コンパニオン・モードからサスペンド・モードに切り替えるには、次の手順に従います。

1　“root” 権限で、cmhaltserv を発行してモニタ・プロセスを無効化し、コンパニオン・サーバをシャットダウンしたときにフェールオーバが実行されないようにします。*primary_package_name* は、プライマリ・パッケージの名前です (プライマリ・コンパニオンの名前と同じです)。

```
cmhaltserv -v primary_package_name
```

2　ノーマル・コンパニオン・モードからサスペンド・モードに切り替えます。セカンダリ・コンパニオンから次のコマンドを発行します。

```
sp_companion primary_server_name, suspend
```

これで、必要なときにプライマリ・コンパニオンをシャットダウンすることができます。このとき、セカンダリ・コンパニオンへのフェールオーバは発生しません。

サスペンド・モードからのノーマル・コンパニオン・モードの再開

サスペンド・モードの 2 つのコンパニオン間でノーマル・コンパニオン・モードを再開するには、次の手順に従います。

1　“root” 権限で、cmhaltpkg をプライマリ・ノードから発行してプライマリ・コンパニオンをシャットダウンします。*primary_package_name* は、プライマリ・パッケージの名前です (プライマリ・コンパニオン・サーバの名前と同じです)。

```
cmhaltpkg primary_package_name
```

2　“root” 権限で、cmmodpkg と cmrunpkg をプライマリ・コンパニオンから発行して、プライマリ・コンパニオンを再起動するパッケージを実行します。*primary_package_name* は、プライマリ・パッケージの名前です (プライマリ・コンパニオン・サーバの名前と同じです)。

```
cmmodpkg -e primary_package_name
cmrunpkg primary_package_name
```

コンパニオン・モードの削除

コンパニオン・モードを削除するには、次のコマンドを発行します。

```
sp_companion companion_name, "drop"
```

コンパニオン・モードを削除したら、二度と元に戻すことはできません。フェールオーバを再確立するには、Adaptive Server のコンパニオン・サーバを再設定する必要があります。ただし、Adaptive Server が動作しているノードは、引き続き高可用性システムでモニタされます。

モニタ・スクリプトの実行中にコンパニオン・モードを削除すると、このスクリプトが引き続きサーバをモニタします。サーバをシャットダウンしてもノードをフェールオーバしたくない場合は、次のコマンドを発行して、モニタ・プロセスを強制終了します。

```
/usr/sbin/cmhaltsrv service_name
```

または、パッケージを停止し、ボリューム・グループを再度アクティブ化してから、コンパニオンだけを再起動します。

モニタ・プロセスを強制終了せずに、コンパニオンが応答を停止したことが検出されると、セカンダリ・ノードにフェールオーバが発生します。このプロセスは、BASIC_FAILOVER の設定に応じて、セカンダリ・ノードでプライマリ・コンパニオンを再起動します。

HP システムでの Sybase フェールオーバのトラブルシューティング

この項では、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。

エラー・メッセージ 18750

コンパニオン・サーバによってエラー・メッセージ 18750 が発行された場合は、サーバの *@@cmpstate* を確認します。プライマリ・コンパニオンがノーマル・コンパニオン・モードになっていても、セカンダリ・コンパニオンがセカンダリ・フェールオーバ・モードになっている場合、クラスタは整合性が失われた状態に陥っています。その場合は、手動でリカバリする必要があります。整合性が失われた状態は、sp_companion 'prepare_failback' コマンドがセカンダリ・コンパニオンで失敗することによって発生する可能性があります。セカンダリ・ノードのログを調べると、この状態になったかどうかを判断できます。エラー 18750 からリカバリするには、次の操作を行います。

1 両パッケージをシャットダウンして、プライマリ・コンパニオンとセカンダリ・コンパニオンを停止します。

2 セカンダリ・コンパニオンのパッケージを起動して、セカンダリ・コンパニオンを再起動します。

3 “suspect” のマークの付いたデータベースをすべて修復します。この状態のデータベースを検索するには、次のコマンドを発行します。

```
select name, status from sysdatabases
```

suspect のマークの付いたデータベースは、status の値が 320 になっています。

4 システム・テーブルの更新を許可するには、以下を入力します。

```
sp_configure “allow updates”, 1
```

5　suspect 状態のフェールオーバ・データベースに対して、それぞれ次を実行します。

```
1> update sysdatabase set status=status-256 where name='database_name'
2> go
1> dbcc traceon(3604)
2> go
1> dbcc dbrecover(database_name)
2> go
```

6　セカンダリ・コンパニオンから、次のコマンドを発行します。

```
sp_companion primary_companion_name, prepare_failback
```

このコマンドが正常に実行されることを確認します。

7　ノーマル・コンパニオン・モードを再開します。プライマリ・コンパニオンから、次のコマンドを発行します。

```
sp_companion secondary_companion, resume
```

失敗した *prepare_failback* からのリカバリ

フェールバック中に、prepare_failback がセカンダリ・コンパニオンで正常に実行されたにもかかわらず、プライマリ・コンパニオンが起動されない場合は、次を実行してロールバックした後に、prepare_failback コマンドを再発行します。

1　プライマリ・コンパニオンのエラー・ログ、HP MC/ServiceGuard package ログ、またはシステム・ログを確認し、サーバが起動できなかった原因を調べ、問題を解決します。

2　プライマリ・コンパニオンのパッケージをプライマリ・ノードで実行すると、このパッケージは停止します。

3　セカンダリ・コンパニオンにログインして、次のコマンドを発行します。

```
dbcc ha_admin ("", "rollback_failback")
dbcc ha_admin ("", "rollback_failover")
```

4　セカンダリ・コンパニオンが、ノーマル・コンパニオン・モードになっていることを確認します。

5　“root” 権限で、プライマリ・コンパニオンがセカンダリ・ノードで起動するよう、パッケージを開始します。

```
/usr/sbin/cmrunpkg -n secondary_node primary_companion_package_name
```

セカンダリ・コンパニオンは、これでフェールオーバ・モードになります。プライマリ・コンパニオンがノーマル・コンパニオン・モードにフェールバックする準備が整ったことを確認したら、sp_companion...prepare_failback を実行します。

エラー・ログのロケーション

Sybase のフェールオーバと HP MC/ServiceGuard には、次のようなエラー・ログが含まれます。

- */var/adm/syslogs/syslog.log* － HP MC/ServiceGuard クラスタのアクティビティや、オペレーティング・システムのアクティビティの出力が含まれます。
- */etc/cmcluster/<package_name>/<package_name>.cntl.log* － HP MC/Service Guard package アクティビティの出力のほか、コンパニオンの起動、停止、モニタ・スクリプトにより発生した Sybase のフェールオーバ・アクティビティの出力が含まれます。

  コンパニオンの起動、停止、およびモニタ・スクリプトからの出力については、“SYBASE HA” を検索します。

  MC/Service Guard package の障害の出力については、“ERROR” という文字列を検索します。
- *$PRIM_CONSOLE_LOG* － このログのロケーションは、*/etc/cmcluster/<package_name>/<package_name>.sh* に定義されています。このエラー・ログには、プライマリの *ASE_HA.sh* スクリプトから Adaptive Server を最後に実行したときの情報が含まれています。
- *SEC_CONSOLE_LOG* － このログのロケーションは、*/etc/cmcluster/<package_name>/<package_name>.sh* に定義されています。このエラー・ログには、セカンダリの *ASE_HA.sh* スクリプトから Adaptive Server を最後に実行したときの情報が含まれています。

Adaptive Server のアップグレード

高可用性設定内の Adaptive Server をアップグレードするには、一時的にプライマリ・コンパニオンとセカンダリ・コンパニオンの間のコンパニオン関係を削除し、Adaptive Server パッケージのモニタリングを無効にする必要があります。これにより、アップグレード・プロセスの間、MC/ServiceGuard クラスタによって予期しないフェールオーバがトリガされることなく、各 Adaptive Server コンパニオンを独立して停止または再起動できます。

注意　-n *node_name* パラメータで別のノードを指定しないかぎり、パッケージを起動する MC/ServiceGuard コマンドはコマンドが実行されるノードに対してそのコマンドが発行されているものとみなします。コマンドを発行する前に、MC/ServiceGuard のマニュアルを参照して、パッケージが適切なノードで起動されていることを確認してください。

アップグレード・プロセスの間は、データベース、オブジェクト、ユーザ、またはログインを追加、削除、修正できません。コンパニオン関係を削除した後、その関係を再確立する前にこのような変更を行うと、アップグレードが失敗する場合があります。または、サーバ間の不整合が原因でクラスタが不安定になることがあります。

❖ モニタリング・サービスの停止とコンパニオン関係の削除

1　コンパニオン関係を削除します。セカンダリ・コンパニオンから、次のコマンドを発行します。

```
sp_companion primary_server_name, "drop"
```

2　プライマリ・コンパニオンから、次のコマンドを発行します。

```
sp_companion secondary_server_name,"drop"
```

3　両方のノードがシングルサーバ・モードであることを確認します。各ノードで次のコマンドを発行します。

```
sp_companion
```

コンパニオンがシングルサーバ・モードである場合は、次のような結果になります。

```
Server 'server_name' is not cluster configured.
Server 'server_name' is currently in 'Single server' mode.
```

4　クラスタ内のすべてのノードの Adaptive Server パッケージで、モニタリング・サービスを停止します。root 権限で、次のコマンドを発行します。

```
cmhaltserv -v primary_package_name
```

❖ Adaptive Server のアップグレード

1 各ノードで、高可用性を無効にします。

```
sp_configure 'enable HA', 0
```

Adaptive Server を再起動して、この変更を有効にします。

2 『インストール・ガイド』の手順に従って各サーバをアップグレードします。

3 すべてのノードで、高可用性を再度有効にします。

```
sp_configure 'enable HA', 1
```

Adaptive Server を再起動して、この変更を有効にします。

4 アップグレードしたサーバに、*installmaster* スクリプトと *installhasvss* スクリプトを再インストールします。詳細については、「installmaster の再インストール」と「installhasvss の再実行」(196 ページ) を参照してください。 *installmaster* を再インストールする場合は、*installhasvss* を再インストールする必要があります。

5 sybha バイナリと *sybhausers* ファイルのパーミッションが正しく設定されていることを確認します。

root 権限で、次のコマンドを *$SYBASE/$SYBASE_ASE/bin* から発行します。

```
chown root sybha
chmod 4550 sybha
```

root 権限で、以下の手順を *$SYBASE/$SYBASE_ASE/install* から実行します。

1 sybase ユーザが *sybhauser* ファイルに含まれていることを確認します。

2 次のコマンドを発行します。

```
chown root sybhauser
chmod 600 sybhauser
```

6 */etc/cmcluster/package_name/package_name.sh* スクリプトで、パッケージ・プロパティまたは新しいインストール環境内の高可用性に関係するファイル (*PRIM_SYBASE*、*PRIM_RUNSCRIPT*、*PRIM_CONSOLE_LOG* など) に変更が正しく反映されます。

❖ コンパニオン関係の再確立とモニタリングの再開

1 各ノードで、Adaptive Server を手動で再起動します。

2 root 権限で、プライマリ・ノードからモニタリング・サービスを再開します。

```
cmmodpkg -e primary_package_name
```

3 「フェールオーバ用コンパニオン・サーバの設定」(63 ページ) で説明されている、コンパニオン関係を確立するための前提条件となる手順が完了していることを確認します。

4 サーバ間のコンパニオン関係を再確立します。
非対称型設定の場合は、セカンダリ・サーバで次のコマンドを発行します。対称型設定の場合は、両方のコンパニオンで次のコマンドを発行します。

```
dbcc traceon (2209)
sp_companion primary_server_name, configure, NULL, user_name, password
```

ユーザ・データベースがセカンダリ・サーバに存在する場合は、次のような警告メッセージが表示されることがあります。このようなメッセージは無視できます。

```
Msg 18739, Level 16, State 1:
Server 'svr2', Procedure 'sp_hacmpcfgvrfy', Line 102:
Database 'svr2_db1': a user database exists. Drop this
database and retry the configuration again.
```

5 root 権限で、パッケージをオフラインにします。

```
dbcc traceoff(2209)
cmhaltpkg "primary_package_name"
cmhaltpkg "secondary_package_name"
```

6 適切なノードでパッケージを再起動します。root 権限で、プライマリ・ノード上で次のコマンドを発行します。

```
cmrunpkg -v "primary_package_name"
```

root 権限で、セカンダリ・ノード上で次のコマンドを入力します。

```
cmrunpkg -v "secondary_package_name"
```

7 sp_companion を実行して、システムがフェールオーバ用に適切に設定されていることを確認します。このコンパニオン・サーバでフェールオーバおよびフェールバックが機能することを確認するには、プライマリ・パッケージをセカンダリ・ノードに移動します。

第 8 章

IBM AIX HACMP システムでフェールオーバを使用できるように Adaptive Server を設定する

この章では、IBM AIX HACMP システム上の Adaptive Server をフェールオーバ用に設定するための情報について説明します。

トピック名	ページ
ハードウェアとオペレーティング・システムの稼働条件	75
高可用性で動作するように Adaptive Server を準備する	77
IBM AIX サブシステムを Sybase フェールオーバ用に設定する	83
フェールオーバ用コンパニオン・サーバの設定	90
Sybase フェールオーバの管理	93
HACMP for AIX でのフェールオーバのトラブルシューティング	98
Adaptive Server のアップグレード	100

ハードウェアとオペレーティング・システムの稼働条件

高可用性を実現するには、次のハードウェアとシステム・コンポーネントが必要です。

- CPU やメモリなどのリソースに関して同様に設定されている、2 台の同種のネットワーク・システム
- 高可用性システム・パッケージと関連ハードウェア
- 両方のノードからアクセス可能なデバイス
- さまざまなクラスタ・ノードに割り当てるデバイス・パス名をユニークに保つための論理ボリューム・マネージャ (LVM)
- メディア障害に対処するためのサードパーティのミラーリング

  プラットフォームに固有の高可用性ソフトウェアのインストールについては、ご使用のハードウェアとソフトウェアのマニュアルを参照してください。

IBM AIX HACMP システムでフェールオーバを使用するための稼働条件

IBM High Availability Cluster Multiprocessing (HACMP) での高可用性の設定に必要な稼働条件は、次のとおりです。

- ハードウェアの互換性があり、HACMP for AIX 5.2 を実行し、同じクラスタに設定されている 2 台のノード。
- 各ノードに、サービス用、起動用、スタンバイ用の 3 つの IP アドレスが設定されていること。スタンバイ用の IP アドレスは、他の 2 つのアドレスと異なるサブネットを使用します。
- 高可用性システム用に設定された共有ディスク・デバイス。ノード間で共有します。
- クラスタ内にデータベース・デバイスをすべて備えるように設定した、共有論理ボリューム・グループ。クラスタに定義した各共有ボリューム・グループのメジャー番号は、両ノードで同じ番号にします。この章では、このようなリソースを次のように呼びます。
  - *shared_vg1* (プライマリ・ノード)
  - *shared_vg2* (セカンダリ・ノード)

  高可用性システムのインストールについては、『HACMP for AIX インストールガイドまたは管理ガイド』を参照してください。

  また Sybase では、次のリソースをあらかじめ定義することをおすすめします。
- プライマリ・コンパニオンのリソース・グループ名 (たとえば、*resgrp1*)
- セカンダリ・コンパニオンのリソース・グループ名 (たとえば、*resgrp2*)
- プライマリ・コンパニオン名
- セカンダリ Adaptive Server コンパニオン名

HACMP for AIX で Adaptive Server を実行する場合の注意事項

プライマリ・コンパニオンが HACMP 5.2 でフェールオーバするとき、プライマリ・コンパニオンだけではなく、ノードがすべてフェールオーバします。ノードがフェールオーバしている間は、サービス中のホスト (プライマリ・ノード) の IP アドレスが、別のスタンバイ・アドレスとスワップします。ネットワーク環境によっては、初期 IP アドレス上の全プロセスがフリーズし、最終的にはタイム・アウトしてしまうことがあります。このため、Sybase のフェールオーバを AIX 上で HACMP とともに使用する場合は、次のようにしてください。

- クライアントには、プライマリ・ノードへの直接ログインを許可しない。
- プライマリ・ノードで一度に実行できる高可用性アプリケーションの数を、1 つに限定する。

高可用性で動作するように Adaptive Server を準備する

この項では、Adaptive Server の高可用性設定を行うための準備について説明します。

Adaptive Server のインストール

まず、Adaptive Server をインストールするノードで HACMP サービスを起動してから、Adaptive Server をインストールします。HACMP ノードは、起動用やスタンバイ用の IP アドレスではなく、サービス用の IP アドレスで実行します。

プライマリ・サーバとセカンダリ・サーバをインストールします。ローカルまたは共有ファイル・システムのどちらかにコンパニオンをインストールできます。

共有ファイル・システムにインストールする場合、ファイル・システムはコンパニオンごとに別の名前にしてください。これは、デバイスのフェールオーバ中に、ファイル・システムを上書きしないようにするためです。たとえば、プライマリ・コンパニオンを *node1_sybase* にインストールした場合、セカンダリ・コンパニオンは */node2_sybase* にインストールします。

ローカル・ファイル・システムにサーバをインストールする場合は、ファイル・システム名を同じにしてください。たとえば、プライマリ・コンパニオンとセカンダリ・コンパニオンを、両方とも */sybase* にインストールできます。

$SYBASE を設定されているファイル・システムは、ローカルまたは共有のどちらかにします。クラスタ内に、*$SYBASE* に設定されているローカル・ファイル・システムや共有ファイル・システムを混在させることはできません。

プライマリ・コンパニオン用のデータベース・デバイスには、プライマリ・ノード上の共有ボリューム・グループにあるデバイス (たとえば、*shared_vg1*) を使用するので、このノードのボリューム・グループは必ず「使用可能」にしてください。「使用可能」とは、ボリューム・グループがこのノード上でアクティブかつアクセス可能であることを意味します。

非対称型の設定を行う場合は、セカンダリ・コンパニオンのデータベース・デバイスとしてどのデバイスでも使用できます。共有ボリューム・グループにあるものでも、ローカル・ボリューム・グループにあるものでもかまいません。対称型の設定を行う場合は、セカンダリ・コンパニオン用のデータベース・デバイスには、セカンダリ・ノード上の共有ボリューム・グループにあるデバイス (たとえば、*shared_vg2*) を使用するので、このノードのボリューム・グループは必ず「使用可能」にしてください。「使用可能」とは、ボリューム・グループがこのノード上でアクティブかつアクセス可能であることを意味します。

プライマリ・コンパニオンは、新しくインストールした Adaptive Server でも、旧バージョンの Adaptive Server からアップグレードして既存のデータベースやユーザなどを受け継いだものでもかまいません。

セカンダリ・コンパニオンには、ユーザ・ログインやユーザ・データベースを使用していない、新しくインストールした Adaptive Server を使用してください。これにより、ユーザ・ログインやデータベース名がクラスタ内で重複しません。フェールオーバ用の設定を完了したら、セカンダリ・コンパニオンにユーザ・ログインやデータベースを追加できます。

Adaptive Server のインストールと設定については、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。

両方の Adaptive Server のエントリを *interfaces* ファイルに追加する

プライマリ・コンパニオンとセカンダリ・コンパニオンの *interfaces* ファイルには、両方のコンパニオンのエントリが必要です。*interfaces* ファイル内のサーバ・エントリには、sysservers に指定されているネットワーク名を使用してください。*interfaces* ファイルへのエントリの追加については、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。

クライアント接続を行うために *interfaces* ファイルにエントリを追加する

フェールオーバしたコンパニオンにクライアントが再接続できるようにするには、*interfaces* ファイルに行を追加します。デフォルトでは、クライアントはサーバ・エントリの query 行にリストされているポートに接続します。サーバのフェールオーバが原因でこのポートが使用できない場合、クライアントはサーバ・エントリの *hafailover* 行で指定されているサーバに接続します。次に示すのは、MONEY1 という名前のプライマリ・コンパニオン用と PERSONNEL1 という名前のセカンダリ・コンパニオン用の *interfaces* ファイルの例です。

```
MONEY1
        master tcp ether FN1 4100
        query tcp ether FN1 4100
        hafailover PERSONNEL1
```

interfaces ファイルにエントリを追加するには、dsedit を使用します。interfaces エントリがすでに存在する場合は、フェールオーバに使用できるように変更します。

dsedit については、『ユーティリティ・ガイド』を参照してください。

$SYBASE の値の設定

ローカル・ファイル・システムに *$SYBASE* をインストールする場合、*$SYBASE* は両方のコンパニオンで同じディレクトリ名を指すようにします。これには、次のいずれかの方法を使用します。

- 各コンパニオン上の *$SYBASE* リリース・ディレクトリが同じディレクトリに作成されていることを確認します。

- 各コンパニオンの *$SYBASE* リリース・ディレクトリが異なるロケーションにある場合は、両方のコンパニオンに同じパスのディレクトリを作成します。このディレクトリは、実際の *$SYBASE* リリース・ディレクトリへのシンボリック・リンクとして機能します。

  たとえば、プライマリ・コンパニオン MONEY1 がリリース・ディレクトリ */usr/u/sybase1* を使用し、PERSONNEL1 がリリース・ディレクトリとして */usr/u/sybase2* を使用していても、*$SYBASE* は同じパスを指すようにします。

  MONEY1 と PERSONNEL1 はどちらも、対応する *$SYBASE* リリース・ディレクトリへのシンボリック・リンクとして確立されている */sybase* を使用します。MONEY1 では、*/sybase* は */usr/u/sybase1* へのリンクであり、PERSONNEL1 では、*/sybase* は */use/u/sybase2* へのリンクです。

$SYBASE をローカル・ファイル・システム上にインストールした場合は、両ノードの *$SYBASE/$SYBASE_ASE/install* に、それぞれのコンパニオンの *RUNSERVER* ファイルもコピーしてください。

sybha 実行プログラム

Adaptive Server High Availability Basis Services ライブラリは、*$SYBASE/$SYBASE_ASE/bin* にある *sybha* を呼び出します。*sybha* は、所有権とパーミッションを変更することにより実行可能になります。*$SYBASE/$SYBASE_ASE/install* の *sybhauser* という名前のファイルも編集する必要があります。このファイルには、そのクラスタに対してシステム管理者権限を持つユーザのリストがあります。クラスタに対するシステム管理者権限を持つユーザの数を制限することを強くおすすめします。

“root” 権限で、次の作業を行います。

1 *sybhagrp* という新しいグループを追加します。このグループを */etc/group* ファイルに追加することも、または NIS マップに追加することもできます。このグループに sybase ユーザを追加します (これは *$SYBASE* ディレクトリを所有するユーザです)。サーバの起動時に sybase ユーザがデータ・サーバを実行します。複数のサーバを実行しており、各サーバの *$SYBASE* ディレクトリを異なるユーザが所有している場合は、すべてのユーザをこのグループに追加してください。

2 “sybase” ユーザをグループ “hacmp” に追加します。このグループは、HACMP クラスタ・ソフトウェアのインストール時に作成されます。

3 *$SYBASE/$SYBASE_ASE/bin* ディレクトリに変更します。

4 *sybha* の所有権を “root” に変更します。

```
chown root sybha
```

5 *sybha* プログラムのグループを *sybhagrp* に変更します。

```
chgrp sybhagrp sybha
```

6 *sybha* のファイル・パーミッションを 4550 に修正します。

```
chmod 4550 sybha
```

7 *$SYBASE/$SYBASE_ASE/install* ディレクトリに変更します。

```
cd $SYBASE/$SYBASE_ASE/install
```

8 sybase ユーザを *sybhauser* ファイルに追加します。追加するユーザのログインは、Adaptive Server のログインではなく、UNIX の形式のログイン ID である必要があります。次に例を示します。

```
sybase
coffeecup
spooner
venting
howe
```

9 *sybhauser* の所有権を “root” に変更します。

```
chown root sybhauser
```

10 *sybhauser* のファイル・パーミッションを修正します。

```
chmod 600 sybhauser
```

設定パラメータの確認

次の設定パラメータを有効にしてから、Adaptive Server をフェールオーバ用に設定します。

- enable CIS － コンポーネント統合サービス (CIS) を有効にします。この設定パラメータは、デフォルトで有効です。
- enable xact coordination － 分散トランザクション管理 (DTM) を有効にします。この設定パラメータは、デフォルトで有効です。
- enable HA － 高可用性システムで、Adaptive Server をコンパニオンとして機能させます。enable HA は、デフォルトでは無効です。この設定は静的であるため、Adaptive Server を再起動して変更内容を有効にする必要があります。また、このパラメータを有効にすると、高可用性システムで Adaptive Server を起動したというメッセージがエラー・ログに書き込まれます。

『システム管理ガイド 第 1 巻』の「第 5 章 設定パラメータ」を参照してください。

マスタ・ログへのスレッショルドの追加

スレッショルドをマスタ・ログに追加していない場合は、追加してください。

1　ダンプ・トランザクションが発生する前に、master データベースのログに対して sp_thresholdaction を定義して実行し、残りのページ数にスレッショルドを設定します。Sybase では sp_thresholdaction を提供していません。このシステム・プロシージャの作成の詳細については、『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。

2　master と sybsystemprocs のログ・セグメントが満杯にならないように、それぞれのスレッショルドを設定します。

```
sp_addthreshold "master", "logsegment", 250, sp_thresholdaction
sp_addthreshold "sybsystemprocs", "logsegment", 250, sp_thresholdaction
```

3　プライマリ・サーバとセカンダリ・サーバで両方の手順を実行し、両方のサーバを再起動して静的設定パラメータを有効にします。

マスタ・デバイス以外の新しいデフォルト・デバイスの作成

新しくインストールした Adaptive Server では、マスタ・デバイスがデフォルト・デバイスです。つまり、データベース (フェールオーバによって使用されるプロキシ・データベースも含む) を作成すると、そのデータベースは自動的にマスタ・デバイス上に作成されます。しかし、ユーザ・データベースをマスタ・デバイスに追加すると、システム障害時に、マスタ・デバイスをリストアすることが困難になります。そのため、マスタ・デバイスには極力余分なユーザ・データベースを置かないようにします。それには、disk init コマンドを使用し、新しいデバイスを作成します。sp_diskdefault を使用して、新しいデバイスをデフォルトとして指定します。

たとえば、money1_default1 という名前の新しいデフォルト・デバイスを MONEY1 Adaptive Server に追加するには、次のように入力します。

```
sp_diskdefault money1_default1, defaulton
```

次のコマンドを発行して、明確にデフォルトとしての設定を無効にしないかぎり、マスタ・デバイスはデフォルト・デバイスのままです。

```
sp_diskdefault master, defaultoff
```

disk init および sp_diskdefault 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。

sysservers へのローカル・サーバの追加

sp_addserver を使用して、ローカル・サーバを sysservers にローカル・サーバとして追加します。サーバには *interfaces* ファイルで指定したネットワーク名を付けます。たとえば、MONEY1 というコンパニオンが、*interfaces* ファイルで指定されている MONEY1 というネットワーク名を使う場合は、次のように入力します。

```
sp_addserver MONEY1, local, MONEY1
```

この変更を有効にするには、Adaptive Server の再起動が必要です。

sysservers へのセカンダリ・コンパニオンの追加

セカンダリ・コンパニオンを sysservers にリモート・サーバとして追加します。

```
sp_addserver server_name
```

デフォルトでは、Adaptive Server は、srvid が 1000 のサーバを追加します。この変更は、Adaptive Server を再起動しなくても有効になります。

installhasvss スクリプトの実行

注意 *installhasvss* を実行する前に、「両方の Adaptive Server のエントリを interfaces ファイルに追加する」(78 ページ) に記載されている作業を行ってください。この作業を行わずに *installhasvss* を実行した場合は、後で *installmaster* を再実行して、すべてのシステム・ストアド・プロシージャを再インストールする必要があります。

installhasvss スクリプトにより、次の作業が行われます。

- フェールオーバに必要なストアド・プロシージャ (sp_companion など) のインストール
- SYB_HACMP サーバの sysservers へのインストール

installhasvss を実行するには、システム管理者権限が必要です。

installhasvss は、*$SYBASE/$SYBASE_ASE/scripts* ディレクトリにあります。スクリプトを実行するには、次のように入力します。

```
$SYBASE/$SYBASE_OCS/bin/isql -Usa -Ppassword -Sservername
<../scripts/installhasvss
```

installhasvss は、ストアド・プロシージャや SYB_HACMP サーバを作成するときにメッセージを表示します。

システム管理者への *ha_role* の割り当て

sp_companion を実行するには、両方の Adaptive Server で ha_role を持っている必要があります。ha_role を割り当てるには、isql から以下を発行します。

```
sp_role "grant", ha_role, sa
```

一度ログアウトしてから、再度 Adaptive Server にログインして、変更内容を有効にしてください。

IBM AIX サブシステムを Sybase フェールオーバ用に設定する

この項では、IBM AIX をフェールオーバ用に設定する手順について説明します。

ASE_HA.sh スクリプトの修正

ASE_HA.sh スクリプトは、高可用性の環境で Adaptive Server の起動、停止、モニタに使用します。このスクリプトは Adaptive Server の *$SYBASE/$SYBASE_ASE/install* ディレクトリにあります。このスクリプトをコピーして、クラスタ内の両方の Adaptive Server の環境に合わせて修正してください。このスクリプトに加える変更内容は、プライマリ・コンパニオン用のスクリプトなのか、セカンダリ・コンパニオン用のスクリプトなのかによって異なります。各ノードでは、このスクリプトのコピーを同じロケーションに置く必要があります (たとえば、両方のノードでこのスクリプトが */usr/u/sybase* にあるようにします)。また、両方のコピーは “root” 権限に対して読み込み、書き込み、実行のパーミッションが必要です。これを簡単に行うには、最初に同じノード上で両方のスクリプトを修正し、両方のスクリプトをもう一方のノードにコピーします。その後で、両方のノード上でスクリプトに適切なパーミッションを設定します。

使用している環境に合わせてスクリプトを修正するには、次の手順に従います。

1　*$SYBASE/$SYBASE_ASE/install* ディレクトリに変更します。

2　“root” として、*ASE_HA.sh* を HACMP イベント・ハンドラのスクリプト・ディレクトリ (通常は */usr/sbin/cluster/events*) にコピーし、次のように入力します。ここで、*servername* はモニタ対象の Adaptive Server です。

```
RUNHA_<servername>.sh
```

3　*RUNHA_<servername>.sh* スクリプトを、使用する環境に合わせて編集します。オリジナルの *ASE_HA.sh* スクリプトには、次に説明する変数があります。サイトに合わせて、“__FILL_IN__” を含む行や、その他編集が必要な行の値を編集してください。

- *MONITOR_INTERVAL* − データ・サーバ・プロセスの動作を確認するチェックの時間間隔 (秒単位)。この間、*RUNHA_<servername>.sh* は待機します。
- *RECOVERY_TIMEOUT* − 高可用性システムの最大待ち時間 (秒単位)。この時間を超過すると、コンパニオンが起動しなかったとみなされます。この値には、ロードされているコンパニオンが再起動するために十分な時間を設定してください。また、*RECOVERY_TIMEOUT* はサブシステムの、フェールオーバからフェールバックまでの処理完了の最大待ち時間としても使用されます。
- *SHUTDOWN_TIMEOUT* − コンパニオンがシャットダウンするまでの高可用性システムの最大待ち時間。この時間を超過すると、コンパニオンを強制終了します。

注意 この値は、HACMP の wait time パラメータが config_too_long ステータスに入る時間よりも、必ず短く設定します。デフォルトでは、360 秒に設定されています。サーバの起動時間がこれよりも長くかかる場合は、次のコマンドを実行してこの値を再設定します。

```
chssys -s clstrmgr -a "-u milliseconds_to_wait"
```

- *RESPONSE_TIMEOUT* − 単純なクエリが結果セットを戻すまでのサブシステムの最大待ち時間。コンパニオン・サーバがハングしているかどうかの診断に使用されます。たとえば、60 秒間経過しても isql で接続を確立できない場合は、自動的にタイム・アウトして終了します。しかし、isql が正常に接続しても、結果セットを戻さない場合は、RESPONSE_TIMEOUT によってコンパニオン・サーバが応答を停止しているとみなされます。デフォルトでは、RESPONSE_TIMEOUT には 999999 が設定されています。
- *ASE_FAILOVER* − 次のように設定します。
  - yes − コンパニオン・サーバのハング・プロセスやデッド・プロセスをモニタし、デバイスがセカンダリ・ノードに対してフェールオーバすると、このノードの HACMP サービスが停止されます。また、サーバで sp_companion configure を実行してください。
  - no − プライマリ・コンパニオンがフェールオーバしても、このノードの HACMP サブシステムを停止させません。この設定は、保守や再設定でコンパニオンを停止する必要がある場合に役立ちます。

    非対称型に設定している場合は、*ASE_FAILOVER* には no を設定してください。

警告！ 両方のサーバで Adaptive Server バージョン 12.0 以降を実行している場合のみ、ASE_FAILOVER を “yes” に設定します。バージョンが 12.0 より前の場合は、ASE_FAILOVER を “no” に設定します。

- *BASIC_FAILOVER* − 次のように設定します。
  - yes − フェールオーバが機能するモードでサーバが稼働している場合は、HACMP サブシステムに備わっているフェールオーバ・メカニズムが使用されます。フェールオーバが発生すると、HACMP サブシステム・モニタは、まず、コンパニオンがフェールオーバを実行できるモードで動作しているかどうかを確認します。コンパニオンで Sybase のフェールオーバが機能しない場合 (つまり、enable ha に 1 が設定されている場合) や、シングルサーバ・モードで実行している場合、またはセカンダリ・コンパニオンが停止している場合は、HACMP サブシステム・モニタは BASIC_FAILOVER が設定されているかどうかを確認します。設定されていると、モニタはプライマリ・コンパニオンをセカンダリ・ノードで起動しようとします。
  - no − Sybase のフェールオーバ基準を満たさない場合でも、フェールオーバのモード 0 に復元されません。つまり、BASIC_FAILOVER に no を設定すると、ノード・レベルでもコンパニオン・レベルでもフェールオーバは発生しません。
- retry − HACMP サブシステムがローカル・ノードで再起動を試みる回数。この回数を超えると、フェールオーバが発生します。非対称型設定の場合は、セカンダリ・コンパニオンがダウンしたときに再起動しやすくするため、大きい数値を設定してください。デフォルトでは 0 が設定されています。これは、コンパニオンがダウンしても、同じノード上では再起動しないことを意味します。
- *SYBASE_ASE* − Sybase Adaptive Server 製品のインストール・ディレクトリ。
- *SYBASE_OCS* − Sybase オープン・クライアント製品のインストール・ディレクトリ。
- *PRIM_SERVER* − プライマリ・コンパニオンの名前。
- *SEC_SERVER* − セカンダリ・コンパニオンの名前。
- *PRIM_HOST* − hostname によって返されるプライマリ・ホストの名前。
- *SEC_HOST* − hostname によって返されるセカンダリ・ホストの名前。

警告！ *PRIM_HOST* と *SEC_HOST* には、hostname コマンドによって返される文字列を設定します。設定されていない場合、フェールオーバまたはフェールバックが適切に行われないことがあります。

- *PRIM_SYBASE* − *$SYBASE* 環境変数を設定するプライマリ・ホスト上のディレクトリ。ローカル・デバイスを使用している場合は、ロケーションは両ノードで同じにします。共有デバイスを使用している場合は、異なるロケーションに設定します。
- *SEC_SYBASE* − *$SYBASE* 環境変数を設定するセカンダリ・ホスト上のディレクトリ。ローカル・デバイスを使用している場合は、ロケーションは両ノードで同じにします。共有デバイスを使用している場合は、異なるロケーションに設定します。
- *PRIM_SYBASE_HOME* − Adaptive Server 製品がインストールされているプライマリ・ホストのディレクトリへのパス。通常は、*$SYBASE/$SYBASE_ASE* です。
- *SEC_SYBASE_HOME* − Adaptive Server 製品がインストールされているセカンダリ・ホストのディレクトリへのパス。通常は、*$SYBASE/$SYBASE_ASE* です。
- *PRIM_ISQL* − プライマリ・ホスト上の isql バイナリへのパス。
- *SEC_ISQL* − セカンダリ・ホスト上の isql バイナリへのパス。
- *HA_LOGIN* − sa_role と ha_role を持つユーザのログイン。これは、プライマリ・コンパニオンとセカンダリ・コンパニオンで同一にします。
- *HA_PWD* − HA_LOGIN のパスワード。これは、プライマリ・コンパニオンとセカンダリ・コンパニオンで同一にします。
- *PRIM_RUNSCRIPT* − プライマリ・コンパニオンを起動するために使用する *RUNSERVER* ファイルの名前。
- *PRIM_CONSOLE_LOG* − 現在のプライマリ・コンパニオン・セッションのエラー・ログへのフル・パス。十分な空き領域があり、“root” によって書き込み可能であれば、どのファイルでもかまいません。デフォルトは *$SYBASE/$SYBASE_ASE/install* です。
- *SEC_CONSOLE_LOG* − 現在のセカンダリ・コンパニオン・セッションのエラー・ログへのフル・パス。十分な空き領域があり、“root” によって書き込み可能であれば、どのファイルでもかまいません。デフォルトは *$SYBASE/$SYBASE_ASE/install* です。

4 スクリプトをプライマリ・コンパニオン用に編集します。

5 スクリプトをセカンダリ・コンパニオン用に編集します。これらの値は、非対称型または対称型のどちらの設定を使用しているかによって異なります。

非対称型設定の場合は、*PRIM_SERVER* の値を *SEC_SERVER* (セカンダリ・コンパニオンの名前) と同じにしてください。*PRIM_HOST* は *SEC_HOST* と同じに、*PRIM_SYBASE* は *SEC_SYBASE* と同じにしてください。

対称型設定の場合は、セカンダリ・コンパニオンのスクリプトにある PRIM_SERVER、PRIM_HOST、PRIM_SYBASE、SEC_SERVER、SEC_HOST、SEC_SYBASE には、プライマリ・コンパニオンのスクリプトの値と逆の値を設定してください。

HACMP でのリソース・グループの設定

注意 この項で説明する手順は、コマンド・ラインからでも、設定ユーティリティ SMIT からでも実行できますが、SMIT ユーティリティの起動と使用については、IBM のマニュアルを参照してください。このマニュアルでは、SMIT を使用してリソース・グループを設定する方法について説明します。以下の手順は HACMP5 2 を使用していることを前提とします。これより後のバージョンを使用している場合は、使用しているオペレーティング・システムのマニュアルを参照してください。コマンド・ラインからリソース・グループを設定する方法の詳細については、IBM HACMP for AIX のマニュアルを参照してください。

両ノードのクラスタ・サービスを「安全」モードでシャットダウンします。その後、プライマリ・ノードの起動用 IP アドレスに “root” 権限でログインし、次の作業を行います。

1　コマンド・ラインから SMIT を起動します (smit hacmp)。

2　[Extended Configuration] メニューから次の操作を行います。

- [Extended Resource Configuration] を選択します。
- [HACMP Extended Resource Group Configuration] を選択します。
- 新しいリソース・グループを作成する場合は、[Add a Resource Group] を選択し、既存のリソース・グループを変更する場合は [Change/Show a Resource Group] を選択します。

3　リソース・グループを定義するときは、適切な起動、フェールオーバ、およびフォールバック方式を選択します。

プライマリ・コンパニオンのリソース・グループは、次のように設定します。

フィールド名	入力値
Resource Group Name	[<*resgrp1*>]
Participating Nodes	[<*primary_node*> <*secondary_node*>]
(デフォルトのノード優先度)	
Start Policy	“Online On Home Node Only”
Fallover Policy	“Fallover To Next Priority Node In The List”
Fallback Policy	“Fallback To Higher Priority Node In The List”

[OK] をクリックします。

セカンダリ・コンパニオンのリソース・グループ (非対称型フェールオーバ設定) は次のように設定します。

フィールド名	入力値
Resource Group Name	[<*resgrp2*>]
Participating Nodes (デフォルトのノード優先度)	[<*secondary_node*>]
Start Policy	“Online On Home Node Only”
Fallover Policy	“Fallover To Next Priority Node In The List”
Fallback Policy	“Fallback To Higher Priority Node In The List”

[OK] をクリックします。

セカンダリ・コンパニオンのリソース・グループ (対称型フェールオーバ設定) は次のように設定します。

フィールド名	入力値
Resource Group Name	[<*resgrp2*>]
Participating Nodes (デフォルトのノード優先度)	[<*secondary_node*> <*primary_node*>]
Start Policy	“Online On Home Node Only”
Fallover Policy	“Fallover To Next Priority Node In The List”
Fallback Policy	“Fallback To Higher Priority Node In The List”

[OK] をクリックします。

注意 非対称型高可用性設定のセカンダリ・コンパニオンの場合は、起動方式と終了方式を “Bring Offline (On Error Node Only)” と “Never Fallback” に設定することもできます。

4 プライマリ・コンパニオンとセカンダリ・コンパニオンを、HACMP のアプリケーション・サーバとして定義します。[Extended Configuration] メニューから次の操作を行います。

- [Extended Resource Configuration] を選択します。
- [HACMP Extended Resources Configuration] を選択します。
- [Configure HACMP Applications] を選択します。
- [Configure HACMP Application Servers] を選択します。
- 新しいアプリケーション・サーバを定義する場合は [Add Application Server]、既存のアプリケーション・サーバを変更する場合は [Change Application Server] を選択し、次の値を入力します。

フィールド名	入力値
Server Name	作成するこのアプリケーション・サーバに対応するコンパニオン・サーバ名。
Start Scrip	「ASE_HA.sh スクリプトの修正」(83 ページ) の項で作成した、このコンパニオン・サーバに対応するスクリプトのフル・パス名。起動スクリプトの引数として “monitor” と入力します。
Stop Script	「ASE_HA.sh スクリプトの修正」(83 ページ) の項で作成した、このコンパニオン・サーバに対応するスクリプトのフル・パス名。 プライマリ・コンパニオン・サーバおよび対称型セカンダリ・コンパニオン・サーバに対応する停止スクリプトの引数として “failover” を入力します。 非対称型セカンダリ・コンパニオン・サーバの引数として “stop” を入力します。
Application Monitor Name(s)	(空白)

情報を入力したら、[OK] をクリックします。

- 次に例を示します。

  プライマリ・コンパニオン・サーバ MONEY1 の場合は、次のようになります。

フィールド名	入力値
Server Name	[MONEY1]
Start Scrip	`[/usr/sbin/cluster/events/RUN_MONEY1.sh monitor]`
Stop Script	`[/usr/sbin/cluster/events/RUN_MONEY1.sh failover]`
Application Monitor Name(s)	[ ]

対称型高可用性設定のセカンダリ・コンパニオン・サーバ PERSONNEL1 の場合は、次のようになります。

Server Name	[PERSONNEL1]
Start Scrip	`[/usr/sbin/cluster/events/RUN_PERSONNEL1.sh monitor]`
Stop Script	`[/usr/sbin/cluster/events/RUN_PERSONNEL1.sh failover]`
Application Monitor Name(s)	[ ]

非対称型高可用性設定のセカンダリ・コンパニオン・サーバ PERSONNEL1 の場合は、次のようになります。

Server Name	[PERSONNEL1]
Start Scrip	`[/usr/sbin/cluster/events/RUN_PERSONNEL1.sh monitor]`
Stop Script	`[/usr/sbin/cluster/events/RUN_PERSONNEL1.sh stop]`
Application Monitor Name(s)	[ ]

非対称型高可用性設定のセカンダリ・リソース・グループでは [Participating Nodes] フィールドにセカンダリ・ノードのみがリストされるのに対し、対称型高可用性設定のプライマリ・リソース・グループでは両方のノードが反対の順序で記載されます。

5 アプリケーション・サーバ用に定義したリソース・グループをそれぞれ設定します。[Extended Configuration] メニューから次の操作を行います。

- [Extended Resource Configuration] を選択します。
- [HACMP Extended Resource Group Configuration] を選択します。
- [Changes/Show Resources and Attributes for a Resource Group] を選択します。
- [Single Select List] から、設定するリソース・グループを選択します。
- [アプリケーション・サーバ] フィールドに、手順 (4) でリソース・グループ用に定義した対応するアプリケーション・サーバ名 (プライマリ・リソース・グループのプライマリ・コンパニオン名、セカンダリ・リソース・グループのセカンダリ・コンパニオン・サーバ名など) を入力します。

  また、[IP Label]、[Volume Groups]、[File systems] といった必須フィールドにもすべて情報を入力します。この手順を各コンパニオンで行ってください。

6 クラスタ・リソースを同期させます。手順 1 から 5 までを実行したノードで smit を使用して、[Extended configuration] 画面を表示し、[Extended Verification and Synchronization] を選択します。これにより、変更内容が同一クラスタ内の全ノードに送信されます。

HACMP サービスを停止して、ノードを 2 台とも再起動してから同期させる必要がある場合もあります。同期によりエラーが発生しないことを確認してください。

フェールオーバ用コンパニオン・サーバの設定

この項では、高可用性システムで Adaptive Server をプライマリ・コンパニオンおよびセカンダリ・コンパニオンとして設定する手順を示します。

do_advisory オプションを指定して *sp_companion* を実行する

十分なリソースを持つセカンダリ・コンパニオンを設定して、フェールオーバ中でもサーバ 2 台分の作業を実行できるようにします。セカンダリ・コンパニオンは、正常なクラスタ・オペレーションを妨げる属性を持っている場合があります。たとえば、プライマリ・コンパニオンとセカンダリ・コンパニオンのユーザ・ログイン数がともに 250 に設定されていると、フェールオーバ中、セカンダリ・コンパニオンには、発生する可能性のあるユーザ・ログインの半分を処理するリソースしかないことになります。したがって、MONEY1 と PERSONNEL1 の両方でユーザ・ログイン数を 500 に設定します。

sp_companion do_advisory は、クラスタ・オペレーション (Adaptive Server をセカンダリ・コンパニオンとして設定するなど) が正常に行われるようにするために、プライマリ・コンパニオンとセカンダリ・コンパニオンの両方の設定オプションを確認します。また、sp_companion do_advisory は、変更する必要のある設定オプションを通知します。

sp_companion do_advisory オプションの詳細については、「第 6 章 do_advisory の実行」を参照してください。

非対称型コンパニオン設定の作成

プライマリ・コンパニオンを非対称型で設定するには、次のコマンドをセカンダリ・コンパニオンから発行します。

sp_companion "*primary_server_name*", configure, NULL, *login_name*, *password*

- *primary_server_name* – *interfaces* ファイルのエントリと sysservers に定義されているプライマリ Adaptive Server の名前。
- *login_name* – このクラスタ・オペレーションを行っているユーザの名前。ユーザは ha_role を持っている必要があります。
- *password* – このクラスタ・オペレーションを行っているユーザのパスワード。

注意　上記コマンドは、セカンダリ・コンパニオンからのみ実行します。

この例では、MONEY1 という名前の Adaptive Server をプライマリ・コンパニオンとして設定します。セカンダリ・サーバ PERSONNEL1 から次のコマンドを発行します。

```
sp_companion "MONEY1", configure, NULL, sa, Odd2Think
Server 'PERSONNEL1' is alive and cluster configured.
Step: Access verified from Server:'PERSONNEL1' to Server:'MONEY1'
Server 'MONEY1' is alive and cluster configured.
Step: Access verified from Server:'MONEY1' to Server:'PERSONNEL1'
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
.....
Step: Companion servers configuration check succeeded
Step: Server handshake succeeded
Step: Master device accessible from companion
Step: Added the servers 'PERSONNEL1' and 'MONEY1' for cluster config
Step: Server configuration initialization succeeded
Step: Synchronizing server logins from companion server
Step: Synchronizing remoteserver from companion server
Step: Synchronizing roles from companion server
```

```
Step: Synchronizing server-wide privs from companion server
Step: User information syncup succeeded
Step: Server configured in normal companion mode
```

sp_companion 設定を行っている間にユーザ・データベースが作成されると、次のようなメッセージが表示されます。

```
Step: Created proxy database 'pubs2'
 Step: Proxy status for database has been set. Please Checkpoint the database 'pubs2'
 Step: Server configured in normal companion modeî
 Starting companion watch thread
```

対称型設定の作成

非対称型フェールオーバを使用できるようにコンパニオンを設定した後、対称型設定に設定できます。対称型設定では、両方のサーバがプライマリ・コンパニオンとしても、セカンダリ・コンパニオンとしても機能します。対称型設定については、図 3-2 (21 ページ) を参照してください。

sp_companion をセカンダリ・コンパニオンから発行して、対称型設定に設定します。sp_companion の構文については、前述の「非対称型コンパニオン設定の作成」を参照してください。

次は、「非対称型コンパニオン設定の作成」(91 ページ) で説明した、MONEY1 という Adaptive Server をセカンダリ・コンパニオンとして、PERSONNEL1 という Adaptive Server に追加する例です。MONEY1 サーバから次のコマンドを発行します。

```
sp_companion 'PERSONNEL1', configure, sa, Think2Odd
Server 'MONEY1' is alive and cluster configured.
Step: Access verified from Server:'MONEY1' to Server:'PERSONNEL1'
Server 'PERSONNEL1' is alive and cluster configured.
Step: Access verified from Server:'PERSONNEL1' to Server:'MONEY1'
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
.....
Step: Companion servers configuration check succeeded
Step: Server handshake succeeded
Step: Master device accessible from companion
Step: Added the servers 'MONEY1' and 'PERSONNEL1' for cluster config
Step: Server configuration initialization succeeded
Step: Synchronizing server logins from companion server
Step: Synchronizing remoteserver from companion server
Step: Synchronizing roles from companion server
Step: Synchronizing server-wide privs from companion server
Step: User information syncup succeeded
Step: Server configured in normal companion mode
```

プライマリ・コンパニオンをモニタ対象リソースとして起動する

この項では、高可用性システムのモニタ対象のサービスとしてプライマリ・コンパニオンを起動する方法について説明します。

注意 保守または他の目的でプライマリ・サーバをシャットダウンする必要がないことを確認してから、プライマリ・コンパニオンのモニタを開始します。いったんモニタを開始したら、プライマリ・コンパニオンを停止する場合はプライマリ・コンパニオンをサスペンド・モードに移行する必要があります。シャットダウンが必要かどうかわからない場合は、*$SYBASE/$SYBASE_ASE/install* にある startserver スクリプトを使用してプライマリ・サーバを起動し、コンパニオンの設定を完了してから、この章に記載してある手順に従ってサーバを再起動します。

フェールオーバのモニタ対象リソースとしてプライマリ・コンパニオンを起動するには、次の手順に従います。

1　プライマリ・ノードの HACMP サービスを停止します。

2　*/tmp/hacmp.out* をチェックして node_down イベントが完了していることを確認します。その後、SMIT を使用するか、または “root” 権限で次のコマンドをコマンド・ラインで実行して、HACMP サービスを再起動します。

```
/usr/sbin/cluster/etc/rc.cluster -boot '-N' '-b'  '-i'
```

これによって、*RUNHA_<servername>.sh* モニタ・スクリプトが自動的に実行され、プライマリ・コンパニオンが起動し、クラッシュ時やハング時にプライマリ・コンパニオンをモニタします。

この処理をセカンダリ・ノードでも行い、セカンダリ・コンパニオンを起動します。

Sybase フェールオーバの管理

この項では、Sybase のフェールオーバの使い方を説明します。以下の手順は HACMP5 2 を使用していることを前提とします。これより後のバージョンを使用している場合は、使用しているオペレーティング・システムのマニュアルを参照してください。

プライマリ・ノードへのフェールバック

HACMP では、フェールバックが自動的に発生します。HACMP をプライマリ・ノードで起動すると、セカンダリ・ノードの *stop_server* イベントによりモニタ・スクリプトが呼び出され、sp_companion 'prepare_failback' が実行されます。

プライマリ・ノードにフェールバックするには、セカンダリ・コンパニオンがセカンダリ・フェールオーバ・モードになっていることを確認し、プライマリ・ノードの HACMP サービスを起動します。sp_companion 'prepare_failback' が正常に実行されたことを確認するには、*/tmp/hacmp.out* に次の文字列があることを検索します。

```
SYBASE HA MONITOR: Prepare_failback was successful.
```

注意 セカンダリ・ノードが立ち上がっていて、セカンダリ・コンパニオンがセカンダリ・フェールオーバ・モードで動作していることを確認してから、プライマリ・ノードで HACMP サービスを起動してください。セカンダリ・コンパニオンまたはセカンダリ・ノードが動作していない場合は、プライマリ・コンパニオンを起動しないでください。また、ノードが両方とも停止しているか、両ノードで HACMP サービスが停止している場合は、必ずセカンダリ・ノードとセカンダリ・ノードの HACMP サービスを再起動してから、プライマリ・ノードを再起動してください。

手動フェールバック

注意 自動フェールバック処理が失敗した場合は、ログを検査して、高可用性システムで次の手順が実行されたかどうか確認してください。実行されていなかった場合は、手動で手順を実行します。必ず次の順序に従って実行します。

---

1 プライマリ・ノードで HACMP サブシステムを takeover モードで停止します (詳細については、IBM のマニュアルを参照してください)。これによってプライマリ・コンパニオンがシャットダウンされ、そのリソースがセカンダリ・コンパニオンにフェールオーバします。

2 セカンダリ・コンパニオンを停止し、再起動します。RETRY に設定されている値が 1 以上の場合は、セカンダリ・コンパニオンを停止すると *RUNHA_<servername>.sh* によってコンパニオンが自動的に再起動されます。

3 *RUNHA_<servername>.sh* に記述されているように、HA_LOGIN として isql を使用してセカンダリ・コンパニオンにログインし、セカンダリ・コンパニオンがセカンダリ・フェールオーバ・モードで動作していることを確認します。

4　sp_companion 'prepare_failback' を発行します。たとえば、セカンダリ・コンパニオン PERSONNEL1 から フェールバックするには次のように入力します。

```
sp_companion MONEY1, 'prepare_failback'
```

5　プライマリ・ノードで HACMP を再起動します。

6　isql を使用してプライマリ・コンパニオンにログインし、プライマリ・コンパニオンがプライマリ・フェールバック・モードで動作していることを確認します。

7　sp_companion 'resume' を発行します。たとえば、プライマリ・コンパニオン MONEY1 のコンパニオン・モードを再開するには、次のように入力します。

```
sp_companion PERSONNEL1, 'resume'
```

注意　sp_companion resume を発行しないと、フェールオーバ・プロパティが設定されたクライアントを接続できません。sp_companion prepare_failback を発行してからクライアントを再接続しようとすると、クライアントは sp_companion resume を発行するまで応答を停止します。

コンパニオン・モードのサスペンド

保守のためにプライマリ・コンパニオンをシャットダウンする必要があるものの、セカンダリ・コンパニオンにフェールオーバしたくない場合は、一時的にコンパニオン・モードをサスペンドします。コンパニオン・モードがサスペンドされると、コンパニオン間の同期が発生せず、プライマリ・コンパニオンはセカンダリ・コンパニオンにフェールオーバできません。しかし、サスペンド・モードは、設定パラメータの変更などの管理タスクを実行するのに役立ちます。

1　次のコマンドを発行して、サスペンド・モードに移行します。

```
sp_companion <primary_server_name>, suspend
```

2　コンパニオン・サーバが停止してもフェールオーバがトリガされないように、モニタ・プロセスを強制終了します。“root” 権限で、次のコマンドを入力します。

```
ps -ef|grep "RUNHA_<servername>.sh monitor"
kill -9 <pid>
```

3　プライマリ・コンパニオンをシャットダウンします。

モニタ・プロセスを強制終了した後は、何度でもコンパニオン・サーバを停止でき、フェールオーバも発生しません。

サスペンド・モード中にコンパニオンを再起動する

$SYBASE/$SYBASE_ASE/install にある起動スクリプトを使用して、プライマリ・コンパニオンをモニタせずに再起動するには次のように入力します。

```
startserver -f ./RUN_<server_name>
```

このスクリプトを使用してコンパニオン・サーバを起動すると、サーバがフェールオーバするように設定されていても、サーバの停止時にフェールオーバは発生しません。この方法は、保守が目的であり、サーバがダウン中にサーバ・データベースにアクセスする必要がない場合にだけ使用してください。

ノーマル・コンパニオン・モードの再開

ノーマル・コンパニオン・モードを再開する手順は、その前のモードがサスペンド・モードか、フェールオーバ・モードかによって若干異なります。

サスペンド・モードからのノーマル・コンパニオン・モードの再開

サスペンド・モードの 2 つのコンパニオン間でノーマル・コンパニオン・モードを再開するには、次の手順に従います。

1 プライマリ・コンパニオンをシャットダウンします。

2 プライマリ・ノードの HACMP サービスを「安全」モードで停止します。

3 プライマリ・ノードで HACMP サービスを再起動します。

フェールオーバ・モードからのノーマル・コンパニオン・モードの再開

フェールオーバ・モードになっている 2 つのコンパニオン間でノーマル・コンパニオン・モードを再開するには、プライマリ・ノードで HACMP サービスを再起動し、次の手順に従います。

1 パラメータを指定せずに sp_companion を発行して、両方のコンパニオンがフェールバック・モードであることを確認します。

2 次のコマンドを発行してノーマル・コンパニオン・モードを再開します。

```
sp_companion secondary_server_name, resume
```

たとえば、プライマリ・コンパニオン PERSONNEL1 のノーマル・コンパニオン・モードを発行するには、次のように入力します。

```
sp_companion PERSONNEL1, resume
Server 'MONEY1' is alive and cluster configured.
Step: Access verified from Server:'MONEY1' to Server:'PERSONNEL1'
Server 'PERSONNEL1' is alive and cluster configured.
Step: Access verified from Server:'PERSONNEL1' to Server:'MONEY1'
Step: Checkin to See if the remote server is up
Step: Synchronizing server logins from companion server
```

```
Step: Synchronizing remoteserver from companion server
Step: Synchronizing roles from companion server
Step: Synchronizing server-wide privs from companion server
Step: User information syncup succeeded
sys_id ses_id      ses_id2     ses_status Purged from s1.
------ ----------- ----------- ---------- ------------------------------

(0 rows affected)
sys_id ses_id      ses_id2     ses_status Copied to s1.
------ ----------- ----------- ---------- ------------------------------

(0 rows affected)
sys_id ses_id      ses_id2     ses_status Purged from s2.
------ ----------- ----------- ---------- ------------------------------
(0 rows affected)
Step: Syssession information syncup succeeded
```

コンパニオン・モードの削除

コンパニオン・モードを削除するには、次のコマンドを発行します。

```
sp_companion companion_name, "drop"
```

コンパニオン・モードを削除したら、二度と元に戻すことはできません。フェールオーバ機能を保持するには、Adaptive Server のコンパニオン・サーバを再設定する必要があります。ただし、Adaptive Server が動作しているノードは、引き続き高可用性システムでモニタされます。

RUNHA_<servername>.sh スクリプトの実行中にコンパニオン・モードを削除しても、サーバの停止または終了状態を、このスクリプトが引き続きモニタします。サーバをシャットダウンしてもノードをフェールオーバしたくない場合は、次のコマンドを発行して、モニタ・プロセスを強制終了します。

```
kill -9 `ps -ef | grep "RUNHA_<servername>.sh monitor" | grep -v grep | awk
     '(print $2}'`
```

モニタ・プロセスを強制終了しないと、モニタ・プロセスは、コンパニオンの停止を検知したときに、リソースのフェールオーバをトリガし、RETRY と BASIC_FAILOVER の設定によって、プライマリ・ノードまたはセカンダリ・ノードからコンパニオンを再起動しようとします。

HACMP for AIX でのフェールオーバのトラブルシューティング

この項では、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。

エラー・メッセージ 18750

コンパニオン・サーバによってエラー・メッセージ 18750 が発行された場合は、サーバの *@@cmpstate* を確認します。プライマリ・コンパニオンがノーマル・コンパニオン・モードになっていても、セカンダリ・コンパニオンがセカンダリ・フェールオーバ・モードになっている場合、クラスタは整合性が失われた状態に陥っています。その場合は、手動でリカバリする必要があります。整合性が失われた状態は、sp_companion 'prepare_failback' コマンドがセカンダリ・コンパニオンで失敗することによって発生する可能性があります。失敗しているかどうかは、セカンダリ・ノードの HACMP ログを調べるとわかります。このログは、*/tmp/hacmp.out* にあります。リカバリするには、次の手順に従います。

1 プライマリ・コンパニオンとセカンダリ・コンパニオンを両方ともシャットダウンします。

2 セカンダリ・コンパニオンを再起動します。

3 “suspect” のマークの付いたデータベースをすべて修復します。この状態のデータベースを検索するには、次のコマンドを発行します。

```
select name, status from sysdatabases
```

suspect のマークの付いたデータベースは、status の値が 320 になっています。

4 システム・テーブルの更新を許可するには、以下を入力します。

```
sp_configure "allow updates", 1
```

5 suspect 状態のフェールオーバ・データベースに対して、それぞれ次を実行します。

```
1> update sysdatabases set status=status-256 where name='database_name'
2> go
1> dbcc traceon(3604)
2> go
1> dbcc dbrecover(database_name)
2> go
```

6 セカンダリ・コンパニオンから、次のコマンドを発行します。

```
sp_companion primary_companion_name, prepare_failback
```

このコマンドが正常に実行されることを確認します。

7 プライマリ・ノードで HACMP サービスを再起動します。

失敗した *prepare_failback* からのリカバリ

フェールバック中に、prepare_failback がセカンダリ・コンパニオンで正常に実行されたにもかかわらず、プライマリ・コンパニオンが起動できない場合は、次を実行してロールバックした後に、prepare_failback コマンドを再発行します。

1　プライマリ・コンパニオンのエラー・ログと HACMP のエラー・ログをチェックし、サーバが起動できなかった原因を調べ、問題を解決します。

2　takeover を使用してプライマリ・ノードの HACMP サービスを停止します。

3　LOGIN_NAME としてセカンダリ・コンパニオンにログインし、次のように発行します。

```
dbcc ha_admin ("", "rollback_failback")
dbcc ha_admin ("", "rollback_failover")
```

両方のコンパニオン・サーバが、フェールオーバ・モードに戻ります。

4　プライマリ・ノードで HACMP を再起動します。

エラー・ログのロケーション

Sybase のフェールオーバには次のログが含まれます。これらのログは、フェールオーバ時に発生したエラーの調査や診断に役立つ場合があります。

- */tmp/hacmp.out* － HACMP アクティビティの出力や、*RUNHA_<servername>.sh* モニタ・スクリプトからの出力を格納します。一般的な HACMP の失敗については、“ERROR” という文字列を探します。*RUNHA_<servername>.sh* スクリプトからの出力については、“SYBASE HA MONITOR” を探します。

  失敗の原因がわかったら、それを訂正します。そして、SMIT の [Cluster Recovery Aids] 画面を表示し、[Recover From Script Failure] を実行してから処理を継続します。

  ノードが特定のファイル・システムに十分な領域を保持してない場合は、フェールオーバ処理かフェールバック処理の途中で HACMP が応答を停止します。そして、このハングが原因で config_too_long がロックされます。このような状態になったら、ディレクトリをすべてクリーンアップしてください。そして、SMIT を起動し、[Cluster Recovery Aids] 画面の [Recover From Script Failure] を実行してから、処理を継続します。

- $PRIM_CONSOLE_LOG － このログのロケーションは *RUNHA_<servername>.sh* モニタ・スクリプトに定義されています。このエラー・ログには、*RUNHA_<servername>.sh* スクリプトを最後に実行したときからの Adaptive Server 情報が含まれています。

Adaptive Server のアップグレード

高可用性設定内の Adaptive Server をアップグレードするには、一時的にプライマリ・コンパニオンとセカンダリ・コンパニオンの間のコンパニオン関係を削除し、Adaptive Server リソース・グループのモニタリングを無効にする必要があります。これにより、アップグレード・プロセスの間、HACMP クラスタによって予期しないフェールオーバがトリガされることなく、各 Adaptive Server を独立して停止または再起動できます。

注意 アップグレード・プロセスの間は、データベース、オブジェクト、ユーザ、またはログインを追加、削除、修正できません。コンパニオン関係を削除した後、その関係を再確立する前にこのような変更を行うと、アップグレードが失敗する場合があります。または、サーバ間の不整合が原因でクラスタが不安定になることがあります。

❖ モニタリング・サービスの停止とコンパニオン関係の削除

1 root 権限で、次のコマンドを発行してリソース・グループをオフラインにします。

```
dbcc traceoff(2209)
clRGmove -g secondary_resource_group -d -s false
clRGmove -g secondary_resource_group -d -s true
clRGmove -g group_name -d -s false
clRGmove -g group_name -d -s true
```

SMIT を使用することもできます (SMIT のユーザ・マニュアルを参照してください)。

2 クラスタ内のすべてのノードで、モニタリング・サービスを停止します。root 権限で、次のコマンドを発行します。

```
ps -ef | grep "RUNHA_server_name.sh monitor"
kill -9 pid
```

モニタ・プロセスを強制終了した後は、何度でもコンパニオン・サーバを停止でき、フェールオーバも発生しません。

3 セカンダリ・コンパニオンから、次のコマンドを発行します。

```
sp_companion primary_server_name, "drop"
```

4 (対称型設定の場合) セカンダリのコンパニオン関係をプライマリ・コンパニオンから削除します。

```
sp_companion secondary_server_name,"drop"
```

5 両方のノードがシングルサーバ・モードであることを確認します。各ノードで次のコマンドを発行します。

```
sp_companion
```

コンパニオンがシングルサーバ・モードである場合は、次のような結果になります。

```
Server 'server_name' is not cluster configured.
Server 'server_name' is currently in 'Single server' mode.
```

❖ **Adaptive Server のアップグレード**

1 各ノードで、高可用性を無効にします。

```
sp_configure 'enable HA', 0
```

Adaptive Server を再起動して、この変更を有効にします。

2 『インストール・ガイド』の手順に従って各サーバをアップグレードします。

3 すべてのノードで、高可用性を再度有効にします。

```
sp_configure 'enable HA', 1
```

Adaptive Server を再起動して、変更を有効にします。

4 アップグレードしたサーバに、スクリプト (*installmaster*、*installhasvss*、*installsecurity* など) を再インストールします。詳細については、「installmaster の再インストール」と「installhasvss の再実行」(196 ページ) を参照してください。*installmaster* を再インストールする場合は、*installhasvss* を再インストールする必要があります。

5 sybha バイナリと *sybhausers* ファイルのパーミッションが正しく設定されていることを確認します。

root 権限で、次のコマンドを *$SYBASE/$SYBASE_ASE/bin* から発行します。

```
chown root sybha
chgrp sybhagrp sybha
chmod 4550 sybha
```

root 権限で、以下の手順を *$SYBASE/$SYBASE_ASE/install* から実行します。

1 sybase ユーザが *sybhauser* ファイルに含まれていることを確認します。

2 次のコマンドを発行します。

```
chown root sybhauser
chmod 600 sybhauser
```

6 次の内容を確認します。

- */usr/sbin/cluster/event/RUNHA_server_name.sh* スクリプトで、リソース、リソース・グループ・プロパティ、または新しいインストール環境内の高可用性に関係するファイル (*PRIM_SYBASE_HOME*、*PRIM_RUNSCRIPT*、*PRIM_CONSOLE_LOG* など) に変更が正しく反映されている。

- 「高可用性で動作するように Adaptive Server を準備する」(77 ページ) および「IBM AIX サブシステムを Sybase フェールオーバ用に設定する」(83 ページ) で説明されている、コンパニオン関係の確立に必要なすべての手順を実施しており、システムがアップグレードの完了後もこれらの変更を保持している。

❖ **コンパニオン関係の再確立とパッケージ・モニタリングの再開**

1 各ノードで、Adaptive Server を手動で再起動します。

2 root 権限で、次のコマンドを発行してクラスタのモニタリング・サービスをリストアします。このコマンドによって、*RUNHA_server_name.sh* モニタリング・スクリプトが自動的に実行されます。

```
/usr/sbin/cluster/etc/rc.cluster -boot '-N' '-b' '-i'
```

3 「フェールオーバ用コンパニオン・サーバの設定」(90 ページ) で説明されている、コンパニオン関係を確立するための前提条件となる手順が完了していることを確認します。

4 サーバ間のコンパニオン関係を再確立します。セカンダリ・サーバで、次のコマンドを発行します。

```
dbcc traceon (2209)
sp_companion primary_server_name,configure
```

注意 対称型設定の場合は、両方のコンパニオンでこのコマンドを発行します。

セカンダリ・サーバにユーザ・データベースが含まれる場合は、次のような警告メッセージが表示されることがあります。このようなメッセージは無視できます。

```
Msg 18739, Level 16, State 1:
Server 'server_name', Procedure 'sp_hacmpcfgvrfy', Line 102:
Database 'database_name': a user database exists. Drop this
database and retry the configuration again.
```

5 適切なノードでリソース・グループを再起動します。root 権限で、プライマリ・ノード上で次のコマンドを入力します。

```
clRGmove -g group_name -u -s false
```

root 権限で、セカンダリ・ノード上で次のコマンドを入力します。

```
clRGmove -g group_name -u -s true
```

6 sp_companion を実行して、システムがフェールオーバ用に適切に設定されていることを確認します。フェールオーバとフェールバックを確認します。

第 9 章

Sun Cluster 3.0 および 3.1 のアクティブ／アクティブ設定

この章では、Sun Cluster 3.0 および 3.1 での Adaptive Server Enterprise のアクティブ／アクティブ設定について説明します。

トピック名	ページ
ハードウェアとオペレーティング・システムの稼働条件	103
アクティブ／アクティブ設定のための Adaptive Server の準備	106
Sun Cluster サブシステムの設定	112
フェールオーバ用コンパニオン・サーバの設定	123
Sybase フェールオーバの管理	128
Sun Cluster での高可用性の確認	130
リソース・グループの手動設定	131
Adaptive Server のアップグレード	136
トラブルシューティング	139

Adaptive Server Enterprise バージョン 15.0 は Sun Cluster バージョン 2.2 をサポートしていません。現在これらのクラスタを設定している場合は、それぞれのクラスタ・バージョンをアップグレードして、Sun Solaris で高可用性が使用できるように Adaptive Server 15.0 を設定する必要があります。

ハードウェアとオペレーティング・システムの稼働条件

高可用性の要件は次のとおりです。

- CPU やメモリなどのリソースに関して同様に設定されている、2 台の同種のネットワーク・システム
- 高可用性パッケージと関連ハードウェア
- 両方のノードからアクセス可能なデバイス
- さまざまなクラスタ・ノードに割り当てるデバイス・パス名をユニークに保つための論理ボリューム・マネージャ (LVM)
- マルチホスト・ディスク上のボリュームまたはディスク・スイート・オブジェクト
- メディア障害に対処するためのサードパーティ・ベンダのミラーリング

- プライマリ・ノードとセカンダリ・ノードにバインドできる論理ホスト名。対称型設定では 2 つの論理ホスト名 (それぞれがプライマリ・コンパニオンに対応) が必要。

稼働条件の詳細、およびプラットフォームに固有の高可用性ソフトウェアのインストールについては、Sun Cluster のマニュアルを参照してください。

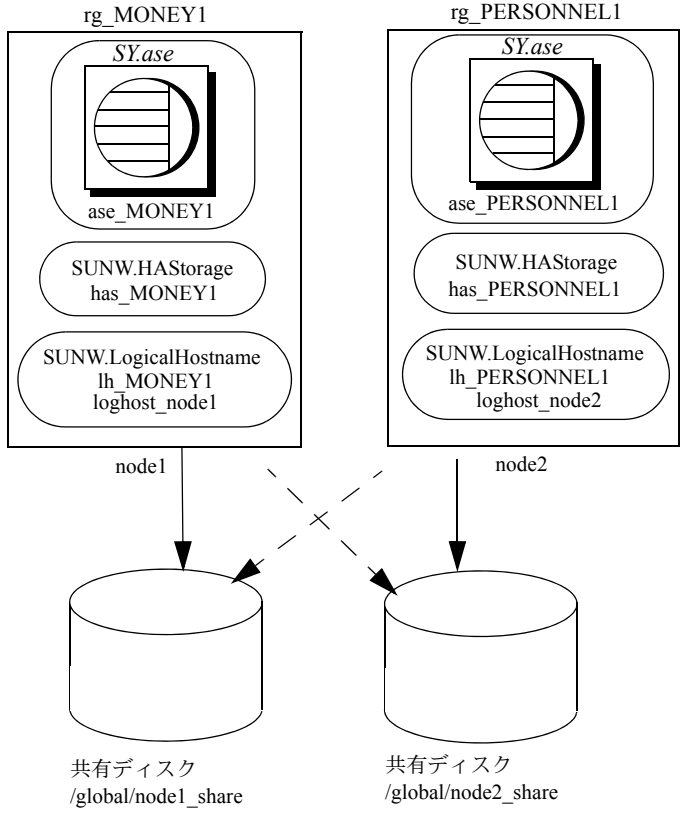
Sun Cluster のアクティブ／アクティブ設定

図 9-1 (105 ページ) は、Sun Cluster のアクティブ／アクティブ設定を示します。

Sun Cluster では、Adaptive Server はデータサービスとして実行され、Sun Cluster のリソースグループマネージャ (RGM) によって管理されます。Adaptive Server は、Adaptive Server リソースと必要なその他すべてのリソース (*SUNW.HAStorage*、*SUNW.HAStoragePlus*、*SUNW.LogicalHostname* など) を含むリソース・グループに関連付けられます。

SY.ase は Adaptive Server リソースであり、タイプ *SY.ase* のリソースについてさまざまな拡張プロパティを定義します。詳細については、「Adaptive Server リソースの拡張プロパティ」(117 ページ) を参照してください。標準リソース・プロパティの詳細については、Sun Cluster のマニュアルを参照してください。

図 9-1: Sun Cluster リソース・グループの設定例



この図では、対称型設定において *rg_MONEY1* と *rg_PERSONNEL1* という 2 つのリソース・グループがコンパニオン・サーバ MONEY1 と PERSONNEL1 に対応しています。

rg_MONEY1 は、リソース・タイプ *SY.ase* の *ase_MONEY1*、リソース・タイプ SUNW.HAStorage の *has_MONEY1*、およびリソース・タイプ SUNW.LogicalHostname の *lh_MONEY1* という 3 つのリソースで構成されています。ストレージ・リソース *has_MONEY1* は共有ディスク上の広域ファイル・システム */global/node1_share* を管理し、論理ホスト・リソース *lh_MONEY1* は論理ホスト名 *loghost_node1* を管理します。Adaptive Server リソース *ase_MONEY1* は、*has_MONEY1* と *lh_MONEY1* を使用します。

rg_PERSONNEL1 は、リソース・タイプ *SY.ase* の *ase_PERSONNEL1*、リソース・タイプ SUNW.HAStorage の *has_PERSONNEL1*、およびリソース・タイプ SUNW.LogicalHostname の *lh_PERSONNEL1* という 3 つのリソースで構成されています。ストレージ・リソース *has_PERSONNEL1* は、共有ディスクの広域ファイル・システム */global/node2_share* を管理し、論理ホスト・リソース *lh_PERSONNEL1* は、論理ホスト名 *loghost_node2* を管理します。Adaptive Server リソース *ase_PERSONNEL1* は、*has_PERSONNEL1* と *lh_PERSONNEL1* を使用します。

アクティブ／アクティブ設定のための Adaptive Server の準備

この項では、アクティブ／アクティブの高可用性を実現するための Adaptive Server の設定方法について説明します。

Adaptive Server のインストール

プライマリ・サーバとセカンダリ・サーバを別々のディスクにインストールし、両方のディレクトリ・パスを同じパスにします。プライマリ・コンパニオンは、新しくインストールした Adaptive Server でも、旧バージョンの Adaptive Server からアップグレードして既存のデータベースやユーザなどを受け継いだものでもかまいません。

セカンダリ・コンパニオンには、ユーザ・ログインやユーザ・データベースを使用しない、新しくインストールした Adaptive Server を使用します。これは、すべてのユーザ・ログインやデータベース名をクラスタ内でユニークにするためです。フェールオーバの設定が完了した後は、セカンダリ・コンパニオンにユーザ・ログインやデータベースを追加できます。

Adaptive Server のインストールと設定については、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。

両方の Adaptive Server のエントリを *interfaces* ファイルに追加する

プライマリ・コンパニオンとセカンダリ・コンパニオンの *interfaces* ファイルには、この両方のコンパニオンのエントリが必要です。*interfaces* ファイル内のサーバ・エントリには、sysservers に指定されているネットワーク名を使用してください。エントリを *interfaces* ファイルへ追加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。

interfaces ファイルに追加する各エントリについて、ホスト名は論理ホストでなければなりません。*/etc/hosts*、NIS ホスト・マップ、ディレクトリ・サービスなど、システムに適したものすべての論理ホストのエントリを作成する必要があります。*interfaces* ファイル内の論理ホスト名は、Sun Cluster サブシステムと連携するように Adaptive Server を設定したときに、SUNW.LogicalHostname リソースを追加する scrgadm コマンドの -l (L の小文字) パラメータで使用したのと同じ名前にしてください。

次に示すのは、MONEY1 という名前のプライマリ・コンパニオン用と PERSONNEL1 という名前のセカンダリ・コンパニオン用の *interfaces* ファイルの例です。

```
MONEY1

        query tcp ether loghost_node1 9865
        master tcp ether loghost_node1 9865
        hafailover PERSONNEL1

PERSONNEL1

        query tcp ether loghost_node2 9866
        master tcp ether loghost_node2 9866
        hafailover MONEY1
```

この *interfaces* ファイルは Adaptive Server クライアントでも使用されます。

次に示すサンプルの */etc/hosts* ファイルには、上記の *interfaces* ファイルで使用されている論理ホスト名に対する適切なエントリが含まれます。

```
#
# Internet host table on machine node1
#
127.0.0.1       localhost
10.22.98.43      node1
10.22.98.44      node2
10.22.98.165     loghost_node1
10.22.98.166     loghost_node2
```

interfaces ファイルにエントリを追加するには、dsedit を使用します。interfaces エントリがすでに存在する場合は、フェールオーバに使用できるように変更します。

dsedit 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。

両方のコンパニオンの *$SYBASE* を同じ値にする

両方のコンパニオンにある *$SYBASE* は同じディレクトリのパス名を指していなければなりません。これには、次の操作を行います。

- 各コンパニオン上の *$SYBASE* リリース・ディレクトリが同じディレクトリに作成されていることを確認します。

- 各コンパニオンの *$SYBASE* リリース・ディレクトリが異なるロケーションにある場合は、両方のコンパニオンに同じパスのディレクトリを作成します。このディレクトリは、実際の *$SYBASE* リリース・ディレクトリへのシンボリック・リンクとして機能します。

  たとえば、プライマリ・コンパニオン MONEY1 とセカンダリ・コンパニオン PERSONNEL1 が、それぞれ */usr/u/sybase1* と */usr/u/sybase2* をリリース・ディレクトリとして使用している場合、各コンパニオンの *$SYBASE* が同じパスを指すようにします。

  MONEY1 と PERSONNEL1 は、対応する *$SYBASE* リリース・ディレクトリへのシンボリック・リンクとして確立する */sybase* を使用します。MONEY1 上では、*/sybase* は */usr/u/sybase1* へのリンクであり、PERSONNEL1 上では、*/sybase* は */usr/u/sybase2* へのリンクです。

sybha の実行

Adaptive Server High Availability Basic Services Library は、*sybha* を呼び出すことによって、各プラットフォームの高可用性クラスタ・サブシステムと対話できるようになります。*sybha* は *$SYBASE/$SYBASE_ASE/bin* にあります。*sybha* は、所有権とパーミッションを変更することにより実行可能になります。

また、*$SYBASE/$SYBASE_ASE/install* にある *sybhauser* という名前のファイルも編集する必要があります。このファイルには、そのクラスタに対してシステム管理者権限を持つユーザのリストがあります。クラスタに対するシステム管理者権限を持つユーザの数を制限することを強くおすすめします。

“root” 権限で、次の作業を行います。

1 sybhagrp という新規グループを、*/etc/group* ファイルまたは NIS マップに追加します。

2 Sybase ユーザを sybhagrp に追加します。これは *$SYBASE* ディレクトリを所有するユーザで、サーバの起動時にこのユーザがデータ・サーバを実行します。*$SYBASE* ディレクトリを所有する異なるユーザが複数のサーバを実行する場合は、これらのユーザをすべて sybhagrp に追加してください。

3 *$SYBASE/$SYBASE_ASE/bin* ディレクトリに変更します。

4 sybha プログラムの所有権を “root” に変更します。

```
chown root sybha
```

5 sybha プログラムのグループを sybhagrp に変更します。

```
chgrp sybhagrp sybha
```

6 sybha のファイル・パーミッションを 4550 に修正します。

```
chmod 4550 sybha
```

7　*$SYBASE/$SYBASE_ASE/install* ディレクトリに変更します。

8　sybase ユーザを *sybhauser* ファイルに追加します。

9　*sybhauser* のパーミッションを “root” に変更します。

```
chown root sybhauser
```

10　*sybhauser* のファイル・パーミッションを修正し、“root” による変更のみを可能にします。

```
chmod 600 sybhauser
```

新しいデフォルト・デバイスの作成

デフォルトでは、新しくインストールされた Adaptive Server のデフォルト・デバイスは master です。つまり、すべてのデータベース (フェールオーバで使用するプロキシ・データベースを含む) は自動的にマスタ・デバイス上に作成されます。しかし、ユーザ・データベースをマスタ・デバイスに作成すると、システム障害からのリストアがより困難になります。

マスタ・デバイス上のユーザ・データベースの数をできるだけ少なくするために、disk init を使用して新しいデバイスを作成します。sp_diskdefault を使用して、新しいデバイスをデフォルトとして指定します。

たとえば、money_default_1 という名前の新しいデフォルト・デバイスを MONEY1 Adaptive Server に追加するには、次のように入力します。

```
sp_diskdefault money1_default1, defaulton
```

次のように、明確にデフォルトとしての設定を無効にしないかぎり、マスタ・デバイスはデフォルト・デバイスのままです。

```
sp_diskdefault master, defaultoff
```

disk init については、『リファレンス・マニュアル：コマンド』を参照してください。また、sp_diskdefault 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。

sysservers へのローカル・サーバの追加

sp_addserver を使用して、ローカル・サーバを sysservers に追加します。サーバには *interfaces* ファイルで指定したネットワーク名を付けます。たとえば、MONEY1 というコンパニオンが、*interfaces* ファイルで指定されている MONEY1 というネットワーク名を使う場合は、次のように入力します。

```
sp_addserver MONEY1, local, MONEY1
```

この変更を有効にするには、Adaptive Server の再起動が必要です。

sysservers へのセカンダリ・コンパニオンの追加

セカンダリ・コンパニオンを sysservers にリモート・サーバとして追加します。

```
sp_addserver server_name
```

デフォルトでは、Adaptive Server は、srvid が 1000 のサーバを追加します。この変更は、Adaptive Server を再起動しなくても有効になります。

システム管理者への *ha_role* の割り当て

sp_companion を実行するために、ha_role を両方の Adaptive Server に割り当てます。ha_role を割り当てるには、isql から次のコマンドを発行します。

```
sp_role "grant", ha_role, sa
```

一度ログアウトしてから、再度 Adaptive Server にログインして、変更内容を有効にしてください。

installhasvss スクリプトの実行

注意 *installhasvss* を実行する前に、「両方の Adaptive Server のエントリを interfaces ファイルに追加する」(106 ページ) に記載されている作業を行ってください。この作業を行わずに *installhasvss* を実行した場合は、*installmaster* を再実行して、システム・ストアド・プロシージャすべてを再インストールしてください。

installhasvss スクリプトにより、次の作業が行われます。

- フェールオーバに必要なストアド・プロシージャ (sp_companion など) のインストール
- *SYB_HACMP* サーバの sysservers へのインストール

installhasvss を実行するには、システム管理者権限が必要です。

installhasvss は、*$SYBASE/$SYBASE_ASE/scripts* にあります。*installhasvss* を実行するには、次のように入力します。

```
$SYBASE/$SYBASE_OCS/bin/isql -Usa -Ppassword
-Sservername
< $SYBASE/$SYBASE_ASE/scripts/installhasvss
```

installhasvss は、ストアド・プロシージャや *SYB_HACMP* サーバを作成するときにメッセージを表示します。

設定パラメータの確認

次の設定パラメータを有効にしてから、Adaptive Server をフェールオーバ用に設定します。

- enable CIS − コンポーネント統合サービス (CIS) を有効にします。この設定パラメータは、デフォルトで有効です。
- enable xact coordination − 分散トランザクション管理 (DTM) を有効にします。この設定パラメータは、デフォルトで有効です。
- enable HA − 高可用性システムで、Adaptive Server をコンパニオンとして機能させます。enable HA は、デフォルトでは無効です。このパラメータを有効にするには、Adaptive Server を再起動してください。また、このパラメータを有効にすると、高可用性システムで Adaptive Server を起動したというメッセージがエラー・ログに書き込まれます。

『システム管理ガイド 第 1 巻』の「第 5 章 設定パラメータ」を参照してください。

マスタ・ログへのスレッショルドの追加

スレッショルドをマスタ・ログに追加していない場合は、追加してください。

1. ダンプ・トランザクションが発生する前に、master データベースのログに対して sp_thresholdaction を定義して実行し、残りのページ数にスレッショルドを設定します。Sybase では sp_thresholdaction を提供していません。このシステム・プロシージャの作成の詳細については、『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。
2. それぞれのスレッショルドをマスタ・ログ・セグメントに置いて、セグメントが満杯にならないようにします。

```
sp_addthreshold "master", "logsegment", 250, sp_thresholdaction
```

3. プライマリ・コンパニオンを再起動して、この静的パラメータを有効にします。

フォールト・モニタ用のユーザとログインの追加

高可用性エージェント・フォールト・モニタ *ase_monitor* は、完全な**プローブ**を実行するために次の処理を行います。

1. Adaptive Server に接続します。
2. テンポラリ・テーブルの作成、テーブルへのエントリ挿入、テーブルの更新および削除を行います。
3. サイクル数が Adaptive Server リソースプロパティ *Connect_cycle_count* で指定した値に達した後で、Adaptive Server から切断します。

モニタが完全プローブ・オペレーションを実行するために使用する特別なユーザとログインを作成または指定します。isql を使ってデータ・サーバに接続し、次のコマンドを発行します。

```
sp_addlogin <user for monitoring ase>, <password>
sp_adduser <user for monitoring ase>
```

注意 システム管理者は、Adaptive Server の設定時に、プローブに使用するユーザとログインにより、他の目的に使用できる接続の合計数が実際には 1 つ減ることを考慮してください。つまり、接続の合計数が 25 の場合、1 つの接続がフォールト・モニタのプローブに使用されるので、ほかの目的に使用できる有効接続数は 24 になります。

Sun Cluster サブシステムの設定

この項では、次のことを前提としています。

- クラスタ・システム・コマンドを実行するときに、PATH 環境変数が */usr/cluster/bin* を含むように設定されている。
- Sun Cluster 高可用性システムがインストールされている。
- 共有ディスクに Adaptive Server がインストールされ、必要なデータベース・デバイス・ファイルが作成されている。
- Adaptive Server が、「アクティブ／アクティブ設定のための Adaptive Server の準備」(106 ページ) の指示に従って設定されている。
- *$SYBASE/SYBASE.sh* が作成され、Adaptive Server で必要な環境に応じて編集されている。

  このファイルは高可用性エージェント・スクリプトで実行されるため、不正なアクセスからファイルを保護し、“root” ユーザのみが読み込みと実行のパーミッションを持つようにしてください。
- *$SYBASE/$SYBASE_ASE/install/RUN_<Dataserver_name>* ファイルが作成されている。このファイルで、-e オプションを使用して Adaptive Server エラー・ログを指定します。

  -s オプションを指定する場合は、エラー・ログを Adaptive Server リソース・プロパティ *Dataserver_name* と同じにします。
- *$SYBASE/$SYBASE_ASE/SC-3_0* が正しくインストールされている。このディレクトリに、Adaptive Server 高可用性エージェントに必要なファイルがすべて含まれる必要があります。

  デフォルトの *$SYBASE/$SYBASE_ASE/SC-3_0/* には次のディレクトリがあります。

- *bin*
- *etc*
- *log*

$SYBASE/$SYBASE_ASE/SC-3_0/bin には次のファイルがあります。

- *ase_start*
- *ase_stop*
- *ase_monitor_start*
- *ase_monitor_stop*
- *ase_update*
- *ase_validate*
- *utils.ksh*
- *ase_monitor*
- *syscadm*

$SYBASE/$SYBASE_ASE/SC-3_0/etc には次のファイルがあります。

- *SY.ase*
- *ase_monitor_action*
- *ase_login_file*
- *sysc_input_file*

$SYBASE/$SYBASE_ASE/SC-3_0/log には最初はファイルがありませんが、Adaptive Server リソースが作成されると、*Callback_log* ファイルと *Monitor_log* ファイルが生成されます。

syscadm スクリプトの使用

syscadm スクリプトでは、Sun Cluster で Adaptive Server リソース・グループとその関連リソースの設定と管理を行います。*syscadm* を使用すると、Adaptive Server リソース・グループとそのリソースの作成や削除を行ったり、非管理状態にしたりできます。*syscadm* スクリプトは *$SYBASE/$SYBASE_ASE/SC-3_0/bin/* にあります。

このスクリプトの create オプションは次のことを行います。

- 必要なリソース・タイプをリソースグループマネージャに登録する。
- 指定したリソース・グループごとに、リソース・グループを作成し、指定のリソースを作成し、作成したリソースをリソース・グループに追加する。

- ストレージリソースと論理ホストリソースに対する Adaptive Server リソースのリソース依存を確立する。

このスクリプトの remove オプションは、指定したリソース・グループとそのリソースを削除します。

unmanage オプションは次のことを行います。

- リソース・グループのすべてのリソースを無効にする。
- リソース・グループをオフライン状態にしてから非管理状態にする。

注意 *syscadm* を実行するには “root” 権限でログインしてください。

---

syscadm は、*sysc_input_file* という入力ファイルとともに機能します。このファイルは、適切な入力値を提供するように、ユーザが設定に応じて編集します。*sysc_input_file* は *$SYBASE/$SYBASE_ASE/SC-3_0/etc/* にあります。

注意 *sysc_input_file* の編集が終了したら、不正に変更されないようにしてください。このファイルの値が適切でない場合は、インストールに影響を与える可能性があります。システム管理者だけがこのファイルを編集できるように、ファイルのパーミッションを変更することをおすすめします。

---

sysc_input_file を編集するときは、次の点に注意します。

- “`<name>=<value>`” エントリの “=” の前後にスペースを入力しない。
- コメントは # で開始する。
- 名前の最後に 1 を付けるとプライマリ・コンパニオンに対応する。
- 名前の最後に 2 を付けるとセカンダリ・コンパニオンに対応する。

sysc_input_file の例については、「sysc_input_file のサンプル」(115 ページ) を参照してください。

この入力ファイルは次の 3 つのセクションに分かれています。

- Section 1 － すべてのエントリの右辺値を入力します。このセクションには、Adaptive Server インストール・ディレクトリ、高可用性設定、データ・サーバ名、Nodelist などのエントリが含まれます。
- Section 2 － 必須エントリの右辺値を入力します。たとえば、SUNW.HAStoragePlus リソースのみを使用する場合は、SUNW.HAStoragePlus に関連するエントリの値を入力してください。使用していないエントリの値は入力しないでください。
- Section 3 － このセクションのすべてのエントリにはデフォルト値が割り当てられます。デフォルトを無効にしない場合は、右辺値を入力する必要はありません。

たとえば、Adaptive Server リソース名のファイルを編集するには、次の行を変更します。

```
ASE_RNAME="ase_$Dataserver_name"
```

上記を次のように変更します。

```
ASE_RNAME="my_ase_name"
```

または、*RUN_SERVER* ファイルを指定して、*Debug_callback* フラグを設定するには、*OTHER_PROPERTIES* のエントリを変更します。この値は、*<name>=<value>* という文字列をスペースで区切って指定します。

```
OTHER_PROPERTIES="RUN_server_file=/mypath/RUN_my_ase Debug_callback=TRUE"
```

sysc_input_file のサンプル

図 9-1 (105 ページ) の Adaptive Server リソース・グループ rg_MONEY とそのリソースの作成と設定に使用される *sysc_input_file* を次に示します。

```
#############################################################################
##NOTE:                                                                    ##
##    1. This file will be executed by ksh to set environment of syscadm ##
##       You will be responsible for executing anything in this file      ##
##       So, make sure THERE ARE NO DANGEROUS COMMANDS IN THIS FILE      ##
##                                                                         ##
##    2. No spaces around = in the <Variable_name>=<value> pairs         ##
##                                                                         ##
##    3. Comments should start with #, like ksh comments                  ##
##                                                                         ##
##    4. Names ending with 1 correspond to primary, and 2 to secondary   ##
#############################################################################

#############################################################
##  Section1: Must specify right hand side values        ##
#############################################################
# Sybase home directory
SYBASE=î/sybaseî

# Valid HA Setups are "ACTIVE_PASSIVE" or "ASYMMETRIC" or "SYMMETRIC"
HA_SETUP="SYMMETRIC"

# Comma separated list of nodes, Ex: "node1,node2"
Nodelist="node1,node2"

# ASE Dataserver name and Dataserver login file
Dataserver_name1="MONEY1"
Dataserver_login_file1="/sybase/ASE-15_0/SC-3_0/etc/ase_login_file"

Dataserver_name2="PERSONNEL1"
Dataserver_login_file2="/sybase/ASE-15_0/SC-3_0/etc/ase_login_file"
```

```
################################################################################
##  Section2: Must specify right hand side values, if required            ##
################################################################################

# if using Logical Hostname
LOGHOST_NAME_OR_FLOATING_IP1="loghost_node1"
LOGHOST_NAME_OR_FLOATING_IP2="loghost_node2"

# if using HAStorage resource
ServicePaths1=î/global/node1_shareî
ServicePaths2=î/global/node2_shareî

# if using HAStoragePlus resource
GlobalDevicePaths1=
FilesystemMountPoints1=

GlobalDevicePaths2=
FilesystemMountPoints2=



################################################################################
## Section3: May specify right hand side values to override defaults      ##
################################################################################

# bin of the cluster commands
CLUSTER_BIN="/usr/cluster/bin"

# ASE Resource Type and corresponding registration file
RT_NAME="SY.ase"
RT_FILE="$SYBASE/ASE-15_0/SC-3_0/etc/$RT_NAME"

# Resource Group names
RG_NAME1="rg_$Dtatserver_name1"
RG_NAME2="rg_$Dataserver_name2"

# ASE Resource names and space separated extended properties
ASE_RNAME1="ase_$Dataserver_name1"
ASE_RNAME2="ase_$Dataserver_name2"

OTHER_PROPERTIES1="RUN_server_file=  Callback_log=  Monitor_log="
OTHER_PROPERTIES2="RUN_server_file=  Callback_log=  Monitor_log="

# Logical Host Resource names
LOGHOST_RNAME1="lh_$Dataserver_name1"
LOGHOST_RNAME2="lh_$Dataserver_name2"

# HA Storage Resource names
HASTORAGE_RNAME1="has_$Dataserver_name1"
HASTORAGE_RNAME2="has_$Dataserver_name2"
```

```
# HA Storage Plus Resource names
HASTORAGE_PLUS_RNAME1="hasp_$Dataserver_name1"
HASTORAGE_PLUS_RNAME2="hasp_$Dataserver_name2"
```

syscadm の構文は次のとおりです。

```
syscadm [-v] -c|r|u [primary|secondary|both] -f <sysc_input_file>
syscadm [-v] -r|u <rg1,rg2,...> [-t <ASE_resource_type>]
```

上記のパラメータの意味は、次のとおりです。

- -c はリソース・グループを作成します。
- -r はリソース・グループを削除します。
- -u はリソース・グループを非管理状態にします。
- -f は入力ファイルを指定します。
- -v は冗長出力 (実行時の Sun Cluster コマンド表示) のことです。
- –t は、*SY.ase* ではない場合の Adaptive Server リソース・タイプ名を指定します (-r コマンドと -u コマンドで入力ファイルが指定されていない場合に役立ちます)。

SUNW.HAStoragePlus リソースは、`AffinityOn=True` を指定すると作成されます。

Adaptive Server リソースの拡張プロパティ

表 9-1 は、Adaptive Server リソースのすべての拡張プロパティを示します。リソースの詳細については、Sun Cluster のそれぞれのマニュアルを参照してください。

表 *9-1: SY.ase* リソースの拡張プロパティ

プロパティ	デフォルト	説明
Sybase_home	なし	Adaptive Server インストール環境のホーム・ディレクトリ。Adaptive Server インストール環境の *$SYBASE* 環境変数と同じ値です。このプロパティは Adaptive Server リソースを作成するときに必須です。
Environment_file	*Sybase_home/SYBASE.sh*	Adaptive Server に渡す環境変数を指定する環境ファイルの絶対パス。 このファイルは、高可用性エージェントが正常に機能するために利用可能にしてください。
Dataserver_name	なし	Adaptive Server データ・サーバの名前。このプロパティは Adaptive Server リソースを作成するときに必須です。
Backup_server_name	None	Backup Server の名前。
Text_server_name	None	全文検索サーバの名前。

プロパティ	デフォルト	説明
Secondary_companion_name	なし	セカンダリ・コンパニオン・サーバの名前。sp_companion コマンドの configure または drop によって、自動的に設定または設定解除されます。アクティブ／アクティブ設定用に予約されています。このプロパティは手動で設定しないでください。
Dataserver_login_file	*Sybase_home/$SYBASE-ASE/SC-3_0/etc/ase_login_file*	データ・サーバのログイン情報が入っているファイルの絶対パス。このファイルは 2 行で構成されています。1 行目はシステム管理者のログインとパスワードで、2 行目はフォールト・モニタ・プログラム *ase_monitor* が完全な**プローブ**に使用するユーザ・ログインとパスワードです。
Action_file	*Sybase_home/$SYBASE_ASE/SC-3_0/etc/ase_monitor_action*	エラー・コードをフォールト・モニタ・プログラム *ase_monitor* のアクションと関連付けるファイルの絶対パス。
RUN_server_file	*Sybase_home/$SYBASE_ASE/install/RUN_<Dataserver_name>*	プロパティ *Dataserver_name* によって指定される Adaptive Server の *RUN_SERVER* ファイルの絶対パス。 このファイルには環境変数を含めないでください。
Thorough_probe_script	無視される。今後のために予約済み。	フォールト・モニタ・プログラムが完全プローブの実行に使用する SQL スクリプトが入っているファイルの絶対パス。
Monitor_log	*Sybase_home/$SYBASE_ASE/SC-3_0/log/ase_monitor_<Dataserver_name>.log*	フォールト・モニタ・プログラム *ase_monitor* のログ・ファイルの絶対パス。
Callback_log	*Sybase_home/$SYBASE_ASE/SC-3_0/log/ase_callback_<Dataserver_name>.log*	*$SYBASE/$SYBASE_ASE/SC-3_0/bin* にある Adaptive Server 高可用性エージェント・コールバック・スクリプトで使用されるログ・ファイルの絶対パス。
Callback_log_max_size	5000000	コールバック・ログ・ファイルの最大サイズ。ログ・サイズがこの制限を超えると、コールバック・ログの拡張子が現在の日付と時刻に変更されます。新しいログ・メッセージが生成されると *Callback_log* に書き込まれます。
Monitor_log_max_size	無視される。今後のために予約済み。	
Probe_timeout	30	フォールト・モニタ・プローブがタイム・アウトになり、エラーを登録してからの経過時間 (単位：秒)。
Restart_delay	30	再起動後に次のプローブを遅延させる時間 (単位：秒)。

プロパティ	デフォルト	説明
Debug_monitor	FALSE	TRUE に設定すると、フォールト・モニタ・プログラム *ase_monitor* が、プロパティ *Monitor_log* で指定されるファイルにデバッグ・メッセージを記録します。
Debug_callback	FALSE	TRUE に設定すると、Adaptive Server 高可用性エージェント・スクリプトが、プロパティ *Callback_log* で指定されるファイルにデバッグ・メッセージを記録します。
Connect_cycle_count	5	Adaptive Server との既存の接続が再使用する完全プローブ・サイクルの数。サイクルがこの数に達すると、既存の接続が切断されて新しい接続が確立されます。
Failback_strategy	無視される。今後のために予約済み。	

Adaptive Server リソース・グループの設定

Sun Cluster で Adaptive Server リソース・グループを設定するには、次の手順を実行します。

1　Adaptive Server のリソース・タイプ登録ファイル *SY.ase* を修正します。このファイルは *$SYBASE/$SYBASE_ASE/SC-3_0/etc/* にあります。Adaptive Server 高可用性エージェントのロケーションを指定するリソース・タイプ・プロパティ *RT_BASEDIR* の行を探します。この値を、*$SYBASE/$SYBASE_ASE/SC-3_0/bin* のインストール・ロケーションを指すように変更します。

次に例を示します。

```
RT_BASEDIR=/sybase/ASE-15_0/SC-3_0/bin/
```

注意　*SY.ase* では環境変数を使用できません。この値にはフル・パスを指定します。*$SYBASE/$SYBASE_ASE/SC-3_0/bin* の *SYBASE*、*SYBASE_ASE* の値は置き換えてください。

2　別のロケーションにある別のファイルを使用する場合は、*SY.ase* リソースの設定時に、リソース拡張プロパティ *Dataserver_login_file* にフル・パスを指定します。システム管理者とフォールト・モニタ用に追加したユーザの Adaptive Server ログイン情報が入っているファイルを作成または編集します。デフォルトのファイルは *$SYBASE/$SYBASE_ASE/SC-3_0/etc/ase_login_file* です。

このファイルは 2 行で構成されています。1 行目はシステム管理者のログインとパスワードで、2 行目は monitor_user のログインとパスワードです。フォールト・モニタ・プログラム ase_monitor は、monitor_user ユーザとして、完全プローブを実行します。

```
login_type <tab> login string
login_type <tab> login_string
```

ログイン・タイプの有効な値は、“encrypted” と “normal” です。*login_type* を “normal” に設定した場合、*login_string* の値は “login_name/password” の形式になります。*login_type* を “encrypted” に設定した場合、*login_string* の値は、haisql ユーティリティ (*$SYBASE/$SYBASE_ASE/bin* にある) から取得する暗号化された文字列です。ファイルの機密情報が十分に保護されるように、*login_type* に “encrypted” を使用することをおすすめします。haisql を使用して、暗号化したログイン文字列を生成するには、次の手順を実行してください。

a haisql に引数を指定しないで実行し、*login_name* と *password* の暗号化した文字列を生成します。

```
/$SYBASE/ASE-12_5/bin/haisql
Enter Username: sa
Enter Password:
TWAS8n1jSF2gBsvayUlw97861.cyTKaS1YhavBRQ2qKcJwtx.TmFBarGS2Kl553WDR7g8m5vrf86
t@K4CU62HEccm4zkeexsP9E=FeuvX
```

b 暗号化した文字列をコピーして *ase_login_file* ファイルに貼り付けます。

次は、“encrypted” ログイン・タイプを使用する *ase_login_file* の例です。

```
encrypted
TWAS8n1jSF2gBsvayUlw97861.cyTKaS1YhavBRQ2qKcJwtx.TmFBarGS2Kl553WDR7g8m5vrf86t@K
4CU62HEccm4zkeexsP9E=FeuvX
encrypted
rX2S8n1jSF2gBuD0q=AXEXKCZvzGcK5K3kWnp_P+e4avf=67kYVSzy7+h640@97FSP_dlkH_oV2Zima
5+7tUyHnsm4zmSIHIUnKSTPoTD
```

次は、“normal” ログイン・タイプを使用する *ase_login_file* の例です。

```
normal    sa/sa_password
normal     monitor_user/monitor_user_password
```

注意 *ase_login_file* の 2 つの行に異なるログイン・タイプを使用できます。

暗号化したログイン文字列を使用しない場合は特に、適切なアクセス・パーミッションを使用して *ase_login_file* ファイルを保護してください。適切な *login_type* 値と *login_string* 値でファイルを編集した後で次のコマンドを実行して、ルート・ユーザだけがこのファイルを読み取ることができるようにします。

```
chmod 400 ase_login_file
chown root ase_login_file
```

```
chgrp sys ase_login_file
```

3　*sysc_input_file* を作成または編集し、次の *syscadm* コマンドを実行します。これによって、リソース・タイプの登録、リソース・グループの作成、リソース・グループへのリソースの追加、リソース依存の確立が行われます。

たとえば、入力ファイル *sysc_input_file* を使用して *syscadm* スクリプトを実行するには、次のように入力します。

```
syscadm -c both -f sysc_input_file
```

syscadm スクリプトの詳細については、「syscadm スクリプトの使用」(113 ページ) を参照してください。

この手順は手動で実行することもできます。詳細については、「リソース・グループの手動設定」(131 ページ) を参照してください。

4　プライマリ Adaptive Server リソース・グループの場合は、scswitch コマンドを実行して次の作業を行います。

- リソース・グループを「管理状態」に移行する。
- すべてのリソースとそのモニタを有効にする。
- プライマリ・ノード上でリソース・グループをオンラインにする。

  scswitch -Z -g *resource_group_name*

  次に例を示します。

  ```
  scswitch -Z -g rg_MONEY1
  ```

5　セカンダリ Adaptive Server リソース・グループの場合は、scswitch コマンドを実行し、手順 4 と同じ作業を行います。

SUNW.HAStoragePlus の使用

Sun Cluster 3.0 Update2 以降を実行している場合は、Adaptive Server リソース・グループで *SUNW.HAStoragePlus* リソースを使用できます。*SUNW.HAStoragePlus* リソースを *SUNW.HAStorage* リソースの代わりに使用するか、*SUNW.HAStorage* リソースと *SUNW.HAStoragePlus* リソースを両方ともリソース・グループで使用できます。

SUNW.HAStoragePlus リソースを Adaptive Server リソース・グループに追加するには、*SUNW.HAStoragePlus* リソース・プロパティ *GlobalDevicePaths* と *FilesystemMountPoints* を必要に応じて設定してください。*syscadm* を使用している場合は、*sysc_input_file* の対応するエントリに値を指定できます。接続を有効にするために、*SUNW.HAStoragePlus* リソース・プロパティ AffinityOn を TRUE に設定してください。

SUNW.HAStoragePlus リソースを手動で追加するには、次の手順に従います。

1 リソース・タイプ *SUNW.HAStoragePlus* を登録します。

```
scrgadm -a -t SUNW.HAStoragePlus
```

2 *SUNW.HAStoragePlus* リソースを Adaptive Server リソース・グループに追加します。

scrgadm -a -j *hasp_resource_name*
-t SUNW.HAStoragePlus
-g *resource_group*
-x FilesystemMountPoints=*shared_disk_filesystem*
-x AffinityOn=TRUE

次に例を示します。

```
scrgadm -a -j hasp_MONEY1
-t SUNW.HAStoragePlus
-g rg_MONEY1
-x fileSystemMountPoints=/global/node1_share
-x Affinityon=TRUE
```

SUNW.HAStoragePlus リソースを使用すると、Adaptive Server データベース・デバイスを、広域ファイル・システムにも *SUNW.HAStoragePlus* リソースで管理されるフェールオーバ・ファイル・システム (FFS) にも作成できます。いずれの場合も、データは共有ディスクに存在する必要があります。*SUNW.HAStoragePlus* リソースを作成するときに、対応するすべてのファイル・システムとデバイス・パスを指定してください。

3 *SUNW.HAStoragePlus* リソースを有効にします。

scswitch -e -j *hasp_resource_name*

次に例を示します。

```
scswitch -e -j hasp_MONEY1
```

4 *SY.ase* リソースと *SUNW.HAStoragePlus* リソースの間にリソース依存を確立します。

```
scrgadm -c -j ase_resource_name
-y Resource_dependencies=hasp_resource_name
```

次に例を示します。

```
scrgadm -c -j ase_MONEY1
-y Resource_dependencies=hasp_MONEY1
```

SUNW.HAStorage リソースと *SUNW.HAStoragePlus* リソースの両方を使用している場合は、すべてのストレージ・リソース名をカンマで区切ったリストとして指定してください。

scrgadm -c -j *ase_resource_name*
-y Resource_dependencies=*hasp_resource_name*,*hastorage_name*

次に例を示します。

```
scrgadm -c -j ase_MONEY1
-y Resource_dependencies=hasp_MONEY1,has_MONEY1
```

SUNW.HAStoragePlus リソース・タイプの詳細については、Sun Cluster のマニュアルを参照してください。

フェールオーバ用コンパニオン・サーバの設定

この項の指示に従って、高可用性システムで Adaptive Server をプライマリ・コンパニオンとセカンダリ・コンパニオンとして設定します。

Adaptive Server 内の高可用性サービス・ライブラリ

Sun Cluster 用の高可用性サービス・ライブラリをロードしてください。

まず、高可用性サービス・ライブラリが使用可能かどうかを確認します。isql を使用して任意の Adaptive Server に接続します。

```
sp_companion "MONEY1", show_cluster
```

次のようなメッセージが表示されます。

```
The default cluster is: SC.
The current cluster is set to default.
Supported cluster systems for SunOS are:
SC
VCS
```

SC3.0 用の高可用性サービス・ライブラリを設定します。たとえば、PERSONNEL1 から次のように入力します。

```
sp_companion "MONEY1", set_cluster, "SC"
The current cluster is set to SC.
```

Adaptive Server と基本クラスタ・システムの対話をチェックします。PERSONNEL1 から次のように入力します。

```
sp_companion
Server 'PERSONNEL1' is alive and cluster configured.
Server 'PERSONNEL1' is configured for HA services.
Server 'PERSONNEL1' is currently in 'Single server' mode.
```

注意 以下の手順は、クラスタ内の 1 つのサーバだけから実行します。高可用性サービス・ライブラリが、クラスタ内の別の Adaptive Server にロードされます。高可用性サービス・ライブラリが別の Adaptive Server にロードされた場合は、サーバ MONEY1 上で sp_companion を発行すると、次のような情報が表示されます。

```
Server 'MONEY1' is alive and cluster configured.
Server 'MONEY1' is configured for HA services.
Server 'MONEY1' is currently in 'Single server' mode.
```

2 つのコンパニオン・サーバはユーザ情報の同期をとって潜在的な矛盾を解決するため、セカンダリ・コンパニオン・サーバ上には完全プローブに使用するユーザ・ログインとパスワードを置かないでください。ログインとパスワードが存在する場合は、ユーザ情報の同期プロセス中に、sp_companion configure と sp_companion do_advisory の両方が失敗します。

セカンダリ・コンパニオン・サーバ上でユーザ・プローブに使用するユーザとログインを削除するには、sp_dropuser と sp_droplogin を使用します。

do_advisory を指定して *sp_companion* を実行する

sp_companion を開始する前に

sp_companion do_advisory と sp_companion configure を実行する前に、次の手順を実行します。

1 セカンダリ Adaptive Server のモニタリングを無効にします。

```
scswitch -n -M -j secondary-resource
```

2 セカンダリ Adaptive Server のモニタ用のユーザとログインを削除します。*secondary_probe_ase* は、「フォールト・モニタ用のユーザとログインの追加」(111 ページ) で作成したログインとユーザです。

```
sp_dropuser secondary_probe_ase
sp_droplogin secondary_probe_ase
```

非対称型設定に対して sp_companion do_advisory と sp_companion configure を正常に実行した後で (詳細については、この後の 2 つの項を参照)、次の手順を実行します。

1 セカンダリ Adaptive Server のモニタ用のユーザとログインを追加します。

```
sp_addlogin secondary_probe_ase, secondary_probe_passwd
sp_adduser secondary_probe_ase
```

secondary_probe_ase は、「フォールト・モニタ用のユーザとログインの追加」(111 ページ) で作成したログインとユーザです。

2　セカンダリ Adaptive Server のモニタリングを有効にします。

```
scswitch -e -M -j secondary-resource
```

十分なリソースを持つセカンダリ・コンパニオンを設定して、フェールオーバ中でもサーバ 2 台分の作業を実行できるようにします。セカンダリ・コンパニオンは、正常なクラスタ・オペレーションを妨げる属性を持っている場合があります。たとえば、プライマリ・コンパニオンとセカンダリ・コンパニオンのユーザ・ログイン数がともに 250 に設定されていると、フェールオーバ中、セカンダリ・コンパニオンには、発生する可能性のあるユーザ・ログインの半分を処理するリソースしかないことになります。したがって、MONEY1 と PERSONNEL1 の両方でユーザ・ログイン数を 500 に設定します。

sp_companion do_advisory は、クラスタ・オペレーション (Adaptive Server をセカンダリ・コンパニオンとして設定するなど) が正常に行われるようにするために、プライマリ・コンパニオンとセカンダリ・コンパニオンの両方の設定オプションを確認します。また、sp_companion do_advisory は、変更する必要のある設定オプションを通知します。

sp_companion do_advisory オプションの詳細については、「第 6 章 do_advisory の実行」を参照してください。

非対称型コンパニオン設定の作成

非対称型設定を設定する前に、scswitch を使用して、プライマリ・リソースとセカンダリ・リソースのモニタリングを無効にする必要があります。

```
scswitch -n -M -j primary_resource
scswitch -n -M -j secondary_resource
```

非対称型設定のプライマリ・コンパニオンを設定するには、sp_companion を使用します。

```
sp_companion "primary_server_name", configure, with_proxydb,
login_name,password
```

- *primary_server_name* − *interfaces* ファイルのエントリと sysservers に定義されているプライマリ Adaptive Server の名前。
- *login_name* − このクラスタ・オペレーションを行っているユーザの名前。ユーザは ha_role を持っている必要があります。
- *password* − このクラスタ・オペレーションを行っているユーザのパスワード。

注意　上記コマンドは、セカンダリ・コンパニオンからのみ実行します。

この例では、MONEY1 という名前の Adaptive Server をプライマリ・コンパニオンとして設定します。セカンダリ・サーバ PERSONNEL1 から次のコマンドを発行します。

```
sp_companion "MONEY1", configure, NULL, sa, Odd2Think
Server 'PERSONNEL1' is alive and cluster configured.
Step: Access verified from Server:'PERSONNEL1' to Server:'MONEY1'
Server 'MONEY1' is alive and cluster configured.
Step: Access verified from Server:'MONEY1' to Server:'PERSONNEL1'
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
.....
Step: Companion servers configuration check succeeded
Step: Server handshake succeeded
Step: Master device accessible from companion
Step: Added the servers 'PERSONNEL1' and 'MONEY1' for cluster config
Step: Server configuration initialization succeeded
Step: Synchronizing server logins from companion server
Step: Synchronizing remoteserver from companion server
Step: Synchronizing roles from companion server
Step: Synchronizing server-wide privs from companion server
Step: User information syncup succeeded
Step: Server configured in normal companion mode
```

sp_companion 設定を行っている間にユーザ・データベースが作成されると、次のようなメッセージが表示されます。

```
Step: Created proxy database 'pubs2'
 Step: Proxy status for database has been set. Please Checkpoint the database 'pubs2'
 Step: Server configured in normal companion mode
 Starting companion watch thread
```

scswitch を使用して、プライマリ・リソースのモニタリングを有効にします。

```
scswitch -e -M -j primary_resource
```

非対称型設定においてセカンダリ・コンパニオン・サーバのフェールオーバを防止するには、フェールオーバ後にセカンダリ・リソースのモニタリングを無効にする必要があります。

非対称型設定の詳細については、「非対称型コンパニオンの設定」(19 ページ)を参照してください。

対称型設定の作成

非対称型フェールオーバを使用できるようにコンパニオンを設定した後、対称型設定に設定できます。対称型設定では、両方のサーバがプライマリ・コンパニオンとしても、セカンダリ・コンパニオンとしても機能します。非対称型設定については、図 3-2 (21 ページ) を参照してください。

対称型設定を設定する前に、scswitch ユーティリティを使用してプライマリ・リソースとセカンダリ・リソースのモニタリングを無効にする必要があります。

```
scswitch -n -M -j primary_resource
scswitch -n -M -j secondary_resource
```

プライマリ・コンパニオンから sp_companion を発行して、対称型設定にします。非対称型設定と同じ構文を使用しますが、with_proxydb を NULL に置き換えます。sp_companion の構文については、「非対称型コンパニオン設定の作成」(125 ページ) を参照してください。

次の例では、PERSONNEL1 が MONEY1 のセカンダリ・サーバです。これは、非対称型設定ですが、後で対称型設定に変更されます。MONEY1 に接続します。

```
sp_companion 'PERSONNEL1', configure, NULL, sa, Think2Odd
Server 'PERSONNEL1' is alive and cluster configured.
Step: Access verified from Server:'MONEY1' to Server:'PERSONNEL1'
Server 'MONEY1' is alive and cluster configured.
Step: Access verified from Server:'PERSONNEL1' to Server:'MONEY1'
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
.....
Step: Companion servers configuration check succeeded
Step: Server handshake succeeded
Step: Master device accessible from companion
Step: Added the servers 'MONEY1' and 'PERSONNEL1' for cluster config
Step: Server configuration initialization succeeded
Step: Synchronizing server logins from companion server
Step: Synchronizing remoteserver from companion server
Step: Synchronizing roles from companion server
Step: Synchronizing server-wide privs from companion server
Step: User information syncup succeeded
Step: Server configured in normal companion mode
```

セカンダリ・リソース・グループの NodeList プロパティに両方のノードが含まれるように変更します。

```
scrgadm -c -g secondary_group -y NodeList=secondary_node,primary_node
```

次の例は、 Adaptive Server PERSONNEL1 を含む、リソース・グループ *rg_PERSONNEL1* の Nodelist プロパティを変更します。

```
scrgadm -c -g rg_PERSONNEL1 -y NodeList=node2,node1
```

scswitch を使用して、プライマリ・リソースとセカンダリ・リソースのモニタリングを有効にします。

scswitch -e -M -j *primary_resource*
scswitch -e -M -j *secondary_resource*

Sybase フェールオーバの管理

この項では、Sybase のフェールオーバの使い方を説明します。

プライマリ・コンパニオンへのフェールバック

フェールバックでは、プライマリ・コンパニオンのリソース・グループをセカンダリ・ノードからプライマリ・ノードに戻し、プライマリ・コンパニオンをプライマリ・ノードで起動します。

1 プライマリ・ホストがプライマリ・コンパニオンを引き継ぐ準備ができたら、scswitch ユーティリティを使用して、セカンダリ・リソースのモニタリングを無効にします (無効にしていない場合)。

```
scswitch -n -M -j secondary_resource
```

2 セカンダリ・コンパニオンから次のコマンドを発行します。

```
sp_companion primary_companion_name, prepare_failback
```

このコマンドは、プライマリ・コンパニオンのリソース・グループをプライマリ・ホストに戻します。

注意 または、次のコマンドを使用して、リソース・グループをフェールバックすることもできます。

scswitch -z -h *primary_host* -g *failed_over_group*

たとえば、*node1* のプライマリ・コンパニオン MONEY1 へのフェールバックを実行するには、セカンダリ・ホストまたはプライマリ・ホストから次のコマンドを発行します (クラスタ制御下でホストが正常に稼働している場合)。

```
scswitch -z -h node1 -g rg_MONEY1
```

---

3 ノーマル・コンパニオン・モードを再開するには、プライマリ・リソースのモニタリングを無効にします。

scswitch -n -M -j *primary_resource*

4 プライマリ・コンパニオンから次のコマンドを発行します。

```
sp_companion secondary_companion_name, resume
```

5　プライマリ・リソースのモニタリングを有効にします。

```
scswitch -e -M -j primary_resource
```

6　対称型モードで作業をしている場合は、scswitch を使用してセカンダリ・リソースのモニタリングを有効にします。

注意　sp_companion resume を発行するまでは、フェールオーバ・プロパティが設定されたクライアントを、高可用性を実現するように設定された Adaptive Server に接続できません。sp_companion prepare_failback を発行した後でクライアントを接続しようとすると、sp_companion resume を発行するまでクライアントがハングします。

ノーマル・コンパニオン・モードのサスペンド

サスペンド・モードでは、一時的にプライマリ・コンパニオンがセカンダリ・コンパニオンにフェールオーバできなくなります。ノーマル・コンパニオン・モードからサスペンド・モードに切り替えるには、次の手順に従います。

1　高可用性システムが、リソースとしてのプライマリ・コンパニオンとセカンダリ・コンパニオンをモニタするのを停止させます。“root” 権限で、次のコマンドを発行します。

```
scswitch -n -M -j primary-resource-name
scswitch -n -M -j secondary-resource-name
```

2　ノーマル・コンパニオン・モードをサスペンドします。セカンダリ・コンパニオンから、次のコマンドを発行します。

```
sp_companion companion_name, suspend
```

ノーマル・コンパニオン・モードの再開

サスペンド・モードからノーマル・コンパニオン・モードに戻るには、次の手順に従います。

1　両方のコンパニオンが実行中であることを確認します。

2　ノーマル・コンパニオン・モードを再開します。セカンダリ・コンパニオンから、次のコマンドを発行します。

```
sp_companion primary_companion_name, resume
```

3　プライマリ・コンパニオンとセカンダリ・コンパニオンのリソースとしてのモニタを開始します。“root” 権限で、次のコマンドを発行します。

```
scswitch -e -M -j primary-resource-name
scswitch -e -M -j secondary-resource-name
```

コンパニオン・モードの削除

1 高可用性システムによるコンパニオンのモニタリングを停止します。次のコマンドを発行します。

```
scwitch -n -M -j primary-resource-name
scswitch -n -M -j secondary-resource-name
```

2 コンパニオン・モードを削除するには、次のコマンドを発行します。

```
sp_companion companion_name, "drop"
```

コンパニオン・モードを削除したら、二度と元に戻すことはできません。したがって、削除する場合は、高可用性システムで再びフェールオーバする前に、Adaptive Server のコンパニオン・サーバを再設定してください。

Sun Cluster での高可用性の確認

Sun Cluster で高可用性を正しく設定していることを確認するには、この項の確認テストを実行します。

次の手順では、2 つの Adaptive Server リソース・グループが非対称型モードに設定されていることを想定しています。

1 Adaptive Server リソース・グループのプライマリ・ノードにログインします。

2 Adaptive Server 環境変数 SYBASE、SYBASE_ASE、SYBASE_OCS などを設定します。*Environment_file* 拡張プロパティを使用して、環境変数を指定します。

3 Adaptive Server リソース・グループがオンラインであることを確認します。

```
scstat -g
```

4 isql を使用してプライマリ・データ・サーバに接続します。

```
isql -Usa -Ppassword -Sprimary-server-name
> select name from sysdatabases
> go
> quit
```

5 プライマリ・リソース・グループをセカンダリ・ノードに切り替えます。これはフェールオーバをシミュレートしています。

```
scswitch -z -g primary-resource-group -h secondary-host
```

6 isql を使用してセカンダリ・データ・サーバに接続し、セカンダリ・データ・サーバがプライマリ・データ・サーバのデータベースを引き継ぎ、そのデータベースにアクセス可能であることを確認します。

```
isql -Usa -Ppassword -Ssecondary-server-name
> select name from sysdatabases
> go
> quit
```

7　「プライマリ・コンパニオンへのフェールバック」(128 ページ) の手順に従って、プライマリ・リソース・グループをフェールバックします。

8　isql を使用してプライマリ・データ・サーバに接続し、プライマリ・データ・サーバがプライマリ・データ・サーバのデータベースを引き継ぎ、そのデータベースにアクセス可能であることを確認します。

```
isql -Usa -Ppassword -Sprimary-server-name
> select name from sysdatabases
> go
> quit
```

リソース・グループの手動設定

この項では、Adaptive Server リソース・グループを作成および設定するために *syscadm* スクリプトで実行されるコマンドについて説明します。

必要に応じてこれらの手順を手動で実行して、Adaptive Server リソース・グループの設定、再設定、トラブルシューティングを行うことができます。*SY.ase* ファイルと *ase_login_file* ファイルを、「Adaptive Server リソース・グループの設定」(119 ページ) の手順 1 と 2の説明どおりに適切に修正したことを確認します。

次に示す Sun Cluster コマンドを実行するには、“root” 権限でログインしてください。

プライマリ・コンパニオン・リソース・グループ

1　*SY.ase* リソース・タイプを登録します。

scrgadm -a -t SY.ase -f *full-path-of-SY.ase-file*

次に例を示します。

```
scrgadm -a -t SY.ase
-f /sybase/ASE-15_0/SC-3_0/etc/SY.ase
```

注意　*SY.ase* リソース・タイプは、各クラスタに 1 回だけインストールします。リソース・タイプがすでにインストールされている場合は、エラー・メッセージが表示されます。

2　プライマリ・コンパニオン・サーバのリソース・グループを作成します。リソース・グループ・プロパティ Nodelist でプライマリ・ノードとセカンダリ・ノードを指定します。

scrgadm -a -g *resource_group*
-y Nodelist=*primary-node*,*secondary-node*

次に例を示します。

```
scrgadm -a -g rg_MONEY1 -y Nodelist=node1,node2
```

3 SUNW.HAStorage リソース・タイプを登録します。

```
scrgadm -a -t SUNW.HAStorage
```

4 *SUNW.HAStorage* リソースを作成して Adaptive Server リソース・グループに追加します。フェールオーバが発生したときにセカンダリ・ノードに移動する、共有ディスク上のファイル・システムとデバイスのパスを指定します。

scrgadm -a -j *hastorage_resource_name*
-t SUNW.HAStorage
-g *resource_group*
-x ServicePaths=*shared-disk-storage-path*

次に例を示します。

```
scrgadm -a -j has_MONEY1 -g rg_MONEY1
-t SUNW.HAStorage
-x ServicePaths=/global/node1_share
```

5 *SUNW.LogicalHostname* リソースを作成して Adaptive Server リソース・グループに追加します。フェールオーバが発生したときにセカンダリ・ノードに移動する論理ホスト名を指定します。

scrgadm -a -L -j *loghost_resource*
-g *resource_group*
-l *logical_hostname*

次に例を示します。

```
scrgadm -a -L -j lh_MONEY1
-g rg_MONEY1
-l loghost_node1
```

6 次のコマンドは、Adaptive Server リソースを作成して、それをリソース・グループに追加します。

scrgadm -a -j *ase_resource_name* -g *resource_group* \
-t SY.ase \
-x Sybase_home=*sybase_home_value* \
-x Environment_file=*environment_file_path* \
-x Dataserver_name=*dataserver_name_value* \
-x Dataserver_login_file=*login_file_path* \
-x RUN_server_file=*run_server_file_path*

次に例を示します。

```
scrgadm -a -j ase_MONEY1 -g rg_MONEY1 \
    -t SY.ase \
    -x Sybase_home=/sybase \
    -x Environment_file=/sybase/SYBASE.sh \
    -x Dataserver_name=MONEY1 \
    -x Dataserver_login_file=/sybase/ASE-15_0/SC-3_0/etc/ase_login_file
    -x RUN_server_file=/sybase/ASE-15_0/install/RUN_MONEY1
```

標準リソースプロパティ値と拡張プロパティ値を指定します。

3 つの拡張プロパティ値 (*Sybase_home*、*Dataserver_name*、*Dataserver_login_file*) を指定してください。ほかの拡張プロパティではデフォルト値を使用できます。

高可用性エージェント・フォールト・モニタで使用される標準リソース・プロパティ *Cheap_probe_interval*、*Thorough_probe_interval*、*Retry_count*、*Retry_interval* を設定できます。

標準リソースプロパティの詳細については、Sun Cluster のマニュアルを参照してください。Adaptive Server リソースの拡張プロパティの詳細については、表 9-1 (117 ページ) を参照してください。

7　*SY.ase* リソースと *SUNW.HAStorage* リソースの間にリソース依存を確立します。つまり、*SUNW.HAStorage* リソースがオンラインになった場合のみ *SY.ase* リソースがオンラインになり、*SY.ase* リソースがオフラインになった後で *SUNW.HAStorage* リソースがオフラインになります。

scrgadm -c -j *ase_resource_name*
-y Resource_dependencies=*hastorage_resource_name*

次に例を示します。

```
scrgadm -c -j ase_MONEY1
-y Resource_dependencies=has_MONEY1
```

注意　リソース・グループに追加されたリソースは、すべて暗黙的にその *SUNW.LogocalHostname* リソースに依存します。

8　プライマリ Adaptive Server リソース・グループの場合は、scswitch を実行して次の作業を行います。

- リソース・グループを管理状態に移行する。
- すべてのリソースとそのモニタを有効にする。
- プライマリ・ノード上でリソース・グループをオンラインにする。

  scswitch -Z -g *resource_group_name*

  次に例を示します。

  ```
  scswitch -Z -g rg_MONEY1
  ```

注意　「SUNW.HAStoragePlus の使用」(121 ページ) を参照して、*SUNW.HAStoragePlus* リソースを作成し、Adaptive Server リソース・グループに追加してください。

セカンダリ・コンパニオン・リソース・グループ

1 セカンダリ・コンパニオン・サーバのリソース・グループを作成します。対称型設定の場合は、リソース・グループ・プロパティ Nodelist でプライマリ・ノードとセカンダリ・ノードを指定します。

scrgadm -a -g *resource_group*
-y Nodelist=*secondary-node, primary-node*

次に例を示します。

```
scrgadm -a -g rg_PERSONNEL1
-y Nodelist=node2,node1
```

NodeList のノードの順序に注意してください。セカンダリ・コンパニオン・サーバ・リソース・グループでは、node2 がプライマリ・ノードで、node1 がセカンダリ・ノードです。

非対称型設定の場合は、次のように指定します。

```
scrgadm -a -g rg_PERSONNEL1
-y Nodelist=node2
```

2 SUNW.HAStorage リソースを作成して Adaptive Server リソース・グループに追加します。

scrgadm -a -j *hastorage_resource_name*
-g *resource_group*
-t SUNW.HAStorage
-x ServicePaths=*shared-disk-storage-path*

次に例を示します。

```
scrgadm -a -j has_PERSONNEL1
-g rg_PERSONNEL1
-t SUNW.HAStorage
-x ServicePaths=/global/node2_share
```

3 *SUNW.LogicalHostname* を作成して Adaptive Server リソース・グループに追加します。

scrgadm -a -L
-j *loghost_resource*
-g *resource_group*
-l *logical_hostname*

次に例を示します。

```
scrgadm -a -L
-j lh_PERSONNEL1
-g rg_PERSONNEL1
-l loghost_node2
```

4　*SY.ase* リソースを作成して Adaptive Server リソース・グループに追加します。

```
scrgadm -a -j ase_resource_name
-g resource_group \
-t  SY.ase \
-x Sybase_home=sybase_home_value \
-x Environment_file=environment_file_path \
-x Dataserver_name=dataserver_name_value \
-x Dataserver_login_file=login_file_path \
-x RUN_server_file=run_server_file_path
```

次に例を示します。

```
scrgadm -a -j ase_PERSONNEL1
-g rg_PERSONNEL1 \
-t SY.ase \
-x Sybase_home=/sybase \
-x Environment_file=/sybase/SYBASE.sh \
-x Dataserver_name=PERSONNEL1 \
-x Dataserver_login_file=/sybase/ASE-15_0/SC-3_0/etc/ase_login_file \
-x RUN_server_file=/sybase/ASE-15_0/install/RUN_PERSONNEL1
```

5　*SY.ase* リソースが常に *SUNW.HAStorage* リソースに依存するように、*SY.ase* と *SUNW.HAStorage* の間にリソース依存を確立します。

```
scrgadm -c -j ase_resource_name
-y Resource_dependencies=hastorage_resource_name
```

次に例を示します。

```
scrgadm -c -j ase_PERSONNEL1
-y Resource_dependencies=has_PERSONNEL1
```

6　セカンダリ Adaptive Server リソース・グループの場合は、scswitch コマンドを実行して次の作業を行います。

- リソース・グループを管理状態に移行する。
- すべてのリソースとそのモニタを有効にする。
- セカンダリ・ノード (セカンダリ・コンパニオン・リソース・グループのプライマリ・ノード) 上でリソース・グループをオンラインにする。

  ```
  scswitch -Z -g resource_group_name
  ```

  次に例を示します。

  ```
  scswitch -Z -g rg_PERSONNEL1
  ```

Adaptive Server のアップグレード

高可用性設定内の Adaptive Server をアップグレードするには、一時的にプライマリ・コンパニオンとセカンダリ・コンパニオンの間のコンパニオン関係を削除し、リソース・グループのモニタリングを無効にする必要があります。これにより、アップグレード・プロセスの間、SunCluster サブシステムによって予期しないフェールオーバがトリガされることなく、各 Adaptive Server を独立して停止または再起動できます。

注意 アップグレード・プロセスの間は、データベース、オブジェクト、ユーザ、またはログインを追加、削除、修正できません。コンパニオン関係を削除した後、その関係を再確立する前にこのような変更を行うと、アップグレードが失敗する場合があります。または、サーバ間の不整合が原因でクラスタが不安定になることがあります。

❖ モニタリング・サービスの停止とコンパニオン関係の削除

1 クラスタ内のすべてのノードで、モニタリング・サービスを停止し、Adaptive Server リソース・グループの管理を停止します。root 権限で、次のコマンドを発行します。

```
scswitch -F -g primary_resourcegroup_name
scswitch -u -g secondary_resourcegroup_name
```

2 セカンダリ・コンパニオンから、次のコマンドを発行します。

```
sp_companion primary_server_name, "drop"
```

3 (対称型設定の場合) セカンダリのコンパニオン関係を削除します。プライマリ・コンパニオンにログインして、次のコマンドを発行します。

```
sp_companion secondary_server_name,"drop"
```

4 両方のノードがシングルサーバ・モードであることを確認します。各ノードで次のコマンドを発行します。

```
sp_companion
```

コンパニオンがシングルサーバ・モードである場合は、次のような結果になります。

```
Server 'server_name' is not cluster configured.
Server 'server_name' is currently in 'Single server' mode.
```

これで、サーバがそのインストール・ノード上で実行されるようになります。サーバは、クラスタがノード間でリソースをフェールオーバしようとすることなく、独立して停止および起動できます。

❖ Adaptive Server のアップグレード

1　各ノードで、高可用性を無効にします。

```
sp_configure 'enable HA', 0
```

Adaptive Server を再起動して、この変更を有効にします。

注意　この代わりに、コンパニオンが停止している場合は、そのサーバ設定ファイル (*server_name.cfg*) を編集して、enable HA の値をゼロに変更できます。

2　『インストール・ガイド』の手順に従って各サーバをアップグレードします。

3　各ノードで、高可用性を再度有効にします。

```
sp_configure 'enable HA', 1
```

Adaptive Server を再起動して、変更を有効にします。『設定ガイド』を参照してください。

4　アップグレードしたサーバに、スクリプト (*installmaster*、*installhasvss*、*installsecurity* など) を再インストールします。詳細については、「installmaster の再インストール」と「installhasvss の再実行」(196 ページ) を参照してください。*installmaster* を再インストールする場合は、*installhasvss* を再インストールする必要があります。

5　sybha バイナリと *sybhausers* ファイルのパーミッションが正しく設定されていることを確認します。
root 権限で、次のコマンドを *$SYBASE/$SYBASE_ASE/bin* から発行します。

```
chown root sybha
chgrp sybhagrp sybha
chmod 4550 sybha
```

root 権限で、以下の手順を *$SYBASE/$SYBASE_ASE/install* から実行します。

1　sybase ユーザが *sybhauser* ファイルに含まれていることを確認します。

2　次のコマンドを発行します。

```
chown root sybhauser
chmod 600 sybhauser
```

6　次の内容を確認します。

- *$SYBASE* のインストール・ロケーション、または新しいインストール環境内の高可用性に関係するファイルにおいて、リソース・グループおよびリソース・プロパティ (たとえば、*Sybase_Home*、runserver ファイル、*Dataserver_login_file* など) に変更が正しく反映されている。

- 「アクティブ／アクティブ設定のための Adaptive Server の準備」(106 ページ) および「Sun Cluster サブシステムの設定」(112 ページ) で説明されている、コンパニオン関係の確立に必要なすべての手順を実施しており、システムがアップグレードの完了後もこれらの変更を保持している。
- 次のファイルが存在し、正しい情報が含まれている。

```
$SYBASE/$SYBASE_ASE/SC-3_0/etc/hacompanion.server_name
$SYBASE/$SYBASE_ASE/SC-3_0/etc/ase_login_file
```

❖ **コンパニオン関係の再確立とリソース・モニタリングの再開**

1. 各ノードで、Adaptive Server を手動で再起動します。
2. root 権限で、次のコマンドを発行してモニタリング・サービスをリストアします。

```
scswitch -z -g primary_resourcegroup_name -h primary_node
scswitch -z -g secondary_resourcegroup_name -h secondary_node
```

3. リソース・グループと Adaptive Server リソースの両方が各ノードでオンラインになっていることを確認します。確認するには、scstat -g コマンドを使用します (Sun のマニュアルを参照してください)。
4. isql を使用してプライマリ・コンパニオンとセカンダリ・コンパニオンにログインし、両方が実行中であることを確認します。
5. *SunCluster3.x* を高可用性ライブラリとして選択します (「Adaptive Server 内の高可用性サービス・ライブラリ」(123 ページ) を参照してください)。
6. サーバ間のコンパニオン関係を再確立します (「非対称型コンパニオン設定の作成」(125 ページ) または「対称型設定の作成」(127 ページ) を参照してください)。

```
dbcc traceon (2209)
sp_companion primary_server_name,configure
dbcc traceoff(2209)
```

注意 対称型設定の場合は、両方のコンパニオンでこのコマンドを発行します。

セカンダリ・サーバにユーザ・データベースが含まれる場合は、次のような警告メッセージが表示されることがあります。このようなメッセージは無視できます。

```
Msg 18739, Level 16, State 1:
Server 'server_name', Procedure 'sp_hacmpcfgvrfy', Line 102:
Database 'database_name': a user database exists. Drop this
database and retry the configuration again.
```

7. sp_companion を実行して、システムが対称型設定または非対称型設定用に適切に設定されていることを確認します。

❖ **フェールオーバとフェールバックの確認**

1　関連付けられているリソース・グループをセカンダリ・ノードに移動することによって、プライマリ・コンパニオンをフェールオーバします。root 権限で、次の作業を行います。

```
scswitch -z -g primary_resourcegroup_name -h secondary_node
```

セカンダリ・コンパニオンにログインし、sp_companion を発行してフェールオーバが正常に完了していることを確認します。

2　「プライマリ・コンパニオンへのフェールバック」(128 ページ) の項の手順に従って、Adaptive Server をフェールバックします。
プライマリ・コンパニオンとセカンダリ・コンパニオンにログインし、sp_companion を発行してフェールバックが正常に完了していることを確認します。

トラブルシューティング

この項では、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。

失敗した *prepare_failback* からのリカバリ

フェールバック中に、prepare_failback がセカンダリ・コンパニオンで正常に実行されたにもかかわらず、プライマリ・コンパニオンが失敗する場合は、ロールバックした後に、prepare_failback コマンドを再発行します。

1　クラスタ・システム・エラー・ログ、コールバック・エラー・ログ、高可用性エージェントのフォールト・モニタ・エラー・ログ、Adaptive Server エラー・ログをチェックし、フェールバックが失敗した原因を調べて問題を解決します。

2　リソース・グループのエラー状態をクリアします。リソース・グループの状態を確認するには、次のように入力します。

scha_resourcegroup_get -O RG_STATE -G *resource_group_name*

次に例を示します。

```
scha_resourcegroup_get -O RG_STATE -G rg_MONEY1
```

リソース・グループの状態を確認するには、次のように入力します。

scha_resource_get
-O RESOURCE_STATE_NODE
-R *resource_name node_name*

たとえば、node2 のリソース ase_MONEY1 の状態を調べるには、次のように入力します。

```
scha_resource_get
-O RESOURCE_STATE_NODE -R ase_MONEY1 node2
```

次のコマンドを発行して STOP_FAILED 状態をクリアします。

scswitch -c -h node_name -j *resource_name* -f STOP_FAILED

3 セカンダリ・コンパニオンにログインして、次のコマンドを発行します。

```
dbcc ha_admin("", "rollback_failback")
```

セカンダリ・コンパニオンでのセカンダリ・フェールオーバからのリカバリ

プライマリ・コンパニオンがノーマル・コンパニオン・モードになっていても、セカンダリ・コンパニオンがフェールオーバ・モードになっている場合、クラスタは整合性が失われた状態に陥っています。その場合は、手動でリカバリしてください。整合性が失われた状態は、sp_companion 'prepare_failback' がセカンダリ・コンパニオンで失敗することによって発生する可能性があります。リカバリするには、次の手順に従います。

1 セカンダリ・コンパニオンで sp_helpdb を発行して、プライマリ・コンパニオン・データベース (たとえば、master_companion) がセカンダリ・コンパニオンでマウントされているかどうかを調べます。

2 プライマリ・データベースにセカンダリ・ノードからアクセスできるようにします。これを行うには、プライマリの SUNW.HAStorage リソースをセカンダリ・ノードに移動します。移動はプライマリ Adaptive Server リソースを無効にして、セカンダリ・ノードでプライマリ・リソース・グループを開始することで実行できます。たとえば、次のように指定すると、プライマリ・リソース・グループ *rg_MONEY1* がセカンダリ・ノードで開始します。

```
scswitch -z -h node2  -g  rg_MONEY1
```

3 ha_admin を発行します。

```
dbcc ha_admin("",  "rollback_failover")
```

セカンダリ・コンパニオンのフェールオーバ防止

フェールオーバの後でモニタリングを無効にしてください。

リソースとリソース・グループの状態の変更

クラスタの保守を行うときは、Adaptive Server リソース・グループのすべてのリソースをオフラインにし、Adaptive Server リソース・グループを非管理状態にします。

```
scswitch -F -g primary-resource-group
scswitch -F -g secondary-resource-group
scswitch -u -g primary-resource-group
scswitch -u -g secondary-resource-group
```

エラー・ログのロケーション

これらのログの情報を使用して、高可用性システムをデバッグします。

- Adaptive Server エラー・ログ ― ロケーションは RUNSERVER ファイルに定義されています。次に例を示します。

  ```
  /sybase/ASE-15_0/install/MONEY1.log
  ```

- Adaptive Server 高可用性エージェント・コールバック・スクリプト・ログ

  $SYBASE/$SYBASE_ASE/SC-3_0/log/
  ase_callback_<*server-name*>.log

  または、Adaptive Server リソース・プロパティ *Callback_log* で指定されます。

- Adaptive Server エージェント・フォールト・モニタ・ログ

  $SYBASE/$SYBASE_ASE/SC-3_0/log/
  ase_monitor_<*server-name*>.log

  または、Adaptive Server リソース・プロパティ *Monitor_log* で指定されます。

- Sun Cluster システム・ログ

  ```
  /var/adm/messages
  ```

第 10 章

Sun Cluster 3.0 および 3.1 のアクティブ／パッシブ設定

この章では、Sun Cluster での Adaptive Server のアクティブ／パッシブ設定について説明します。

トピック名	ページ
ハードウェアとオペレーティング・システムの稼働条件	144
Sun Cluster のアクティブ／パッシブ設定	144
アクティブ／パッシブ設定のための Adaptive Server の準備	147
Sun Cluster サブシステムの設定	153
リソース・グループの手動設定	163
Adaptive Server のアップグレード	165
エラー・ログのロケーション	167

Adaptive Server Enterprise バージョン 15.0 は Sun Cluster バージョン 2.2 をサポートしていません。現在これらのクラスタを設定している場合は、それぞれのクラスタ・バージョンをアップグレードして、Sun Solaris で高可用性が使用できるように Adaptive Server 15.0 を設定する必要があります。

アクティブ／パッシブ設定は、2 つのノードと 1 つの Adaptive Server を含みます。通常の状態で Adaptive Server のホストとなるノードはプライマリ・ノードと呼ばれ、Adaptive Server のホストとなる可能性があるノードはセカンダリ・ノードと呼ばれます。

Adaptive Server または Adaptive Server が依存するリソース (ディスクまたはノードそのもの) がクラッシュすると、Adaptive Server は必要なリソースとともにセカンダリ・ノードに移動して再起動します。このようにプライマリ・ノードからセカンダリ・ノードに移動することを「フェールオーバ」と呼びます。

フェールオーバ後は、システム管理者が予定されたフェールバックを実行するか、新しいプライマリ・ノードの Adaptive Server で障害が発生してもう 1 回フェールオーバが発生するまで、Adaptive Server のホストとなるノードがプライマリ・ノードとみなされます。

フェールオーバが発生すると、既存のクライアント接続はすべて切断されます。Adaptive Server がセカンダリ・ノードで開始されるとすぐに、クライアントは接続を再確立して、コミットしていないトランザクションを再送信する必要があります。クライアントの接続フェールオーバは、高可用性接続を使用し、*interfaces* ファイルの `hafailover` エントリを自己参照することで、自動的に実行できます。詳細については、「クライアント側での interfaces ファイルの設定」(150 ページ) を参照してください。

ハードウェアとオペレーティング・システムの稼働条件

高可用性の要件は次のとおりです。

- CPU やメモリなどのリソースに関して同様に設定されている、2 台の同種のネットワーク・システム
- 高可用性パッケージと関連ハードウェア
- 両方のノードからアクセス可能なデバイス
- さまざまなクラスタ・ノードに割り当てるデバイス・パス名をユニークに保つための論理ボリューム・マネージャ (LVM)
- マルチホスト・ディスク上のボリュームまたはディスク・スイート・オブジェクト
- メディア障害に対処するためのサードパーティ・ベンダのミラーリング
- 任意のプライマリ・ノードとセカンダリ・ノードにバインドできる論理ホスト名または浮動 IP アドレス

Sun Cluster の実行に関する要件の詳細については、Sun Cluster のマニュアルを参照してください。

プラットフォームに固有の高可用性ソフトウェアのインストールについては、ご使用のハードウェアとソフトウェアのマニュアルを参照してください。

Sun Cluster のアクティブ／パッシブ設定

2 ノードのアクティブ／パッシブ設定を図 10-1 に示します。

Sun Cluster では、Adaptive Server はデータサービスとして実行され、Sun Cluster のリソースグループマネージャ (RGM) によって管理されます。Adaptive Server は、Adaptive Server リソースと必要なその他すべてのリソース (*SUNW.HAStorage*、*SUNW.HAStoragePlus*、*SUNW.LogicalHostname* リソースなど) を含むリソース・グループに関連付けられます。

SY.ase は Adaptive Server リソースであり、さまざまな拡張プロパティを定義します。詳細については、「Adaptive Server リソースの拡張プロパティ」(117 ページ) を参照してください。標準リソース・プロパティの詳細については、Sun Cluster のマニュアルを参照してください。

図 10-1: Sun Cluster のアクティブ／パッシブ設定 (フェールオーバ前)

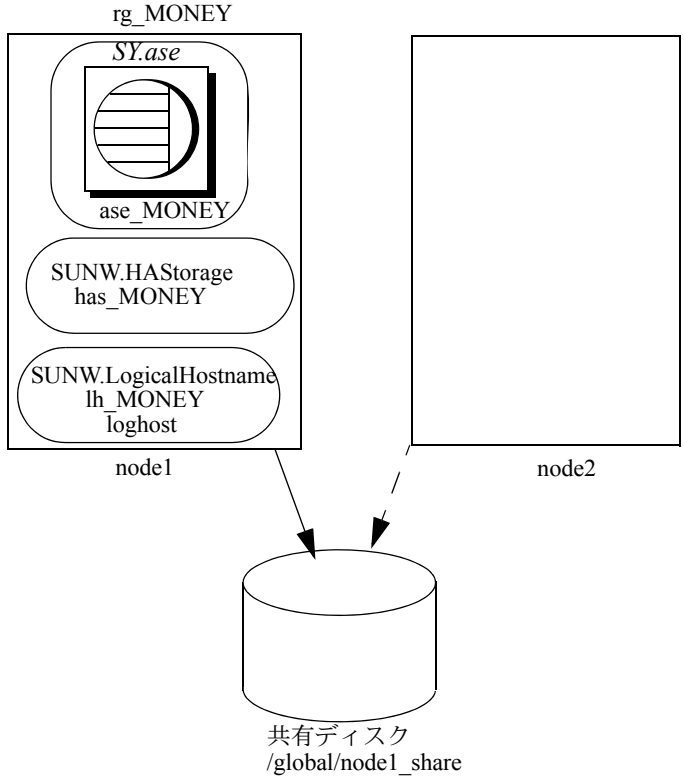


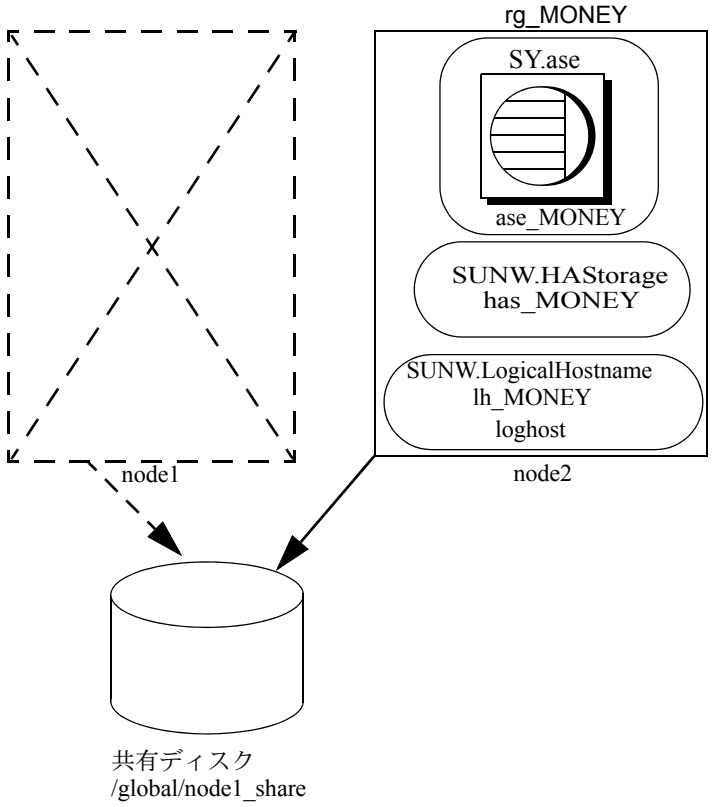
図 10-1 で、Adaptive Server MONEY は、リソース・グループ *rg_MONEY* に関連付けられています。このグループは、リソース・タイプ *SY.ase* の Adaptive Server リソース *ase_MONEY*、リソース・タイプ *SUNW.HAStorage* のストレージ・リソース *has_MONEY*、リソース・タイプ *SUNW.LogicalHostname* の論理ホスト・リソース *lh_MONEY* という 3 つのリソースで構成されています。*has_MONEY* は、共有ディスクの広域ファイル・システム */global/node1_share* を管理します。論理ホスト・リソースは、論理ホスト名または浮動 IP アドレス *loghost* に関連付けられます。*ase_MONEY* は *has_MONEY* および *lh_MONEY* を使用します。

最初、Adaptive Server リソース・グループ *rg_MONEY* のホストはプライマリ・ノード node1 です。Adaptive Server MONEY は、*lh_MONEY* に関連付けられている論理ホスト名 *loghost* を使用して、クライアントにサービスを提供します。

node1 がクラッシュすると、リソース・グループ *rg_MONEY* とそのすべてのリソースは、図 10-2 に示すようにセカンダリ・ノードに移動され、再起動されます。

フェールオーバ後、Adaptive Server は node2 上で実行され、同じ *loghost* を使用してクライアントへのサービスを続けます。

図 10-2: Sun Cluster のアクティブ／パッシブ設定 (フェールオーバ後)



リソース・グループ・プロパティ Pingpong_interval と Global_resources_used がフェールオーバに影響する場合があります。たとえば、Sun Cluster 3.0 update1 のマニュアルによると、Adaptive Server リソース・グループ *rg_MONEY* がプライマリ・ノードとセカンダリ・ノードの間を (300 秒以内に) 何度も移動すると、次のエラーが発生して RGM が Adaptive Server リソース・グループのフェールオーバを停止する場合があります。

```
608202 :scha_control: resource group ase_MONEY was frozen on
Global_resources_used within the past 300 seconds; exiting
```

アクティブ／パッシブ設定でのフェールバック

プライマリ・ノードがリカバリし、Adaptive Server リソース・グループのホストとして正常に稼働できるようになったら、Adaptive Server リソース・グループをプライマリ・ノードに移動できます。この動作をフェールバックといいます。アクティブ／パッシブ設定でのフェールバックは、プライマリ・ノードへのフェールオーバと同じです。つまり、現在のノード上の Adaptive Server とリソースを停止してから、プライマリ・ノードに移動して起動します。フェールバックは必須ではありませんが、管理だけの目的で実行できます。フェールバックを実行しない場合、リカバリされたプライマリ・ノードは次にフェールオーバが発生するまでセカンダリ・ノードとして稼働します。

アクティブ／パッシブ設定のクライアント

フェールオーバまたはフェールバックが発生すると、クライアントの既存の接続はすべて切断されます。この 2 つのイベントの違いはクライアントにはわかりません。ただし、クライアント接続フェールオーバが発生するしくみは、クライアントが Adaptive Server との間に確立している接続のタイプによって異なります。クライアント接続には、高可用性接続と非高可用性接続があります。

高可用性接続では、CS_HAFAILOVER プロパティが接続ハンドルに設定され、`hafailover` エントリが *interfaces* ファイルに設定される必要があります。高可用性接続を使用するクライアントでは、フェールオーバは透過的です。セカンダリー・ノードで Adaptive Server が再起動すると、切断された接続は自動的に再確立されます。ただし、クライアントは、コミットされていないトランザクションを再送信する必要があります。

非高可用性接続では自動的に再接続されません。クライアントは Adaptive Server への接続を再確立し、コミットされていないトランザクションを再送信する必要があります。

詳細については、「クライアント側での interfaces ファイルの設定」(150 ページ) を参照してください。

アクティブ／パッシブ設定のための Adaptive Server の準備

この項では、アクティブ／パッシブの高可用性を実現するための Adaptive Server の設定方法について説明します。

Adaptive Server のインストール

Adaptive Server は、プライマリ・ノードとセカンダリ・ノードの広域ファイル・システムまたはローカル・ファイル・システムにインストールできます。

広域ファイルシステムへのインストール

Adaptive Server を広域ファイルシステムにインストールする利点は、1 つのサーバのインストール環境のみを保守すればよいということです。ただし、Adaptive Server リソース・グループの *SUNW.HAStorage* または *SUNW.HAStoragePlus* リソースで管理されている広域ファイルシステムに Adaptive Server をインストールしてください。こうすることで、フェールオーバ時にインストール・ディレクトリ *$SYBASE* もセカンダリ・ノードに移動されます。

注意 SUNW.HAStoragePlus リソースで管理されるフェールオーバ・ファイル・システムに *$SYBASE* をインストールしないでください。

ローカル・ファイル・システムへのインストール

Adaptive Server をローカル・ファイル・システムにインストールする場合は、次のようにします。

- インストール・ディレクトリ *$SYBASE* のディレクトリ・パスをプライマリ・ノードとセカンダリ・ノードで同じにすること。*$SYBASE* ディレクトリがノードによって異なるロケーションにある場合は、プライマリ・ノードとセカンダリ・ノードで同じパスのディレクトリを作成します。このディレクトリは、実際の各 *$SYBASE* リリース・ディレクトリへのシンボリック・リンクとして機能します。

  たとえば、ディレクトリが、node1 では */usr/sybase1*、node2 では */usr/sybase2* である場合、各 *$SYBASE* リリース・ディレクトリへのシンボリック・リンク */sybase* を両方のノードに作成します。

  node1 では */sybase* は */usr/sybase1* へのリンクであり、node2 では */sybase* は */usr/sybase2* へのリンクです。このように、*$SYBASE* の値は、プライマリ・ノードでもセカンダリ・ノードでも同じパスを指します。

- プライマリ・ノードとセカンダリ・ノードすべての *$SYBASE* の以下の内容が同じであること。
  - RUNSERVER、interfaces、*SYBASE.sh*、サーバ設定ファイル、*<servername>.cfg* などのファイルの内容が同じである。
  - *$SYBASE/$SYBASE-ASE/SC-3_0* の内容、特に *etc* ディレクトリと *bin* ディレクトリのファイルが同じである。
  - アップグレードとパッチが同じように適用されている。
- 1 つのノードが Adaptive Server リソース・グループのホストとして稼働するときにはいつでも、各種ログ・ファイルが両方のノードに作成される。たとえば、*Callback_log*、*Monitor_log*、Adaptive Server と補助サーバ・エラー・ログなど。これらのファイルとすべての関連ファイルの一貫性を保つ必要があります。ファイルがデフォルト・ディレクトリにある場合も、対応する Adaptive Server リソース・プロパティを使用してファイルに別のディレクトリ・パスを指定した場合も同様です。

Adaptive Server への環境の引き渡し

SYBASE.sh ファイルを使用して、Adaptive Server に渡す環境を指定します。不正なアクセスから *SYBASE.sh* を保護するため、“root” のみが読み込みと実行のパーミッションを持つようにしてください。

高可用性エージェントは、*$SYBASE* 内、または Adaptive Server リソース・プロパティ *Environment_file* で指定されたロケーションでこのファイルを探します。*SYBASE.sh* が見つからない場合、高可用性エージェントが予想どおりに動作しない場合があります。

注意　*SYBASE.csh* ファイルはサポートされていません。

クラスタでの *SySam* License Manager の実行

SySam License Manager は、クラスタ内のプライマリ・ノードとセカンダリ・ノードで実行する必要があります。*$SYBASE* がローカル・ファイル・システムにインストールされている場合は、このための追加の作業は不要です。

$SYBASE が広域ファイル・システムにインストールされている場合は、次の手順に従って、両方のノードで同じ *license.dat* ファイルを使用して License Manager を実行します。

- */etc/hosts* ファイルで、プライマリ・ノードとセカンダリ・ノードの各物理ホスト名に対して同じエイリアスを作成します。

  たとえば、node1 と node2 がプライマリ・ノードとセカンダリ・ノードのホスト名である場合は、それぞれの */etc/hosts* ファイルでノードに対して同じエイリアス (*license_host* など) を追加します。

  たとえば、node1 の */etc/hosts* は次のようになります。

  ```
  10.22.98.43     node1   license_host
  10.22.98.44     node2
  ```

  node2 の */etc/hosts* は次のようになります。

  ```
  10.22.98.43     node1
  10.22.98.44     node2    license_host
  ```

- *$SYBASE/$SYBASE_SYSAM/licenses* 内、または環境変数 LM_LICENSE_FILE で指定されたロケーションにある *license.dat* ファイルを編集します。

  SERVER 行のホスト名を、*/etc/hosts* ファイルに定義したエイリアス・ホスト名に変更します。上記の例に続き、SERVER 行を次のように変更します。

  ```
  SERVER node1 any 1700

  SERVER license_host any 1700
  ```

SySAM の詳細は、『Sybase ソフトウェア資産管理 (SySAM) ユーザーズ・ガイド』を参照してください。

Adaptive Server のエントリを *interfaces* ファイルに追加する

interfaces ファイルは、サーバ側とクライアント側の両方で保持する必要があります。*interfaces* ファイルで Adaptive Server エントリに指定するホスト名は、プライマリ・ノードとセカンダリ・ノードの間で移動できる論理ホスト名または浮動 IP アドレスであることが必要です。

サーバ側での *interfaces* ファイルの設定

浮動 IP アドレスまたは論理ホスト名を使用して、サーバ・エントリの *interfaces* ファイルを修正します。サーバ側の *interfaces* ファイルのサーバ・エントリには *retry* オプションと *timeout* オプションを指定しないでください。次に、論理ホスト名 loghost を使用した、サーバ側の *interfaces* ファイルの例を示します。

```
MONEY
    master tcp ether loghost 4010
    query tcp ether loghost 4010
```

/etc/hosts または NIS ホスト・マップのファイルと */etc/nsswitch.conf* ファイルを適切に更新して、プライマリ・ノードとセカンダリ・ノードすべてからこの論理ホスト名にアクセスできるようにしてください。

注意 クラスタ環境では、NIS サーバへの不必要な依存を避けるために、NIS ホスト・マップではなくローカルの */etc/hosts* を使用することをおすすめします。*/etc/nsswitch.conf* ファイルを適切に修正してください。

たとえば、図 10-1 (145 ページ) の設定の */etc/hosts* ファイルは次のようになります。

```
#
#internet host table
#
10.22.98.43          node1
10.22.98.44         node2
10.22.98.165         loghost
```

/etc/nsswitch.conf ファイルのホスト・エントリは次のようになります。

```
hosts:    files nis dns
```

クライアント側での *interfaces* ファイルの設定

クライアント接続には、高可用性接続と非高可用性接続があります。どちらの場合もクライアント接続では次の設定が必要です。

- *interfaces* ファイルの *retry* オプションと *timeout* オプションには適切な値を指定すること。これらの値を決めるときは、セカンダリ・ノードでの Adaptive Server の起動やリカバリ時間などフェールオーバ時の遅延を考慮に入れます。
- 論理ホスト名がクライアント・マシンからアクセスできること。

非高可用性接続

非高可用性接続の *interfaces* ファイルには *hafailover* エントリが含まれません。また、クライアント接続には CS_HAFAILOVER プロパティが設定されていません。非高可用性接続が切断されると、クライアントは障害の後で Adaptive Server に再接続する必要があります。接続を再確立するには、フェールオーバが完了して Adaptive Server がセカンダリ・ノードで実行されるようになるまで、クライアントは何度もリトライするか、十分に待機時間をとってからリトライする必要があります。

サーバに再接続するために、クライアントは *interfaces* ファイルの *retry* オプションと *timeout* オプション、または対応する接続プロパティを使用できます。次の *interfaces* ファイルの例では、リトライ回数は 10 で、各リトライまでのタイムアウト遅延は 20 秒です。

```
MONEY    10    20
    master tcp ether loghost 4010
    query tcp ether loghost 4010
```

高可用性接続

高可用性接続は、以下によって確立されます。

- 接続レベルまたはコンテキスト・レベルで設定された CS_HAFAILOVER プロパティ (isql の -Q オプションと等価)。
- *interfaces* ファイルの *hafailover* エントリ (フェールオーバ時にコンタクトされる Adaptive Server エントリを指定する)。

アクティブ／パッシブ設定では、クライアントはフェールオーバ後に同じ Adaptive Server に再接続するため、*hafailover* エントリを自己参照する必要があります。つまり、同じ Adaptive Server がセカンダリ・ノードで再起動されるため、*interfaces* ファイルで *hafailover* サーバとして同じサーバ名を設定する必要があります。

たとえば、上記の例の Adaptive Server エントリは次のように自己参照できます。

```
MONEY    10    20
    master tli tcp loghost 4010
    query tli tcp loghost 4010
    hafailover MONEY
```

フェールオーバ・プロパティを使用するクライアント接続の設定については、「付録 A フェールオーバ設定での Open Client の機能」を参照してください。

設定パラメータの確認

Adaptive Server をアクティブ／パッシブ設定にするには、enable HA 設定パラメータを 2 に設定してください。デフォルトでは enable HA は 0 に設定されます。

enable HA を 2 に設定するには次のように入力します。

```
sp_configure "enable HA", 2
```

このパラメータを有効にするには、Adaptive Server の再起動が必要です。

『システム管理ガイド 第 1 巻』の「第 5 章 設定パラメータ」を参照してください。

マスタ・ログへのスレッショルドの追加

スレッショルドをマスタ・ログに追加していない場合は、追加してください。

1 ダンプ・トランザクションが発生する前に、master データベースのログに対して sp_thresholdaction を定義して実行し、残りのページ数にスレッショルドを設定します。Sybase では sp_thresholdaction を提供していません。このシステム・プロシージャの作成の詳細については、『システム管理ガイド 第 2 巻』の「第 16 章 スレッショルドによる空き領域の管理」と『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。

2 それぞれのスレッショルドをマスタ・ログ・セグメントに置いて、セグメントが満杯にならないようにします。

```
sp_addthreshold "master", "logsegment", 250, sp_thresholdaction
```

3 Adaptive Server を再起動して、この静的パラメータを有効にします。

フォールト・モニタ用のユーザとログインの追加

高可用性エージェント・フォールト・モニタ *ase_monitor* は、Sun Cluster 3.0 以上で thorough_probe を実行するとき、Adaptive Server のパフォーマンスを綿密に検査します。thorough_probe は、次のことを実行します。

1 Adaptive Server に接続します。

2 テンポラリ・テーブルの作成、テーブルへのエントリ挿入、テーブルの更新および削除を行います。

3 thorough_probe が *Connect_cycle_count* に指定された回数実行された後、Adaptive Server から切断します。次に、thorough_probe は新しい接続を確立します。

モニタが thorough_probe オペレーションを実行するために使用する特別なユーザとログインを作成または指定します。isql を使って dataserver に接続し、次のコマンドを発行します。

sp_addlogin *user for monitoring ase*, *password*
sp_adduser *user for monitoring ase*

次に例を示します。

```
sp_addlogin ase_monitor_user,ase_monitor_user_password
sp_adduser ase_monitor_user
```

注意　システム管理者は、Adaptive Server の設定時に、thorough_probe に使用するユーザとログインにより、他の目的に使用できる接続の合計数が実際には 1 つ減ることを考慮してください。つまり、接続の合計数が 25 の場合、1 つの接続がフォールト・モニタのプローブに使用されるので、ほかの目的に使用できる有効接続数は 24 になります。

Sun Cluster サブシステムの設定

高可用性システムのインストールについては、『Sun Cluster Installation Guide』を参照してください。

この項では、次のことを前提としています。

- クラスタ・システム・コマンドを実行するときに、PATH 環境変数が */usr/cluster/bin* を含むように設定されている。
- Sun Cluster 高可用性システムがインストールされている。
- 共有ディスクに Adaptive Server がインストールされ、必要なデータベース・デバイス・ファイルが作成されている。
- Adaptive Server が、「アクティブ／パッシブ設定のための Adaptive Server の準備」(147 ページ) の手順に従って設定されている。
- *$SYBASE/SYBASE.sh* が作成され、Adaptive Server で必要な環境に応じて編集されている。このファイルは高可用性エージェント・スクリプトで実行されるため、不正なアクセスからファイルを保護し、“root” ユーザのみが読み込みと実行のパーミッションを持つようにしてください。
- *$SYBASE/$SYBASE_ASE/install/RUN_<Dataserver_name>* ファイルが作成されている。このファイルで、-e オプションを使用して Adaptive Server エラー・ログを指定します。-s を指定する場合は、エラー・ログを Adaptive Server リソース・プロパティ *Dataserver_name* と同じにします。
- *$SYBASE/$SYBASE_ASE/SC-3_0* が正しくインストールされている (Adaptive Server とともに自動的にインストールされる)。このディレクトリに、Adaptive Server 高可用性エージェントに必要なファイルがすべて含まれる必要があります。

デフォルトの *$SYBASE/$SYBASE_ASE/SC-3_0/* には次のディレクトリがあります。

- *bin*
- *etc*

$SYBASE/$SYBASE_ASE/SC-3_0/bin には次のファイルがあります。

- *ase_start*
- *ase_stop*
- *ase_monitor_start*
- *ase_monitor_stop*
- *ase_update*
- *ase_validate*
- *utils.ksh*
- *ase_monitor*
- *syscadm*

$SYBASE/$SYBASE_ASE/SC-3_0/etc には次のファイルがあります。

- *SY.ase*
- *ase_monitor_action*
- *ase_login_file*
- *sysc_input_file*

$SYBASE/$SYBASE_ASE/SC-3_0/log には最初はファイルはありませんが、Adaptive Server リソースが作成されると、*Callback_log* ファイルと *Monitor_log* ファイルが生成されます。

log ディレクトリが存在しない場合は、*callback_log* ファイルと *monitor_log* ファイルを保管するためにこのディレクトリを作成する必要があります。

syscadm スクリプトの使用

syscadm スクリプトでは、Sun Cluster で Adaptive Server リソース・グループとその関連リソースの設定と管理を行います。*syscadm* を使用すると、Adaptive Server リソース・グループとそのリソースの作成、削除、無効化を行うことができます。*syscadm* スクリプトは *$SYBASE/$SYBASE_ASE/SC-3_0/bin/* にあります。

このスクリプトの create オプションは次のことを行います。

- 必要なリソース・タイプをリソースグループマネージャに登録する。

- 指定したリソース・グループごとに、リソース・グループを作成し、指定のリソースを作成し、作成したリソースをリソース・グループに追加する。
- ストレージリソースと論理ホストリソースに対する Adaptive Server リソースの依存性を確立する。

このスクリプトの remove オプションは、指定したリソース・グループとそのリソースを削除します。

unmanage オプションは次のことを行います。

- リソース・グループのすべてのリソースを無効にする。
- リソース・グループをオフライン状態にする。
- リソース・グループを非管理状態にする。

注意　*syscadm* を実行するには “root” 権限でログインしてください。

---

syscadm は、*sysc_input_file* という入力ファイルとともに機能します。このファイルは、適切な入力値を提供するように、ユーザが設定に応じて編集します。*sysc_input_file* は *$SYBASE/$SYBASE_ASE/SC-3_0/etc/* にあります。

注意　*sysc_input_file* の編集が終了したら、不正に変更されないようにしてください。このファイルの値が適切でない場合は、インストールに影響を与える可能性があります。システム管理者だけがこのファイルを編集できるように、ファイルのパーミッションを変更することをおすすめします。

---

sysc_input_file を編集するときは、次の点に注意してください。

- “<name>=<value>” エントリの “=” の前後にスペースを入力しない。
- コメントは # で開始する。
- プライマリ・コンパニオンの名前の末尾には 1 を付け、セカンダリ・コンパニオンには 2 を付ける。

sysc_input_file のサンプル

図 10-1 (145 ページ) の Adaptive Server リソース・グループ rg_MONEY とそのリソースを作成し、設定するために使用される *sysc_input_file* を次に示します。

```
##############################################################################
##NOTE:                                                                      ##
##    1. This file will be executed by ksh to set environment of syscadm ##
##       You will be responsible for executing anything in this file      ##
##       So, make sure THERE ARE NO DANGEROUS COMMANDS IN THIS FILE      ##
##                                                                        ##
##    2. No spaces around = in the <Variable_name>=<value> pairs         ##
##                                                                        ##
```

```
##    3. Comments should start with #, like ksh comments                 ##
##                                                                        ##
##    4. Names ending with 1 correspond to primary, and 2 to secondary   ##
############################################################################

############################################################
##  Section1: Must specify right hand side values        ##
############################################################
# Sybase home directory
SYBASE="/sybase"

# Valid HA Setups are "ACTIVE_PASSIVE" or "ASYMMETRIC" or "SYMMETRIC"
HA_SETUP="ACTIVE_PASSIVE"

# Comma separated list of nodes, Ex: "node1,node2"
Nodelist="node1,node2"

# ASE Dataserver name and Dataserver login file
Dataserver_name1="MONEY"
Dataserver_login_file1="/sybase/ASE-15_0/SC-3_0/etc/ase_login_file"

Dataserver_name2=
Dataserver_login_file2=

############################################################################
##  Section2: Must specify right hand side values, if required            ##
############################################################################

# if using Logical Hostname or Virtual/Floating IP address
LOGHOST_NAME_OR_FLOATING_IP1="loghost"
LOGHOST_NAME_OR_FLOATING_IP2=

# if using HAStorage resource
ServicePaths1="/global/node1_share"
ServicePaths2=

# if using HAStoragePlus resource
GlobalDevicePaths1=
FilesystemMountPoints1=

GlobalDevicePaths2=
FilesystemMountPoints2=

############################################################################
## Section3: May specify right hand side values to override defaults      ##
############################################################################

# bin of the cluster commands
CLUSTER_BIN="/usr/cluster/bin"

# ASE Resource Type and corresponding registration file
```

```
RT_NAME="SY.ase"
RT_FILE="$SYBASE/ASE-15_0/SC-3_0/etc/$RT_NAME"



# Resource Group names
RG_NAME1="rg_$Dtatserver_name1"
RG_NAME2="rg_$Dataserver_name2"

# ASE Resource names and space separated extended properties
ASE_RNAME1="ase_$Dataserver_name1"
ASE_RNAME2="ase_$Dataserver_name2"

OTHER_PROPERTIES1="RUN_server_file=/sybase/ASE-15_0/install/RUN_MONEY"
OTHER_PROPERTIES2="RUN_server_file=  Callback_log=  Monitor_log="

# Logical Host Resource names
LOGHOST_RNAME1="lh_$Dataserver_name1"
LOGHOST_RNAME2="lh_$Dataserver_name2"

# HA Storage Resource names
HASTORAGE_RNAME1="has_$Dataserver_name1"
HASTORAGE_RNAME2="has_$Dataserver_name2"

# HA Storage Plus Resource names
HASTORAGE_PLUS_RNAME1="hasp_$Dataserver_name1"
HASTORAGE_PLUS_RNAME2="hasp_$Dataserver_name2"
```

この入力ファイルは次の 3 つのセクションに分かれています。

- Section 1 － すべてのエントリの右辺値を入力します。このセクションには、Adaptive Server インストール・ディレクトリ、高可用性設定、データ・サーバ名、ノード・リストなどのエントリが含まれます。
- Section 2 － 必須エントリの右辺値を入力します。たとえば、SUNW.HAStoragePlus リソースのみを使用する場合は、SUNW.HAStoragePlus に関連するエントリの値を入力してください。使用していないエントリの値は入力しないでください。
- Section 3 － このセクションのすべてのエントリにはデフォルト値が割り当てられます。デフォルトを無効にしない場合は、右辺値を入力する必要はありません。

たとえば、Adaptive Server リソース名のファイルを編集するには、次の行を変更します。

```
ASE_RNAME="ase_$Dataserver_name"
```

次のように変更します。

```
ASE_RNAME="MONEY_RNAME"
OTHER_PROPERTIES="RUN_server_file=/mypath/RUN_MONEY
Debug_callback=TRUE"
```

または、*RUN_SERVER* ファイルを指定して、*Debug_callback* フラグを設定するには、*OTHER_PROPERTIES* のエントリを変更します。この値は、<*name*>=<*value*> という文字列をスペースで区切って指定します。

syscadm の構文は次のとおりです。

```
syscadm [-v] -c|r|u [primary|secondary|both] -f <sysc_input_file>
syscadm [-v] -r|u <rg1,rg2,...> [-t <ASE_resource_type>]
```

各パラメータの意味は次のとおりです。

- -c はリソース・グループを作成します。
- -r はリソース・グループを削除します。
- -u はリソース・グループを非管理状態にします。
- -f は入力ファイルを指定します。
- -v は冗長出力 (実行時の Sun Cluster コマンド表示) のことです。
- –t は、*SY.ase* ではない場合の Adaptive Server リソース・タイプ名を指定します (入力ファイルが指定されていない場合に、-r コマンドと -u コマンドで役立ちます)。

SUNW.HAStoragePlus リソースは、`AffinityOn=True` を指定すると作成されます。

注意 アクティブ／パッシブ設定では、プライマリでのみ、-c オプションを指定して Adaptive Server リソース・グループを作成してください。

Adaptive Server リソース・グループの設定

1 Adaptive Server のリソース・タイプ登録ファイル *SY.ase* を修正します。このファイルは *$SYBASE/$SYBASE_ASE/SC-3_0/etc/* にあります。Adaptive Server 高可用性エージェントのロケーションを指定するリソース・タイプ・プロパティ *RT_BASEDIR* の行を探します。この値を、*$SYBASE/$SYBASE_ASE/SC-3_0/bin* のインストール・ロケーションを指すように変更します。

次に例を示します。

```
RT_BASEDIR=/sybase/ASE-15_0/SC-3_0/bin/
```

2　システム管理者とフォールト・モニタ用に追加したユーザの Adaptive Server ログイン情報が入っているファイルを作成または編集します。デフォルトのファイルは *$SYBASE/$SYBASE_ASE/SC-3_0/etc/ase_login_file* です。別のロケーションにある別のファイルを使用する場合は、*SY.ase* リソースの設定時に、リソース拡張プロパティ *Dataserver_login_file* にフル・パスを指定します。このファイルは 2 行で構成されています。1 行目はシステム管理者のログインとパスワードで、2 行目はモニタ・ユーザのログインとパスワードです。

login_type <tab> *login_string*
login_type <tab> *login_string*

ログイン・タイプの有効値は normal のみです。login string の値は、login-name/password という形式で指定します。*$SYBASE/$SYBASE_ASE/SC-3_0/etc/ase_login_file* の例を次に示します。

normal <tab> *sa/sa-password*
normal <tab> *ase_monitor_user/ase_monitor_user_password*

注意　適切な値で編集したら、“root” ユーザしかこのファイルを読めないようにしてください。

```
chmod 400 ase_login_file
chown root ase_login_file
chgrp sys ase_login_file
```

3　*sysc_input_file* を作成または編集し、syscadm を実行します。これによって、リソース・タイプの登録、リソース・グループの作成、リソース・グループへのリソースの追加、リソース依存の確立が行われます。次に例を示します。

```
syscadm -c primary
-f $SYBASE/$SYBASE_ASE/SC-3_0/etc/sysc_input_file
```

詳細については、「syscadm スクリプトの使用」(113 ページ) を参照してください。

syscadm コマンドで行われる手順を手動で実行することもできます。詳細については、「リソース・グループの手動設定」(163 ページ) を参照してください。

注意　拡張プロパティのリストについては、表 9-1 (117 ページ) を参照してください。

4　scswitch を実行して次の作業を行います。

- リソース・グループを管理状態に移行する。
- すべてのリソースとそのモニタを有効にする。

- プライマリ・ノード上でリソース・グループをオンラインにする。

  scswitch -Z -g *resource_group_name*

次に例を示します。

```
scswitch -Z -g rg_MONEY
```

SUNW.HAStoragePlus の使用

Sun Cluster 3.0 Update2 以降を実行している場合は、Adaptive Server リソース・グループで *SUNW.HAStoragePlus* リソースを使用できます。*SUNW.HAStoragePlus* リソースを *SUNW.HAStorage* リソースの代わりに使用するか、*SUNW.HAStorage* リソースと *SUNW.HAStoragePlus* リソースを両方ともリソース・グループで使用できます。

SUNW.HAStoragePlus リソースを Adaptive Server リソース・グループに追加するには、*SUNW.HAStoragePlus* リソース・プロパティ *GlobalDevicePaths* と *FilesystemMountPoints* を必要に応じて設定してください。*syscadm* を使用している場合は、*sysc_input_file* の対応するエントリに値を指定できます。接続を有効にするために、*SUNW.HAStoragePlus* リソース・プロパティ AffinityOn を TRUE に設定してください。

SUNW.HAStoragePlus リソースを手動で追加するには、次の手順に従います。

1 リソース・タイプ *SUNW.HAStoragePlus* を登録します。

```
scrgadm -a -t SUNW.HAStoragePlus
```

2 *SUNW.HAStoragePlus* リソースを Adaptive Server リソース・グループに追加します。

scrgadm -a -j *hasp_resource_name*
-t SUNW.HAStoragePlus
-g resource_type
-x FilesystemMountPoints=*shared_disk_filesystem*
-x AffinityOn=TRUE

次に例を示します。

```
scrgadm -a -j hasp_MONEY
-t SUNW.HAStoragePlus
-g rg_MONEY
-x fileSystemMountPoints=\global\node1_share
-x Affinityon=TRUE
```

SUNW.HAStoragePlus リソースを使用すると、Adaptive Server データベース・デバイスを、広域ファイル・システムにも *SUNW.HAStoragePlus* リソースで管理されるフェールオーバ・ファイル・システム (FFS) にも作成できます。いずれの場合も、データは共有ディスクに存在する必要があります。*SUNW.HAStoragePlus* リソースを作成するときに、対応するすべてのファイル・システムとデバイス・パスを指定してください。

- *SUNW.HAStoragePlus* リソースを有効にします。

  scswitch -e -j *hastorageplus_name*

  次に例を示します。

  ```
  scswitch -e -j hasp_MONEY
  ```

- *SY.ase* リソースと *SUNW.HAStoragePlus* リソースの間にリソース依存を確立します。

  scrgadm -c -j *ase_resource_name*
  -y Resource_dependencies=*hastorageplus_name*

  次に例を示します。

  ```
  scrgadm -c -j ase_MONEY
  -y Resource_dependencies=hasp_MONEY
  ```

  SUNW.HAStorage リソースと *SUNW.HAStoragePlus* リソースの両方を使用している場合は、すべてのストレージ・リソース名をカンマで区切ったリストとして指定してください。

  scrgadm -c -j *ase_resource_name*
  -y Resource_dependencies=*hastorageplus-name*, *hastorage-name*

  次に例を示します。

  ```
  scrgadm -c -j MONEY
  -y Resource_dependencies=has_MONEY,hasp_MONEY
  ```

SUNW.HAStoragePlus リソース・タイプの詳細については、Sun Cluster のマニュアルを参照してください。

アクティブ／パッシブ設定の確認

次のテストを実行して、Sun Cluster でアクティブ／パッシブの高可用性を実現できるように Adaptive Server を正しくインストールして設定したことを確認します。

1　プライマリ・ノードのリソース・グループをオンラインにし、リソース・グループのすべてのリソースとフォールト・モニタを有効にします。次に例を示します。

```
scswitch -Z -g rg_MONEY
```

2　isql などのクライアントが論理ホストを使用して Adaptive Server に接続することを確認します。クライアント接続フェールオーバを確認するために、Adaptive Server に接続します。isql を使用して高可用性接続を確立します (必要であれば、*interfaces* ファイルを修正して、hafailover エントリを自己参照します)。

```
isql -Usa -Ppassword -SMONEY -Q
1> select @@servername
2> go
```

```
------------------------------
MONEY

(1 row affected)
```

3 次のいずれかの方法でフェールオーバをシミュレートします。サーバを停止します。

```
isql -Usa -Ppassword -SMONEY
1> shutdown with nowait
2> go
```

または、Adaptive Server リソース・グループをセカンダリ・ノードに移動します。

```
scswitch -z -h node2 -g rg_MONEY
```

4 手順 2 で起動した isql セッションで次のコマンドを発行して、接続フェールオーバを確認します。

```
1> select @@servername
2> go
CT-LIBRARY error:
         ct_results(): user api layer: internal Client
Library error:
HAFAILOVER:Trying to connect to MONEY server.
1> select @@servername
2> go
------------------------------
MONEY

(1 row affected)
```

5 リソース・グループをプライマリ・ノードに戻して、フェールバックをシミュレートします。

```
scswitch -z -h node1 -g rg_MONEY
```

6 手順 2 で起動した isql セッションで次のコマンドを発行して、接続フェールオーバを確認します。

```
1> select @@servername
2> go
CT-LIBRARY error:
         ct_results(): user api layer: internal Client
Library error:
HAFAILOVER:Trying to connect to MONEY server.
1> select @@servername
2> go
------------------------------
MONEY

(1 row affected)
```

リソース・グループの手動設定

この項では、Adaptive Server リソース・グループを作成し、設定するために *syscadm* スクリプトで実行されるコマンドについて説明します。

必要に応じてこれらの手順を手動で実行して、Adaptive Server リソース・グループの設定、再設定、トラブルシューティングを行うことができます。*SY.ase* ファイルと *ase_login_file* ファイルを「Adaptive Server リソース・グループの設定」(158 ページ)の手順 1 と 2 で適切に修正したことを確認します。

次に示す Sun Cluster コマンドを実行するには、“root” 権限でログインしてください。

1　*SY.ase* リソース・タイプを登録します。

```
scrgadm -a -t SY.ase -f full-path-of-SY.ase-file
```

次に例を示します。

```
scrgadm -a -t SY.ase
-f /sybase/ASE-15_0/SC-3_0/etc/SY.ase
```

2　Adaptive Server リソース・グループを作成します。リソース・グループ・プロパティ Nodelist でプライマリ・ノードとセカンダリ・ノードを指定します。

```
scrgadm -a -g resource_group
-y Nodelist=primary-node,secondary-node
```

次に例を示します。

```
scrgadm -a -g rg_MONEY -y Nodelist=node1,node2
```

3　*SUNW.HAStorage* リソース・タイプを登録します。

```
scrgadm -a -t SUNW.HAStorage
```

4　*SUNW.HAStorage* リソースを作成して Adaptive Server リソース・グループに追加します。フェールオーバが発生したときにセカンダリ・ノードに移動する、共有ディスク上のファイル・システムとデバイスのパスを指定します。

```
scrgadm -a -j hastorage_resource_name
-t SUNW.HAStorage
-g resource_group
-x ServicePaths=shared-disk-storage-path
```

次に例を示します。

```
scrgadm -a -j has_MONEY -g rg_MONEY
-t SUNW.HAStorage
-x ServicePaths=/global/node1_share
```

5 *SUNW.LogicalHostname* リソースを作成して Adaptive Server リソース・グループに追加します。フェールオーバが発生したときにセカンダリ・ノードに移動する論理ホスト名または浮動 IP アドレスを指定します。

```
scrgadm -a -L -j loghost_resource_name
-g resource_group
-l logicalhostname
```

次に例を示します。

```
scrgadm -a -L -j lh_MONEY -g rg_MONEY -l loghost
```

6 *SY.ase* リソースを作成して Adaptive Server リソース・グループに追加します。Adaptive Server リソースに対して標準リソースプロパティ値と拡張プロパティ値を指定します。3 つの拡張プロパティ値 (*Sybase_home*、*Dataserver_name*、*Dataserver_login_file*) を指定してください。指定しないとコマンドが失敗します。

ほかの拡張プロパティではデフォルト値を使用できます。高可用性エージェント・フォールト・モニタで使用される標準リソース・プロパティ *Cheap_probe_interval*、*Thorough_probe_interval*、*Retry_count*、*Retry_interval* を設定します。

次のコマンドは、Adaptive Server リソースを作成して、それをリソース・グループに追加します。

scrgadm -a -j *ase_resource_name* -g *resource_group* \
-t SY.ase \
-x Sybase_home=*sybase_home_value* \
-x Environment_file=*environment_file_path* \
-x Dataserver_name=*dataserver_name_value* \
-x Dataserver_login_file=*login_file_path* \
-x RUN_server_file=*run_server_file_path*

次に例を示します。

```
scrgadm -a -j ase_MONEY -g rg_MONEY \
    -t SY.ase \
    -x Sybase_home=/sybase \
    -x Environment_file=/sybase/SYBASE.sh \
    -x Dataserver_name=MONEY \
    -x Dataserver_login_file=/sybase/ASE-15_0/SC-    3_0/etc/ase_login_file\
    -x RUN_server_file=/sybase/ASE-15_0/install/RUN_MONEY
```

標準リソースプロパティの詳細については、Sun Cluster のマニュアルを参照してください。Adaptive Server リソースの拡張プロパティの詳細については、表 9-1 (117 ページ) を参照してください。

7 *SY.ase* リソースと *SUNW.HAStorage* リソースの間にリソース依存を確立します。つまり、*SUNW.HAStorage* リソースがオンラインになった後にのみ *SY.ase* リソースがオンラインになり、*SY.ase* リソースがオフラインになった後で *SUNW.HAStorage* リソースがオフラインになります。

scrgadm -c -j *ase_resource_name* -y
Resource_dependencies=*hastorage_resource_name*

次に例を示します。

```
scrgadm -c -j ase_MONEY -y Resource_dependencies=has_MONEY
```

注意　リソース・グループに追加されたリソースは、すべて暗黙的にその *SUNW.LogocalHostname* リソースに依存します。

8　scswitch を実行して次の作業を行います。

- リソース・グループを管理状態に移行する。
- すべてのリソースとそのモニタを有効にする。
- プライマリ・ノード上でリソース・グループをオンラインにする。

  scswitch -Z -g *resource_group_name*

次に例を示します。`scswitch -Z -g rg_MONEY`

注意　「SUNW.HAStoragePlus の使用」(121 ページ) を参照して、*SUNW.HAStoragePlus* リソースを作成し、Adaptive Server リソース・グループに追加してください。

Adaptive Server のアップグレード

高可用性設定内の Adaptive Server をアップグレードするには、一時的にプライマリ・コンパニオンとセカンダリ・コンパニオンの間のコンパニオン関係を削除し、Adaptive Server リソース・グループのモニタリングを無効にする必要があります。これにより、アップグレード・プロセスの間、SunCluster サブシステムによって予期しないフェールオーバがトリガされることなく、各 Adaptive Server を独立して停止または再起動できます。

注意　アップグレード・プロセスの間は、データベース、オブジェクト、ユーザ、またはログインを追加、削除、修正できません。コンパニオン関係を削除した後、その関係を再確立する前にこのような変更を行うと、アップグレードが失敗する場合があります。または、サーバ間の不整合が原因でクラスタが不安定になることがあります。

❖ モニタリング・サービスの停止とコンパニオン関係の削除

クラスタ内の両方のノードで、モニタリング・サービスを停止し、Adaptive Server リソース・グループの管理を停止します。

- root 権限で次のコマンドを発行して、Adaptive Server に関連付けられている Sun Cluster リソース・グループをオフラインにし、Adaptive Server を停止します。

```
scswitch -F -g primary_resourcegroup_name
scswitch -u -g secondary_resourcegroup_name
```

❖ Adaptive Server のアップグレード

1 両方のノードで、高可用性を無効にします。

```
sp_configure 'enable HA', 0
```

Adaptive Server を再起動して、この変更を有効にします。

注意 この代わりに、コンパニオンが停止している場合は、そのサーバ設定ファイル (*server_name.cfg*) を編集して、enable HA の値を 0 に変更します。

---

2 『インストール・ガイド』の手順に従って各サーバをアップグレードします。

3 両方のノードで、高可用性を再度有効にします。

```
sp_configure 'enable HA', 2
```

変更を有効にするには、Adaptive Server を再起動します。使用しているプラットフォームの『設定ガイド』を参照してください。

4 次の内容を確認します。

- *$SYBASE* のインストール・ロケーション、または新しいインストール環境内の高可用性に関係するファイルにおいて、リソース・グループおよびリソース・プロパティ (たとえば、*Sybase_Home*、runserver ファイル、*Dataserver_login_file* など) に変更が正しく反映されている。
- 「アクティブ／パッシブ設定のための Adaptive Server の準備」(147 ページ) および「Sun Cluster サブシステムの設定」(153 ページ) で説明されている、コンパニオン関係の確立に必要なすべての手順を実施しており、システムがアップグレードの完了後もこれらの変更を保持している。

❖ コンパニオン関係の再確立とリソース・モニタリングの再開

リソース・グループ (Adaptive Server がまだ起動されていない場合に、Adaptive Server を起動するリソース・グループ) を開始し、モニタリング・サービスをリストアします。

- root 権限で、次のコマンドを発行します。

```
scswitch -z -g ase_resourcegroup_name -h primary_node
```

❖ フェールオーバとフェールバックの確認

1　関連付けられているリソース・グループをセカンダリ・ノードに移動することによって、プライマリ・コンパニオンをフェールオーバします。root 権限で、次のコマンドを発行します。

```
scswitch -z -g ase_resourcegroup_name -h secondary_node
```

Adaptive Server オペレーティング・システム・プロセスをチェックし、Adaptive Server に isql でログインすることによって、Adaptive Server がセカンダリ・ノードで正常に実行されていることを確認します。

2　関連付けられているリソース・グループをプライマリ・ノードに戻すことによって、Adaptive Server をフェールバックします。root 権限で、次のコマンドを発行します。

```
scswitch -z -g ase_resourcegroup_name -h primary_node
```

Adaptive Server オペレーティング・システム・プロセスをチェックし、Adaptive Server に isql でログインすることによって、Adaptive Server がプライマリ・ノードで正常に実行されていることを確認します。

エラー・ログのロケーション

次の情報を使用して、高可用性システムをデバッグします。

- Adaptive Server エラー・ログ － ロケーションは RUNSERVER ファイルに指定されています。次に例を示します。

  ```
  /sybase/ASE-15_0/install/MONEY.log
  ```

- Adaptive Server 高可用性エージェント・コールバック・スクリプト・ログ

  $SYBASE/$SYBASE_ASE/SC-3_0
  /log/ase_callback_<*server-name*>.log

  または、Adaptive Server リソース・プロパティ *Callback_log* で指定されます。

- Adaptive Server エージェント・フォールト・モニタ・ログ

  $SYBASE/$SYBASE_ASE/SC-3_0/log
  /ase_monitor_<*server-name*>.log

  または、Adaptive Server リソース・プロパティ *Monitor_log* で指定されます。

- Sun Cluster システム・ログ

  ```
  /var/adm/messages
  ```

第 11 章

Veritas 5.0 以降でフェールオーバを使用できるように Adaptive Server を設定する

この章では、Veritas Cluster Server (VCS) バージョン 5.0 以降で Linux Adaptive Server をフェールオーバ用に設定する方法について説明します。

トピック名	ページ
ハードウェアとオペレーティング・システムの稼働条件	170
高可用性で動作するように Adaptive Server を準備する	171
Veritas サブシステムを Sybase フェールオーバ用に設定する	177
フェールオーバ用コンパニオン・サーバの設定	181
Sybase フェールオーバの管理	185
リソース・タイプ Sybase のエージェントからのアップグレード	187
Adaptive Server のアップグレード	188
Veritas クラスタでのフェールオーバのトラブルシューティング	191

注意 Adaptive Server Enterprise バージョン 15.5 は、Sun Cluster 2.2, 3.0, 3.1 をサポートしません。これらのクラスタを現在設定している場合は、Veritas で Adaptive Server の高可用性を設定するために、対応するクラスタのバージョンをアップグレードする必要があります。

Veritas のユーザ・マニュアルを読み、Veritas のクラスタについてよく理解してから、この章の手順を実行してください。

注意 Veritas をアップグレードする場合は、「リソース・タイプ Sybase のエージェントからのアップグレード」(187 ページ) の内容を確認してから、この章の作業を実行してください。

ハードウェアとオペレーティング・システムの稼働条件

高可用性を実現するには、次のハードウェアとシステム・コンポーネントが必要です。

- CPU やメモリなどのリソースに関して同様に設定されている、2 台の同種のネットワーク・システム。インストールの設定については、「第 2 章 高可用性の概要」を参照してください。

  設定と管理を容易にする VCS グラフィカル・ユーザ・インタフェースもインストールします。

- VCS にインポートされたリソース・タイプ *HAase*。
- 高可用性の設定が行われている Adaptive Server のデータベースを格納する共有マルチホスト・ディスクへのアクセス。
- ディスクの管理および DiskGroup や Volume などのリソースの作成を行うための Veritas Volume Manager 3.1 以降。
- メディア障害に対処するためのサードパーティ・ベンダのミラーリング。
- システムごとのサービス・グループ。サービス・グループは固有のサービスを提供する一連のリソースです。高可用性の設定が行われている Adaptive Server のサービスを提供するために、サービス・グループには Adaptive Server 用の DiskGroup、Volume、Mount、IP、NIC、*HAase* などのリソースを含めてください。サービス・グループのサンプルとリソース依存を図 11-1 に示します。サービス・グループの作成方法とリソースのサービス・グループへの追加方法の詳細については、『Veritas Cluster Server User's Guide』を参照してください。

  注意 各サービス・グループには少なくとも 2 つのリソースを含め、そのうちの 1 つはリソース・タイプ *HAase* にします。リソース・タイプ *HAase* が他のリソースに依存するように、クラスタ・コマンドを使用してリソース依存を確立します。

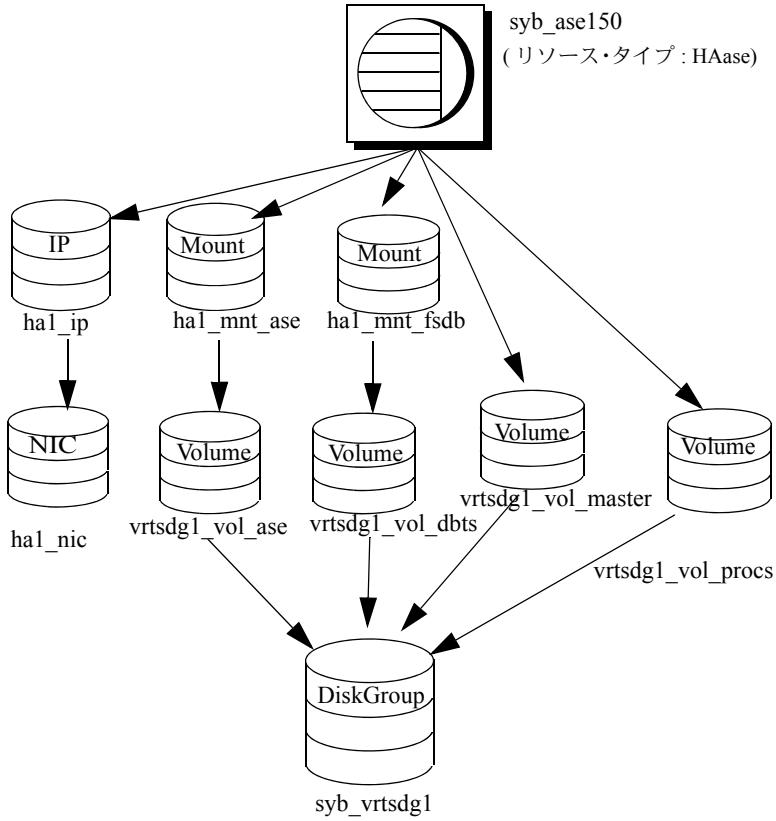
- 両方のノード上に、パブリックとプライベートの両ネットワーク。

プラットフォームに固有の高可用性ソフトウェアのインストールについては、ご使用のハードウェアとソフトウェアのマニュアルを参照してください。

図 11-1 (171 ページ) のサービス・グループの設定では、1 つの DiskGroup である *syb_vrtsdg1* 上に 4 つのボリュームが作成されています。ボリュームは、Adaptive Server インストール用に 1 つ、ファイル・システム上に作成されるデータベース用に 1 つと、後の 2 つはロー・デバイスに作成されるデータベース用です。2 つのマウント・リソースは、ボリューム・リソースの上のレイヤにあるタイプ ufs のファイル・システム用に作成されます。タイプ *HAase* のリソース *syb_ase150* は、Adaptive Server のインストール環境であり、マウント・リソースの上に位置します。また、*syb_ase150* はリソース IP も必要とし、これはさらにパブリック・ネットワーク・アクセス用のリソース NIC を必要とします。

図 11-1 (171 ページ) には表示されていませんが、サービス・グループ *SybASE* はプライマリ・ノードで動作し、サービス・グループ *SybASE2* は同様の設定でセカンダリ・ノードで動作しています。

図 11-1: Veritas Cluster Server で動作するサービス・グループのサンプル



高可用性で動作するように Adaptive Server を準備する

この項では、Adaptive Server を高可用性に設定するために必要な作業について説明します。

Adaptive Server のインストール

プライマリ・サーバとセカンダリ・サーバの両方をインストールします。共有ディスクまたはローカル・ディスクにインストールできます。プライマリ・コンパニオンは、新しくインストールした Adaptive Server でも、旧バージョンの Adaptive Server からアップグレードして既存のデータベースやユーザなどを受け継いだものでもかまいません。セカンダリ・コンパニオンは、新しくインストールした Adaptive Server である必要があり、すべてのユーザ・ログインやデータベース名がクラスタ内でユニークであることを確保するため、ユーザ・ログインまたはユーザ・データベースを持つことができません。フェールオーバ用の設定を完了したら、セカンダリ・コンパニオンにユーザ・ログインやデータベースを追加できます。

ローカル・ディスクにインストールする場合は、すべてのデータベースがマルチホスト・ディスク上に作成されていることを確認してください。

Adaptive Server のインストールと設定については、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。

両方の Adaptive Server のエントリを *interfaces* ファイルに追加する

プライマリ・コンパニオンとセカンダリ・コンパニオンの *interfaces* ファイルには、この両方のコンパニオンのエントリが必要です。*interfaces* ファイル内のサーバ・エントリには、sysservers に指定されているネットワーク名を使用してください。*interfaces* ファイルへのエントリの追加については、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。

フェールオーバ中にクライアント接続を行うために *interfaces* ファイルにエントリを追加する

フェールオーバしたコンパニオンにクライアントが再接続できるようにするには、*interfaces* ファイルに行を追加します。デフォルトでは、クライアントはサーバ・エントリの query 行にリストされているポートに接続します。サーバのフェールオーバが原因でこのポートが使用できない場合、クライアントはサーバ・エントリの *hafailover* 行で指定されているサーバに接続します。次に示すのは、MONEY1 という名前のプライマリ・コンパニオン用と PERSONNEL1 という名前のセカンダリ・コンパニオン用の *interfaces* ファイルの例です。

```
MONEY1
        master tli tcp MONEY 9678
        query tli tcp MONEY 9678
        hafailover PERSONNEL1

PERSONNEL1
        master tli tcp PERSONNEL 9679
        query tli tcp PERSONNEL 9679
```

interfaces ファイルにエントリを追加するには、dsedit を使用します。interfaces エントリがすでに存在する場合は、フェールオーバに使用できるように変更します。

dsedit については、『ユーティリティ・ガイド』を参照してください。

sybha 実行プログラム

sybha 実行プログラムによって、Adaptive Server High Availability Basis Services ライブラリは、各プラットフォームの高可用性クラスタ・サブシステムと対話できるようになります。Adaptive Server High Availability Basis Services ライブラリは、*$SYBASE/$SYBASE_ASE/bin* にある *sybha* を呼び出します。*sybha* は、所有権とパーミッションを変更することにより実行可能になります。*$SYBASE/$SYBASE_ASE/install* の *sybhauser* という名前のファイルも編集する必要があります。*sybhauser* には、そのクラスタに対してシステム管理者権限を持つユーザのリストがあります。クラスタに対するシステム管理者権限を持つユーザの数を制限することを強くおすすめします。

“root” 権限で、次の作業を行います。

1　*sybhagrp* という新しいグループを追加します。このグループを */etc/group* ファイルに追加することも、または NIS マップに追加することもできます。このグループに sybase ユーザを追加します (これは *$SYBASE* ディレクトリを所有するユーザです)。サーバの起動時に sybase ユーザがデータ・サーバを実行します。複数のサーバを実行し、各サーバの *$SYBASE* ディレクトリを異なるユーザが所有する場合は、これらのユーザをすべてこのグループに追加してください。

2　*$SYBASE/$SYBASE_ASE/bin* ディレクトリに変更します。

```
cd $SYBASE/$SYBASE_ASE/bin
```

3　*sybha* の所有権を “root” に変更します。

```
chown root sybha
```

4　*sybha* プログラムのグループを *sybhagrp* に変更します。

```
chgrp sybhagrp sybha
```

5　*sybha* のファイル・パーミッションを 4550 に修正します。

```
chmod 4550 sybha
```

6　*$SYBASE/$SYBASE_ASE/install* ディレクトリに変更します。

```
cd $SYBASE/$SYBASE_ASE/install
```

7　sybase ユーザを *sybhauser* ファイルに追加します。追加するユーザのログインは、Adaptive Server のログインではなく、UNIX の形式のログイン ID である必要があります。次に例を示します。

```
sybase
coffeecup
spooner
venting
howe
```

8 *sybhauser* の所有権を “root” に変更します。

```
chown root sybhauser
```

9 *sybhauser* のファイル・パーミッションを修正します。

```
chmod 600 sybhauser
```

新しいデフォルト・デバイスの作成

新しくインストールされた Adaptive Server のデフォルト・デバイスは master です。つまり、作成するすべてのデータベース (フェールオーバによって使用されるプロキシ・データベースも含む) は、自動的にマスタ・デバイス上に作成されます。ユーザ・データベースをマスタ・デバイスに追加すると、システム障害時に、マスタ・デバイスをリストアすることが困難になります。そのため、マスタ・デバイスには極力余分なユーザ・データベースを置かないようにします。それには、disk init コマンドを使用し、新しいデバイスを作成します。sp_diskdefault を使用して新しいデバイスをデフォルトに指定してから、Adaptive Server をフェールオーバ用のコンパニオンとして設定します。

たとえば、money_default1 という名前の新しいデフォルト・デバイスを MONEY1 Adaptive Server に追加するには、次のように入力します。

```
sp_diskdefault money1_default1, defaulton
```

次のコマンドを発行してデフォルトとしての設定を無効にしないかぎり、マスタ・デバイスも引き続きデフォルト・デバイスとして設定されています。

```
sp_diskdefault master, defaultoff
```

disk init については、『リファレンス・マニュアル：コマンド』を参照してください。sp_diskdefault 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。

sysservers へのローカル・サーバの追加

sp_addserver を使用して、ローカル・サーバを sysservers に追加します。サーバには *interfaces* ファイルで指定したネットワーク名を付けます。たとえば、MONEY1 というコンパニオンが、*interfaces* ファイルで指定されている MONEY1 というネットワーク名を使う場合は、次のように入力します。

```
sp_addserver MONEY1, local, MONEY1
```

この変更を有効にするには、Adaptive Server を再起動してください。

sysservers へのセカンダリ・コンパニオンの追加

セカンダリ・コンパニオンを sysservers にリモート・サーバとして追加します。

```
sp_addserver server_name
```

デフォルトでは、Adaptive Server は srvid の値が 1000 のサーバを追加します。この変更は Adaptive Server を再起動しなくても有効になります。

ha_role の割り当て

sp_companion を実行するには、両方の Adaptive Server で ha_role を持っている必要があります。ha_role を割り当てるには、isql から次のコマンドを発行します。

```
sp_role "grant", ha_role, sa
```

sa_role を使用すると、このセッションでの ha_role のオンとオフを切り替えることができます。

変更を有効にするには、ログアウトしてからログインし直してください。

高可用性ストアド・プロシージャのインストール

注意　両方のサーバを *interfaces* ファイルに追加してから、高可用性ストアド・プロシージャをインストールしてください。この作業を行わずに *installhasvss* を実行した場合は、すべてのシステム・ストアド・プロシージャを再インストールする必要があります。

---

installhasvss スクリプトにより、次の作業が行われます。

- フェールオーバに必要なストアド・プロシージャ (sp_companion など) のインストール
- *SYB_HACMP* サーバの sysservers へのインストール

installhasvss を実行するには、システム管理者権限が必要です。

installhasvss は、*$SYBASE/$SYBASE_ASE/scripts* にあります。*installhasvss* を実行するには、次のように入力します。

```
$SYBASE/$SYBASE_OCS/bin/isql -Usa -Ppassword -Sservername
<../scripts/installhasvss
```

installhasvss は、ストアド・プロシージャや SYB_HACMP サーバを作成するときにメッセージを表示します。

設定パラメータの確認

次の設定パラメータを有効にしてから、Adaptive Server をフェールオーバ用に設定します。

- enable CIS － コンポーネント統合サービス (CIS) を有効にします。この設定パラメータは、デフォルトで有効です。
- enable xact coordination － 分散トランザクション管理 (DTM) を有効にします。この設定パラメータは、デフォルトで有効です。
- enable HA － 高可用性システムで、Adaptive Server をコンパニオンとして機能させます。enable HA は、デフォルトでは無効です。このパラメータを有効にするには、Adaptive Server を再起動してください。また、このパラメータを有効にすると、高可用性システムで Adaptive Server を起動したというメッセージがエラー・ログに書き込まれます。ASE_HA ライセンス・オプションは別途購入します。ASE_HA ライセンスの有効化については、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。

『システム管理ガイド 第 1 巻』の「第 5 章 設定パラメータ」を参照してください。

マスタ・ログへのスレッショルドの追加

フェールオーバ、フェールバック、プロキシ・データベースの作成などでは、ログが大量に生成されます。適度なログ領域がないと、これらの処理が失敗する可能性があります。スレッショルドをマスタ・ログに追加していない場合は、追加してください。

1 ダンプ・トランザクションが発生する前に、master データベースのログに対して sp_thresholdaction を定義して実行し、残りのページ数にスレッショルドを設定します。Sybase では sp_thresholdaction を提供していません。このシステム・プロシージャの作成の詳細については、『システム管理ガイド 第 2 巻』の「第 16 章 スレッショルドによる空き領域の管理」と『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。

2 それぞれのスレッショルドをマスタ・ログ・セグメントに置いて、セグメントが満杯にならないようにします。

```
sp_addthreshold "master", "logsegment", 250, sp_thresholdaction
```

3 プライマリ・コンパニオンを再起動して、この静的パラメータを有効にします。

Veritas サブシステムを Sybase フェールオーバ用に設定する

この項は、高可用性システムがすでにインストールされていることを前提としています。Veritas Cluster Server 高可用性システムのインストールと使用については、『VCS Installation Guide』と『VCS User's Guide』を参照してください。

HAase エージェントのインストール

HAase エージェントをクラスタの各ノードにインストールするには、次の操作を行います (次のコマンドを実行するには "root" パーミッションが必要です)。

1　*$SYBASE/$SYBASE_ASE/install/veritas/HAase* ディレクトリに変更します。

```
cd $SYBASE/$SYBASE_ASE/install/veritas/HAase
```

2　次のインストール・スクリプトを実行します。

```
perl installHAase.pl
```

インストール・スクリプトにより、次の作業が行われます。

- *HAase* リソース・タイプ・ファイル *HaaseTypes.cf* をローカル・システムの */etc/VRTSvcs/conf/config/* にコピーする。
- ディレクトリ */opt/VRTSvcs/bin/HAase* がまだ作成されていない場合は新しく作成する。
- ローカル・システムの */opt/VRTSvcs/bin/HAase/* に次のエージェント・バイナリとスクリプトをコピーする。
  - *HAaseAgent*
  - *online*
  - *offline*
  - *clean*
  - *sybhautil.pm*
  - *attr_changed*

Adaptive Server ログイン・ファイルの作成

システム管理者とフォールト・モニタ用に追加したユーザの Adaptive Server ログイン情報が入っているファイルを作成します。この情報のテンプレートを含むサンプル・ファイルは、*$SYBASE/$SYBASE_ASE/install/veritas/HAase/ase_login_file* にあります。

このファイルは 2 行で構成されています。1 行目はシステム管理者のログインとパスワードで、2 行目はモニタ・ユーザのログインとパスワードです。

login-type<tab>*login string*
login-type<tab>*login string*

login-type と *login string* は tab 文字で区切ります。

注意 別のロケーションにある別のファイルを使用する場合は、*HAase* リソースの設定時に、リソース拡張プロパティ *Dataserver_login_file* にフル・パスを指定します。

---

login-type のデフォルト値は normal です。*login string* の値は、*login-name/password* という形式で指定します。次に例を示します。

```
normal      sa/sa-password
normal      probe-user/probe-password
```

セキュリティ上の理由から、read および write のアクセス・パーミッションは “root” に限定し、*ase_login_file* を保護する必要があります。

```
chmod 400 ase_login_file
chown root ase_login_file
chgrp  sys  ase_login_file
```

注意 パスワードを使用することを強くおすすめします。空のパスワードを使用すると、エージェント・スクリプトが警告メッセージを生成します。

HAase リソース・タイプのインポート

HAase リソース・タイプをインポートするには次の 2 つの方法があります。

- クラスタ GUI ツールを使用して新しいリソース・タイプ *HAase* をインポートする方法。詳細については、『VCS User’s Guide』を参照してください。
- コマンド・ラインでクラスタ・コマンド hatype と haattr を使用して、新しいリソース・タイプを手動でインポートする方法。詳細については、『VCS User Guide』を参照してください。

HAase エージェントの起動

HAase エージェントは次のいずれかの方法で起動できます。

- Veritas クラスタを再起動する方法。
- クラスタ・コマンドを使用して、手動で *HAase* エージェントを起動する方法。

2 番目の方法では中断が起きません。*HAase* エージェントを手動で起動するには、次の手順に従います。

1　*haagent* ユーティリティを使用して、*HAase* エージェントのステータスを調べます。

```
#haagent -display HAase
#Agent   Attribute  Value
HAase     AgentFile
HAase     Faults     0
HAase     Running    No
HAase      Started      No
```

2　*haagent* ユーティリティを使用して *myhost* で *HAase* エージェントを起動します。

```
# haagent -start HAase -sys myhost
VCS:10001:Please look for messages in the log file
```

3　*HAase* エージェントのステータスを *haagent* ユーティリティで調べます。

```
# haagent -display HAase
#Agent    Attribute   Value
HAase     AgentFile
HAase     Faults      0
HAase     Running     Yes
HAase      Started       Yes
```

HAase リソースの追加

各サービス・グループは *HAase* リソースを含んでいる必要があります。表 11-1 は、*HAase* リソースの属性を示します。

表 11-1: HAase リソース

プロパティ	データ型、次元、デフォルト	説明
Sybase_home	string、スカラ、null	Adaptive Server インストール環境のホーム・ディレクトリ。Adaptive Server インストール環境の SYBASE 環境変数と同じ値です。
Dataserver_name	string、スカラ、null	設定時に提供される Adaptive Server の名前。
Backup_server_name	string、スカラ、null	設定時に提供される Backup Server の名前。
Textserver_name	string、スカラ、null	設定時に提供される全文検索サーバの名前。
Secondary_companion_name	string、スカラ、null	sp_companion configure コマンドの実行時に設定されるセカンダリ・コンパニオン・サーバの名前。
Dataserver_login_file	string、スカラ、null	現在のデータ・サーバのログイン情報が入っているファイルの絶対パス。このファイルは 2 行で構成されています。1 行目はシステム管理者のログインとパスワードで、2 行目は高可用性エージェント・モニタが完全プローブに使用するユーザ・ログインとパスワードです。

プロパティ	データ型、次元、デフォルト	説明
RUN_server_file	string、スカラ、null	代替 *RUN_server* ファイルの絶対パス。デフォルトの *$SYBASE/$SYBASE_ASE/install/RUN_SERVER* を上書きします。
Thorough_probe_cycle	int、スカラ、3	完全プローブが実行される前の「単純な」プローブの数。
Thorough_probe_script	string、スカラ、null	フォールト・モニタ・プログラムが完全プローブの実行に使用する SQL スクリプトが入っている代替ファイルの絶対パス。null に設定されている場合、エージェントはデフォルトの SQL コマンドを使用します。 セキュリティ上の理由から、このファイルの書き込みアクセスは *$SYBASE* ディレクトリの所有者のみに限定してください。 **注意** この値は *HAase* リソースでは無視されます。
Debug	Boolean、スカラ、0	1 (真) に設定されている場合、モニタはデバッグ・メッセージを *$VCS_LOG/log/HAase_A.log* に記録し、他のスクリプトはデバッグ・メッセージを *$VCS_LOG/log/engine_A.log* に記録します。メッセージ番号の範囲は 2,000,001 以上です。
Log_max_size	int、スカラ、5000000	*$VCS_LOG/log/HAase_A.log* ファイルの最大サイズ。
Failback_strategy	string、スカラ、null	今後のために予約済み。
HA_config	Boolean、スカラ、0	今後のために予約済み。
Cmpstate	Boolean、スカラ、0	今後のために予約済み。

注意 *$VCS_LOG* のデフォルト値は */var/VRTSvcs* です。

各サービス・グループの *HAase* リソースのインスタンスの設定

HAase リソースのインスタンスを次のいずれかの方法で設定します。

- クラスタ GUI ツールを使用して *HAase* のインスタンスを設定する方法 (詳細については、『VCS User's Guide』を参照してください)。
- 次に示すように、クラスタ・コマンドを使用して新しいリソースを手動で追加し、その属性を設定する方法。サービス・グループ *SybASE* の設定については、図 11-1 (171 ページ) を参照してください。
    - *HAase* リソースを追加します。

```
#hares -add syb_ase150 HAase SybASE
VCS:10245:Resource added
NameRule and Enabled attributes must be set before agent monitors
# hares -modify syb_ase150 Dataserver_name MONEY1
# hares -modify syb_ase150 RUN_server_file  /release/rel150/ASE-
```

```
15_0/install/RUN_MONEY1
# hares -modify syb_ase150 Log_max_size 5000000
# hares -modify syb_ase150 Dataserver_login_file /release/rel150/ASE-
15_0/install/MONEY1_login
# hares -modify syb_ase150 Sybase_home  /release/rel150
# hares -modify syb_ase150 Thorough_probe_cycle 3
```

- リソース *syb_ase150* のステータスをモニタするようにエージェントを設定します。

  ```
  # hares -modify syb_ase150 Enabled 1
  ```

新しいリソースをサービス・グループに追加したら、*HAase* リソースと、その他のストレージ・リソースやネットワーク・アクセス・リソースのアクセスの間にリソース依存を確立します。

次のクラスタ・コマンドを使用して、*syb_ase150* と、タイプが *Mount*、*Volume*、*IP* のリソースとの間にリソース依存を確立します (詳細については、図 11-1 を参照してください)。

```
# hares -link syb_ase150 ha1_mnt_ase
# hares -link syb_ase150 ha1_mnt_fsdb
# hares -link syb_ase150 vrtsdg1_vol_master
# hares -link syb_ase150 vrtsdg1_vol_procs
# hares -link syb_ase150 ha1_ip
```

フェールオーバ用コンパニオン・サーバの設定

この項では、高可用性システムで Adaptive Server をプライマリ・コンパニオンとセカンダリ・コンパニオンとして設定する方法について説明します。

高可用性モニタ用のユーザとログインの追加

HAase リソースと関連付けられた各データ・サーバに対して、モニタ用に特別なユーザとログインを作成します。isql を使ってデータ・サーバに接続し、次のコマンドを発行します。

sp_addlogin *probe_ase*, *password*
sp_adduser *probe_ase*

注意　システム管理者は、Adaptive Server の設定時に、プローブに使用するユーザとログインにより、他の目的に使用できる接続の合計数が実際には 1 つ減ることを考慮してください。

モニタのログイン情報の格納の詳細については、「Adaptive Server ログイン・ファイルの作成」(177 ページ) を参照してください。

do_advisory オプションを指定して *sp_companion* を実行する

十分なリソースを持つセカンダリ・コンパニオンを設定して、フェールオーバ中でもサーバ 2 台分の作業を実行できるようにします。セカンダリ・コンパニオンは、正常なクラスタ・オペレーションを妨げる属性を持っている場合があります。たとえば、プライマリ・コンパニオンとセカンダリ・コンパニオンのユーザ・ログイン数がともに 250 に設定されていると、フェールオーバ中、セカンダリ・コンパニオンには、発生する可能性のあるユーザ・ログインの半分を処理するリソースしかないことになります。したがって、MONEY1 と PERSONNEL1 の両方でユーザ・ログイン数を 500 に設定します。

sp_companion do_advisory は、クラスタ・オペレーション (Adaptive Server をセカンダリ・コンパニオンとして設定するなど) が正常に行われるようにするために、プライマリ・コンパニオンとセカンダリ・コンパニオンの両方の設定オプションを確認します。また、sp_companion do_advisory は、変更する必要のある設定オプションを通知します。

sp_companion do_advisory オプションの詳細については、「第 6 章 do_advisory の実行」を参照してください。

高可用性エージェントの確認

Solaris オペレーティング・システムを搭載したマシンはさまざまなクラスタ・ソフトウェアをサポートできるため、sp_companion には、実行中の高可用性エージェントを問い合わせるための show_cluster オプションと、高可用性エージェントを設定するための set_cluster オプションが含まれています。

Veritas Cluster Server サブシステムを実行している場合は、sp_companion を使用してクラスタを指定してください。Adaptive Server は、別のクラスタが指定されないかぎり、使用しているオペレーティング・システム用のクラスタ・ソフトウェアが実行されていると想定します。

構文は次のとおりです。

```
sp_companion companion_server_name, [show_cluster]
sp_companion companion_server_name, [set_cluster, "SC"|"VCS"]
```

Veritas クラスタ用の *HAase* エージェントを使用するように Adaptive Server を変更するには、次のように指定します。

```
sp_companion MONEY1, set_cluster, "VCS"

The current cluster is set to VCS
```

注意 Adaptive Server が VCS システムのノーマル・コンパニオン・モード用に設定されている場合には、別の高可用性エージェント・タイプに変更しないでください。

非対称型コンパニオン設定の作成

プライマリ・コンパニオンを非対称型で設定するには、次のコマンドをセカンダリ・コンパニオンから発行します。

sp_companion "*primary_server_name*", configure, NULL, *login_name*, *password*

各パラメータの意味は次のとおりです。

- *primary_server_name* − *interfaces* ファイルのエントリと sysservers に定義されているプライマリ Adaptive Server の名前。
- *login_name* − このクラスタ・オペレーションを行っているユーザの名前。ユーザは ha_role を持っている必要があります。
- *password* − このクラスタ・オペレーションを行っているユーザのパスワード。

注意　上記コマンドは、セカンダリ・コンパニオンからのみ実行します。

この例では、MONEY1 という名前の Adaptive Server をプライマリ・コンパニオンとして設定します。セカンダリ・サーバ PERSONNEL1 から次のコマンドを発行します。

```
sp_companion "MONEY1", configure, NULL, sa, Odd2Think
Server 'PERSONNEL1' is alive and cluster configured.
Step: Access verified from Server:'PERSONNEL1' to Server:'MONEY1'
Server 'MONEY1' is alive and cluster configured.
Step: Access verified from Server:'MONEY1' to Server:'PERSONNEL1'
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
.....
Step: Companion servers configuration check succeeded
Step: Server handshake succeeded
Step: Master device accessible from companion
Step: Added the servers 'PERSONNEL1' and 'MONEY1' for cluster config
Step: Server configuration initialization succeeded
Step: Synchronizing server logins from companion server
Step: Synchronizing remoteserver from companion server
Step: Synchronizing roles from companion server
Step: Synchronizing server-wide privs from companion server
Step: User information syncup succeeded
Step: Server configured in normal companion mode
```

sp_companion 設定を行っている間にユーザ・データベースが作成されると、次のようなメッセージが表示されます。

```
Step: Created proxy database 'pubs2'
 Step: Proxy status for database has been set. Please Checkpoint the database 'pubs2'
 Step: Server configured in normal companion mode Starting companion watch thread
```

対称型設定の設定

非対称型フェールオーバを使用できるようにコンパニオンを設定した後、対称型設定に設定できます。対称型設定では、両方のサーバがプライマリ・コンパニオンとしても、セカンダリ・コンパニオンとしても機能します。対称型設定については、図 3-2 (21 ページ) を参照してください。

プライマリ・コンパニオンから sp_companion を発行して、対称型設定に設定します。非対称型設定の場合と同じ構文を使用します。ただし、with_proxydb オプションは使用できません。sp_companion の構文については、非対称型コンパニオン設定の作成を参照してください。

次は、MONEY1 という Adaptive Server を、セカンダリ・コンパニオンとして PERSONNEL1 という Adaptive Server に追加する例です (このコマンドはプライマリ・コンパニオン MONEY1 から発行します)。

```
sp_companion 'PERSONNEL1', configure, null, sa, Think2Odd
Server 'MONEY1' is alive and cluster configured.
Step: Access verified from Server:'MONEY1' to Server:'PERSONNEL1'
Server 'PERSONNEL1' is alive and cluster configured.
Step: Access verified from Server:'PERSONNEL1' to Server:'MONEY1'
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
.....
Step: Companion servers configuration check succeeded
Step: Server handshake succeeded
Step: Master device accessible from companion
Step: Added the servers 'MONEY1' and 'PERSONNEL1' for cluster config
Step: Server configuration initialization succeeded
Step: Synchronizing server logins from companion server
Step: Synchronizing remoteserver from companion server
Step: Synchronizing roles from companion server
Step: Synchronizing server-wide privs from companion server
Step: User information syncup succeeded
Step: Server configured in normal companion mode
```

注意 前述の sp_companion configure コマンドの *login_name* と *password* を null にすることはできません。sp_companion configure が正常に実行されたら、オペレーティング・システムによって新しいファイル */etc/VRTSvcs/conf/config/ha_companion.remote_server_name* が作成されます。このファイルの読み込みおよび書き込みアクセスを、サーバを実行するユーザだけに限定してください。そうしない場合、セキュリティに問題が発生する可能性があります。

Sybase フェールオーバの管理

この項では、Sybase のフェールオーバの使い方を説明します。

フェールオーバ中

プライマリ・ノードがセカンダリ・ノードにフェールオーバすると、プライマリ・ノードでオンラインのサービス・グループがセカンダリ・ノードに切り替えられます。この時点で、Adaptive Server のバイナリを除くすべてのリソースが、セカンダリ・ノード上でオンラインになります。セカンダリ・ノードの Adaptive Server がこれらのリソースを引き継ぎます。

注意 あるサービス・グループがプライマリ・ホストからセカンダリ・ホストにフェールオーバすると、セカンダリ・ホストの Adaptive Server がそのグループのすべてのプライマリ・リソースを引き継ぎますが、フェールオーバしたグループの Adaptive Server は起動されません。

プライマリ・コンパニオンへのフェールバック

フェールバックでは、当初プライマリ・ノードに所属していたサービス・グループを、セカンダリ・ノードからプライマリ・ノードに戻して、そのグループをオンラインにします。

フェールバックを開始するには、次の手順に従います。

- プライマリ・ノードがサービス・グループを引き継ぐ準備ができたら、セカンダリ・コンパニオンから次のコマンドを発行します。

  sp_companion *primary_companion_name*, prepare_failback

  primary_companion_name は、プライマリ・コンパニオンの名前です。このコマンドは、プライマリ・ノードのサービス・グループを、セカンダリ・ノードからプライマリ・ノードに戻します。

- コマンド・ラインで次のコマンドを発行して、プライマリ・ノードのサービス・グループが正常にプライマリ・ノードに切り替えられたことを確認します。

  hastatus -group *service_group_name*

  このコマンドを使用すると、プライマリ・ノードの**サービス・グループ**のステータスが表示されます。

- ノーマル・コンパニオン・モードを再開するには、次のコマンドをプライマリ・コンパニオンから発行します。

  sp_companion *secondary_companion_name*, resume

secondary_companion_name は、セカンダリ・コンパニオン・サーバの名前です。

注意 sp_companion resume を発行しないと、フェールオーバ・プロパティが設定されたクライアントを接続できません。sp_companion prepare_failback を発行してからクライアントを再接続しようとすると、クライアントは sp_companion resume を発行するまで応答を停止します。

ノーマル・コンパニオン・モードのサスペンド

サスペンド・モードでは、一時的にプライマリ・コンパニオンがセカンダリ・コンパニオンにフェールオーバできなくなります。ノーマル・コンパニオン・モードからサスペンド・モードに切り替えるには、次の手順に従います。

1 “root” 権限で hares を使用して、プライマリ・ノードの *Sybase* リソースの属性 *Critical* を 0 に変更します。構文は次のとおりです。

```
hares -modify name_of_Sybase_resource Critical 0
```

2 ノーマル・コンパニオン・モードをサスペンドします。セカンダリ・コンパニオンから、次のコマンドを発行します。

```
sp_companion companion_name, suspend
```

ノーマル・コンパニオン・モードの再開

サスペンド・モードからノーマル・コンパニオン・モードに戻るには、次の手順に従います。

1 両方のコンパニオンが実行中であることを確認します。“root” 権限で、次のコマンドを発行します。

```
hastatus
```

2 プライマリ・ノードの *Sybase* リソースの *Critical* 属性を 1 に変更します。“root” 権限で、次のコマンドを発行します。

```
hares -modify name_of_Sybase_resource Critical 1
```

3 ノーマル・コンパニオン・モードを再開します。セカンダリ・コンパニオンから、次のコマンドを発行します。

```
sp_companion primary_companion_name, resume
```

注意 sp_companion resume を発行しないと、フェールオーバ・プロパティが設定されたクライアントを接続できません。sp_companion prepare_failback を発行してからクライアントを再接続しようとすると、クライアントは sp_companion resume を発行するまでハングします。

コンパニオン・モードの削除

コンパニオン・モードを削除したら、二度と元に戻すことはできません。したがって、削除する場合は、高可用性システムをフェールオーバさせる前にコンパニオン・サーバを再設定し、Sybase のフェールオーバに装備されている機能をすべて保持する必要があります。しかし、その後も、コンパニオン・サーバは引き続き高可用性エージェントによってモニタされます。最初に Adaptive Server をモニタするエージェントを無効にしてから、コンパニオン・モードを削除してください。次のコマンドを発行します。

```
hares -modify Sybase_resource_name Enabled 0
```

コンパニオン・モードを削除するには、次のコマンドを発行します。

```
sp_companion companion_name, "drop"
```

リソース・タイプ *Sybase* のエージェントからのアップグレード

以前のリリースの VCS の高可用性エージェントのうち、リソース・タイプ *Sybase* のエージェントを使用している場合に、リソース・タイプ *HAase* の新しいエージェントを使用するときは、古いエージェントを新しいエージェントに切り替えます。

1. リソース・タイプ　*HAase*　の新しいエージェントをインストールします。「HAase エージェントのインストール」(177 ページ) を参照してください。
2. 新しいリソース・タイプ *HAase* をインポートします。「HAase リソース・タイプのインポート」(178 ページ) を参照してください。
3. リソース・タイプ *HAase* の新しいエージェントを起動します。「HAase エージェントの起動」(178 ページ) を参照してください。
4. Sybase のリソース・モニタリングを無効にします。

   ```
   haconf -makerw
   hares -modify Sybase_resource_name Enabled 0
   haconf -dump -makero
   ```

5. Sybase の既存のリソース・インスタンスをサービス・グループから削除します。

   ```
   haconf -makerw
   hares -delete sybase_resource_name
   haconf -dump -makero
   ```

6. リソース・タイプ *HAase* の新しいリソース・インスタンスを設定します。「HAase リソースの追加」(179 ページ) を参照してください。

7 新しい *HAase* リソースを有効にします。

```
haconf -makerw
hares -modify Sybase_resource_name Enabled 1
haconf -dump -makero
```

Adaptive Server のアップグレード

高可用性設定内の Adaptive Server をアップグレードするには、一時的にプライマリ・コンパニオンとセカンダリ・コンパニオンの間のコンパニオン関係を削除し、VCS の Adaptive Server サービス・グループのモニタリングを無効にする必要があります。これにより、アップグレード・プロセスの間、VCS によって予期しないフェールオーバがトリガされることなく、各 Adaptive Server を独立して停止または再起動できます。

注意 アップグレード・プロセスの間は、データベース、オブジェクト、ユーザ、またはログインを追加、削除、修正できません。コンパニオン関係を削除した後、その関係を再確立する前にこのような変更を行うと、アップグレードが失敗する場合があります。または、サーバ間の不整合が原因でクラスタが不安定になることがあります。

❖ モニタリング・サービスの停止とコンパニオン関係の削除

1 クラスタ内のすべてのノードで、モニタリング・サービスを停止します。root 権限で、次のコマンドを発行します。

```
hares -modify primary_resource Enabled 0
hares -modify primary_resource Critical 1
```

システムを対称型フェールオーバ用に設定した場合は、セカンダリ・リソースのモニタリングを無効にします。

```
hares -modify secondary_resource Enabled 0
hares -modify secondary_resource Critical 0
haconf -dump -makero
```

2 root 権限で、次のコマンドを発行します。

```
hares -offline primary_resource -sys primary_system_name
hares -offline secondary_resource -sys secondary_system_name
```

3 セカンダリ・コンパニオンから、次のコマンドを発行します。

```
sp_companion primary_server_name, "drop"
```

4 (対称型設定の場合) セカンダリのコンパニオン関係を削除します。プライマリ・コンパニオンにログインし、次のコマンドを発行します。

```
sp_companion secondary_server_name,"drop"
```

5　両方のノードがシングルサーバ・モードであることを確認します。

```
sp_companion
```

コンパニオンがシングルサーバ・モードである場合は、次のような結果になります。

```
Server 'server_name' is not cluster configured.
Server 'server_name' is currently in 'Single server' mode.
```

これで、サーバがそのインストール・ノード上で実行されるようになります。サーバは、VCS がノード間でリソースをフェールオーバしようとすることなく、独立して停止および起動できます。

❖ Adaptive Server のアップグレード

1　各ノードで、高可用性を無効にします。

```
sp_configure 'enable HA', 0
```

Adaptive Server を再起動して、この変更を有効にします。

2　『インストール・ガイド』の手順に従って各サーバをアップグレードします。

3　すべてのノードで、高可用性を再度有効にします。

```
sp_configure 'enable HA', 1
```

Adaptive Server を再起動して、この変更を有効にします。

4　sybha バイナリと *sybhausers* ファイルのパーミッションが正しく設定されていることを確認します。
root 権限で、次のコマンドを *$SYBASE/$SYBASE_ASE/bin* から発行します。

```
chown root sybha
chgrp sybhagrp sybha
chmod 4550 sybha
```

root 権限で、以下の手順を *$SYBASE/$SYBASE_ASE/install* から実行します。

1　sybase ユーザが *sybhauser* ファイルに含まれていることを確認します。

2　次のコマンドを発行します。

```
chown root sybhauser
chmod 600 sybhauser
```

5　次の内容を確認します。

- *$SYBASE* のインストール・ロケーション、または新しいインストール環境内の高可用性に関係する関連ファイルにおいて、サービス・グループおよびリソース・プロパティ (たとえば、*Sybase_Home*、runserver ファイル、*Dataserver_login_file* など) に変更が正しく反映されている。

- 「高可用性で動作するように Adaptive Server を準備する」(171 ページ) および「Veritas サブシステムを Sybase フェールオーバ用に設定する」(177 ページ) で説明されている、コンパニオン関係の確立に必要なすべての手順を実施しており、システムがアップグレードの完了後もこれらの変更を保持している。

❖ **コンパニオン関係の再確立とリソース・モニタリングの再開**

1 各ノードで、Adaptive Server を手動で再起動します。

2 root 権限で、プライマリ・ノードからモニタリング・サービスをリストアします。

```
haconf -makerw
hares -modify primary_resource Enabled 1
hares -modify primary_resource Critical 1
```

システムを対称型フェールオーバ用に設定した場合は、セカンダリ・リソースのモニタリングを有効にします。

```
hares -modify secondary_resource Enabled 1
hares -modify secondary_resource Critical 1
haconf -dump -makero
```

3 「フェールオーバ用コンパニオン・サーバの設定」(181 ページ) で説明されている、コンパニオン関係を確立するための前提条件となる手順が完了していることを確認します。

4 サーバ間のコンパニオン関係を再確立します (「非対称型コンパニオン設定の作成」(183 ページ) または「対称型設定の設定」(184 ページ) を参照してください)。

```
dbcc traceon (2209)
sp_companion primary_server_name,configure
dbcc traceoff(2209)
```

注意 対称型設定の場合は、両方のコンパニオンでこのコマンドを発行します。

セカンダリ・サーバにユーザ・データベースが含まれる場合は、次のような警告メッセージが表示されることがあります。このようなメッセージは無視できます。

```
Msg 18739, Level 16, State 1:
Server 'server_name', Procedure 'sp_hacmpcfgvrfy', Line 102:
Database 'database_name': a user database exists. Drop this
database and retry the configuration again.
```

5 適切なノードで Adaptive Server リソースを再起動します。root 権限で、プライマリ・ノード上で次のコマンドを入力します。

```
hares -online primary_resource -sys primary_system_name
```

root 権限で、セカンダリ・ノード上で次のコマンドを入力します。

```
hares -online secondary_resource -sys secondary_system_name
```

6　sp_companion を実行して、システムが対称型フェールオーバまたは非対称フェールオーバ用に適切に設定されていることを確認します。

Veritas クラスタでのフェールオーバのトラブルシューティング

この項では、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。

- Adaptive Server のデバッグをオンにします。トレース・フラグ 2205 を使用して、高可用性関連のデバッグ情報を取得します。次の isql セッションは、デバッグをオンにして、メッセージをコンソールにリダイレクトします。

  ```
  dbcc traceon(2205)
  dbcc traceon(3604)
  ```

- エラーが報告されたら、最初にエラー・ログをチェックします。ID が 2,000,000 より大きいすべてのエラー・メッセージが *HAase* エージェントのエラー・メッセージです。
- VCS エラー・ログは */var/VRTSvcs/log/log_name.log* にあります。このうち、*engine_A.log* は重要な情報源です。

  システム・エラー・ログは、*/var/log/syslog* にあります。
- システムの情報を調べるには、次の監視ツールの使用をおすすめします。
  - *hagui* － GUI ツール。
  - *hastatus* － コマンド・ライン・ツール。
  - VCS システムでのイベントについて警告するトリガ・スクリプト (*injeopardy*、*preonline*、*postonline*、*postoffline*、*resnotoff*、*resfault*、*sysoffline*、*violation*)。
- あるサービス・グループがプライマリ・ホストからセカンダリ・ホストにフェールオーバすると、セカンダリ・ホストの Adaptive Server がそのグループのすべてのリソースを引き継ぎますが、フェールオーバしたグループの Adaptive Server は起動されず、セカンダリ・ホストの HAase リソースに「障害が発生している」ことが VCS によって示される可能性があります。セカンダリ・ホストで次のコマンドを使用して、フェールオーバ後にステータスをクリアしてください。

  ```
  hares -clear sybase_res_name -sys secondary_host_name
  ```

失敗した *prepare_failback* からのリカバリ

フェールバック中に、prepare_failback がセカンダリ・コンパニオンで正常に実行されたにもかかわらず、プライマリ・コンパニオンが起動しない場合は、次の手順に従います。

1 プライマリ・コンパニオンのエラー・ログとクラスタ・エラー・ログをチェックし、サーバが起動しなかった原因を特定し、問題を解決します。

2 *HAase* リソースの FAULTED ステータスをクリアするには、次のコマンドを発行します。

```
hares -clear primaryHAase_resname  -sys  primary_hostname
```

3 “root” 権限で以下を発行し、プライマリの論理ホストをセカンダリ・ノードに戻します。

```
hagrp -switch primary_service_group -to secondary_host_name
```

4 セカンダリ・コンパニオンにログインして、次のコマンドを発行します。

```
dbcc ha_admin ("", "rollback_failback")
```

両方のコンパニオン・サーバが、フェールオーバ・モードに戻ります。dbcc ha_admin の詳細については、「高可用性システムの dbcc オプション」(205 ページ) を参照してください。

5 セカンダリ・コンパニオン上で sp_companion...prepare_failback を再発行します。

ログのロケーション

これらのログの情報を使用して、高可用性システムをデバッグします。

- Adaptive Server エラー・ログ (RUNSERVER ファイルで定義)
- Veritas クラスタ・ログ (*/var/VRTSvcs/log/engine_A.log*)
- オペレーティング・システム・メッセージ (*/var/log/syslog*)
- *HAase* エージェント・ログ (*/var/VRTSvcs/log/HAase_A.log*)

付 録 A

フェールオーバ設定での Open Client の機能

この付録では、Open Client で Sybase フェールオーバを機能させるために必要な変更について説明します。

CTLIB アプリケーションの変更

注意 クラスタにインストールされているアプリケーションは、プライマリ・コンパニオンとセカンダリ・コンパニオンの両方で実行可能でなければなりません。つまり、並列設定が必要なアプリケーションをインストールする場合は、セカンダリ・コンパニオンも並列処理ができるように設定する必要があります。これによって、フェールオーバ中にアプリケーションを実行できるようになります。

CTLIB API 呼び出しを使用して作成されたすべてのアプリケーションは、Sybase のフェールオーバで実行できるようにするには変更が必要です。次の操作を行います。

1 Client-Library API 呼び出しの ct_config と ct_con_props を使用して、CS_HAFAILOVER プロパティを設定します。このプロパティは、コンテキスト・レベルと接続レベルのどちらでも設定できます。次の構文を使用します。

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

2 クライアントがセカンダリ・コンパニオンにフェールオーバできるように、*interfaces* ファイルを修正します。

interfaces ファイルには *hafailover* というラベルの行が入っており、プライマリ・コンパニオンのクラッシュ、または shutdown with nowait の発行によりフェールオーバが発生した場合、クライアントはこれを使用してセカンダリ・コンパニオンに再接続できます。

この行の追加方法については、使用しているプラットフォーム用の章で「両方の Adaptive Servers のエントリを *interfaces* ファイルに追加する」の項を参照してください。

3 次の要因に従って、アプリケーションのフェールオーバ・メッセージを記述します。

- コンパニオンがシャットダウン処理を始めると、クライアントはフェールオーバが発生するという情報メッセージを受け取ります。これは、クライアント・エラー・ハンドラの情報メッセージとして扱います。
- フェールオーバのプロパティを設定 (手順 1 以降) し、*interfaces* ファイルに *hafailover* サーバに有効なエントリが入っている場合、クライアント接続はフェールオーバ接続となり、クライアントはセカンダリ・コンパニオンに正しく再接続します。

  ただし、フェールオーバのプロパティが設定されていても *interfaces* ファイルに hafailover サーバに対するエントリが入っていない (またはその逆の) 場合は、フェールオーバ接続にはなりません。この場合は、フェールオーバ・プロパティがオフになった、通常の接続になります。フェールオーバ接続になっているかどうかを知るには、フェールオーバのプロパティを確認してください。

4 リターン・コードを追加します。

フェールオーバが正常に実行されると、クライアントは CS_RET_HAFAILOVER という戻り値を発行します。この戻り値は、次の 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)
```

同期接続では、API 呼び出しから CS_RET_HAFAILOVER が返されます。非同期接続では、API が CS_PENDING を発行し、callback 関数によって CS_RET_HAFAILOVER が返されます。リターン・コードに応じて、次のコマンドを送信するなどの必要な処理を実行できます。

a アプリケーションを再構築し、フェールオーバ・ソフトウェアに含まれるライブラリにリンクさせます。

注意 sp_companion resume を発行しないと、フェールオーバ・プロパティが設定されたクライアントを接続できません。sp_companion prepare_failback を発行してからクライアントを再接続しようとすると、クライアントは sp_companion resume を発行するまで応答を停止します。

付 録 B

2 次ポイント障害のトラブルシューティング

この付録では、高可用性システムにおける、2 次ポイント障害から発生する一般的な問題について説明します。

dbcc ha_admin を使ったトラブルシューティング

プライマリ・コンパニオンがすでにフェールオーバ・モードになっていて、システムの別のポイントで障害が発生すると、高可用性システムで 2 次ポイント障害が発生します。Sybase フェールオーバには、2 次ポイント障害に対処する dbcc ha_admin が含まれています。

dbcc ha_admin の構文と全オプションのリストについては、「高可用性システムの dbcc オプション」(205 ページ) を参照してください。

コンパニオン・サーバで *installhasvss* を実行した後、このスクリプトを再実行するのは、このスクリプトで作成したストアド・プロシージャが破壊された場合か、*installhasvss* の新しいバージョンをインストールする必要がある場合のみにします。dbcc ha_admin (' ', state_machine) は、*installhasvss* がストアド・プロシージャを安全に再インストールまたは更新できるように、コンパニオンを一時的にシングルサーバ・モードに切り換えます。dbcc ha_admin を実行しないで *installhasvss* を実行しようとすると、次のようなエラー・メッセージが表示されます。

```
Server is not in single-server mode.
Please run dbcc ha_admin (' ', 'state_machine', 'halt') and try again
```

注意 dbcc ha_admin はコンパニオンをシングルサーバ・モードに切り換えるので、このコマンドを実行するのは、同期アクティビティがない場合だけにしてください。

installmaster の再インストール

コンパニオン・サーバに *installmaster* をインストールした後でこのスクリプトを再実行するのは、このスクリプトで作成したストアド・プロシージャが破壊された場合、または *installmaster* の新しいバージョンをインストールする必要がある場合のみにします。dbcc ha_admin (' ', state_machine) は、*installmaster* がストアド・プロシージャを安全に再インストールまたは更新できるように、コンパニオンを一時的にシングルサーバ・モードに切り換えます。*installmaster* を実行するときは、必ず dbcc ha_admin を実行してください。

注意 dbcc ha_admin はコンパニオンをシングルサーバ・モードに切り換えるので、このコマンドを実行するのは、同期アクティビティがない場合だけにしてください。

---

installmaster を再インストールするには、次の手順に従います。

- dbcc ha_admin を実行して、ローカル・コンパニオン・サーバをシングルサーバ・モードに切り換えます。

  ```
  dbcc ha_admin (' ', 'state_machine', 'halt')
  ```

  ' ' は空のプレースホルダとして使用します。

- *installmaster* を再実行します。
- dbcc ha_admin を実行して、コンパニオン・サーバを元のモードに戻します。

  ```
  dbcc ha_admin (' ', 'state_machine', 'restart')
  ```

- *installmaster* を再インストールした後、installhasvss を再インストールする必要があります。以下を参照してください。

installhasvss の再実行

1. sysservers の SYB_HACMP エントリの srvnetname を記録してください。Sybase のフェールオーバを使用できるように設定されている場合は、SYB_HACMP はコンパニオン・サーバの svrnetname を指します (たとえば、コンパニオン・サーバ MONEY1 の SYB_HACMP エントリの srvnetname は PERSONNEL1 です)。
2. dbcc ha_admin を実行して、コンパニオンをシングルサーバ・モードに切り換えます。

   ```
   dbcc ha_admin (' ', 'state_machine', 'halt')
   ```

   ' ' は空のプレースホルダとして使用します。

3　*installhasvss* を再実行します。*installhasvss* の再実行が完了すると、コンパニオン・サーバは元のモードに戻ります。

前述の手順 2 を行った後にノードがクラッシュすると、sysservers からリモート・サーバの srvnetname が削除されます。そのような場合は、次を発行して、sysservers にリモート・サーバの名前を追加してください。

```
sp_addserver SYB_HACMP, null, 'remote_server_svrnetname'
```

dbcc ha_admin を実行して、コンパニオン・サーバを元のモードに戻します。

```
dbcc ha_admin (' ', 'state_machine', 'restart')
```

フェールオーバ・コマンドのロールバックのための *dbcc ha_admin* の使用

dbcc ha_admin には、rollback_failover オプションと rollback_failback オプションがあります。dbcc のこの 2 つのオプションは、高可用性システムに精通したシステム管理者が使用する最後の手段です。それ以外の方は使用しないでください。

これらのオプションを使用すると、次の場合に実行された手順をロールバックできます。

- 高可用性システムにおける問題 (たとえば、フェールオーバ中にすべてのディスクが使用できなかったために、すべてのデータベースが損傷を受けている疑いがあるとコンパニオンがマークするなど) や、フェールオーバ中にクラッシュしたセカンダリ・コンパニオンが原因で完了しなかったフェールオーバ。
- 高可用性システムにおける問題、またはフェールバック手順の実行中にプライマリ・コンパニオンが再起動されなかったために完了しなかった sp_companion...prepare_failback の処理。

dbcc ha_admin rollback_failover や rollback_failback を発行する前に実行する必要がある、プラットフォームに固有の手順があります。詳細については、このマニュアル内に記述されている、使用しているプラットフォームの設定に関する章を参照してください。

@@hacmpservername の使用

コンパニオン・サーバ名を確認するには、グローバル変数 *@@hacmpservername* を使用します。

```
select @@hacmpservername
```

たとえば、プライマリ・コンパニオンの MONEY1 からこのコマンドを実行すると、次のようなメッセージが出力されます。

```
select @@hacmpservername
------------------------------
PERSONE1
(1 row affected)
```

エラー・メッセージ

表示される一般的なエラー・メッセージは、次のとおりです。

- エラー・メッセージ 18805

```
Warning: Server '%1!' is configured for ASE HA services. The network name
in its SYB_HACMP entry does not point to the local server. If this is due
to an earlier failed cluster command, refer to the System Administration
Guide.
```

SYB_HACMP のネットワーク名が別のサーバのネットワーク名に設定されていると、このエラーが発生します。sp_addserver を使用して、SYB_HACMP の srvnetname をローカル・サーバのネットワーク名に設定します。通常のコンパニオン・モードでは、SYB_HACMP の svrnetname は常にリモート・コンパニオンのネットワーク名を指します。絶対に変更しないでください。

- エラー・メッセージ 18769

```
The HA cluster is currently in use for other cluster operations. Retry
the command later. If the problem persists, it may be due to an earlier
failed cluster command; check the System Administration Guide (Error
%1!).
```

すべてのクラスタ・オペレーションがクラスタ全体のロックを受信し、クラスタ・オペレーションが完了すると、ロックを解放します。直前のクラスタ・オペレーションがクラスタ全体のロックを解放する前にクラスタ・オペレーションを実行すると、このエラーが発生します。クラスタ全体のロックの解放の詳細については、「高可用性ノードにおけるクラスタのロック」(6 ページ) を参照してください。

- エラー・メッセージ 18836

```
Configuration operation '%1!' can not proceed due to Quorum AdvisoryCheck
failure. Please run 'do_advisory' command to find the incompatible attribute and
fix it
```

sp_companion は、一連の属性をチェックして、コンパニオン・サーバ間の互換性を確立します。コンパニオン・サーバのうちの 1 台に、他のコンパニオン・サーバと互換性のない属性が設定されています。do_advisory を実行して、問題のある属性のリストを表示してください。「第 6 章 do_advisory の実行」 を参照してください。

付 録 C

コマンド、システム・プロシージャ、データベースの変更

この付録では、Adaptive Server をフェールオーバ用に設定している場合のコマンド、システム・プロシージャ、システム・データベースの動作の変化について説明します。

コマンドの変更

表 C-1: コマンドの変更

コマンド	非対称型の設定	対称型の設定
create role add role drop role alter role	ノーマル・コンパニオン・モードでは、これらのコマンドによるプライマリ・コンパニオンへの変更は、すべてセカンダリ・コンパニオン・サーバに同期されます。 フェールオーバ・モード中は、セカンダリ・コンパニオンは create role、alter role の変更に従って更新されます。フェールバック・モード中は、プライマリ・コンパニオンは、その情報に従って更新されます。 フェールオーバ・モード中に、drop role は実行できません。 サスペンド・モード中に、これらのコマンドを実行することはできません。	これらのコマンドは、対称型設定でも非対称型の設定時と同じ動作をします。
create database	ノーマル・コンパニオン・モードでは、create database はセカンダリ・コンパニオン上にプロキシ・データベースを作成します。 フェールオーバ・モード中は、プライマリ・コンパニオンの model データベースがフェールオーバ・モードではないため、create database は実行できません。 フェールバック・モード中は、create database は特殊な状況下でのみ使用できます。 サスペンド・モード中に create database は実行できません。	create database は、対称型の設定でも非対称型の設定時と同じ動作をします。
alter database	ノーマル・コンパニオン・モード中に alter database を実行すると、データベースに 2MB の容量が追加されます。	alter database は、対称型の設定でも非対称型の設定時と同じ動作をします。

コマンド	非対称型の設定	対称型の設定
disk init	ノーマル・コンパニオン・モードでは、disk init は対称型の設定時と同じ動作をします。 フェールオーバ・モードでは、ユニークなデバイス名領域を確保すると、セカンダリ・サーバを使用して、そのローカル・セットにデバイスを追加できます。 サスペンド・モード中は、disk init を実行できません。	ノーマル・コンパニオン・モードでは、disk init を使用して、セカンダリ・コンパニオンに同じ物理名と論理名を持つディスクが存在しないこと、セカンダリ・コンパニオン・サーバがデバイスにアクセスできることを確認できます。 フェールオーバ・モード中は、disk init を実行できません。これは、このコマンドがプライマリ・コンパニオン上のディスクへのアクセスを検証できないためです。しかし、disk init を使用して、ログ拡張などの特別な作業を行うことは可能です。 サスペンド・モード中は、disk init を実行できません。
disk mirror disk remirror disk unmirror	Sybase ミラーリングは、高可用性システムではサポートされていません。	
disk resize	disk resize コマンドを実行しても、高可用性環境で実行中の Adaptive Server の動作に影響はありません。Adaptive Server は、ファイル・システムによって割り付けられるディスク領域は、ローカル・ディスクからプライマリ・サーバに割り付けられるのではなく、共有物理ディスクから割り付けられるものとみなします。	
drop database	ノーマル・コンパニオン・モード中に drop database は、コンパニオン・サーバにデータベースのネーム・スペースを空けるよう通知したり、またプロキシ・データベースを削除するよう要求したりできます。 フェールオーバ・モード中は、drop database コマンドに制限はありません。 サスペンド・モード中は、drop database は実行できません。	このコマンドは、対称型の設定でも非対称型の設定時と同じ動作をします。
grant revoke	ノーマル・コンパニオン・モード中に、これらのコマンドを使用して行われたパーミッションの変更は、コンパニオン・サーバ間で同期されます。 フェールオーバ・モード中は、grant に制限はありません。フェールオーバ・モード中は、revoke を実行できません。 サスペンド・モード中は、grant も revoke も実行できません。	このコマンドは、対称型の設定でも非対称型の設定時と同じ動作をします。
shutdown shutdown with nowait		

システム・プロシージャの変更

プロキシ・データベースを使用すると、クラスタによってユニークなデータベース名は保証されますが、ユニークなデータベース ID は保証されません。同じデータベースでも、フェールオーバの前後でデータベース ID が異なる場合があります。このように、データベース ID が変更される可能性があるため、フェールオーバ後にシステム・プロシージャを自動的に再コンパイルすることによって、間違った、または古いデータベース ID やオブジェクト ID が sysprocedures で使用されないようにします。

フェールオーバ・モード中、2 つの Adaptive Server に重複した名前のシステム・プロシージャが存在する場合は、Adaptive Server がドメイン・チェックを行って、正しいドメインのシステム・プロシージャが動作していることを確認します。このドメイン・チェックは、フェールオーバ・モードだけで実行されます。

テーブル・ロックを保持するシステム・プロシージャ

システム・プロシージャは、システム・テーブルの明示的なテーブル・ロックを取得できません。しかし、Sybase のフェールオーバを使用しているシステムでは、両方のコンパニオンのシステム・プロシージャがシステム・テーブルを同時に変更しようとする可能性があります。

システム・プロシージャを実行してシステム・テーブルを変更する場合、変更中のシステム・テーブルのプロキシ・テーブルに対するテーブル・ロックが取得されます。つまり、システム・プロシージャを実行してプライマリ・コンパニオン MONEY1 上の syslogins システム・テーブルを修正すると、システム・プロシージャはセカンダリ・コンパニオン PERSONNEL1 上の syslogins プロキシ・テーブルに対するテーブル・ロックを取得します。

次に、システム・プロシージャは、PERSONNEL1 上の syslogins プロキシ・テーブルを修正し、syslogins プロキシ・テーブルが MONEY1 上の syslogins システム・テーブルを更新します。変更がコミットされた後、syslogins に対するテーブル・ロックが解放されます。

同じシステム・テーブルを変更する必要がある他のシステム・プロシージャは、いずれもそのテーブルのキュー内に存在します。ロックが解放されると、システム・プロシージャはテーブル・ロックを取得します。

sp_configure “dtm lock timeout period” コマンドを使用して、システム・プロシージャが、ロックされたプロキシ・システム・テーブルをキュー内で待機する時間を設定できます。『システム管理ガイド 第 1 巻』の「第 5 章 設定パラメータ」を参照してください。

変更を同期するシステム・プロシージャ

プライマリ・コンパニオンとセカンダリ・コンパニオン間で変更を同期するシステム・プロシージャを次に示します。たとえば、sp_droplanguage を使用してプライマリ・コンパニオンからフランス語を削除すると、sp_droplanguage はセカンダリ・コンパニオンのフランス語も削除します。

これらのシステム・プロシージャは、どのデータベースからでも発行できます。

sp_addexternlogin	sp_dropremotelogin
sp_addlanguage	sp_drop_resource_limit
sp_addlogin	sp_dropserver
sp_addremotelogin	sp_drop_time_range
sp_add_resource_limit	sp_locklogin
sp_addserver	sp_modifylogin
sp_add_time_range	sp_modify_resource_limit
sp_defaultdb	sp_modify_time_range
sp_defaultlanguage	sp_password
sp_dropexternlogin	sp_remoteoption
sp_droplanguage	sp_serveroption
sp_droplogin	sp_setlangalias

次のシステム・プロシージャを master データベースから発行した場合には、プライマリ・コンパニオンとセカンダリ・コンパニオン間で変更が同期されます。

- sp_addalias
- sp_addgroup
- sp_addtype
- sp_adduser
- sp_changegroup
- sp_dropalias
- sp_dropgroup
- sp_droptype
- sp_dropuser

システム・プロシージャのその他の変更

この項では、Adaptive Server をフェールオーバ用に設定すると、動作が変わるシステム・プロシージャについて説明します。Adaptive Server をコンパニオン・サーバとして設定すると、次のように動作が変わります。

- シングルサーバ・モードで動作している場合、システム・プロシージャのデフォルトの機能は変更されない。
- フェールバック・モード中は、表 C-2 または表 C-3 に表示されているシステム・プロシージャはいずれも動作しない。
- 非対称型のプライマリ、非対称型のセカンダリ、または対称型のコンパニオンから発行されたシステム・プロシージャには、表 C-2 と表 C-3 の「ノーマル・コンパニオン・モード」の列に記述されているような動作変化がある。
- 非対称型のセカンダリ・フェールオーバ、または対称型のフェールオーバのいずれかの間に発行されたシステム・プロシージャには、表 C-2 と表 C-3 の「フェールオーバ・モード」の列に記述されているような動作変化がある。

表 C-2 は、サーバワイドな属性 (たとえばデフォルトの言語やリソースの制限など) を変更するシステム・プロシージャを示します。

- ノーマル・コンパニオン・モードでは、表 C-2 にリストされているシステム・プロシージャはすべて master から実行する必要がある。
- これらのシステム・プロシージャは、非対称型のセカンダリ・サスペンド・モードや対称型のサスペンド・モードでは実行できない。
- X は、リストされたモードではシステム・プロシージャが動作しないことを示す。

表 C-2: サーバワイドな属性を変更するシステム・プロシージャの動作変化

システム・プロシージャ	ノーマル・コンパニオン・モード	非対称型のプライマリ・サスペンド・モード	フェールオーバ・モード
sp_drop_resource _limit	このシステム・プロシージャをリモート・サーバ上でも手動で実行して、2 つのコンパニオンを同期させる必要がある。	X	X
sp_drop_time_range		X	X
sp_dropexternlogin		X	X
sp_droplanguage		X	X
sp_droplogin		X	X
sp_dropremotelogin		X	X
sp_dropserver		X	X
sp_locklogin			
sp_modify_resource _limit	このシステム・プロシージャをリモート・サーバ上でも手動で実行して、2 つのコンパニオンを同期させる必要がある。		

表 C-3 は、現在のデータベースへのユーザ、エイリアス、グループの追加など、実行先となるデータベースの属性を変更するシステム・プロシージャを示します。セカンダリ・サスペンド・モード、または対称型のサスペンド・モードでは、master からシステム・プロシージャを実行できません。X は、そのモードではシステム・プロシージャが動作しないことを示します。

表 C-3: master で動作しているときにデータベース全体の属性を変更する動作変化

システム・プロシージャ	ノーマル・コンパニオン・モード	非対称型のプライマリ・サスペンド・モード	フェールオーバ・モード	注意
sp_changedbowner			X	このシステム・プロシージャに関する追加の制限については下を参照。
sp_changegroup	このシステム・プロシージャをリモート・サーバ上でも手動で実行して、2 つのコンパニオンを同期させる必要がある。			
sp_dropalias		X	X	
sp_dropgroup		X	X	
sp_droptype		X	X	
sp_dropuser		X	X	
sp_renamedb			X	このシステム・プロシージャに関する追加の制限については下を参照。

sp_changedbowner と sp_renamedb は、フェールオーバ・モードで動作します。また、次のような動作変化もあります。

- sp_changedbowner – 次の条件に当てはまる場合は、このプロシージャを、ローカル・コンパニオン上で実行してからリモート・サーバ上でも手動で実行し、2 つのコンパニオンを同期させる必要があります。
  - このコマンドを master で実行していない。
  - コンパニオンがサスペンド・モードかノーマル・コンパニオン・モードになっている。
  - コンパニオンが、with_proxydb オプションを使用して設定されている。
- sp_renamedb – 次の条件に当てはまる場合、このシステム・プロシージャは、まずプライマリ・データベースで実行し、その後リモート・サーバ上のプロキシ・データベースで実行する必要があります。
  - このコマンドを master で実行していない。
  - コンパニオンがサスペンド・モードかノーマル・コンパニオン・モードになっている。
  - コンパニオンが、with_proxydb オプションを使用して設定されている。

高可用性システムの *dbcc* オプション

表 C-4 は、dbcc ha_admin オプションに関する情報を示します。

表 C-4: dbcc ha_admin オプション

オプション名	機能	構文とコメント
rollback_failback	sp_companion... prepare_failback の結果をロールバックし、コンパニオンをフェールオーバ・モードに戻します。このコマンドは、prepare_failback コマンドの結果に関係なく動作します。	`dbcc ha_admin (" ", rollback_failback)` " " は必要な空のプレースホルダです。 • フェールバック・モードでのみ使用できます。 • resume コマンドを待っているフェールバック・スレッドはすべて、このコマンドが実行されると強制終了されます。 • コンパニオンで rollback_failback オプションを使用するために必要な手順は、プラットフォームごとに異なります。詳細については、このマニュアル内の該当する章を参照してください。 • このコマンドは、セカンダリ・コンパニオンからのみ発行できます。
rollback_failover	プライマリ・コンパニオンからフェールオーバの結果をロールバックし、ノーマル・コンパニオン・モードに戻します。rollback_failover はセカンダリ・コンパニオンには影響しません。	`dbcc ha_admin (" ", rollback_failover)` " " は必要な空のプレースホルダです。 • このコマンドは、フェールオーバ・モードのみで使用できます。 • コンパニオンで rollback_failback オプションを使用するために必要な手順は、プラットフォームごとに異なります。詳細については、このマニュアル内の該当する章を参照してください。 • rollback_failover は、失敗したコンパニオン・サーバには影響しません。失敗したコンパニオンの作業を引き継ぐコンパニオン・サーバがノーマル・コンパニオン・モードを再開します。 • このコマンドは、セカンダリ・コンパニオンからのみ発行できます。 • このコマンドは、フェールオーバがデータベースに “suspect” のマークをつけたときにも動作します。
drop_failoverdb	フェールオーバ・モードでのみ使用されます。drop_failoeverdb はフェールオーバされたデータベースのうち、drop database コマンドで削除できなかったものを削除します。また、このコマンドは、削除されたデータベースに関連するすべてのメタデータの master_companion をクリーンアップします。	`dbcc ha_admin (" ", drop_failedoverdb, database_name)` " " は必要な空のプレースホルダで、*database_name* は削除するデータベースの名前です。 • 他のデータベースのロードを完了するためにデータベースを削除しなければならない場合に、最後の手段としてのみ使用します。
clusterlock	クラスタ・オペレーション時にクラスタ全体のロックを取得または解放します。	`dbcc ha_admin (" ", clusterlock, [acquire | release])` クラスタ全体のロックとその解放の詳細については、「高可用性ノードにおけるクラスタのロック」(6 ページ) を参照してください。

オプション名	機能	構文とコメント
state_machine	コンパニオン・サーバをシングルサーバ・モードに移行します。	`dbcc ha_admin (' ', 'state_machine', 'halt')` " " は必要な空のプレースホルダです。このオプションの使用の詳細については、付録 A を参照してください。
session	sp_companion...resume が失敗した結果スリープしているクライアントを呼び出します。呼び出されたクライアントはセカンダリ・コンパニオンとの接続を切断し、プライマリ・コンパニオンに接続します。	`dbcc ha_admin (SYB_HACMP, session, "drop")`

dbcc dbrepair オプション

Sybase のフェールオーバは、dbcc dbrepair に dropproxydb オプションを追加します。

表 C-5: dbcc dbrepair の dropproxydb オプション

オプション名	機能	構文とコメント
droppproxydb	プロキシ・データベースを削除します。	`dbcc dbrepair(`*`database_name`*`, dropproxydb)` *database_name* は、削除するプロキシ・データベースの名前です。

用語解説

この用語解説では、このマニュアルで使用されている用語についてのみ説明します。Adaptive Server と SQL の用語については、『ASE 用語解説』を参照してください。

thorough_probe

ase_monitor によって実行されるユーティリティ。thorough_probe を実行して、Adaptive Server のパフォーマンスを綿密に検査します。

アクティブ／アクティブ (active-active)

システムは 2 つのノードで構成されます。クラスタ内のどちらのノードでも Adaptive Server が独立して負荷を管理しており、一方に障害が発生したときにもう一方がその負荷を引き受けることができます。

アクティブ／パッシブ (active-passive)

マルチ・ノード設定です。この構成には、1 つの Adaptive Server、Adaptive Server がプライマリとして動作するプライマリ・ノード、必要に応じて Adaptive Server とそのリソースをホストできる一連のセカンダリ・ノードが含まれます。

クラスタ (cluster)

高可用性システムのノードの集合。Adaptive Server の高可用性システムのクラスタは、2 つ以上のノードから構成されます。

コンパニオン・サーバ (companion server)

高可用性システムの各 Adaptive Server はコンパニオンです。Adaptive Server の 1 つをコンパニオン・サーバ、もう 1 つをセカンダリ・コンパニオン・サーバと呼びます。

サスペンド・コンパニオン・モード (suspended companion mode)

コンパニオン・モードがサスペンドされた後の Adaptive Server のモード。このモードの間、Adaptiver Server はフェールオーバできず、コンパニオン Adaptive Server とは独立して稼働します。

シングルサーバ・モード (single-server mode)

Adaptiver Server に高可用性の設定が行われているときのモード。このモードの間は、Adaptive Server はフェールオーバできません。

ステーブル・モード (stable mode)

Adaptive Server が長期間存続できるシステム状態 (Adaptive Server の日常のオペレーションなど)。

セカンダリ・コンパニオン (secondary companion)

フェールオーバされるプライマリ Adaptive Server を受け入れるように設定された Adaptive Server。

ノード (node)

高可用性システム内に存在するマシン。

ノーマル・コンパニオン・モード (normal companion mode)

高可用性システムの 2 つの Adaptive Server が独立したサーバとして機能し、定期保守やシステム障害時にフェールオーバされるように設定されているモード。

フェールオーバ (failover または fail over)
Adaptive Server が別のマシンにマイグレートし、そのマシンが障害の発生したサーバの管理を引き受けるプロセス。フェールオーバは、定期保守、Adaptive Server の障害、または Adaptive Server を実行するマシンの障害によって発生する可能性があります。

フェールオーバ・モード (failover mode)
プライマリ・コンパニオンがフェールオーバされてセカンダリ・コンパニオン上で実行されるようになった後の、プライマリ・コンパニオンのモード。

フェールバック (failback または fail back)
フェールオーバが発生した後、Adaptive Server がプライマリ・コンパニオンにマイグレートされ再起動する、予定されたイベント。このイベントでは、フェールオーバされたデータベース、デバイス、クライアント接続が、セカンダリ・コンパニオンから再起動したプライマリ・コンパニオンへ移動します。

プライマリ・コンパニオン (primary companion)
Adaptive Server の 1 つ。フェールオーバ時にそのデータベースと接続がセカンダリの Adaptive Server にマイグレートされます。

プロキシ・データベース (proxy databases)
プライマリ・コンパニオン上のユーザ・データベースごとに、セカンダリ・コンパニオン上に作成されるプレースホルダ・データベース。プロキシ・データベースはデータベース名を予約して、フェールオーバ時にシステム上ですべてのデータベース名がユニークであるようにします。

移行モード (transitional mode)
Adaptive Server がフェールオーバ・モードからノーマル・コンパニオン・モードに移行するときに発生するモードです。通常は非常に短時間です。

高可用性 (high availability)
ダウン時間を減らすように設計されたシステム。

接続フェールオーバ (connection failover)
フェールオーバ・プロパティがあり、セカンダリ・コンパニオンにフェールオーバする接続。

対称型 (symmetric)
2 つの独立した Adaptive Server が互いにフェールオーバ・サーバとして機能する高可用性システム。つまり、各 Adaptive Server がプライマリ・コンパニオンとセカンダリ・コンパニオンの両方として機能します。

非対称型 (asymmetrical)
1 つのプライマリ・コンパニオンと 1 つのセカンダリ・コンパニオンから構成される高可用性システム。非対称型システムでは、プライマリ・コンパニオンだけがフェールオーバ可能です。セカンダリ Adaptive Server を「ホット・スタンバイ」ともいいます。

索引

記号

$SYBASE の値の設定
  HP 47
  IBM 78
@@cmpstate グローバル変数 28
@@hacmpservername グローバル変数 197

数字

18750 エラー・メッセージ、IBM システム 98
2 フェーズ・コミット・トランザクション
  Adaptive Server 18

A

Adaptive Server
  2 フェーズ・コミット・トランザクション 18
  IBM に関する考慮事項 76
  パフォーマンス、対称型設定 22
  パフォーマンス、非対称型設定 20
Adaptive Server interfaces ファイルのエントリ追加
  HP 46
  IBM 78
  Sun Cluster、アクティブ／パッシブ 150
  Veritas システム 172
  フェールオーバ中のアクティブ／アクティブ 107
add role コマンド 199
alter role コマンド 199
ASE_HA.sh スクリプトの編集
  HP 53
  IBM 83

C

create database コマンド 199
  低下、パフォーマンス 20
create role コマンド 199
CTLIB API 呼び出し、フェールオーバ用の変更 193

D

dbcc dbrepair オプション
  dropproxydb オプション 206
dbcc ha_admin オプション 6, 205
  2 次ポイント障害 197
  clusterlock 205
  drop_failoverdb オプション 205
  prepare_failback オプション 197
  rollback_failback オプション 197, 205
  rollback_failover オプション 197
  session 206
  state_machine オプション 206
  説明 195
dbcc ha_admin オプションの clusterlock オプション 205
dbcc ha_admin オプションの drop_failoverdb オプション 205
dbcc ha_admin オプションの session オプション 206
dbcc ha_admin オプションの state_machine オプション 206
disk init コマンド 200
disk mirror コマンド 200
disk remirror コマンド 200
disk resize コマンド 200
disk unmirror コマンド 200
do_advisory オプション
  Sun 41
  グループ属性 38
  出力 41
  説明 37
  フェールバック、実行 5
drop database コマンド 200
drop role コマンド 199
dropproxydb オプション
  dbcc dbrepair オプション 206
dtm lock timeout period コマンド 201

G

grant コマンド 200

H

ha_role sp_companion　110
　HP　50
　IBM　83
　Veritas システム　175
HACMP
　IBM AIX システムでのトラブルシューティング　98
　IBM システムでリソース・グループを設定　87
hafailover ラベル、interfaces ファイルへの追加　3
HP システムの設定　45–70
　$SYBASE　47
　Adaptive Server のインストール　46
　Adaptive Server の準備　46
　ASE_HA.sh スクリプトの編集　53
　ha_role sp_companion　50
　installhasvss スクリプト　50
　interfaces ファイル、エントリを追加　46
　sybha 実行プログラム　47
　sysservers へのセカンダリ・コンパニオンの追加　49
　sysservers へのローカル・サーバの追加　49
　新しいデフォルト・デバイスの作成　49
　稼働条件　45
　コンパニオン・モードの削除　67
　パッケージ制御スクリプト　59
　パッケージの設定　52
　パラメータ、確認　50
　フェールオーバ　51

I

IBM システムでのエラー・メッセージ 18750　98
IBM システムの設定　75–99
　$SYBASE　78
　Adaptive Server のインストール　77
　Adaptive Server の準備　77
　Adaptive Server、インストール　77
　ASE_HA.sh スクリプトの編集　83
　ha_role sp_companion　83
　HACMP リソース・グループ　87
　installhasvss スクリプト　82
　interfaces ファイル、エントリを追加　78
　sybha 実行プログラム　79
　sysservers にローカル・サーバを追加　82
　新しいデフォルト・デバイスの作成　81
　エラー・メッセージ 18750　98
　稼働条件　76
　コンパニオン・モードの削除　97
　失敗した prepare_failback からのリカバリ　99
　シャットダウンしたコンパニオンのサスペンド・モード中の再起動　96
　手動フェールバック　94
　スレッショルドをマスタ・ログに追加　81
　セカンダリ・コンパニオンを sysservers に追加　82
　トラブルシューティング、HACMP でのフェールオーバ　98
　ノーマル・コンパニオン・モードの再開　96
　パラメータ、確認　80
　フェールオーバ用コンパニオン・サーバ　90
　フェールオーバ・ログのロケーション　99
　モニタ対象リソースとしてのプライマリ・コンパニオン　93
installhasvss スクリプト
　IBM　82
　Veritas システム　175
　インストール、ストアド・プロシージャ　17
　再インストール　195
installmaster スクリプト
　installhasvss の実行　17
　フェールオーバ用のストアド・プロシージャ　17
interfaces ファイル
　hafailover ラベル、追加　3
　HP　46
　IBM　78
　Sun Cluster、アクティブ／パッシブ　150
　Sun アクティブ／アクティブ　106
　Veritas システム　172
　Veritas システム、フェールオーバ中　172
　エントリ追加、アクティブ／アクティブのフェールオーバ中　107

P

prepare_failback
　dbcc ha_admin オプション　197
　sp_companion、構文　5
　sp_companion、発行　5
　リカバリ　139
prepare_failback、リカバリ
　IBM　99
　Veritas システム　192

R

revoke コマンド　200
rollback_failback オプション
　dbcc ha_admin　197
　dbcc ha_admin オプション　205

S

shutdown with nowait コマンド　200
shutdown コマンド　200
　IBM システムでのサスペンド・モード時の再起動　96
sp_companion　28
　do_advisory オプションの説明　37
　定属性　42
　フェールバック、発行　5
　フェールバック、発行する構文　5
sp_companion do_advisory オプション　125
sp_companion ha_role
　HP　50
　IBM　83
　Sun アクティブ／アクティブ　110
　Veritas システム　175
sp_companion prepare_failback コマンド　5
sp_companion resume コマンド　5
sp_companion の定属性　42
sp_configure コマンド
　number of open databases　13
　number of user connections　13
sp_dboption コマンドとプロキシ・データベース　35
srids の条件　13
Sun　103
Sun Cluster アクティブ／アクティブ
　ha_role と sp_companion、アクティブ／アクティブ　110
　installhasvss スクリプト、アクティブ／アクティブ　110
　interfaces ファイル、アクティブ／アクティブのエントリの追加　106
　sp_companion と do_advisory オプション　125
　Sun Cluster でのコンパニオン・モードの削除　130
　sybha 実行プログラム、アクティブ／アクティブ　108
　アクティブ／アクティブの sysservers へのローカル・サーバの追加　109
　アクティブ／アクティブのオペレーティング・システム　103
　アクティブ／アクティブのハードウェア　103
　アクティブ／アクティブ用の新しいデフォルト・デバイスの作成　109
　高可用性アクティブ／アクティブの Adaptive Server のインストール　106
　高可用性アクティブ／アクティブの Adaptive Server の準備　106
　高可用性の設定　112
　失敗した prepare_failback からのリカバリ　139
　スレッショルドをマスタ・ログに追加　111
　セカンダリ・コンパニオンをアクティブ／アクティブの sysservers に追加　110
　対称型設定　127
　手順　119
　ノーマル・コンパニオン・モードの再開　129
　パラメータ、確認　111
　フェールオーバ中の interfaces ファイル、エントリ追加、アクティブ／アクティブ　107
　フェールオーバのトラブルシューティング　139
　フェールオーバ用コンパニオン・サーバ　123
　プローブ用のログインの追加　111
　ユーザの追加　111
　リソース・グループの手動設定　131
Sun Cluster のアクティブ／パッシブ　143–165
　interfaces ファイル、エントリを追加　150
　Sun Cluster のアクティブ／パッシブ設定　153
　アクティブ／パッシブの interface エントリ　150
　クライアント設定　147
　高可用性接続　151
　スレッショルドをマスタ・ログに追加　152
　設定　146, 147, 153
　手順　158
　パラメータ、確認　152
　非高可用性接続　151
　フェールバック　147
　プローブ用のログインの追加　152
　ユーザの追加　152
　リソース・グループの手動設定　163
　ログのロケーション　167
Sun システムの設定
　プローブ用のログインの追加　207
　ユーザの追加　207
SY.ase ファイルのプロパティ　117, 179
SYB_HACMP　17
　installhasvss スクリプト　17
　誤って削除した場合の処理　17

sybha 実行プログラム
HP での実行　47
IBM システムでの実行　79
Sun アクティブ／アクティブでの実行　108
Veritas システムでの実行　173
sybsecurity とフェールオーバ　23
sysdevices、フェールオーバ時のマッピング　2
sysservers でリモート・サーバを追加
HP　49
sysservers にセカンダリ・コンパニオンを追加
HP　49
sysservers、セカンダリ・コンパニオンの追加
IBM　82
Sun アクティブ／アクティブ　110
Veritas システム　175
sysservers、ローカル・サーバの追加
HP　49
IBM　82
Sun アクティブ／アクティブ　109
Veritas システム　174

V

Veritas システムの設定　170–192
Adaptive Server のインストール　172
ha_role sp_companion　175
installhasvss スクリプト　175
interfaces ファイル (フェールオーバ中)、エントリを追加　172
interfaces ファイル、エントリを追加　172
sybha 実行プログラム　173
sysservers にローカル・サーバを追加　174
Veritas クラスタでのフェールオーバのトラブルシューティング　191
Veritas クラスタの設定　177
新しいデフォルト・デバイスの作成　174
コンパニオン・モードの削除　187
失敗した prepare_failback からのリカバリ　192
スレッショルドをマスタ・ログに追加　176
セカンダリ・コンパニオンを sysservers に追加　175
対称型設定　184
ノーマル・コンパニオン・モードの再開　186
パラメータ、確認　176
フェールオーバの管理　185
フェールオーバ用コンパニオン・サーバ　181
ログのロケーション　192

あ

アクティブ／アクティブ設定
Sun Cluster の設定　112
アクティブ／パッシブ設定　147
interfaces エントリ、Sun Cluster　150
Sun Cluster　143–167
Sun Cluster の設定　146
クライアント設定　147

い

移行フェールオーバ・モード　25
インストール
HP　46
IBM　77
Veritas システム　172

お

オープン・データベース、必要数　13
オペレーションの再開　4, 9

か

稼働条件
フェールオーバ　12
監査
設定パラメータ　22
設定、オプション　23
監査証跡とフェールオーバ　23

く

クライアント接続
クラスタのロック　6
フェールオーバ　3
クラスタ
説明　14
定義　207
クラスタのロック
解放　6
クライアント接続　6
クラスタ全体のロック　6
高可用性ノード　6

こ

高可用性　9, 208
　HP でのストアド・プロシージャのインストール　50
　サブシステム　14
　接続、アクティブ／パッシブ Sun Cluster　151
　設定の図　15
　定義　208
　ノード、クラスタのロック　6
　フェールオーバ　14
高可用性の設定
　HP　45–70
　IBM　75–99
　Veritas システム　170–192
更新、プロキシ・データベース　35
コンパニオン
　ディスク・ミラーリング　17
　非対称型設定　64, 91, 125, 183
　フェールオーバ　1
コンパニオン・サーバ
　IBM 用の設定　90
　Sun アクティブ／アクティブの設定　123
　Veritas システムでの設定　181
　サスペンド・モード　27
　シングルサーバ・モード　26
　定義　207
　ノーマル・コンパニオン・モード　26
　ノーマル・コンパニオン・モードからサスペンド・モードへの再開　27
　フェールバック・モード　27
　命名、@@hacmpservername　197
　モード　26
　モードの確認　28
コンパニオン・モード
　HP　67
　IBM　97
　Sun アクティブ／アクティブのサスペンド　129
　Veritas システム　187
　削除　130
コンパニオン・モード、サスペンド
　IBM　95
　Sun アクティブ／アクティブ　129
　Veritas 3.5 システム　186
コンポーネント統合サービス (CIS)、プロキシ・データベースの作成　32

さ

サーバ
　HP システムで sysservers にリモート・サーバを追加　49
　HP システムで sysservers にローカル・サーバを追加　49
　IBM システムで sysservers にセカンダリ・コンパニオンを追加　82
　SYB_HACMP　17
　Veritas システムで sysservers にセカンダリ・コンパニオンを追加　175
　サスペンド・モード　27
　セカンダリ・コンパニオンを Sun アクティブ／アクティブの sysservers に追加　110
　ノーマル・コンパニオン・モード　26
　ノーマル・コンパニオン・モードからサスペンド・モードへの再開　27
　フェールバック・モード　27
サーバ、コンパニオン
　命名、@@hacmpservername　197
　モード　26
再インストール
　installhasvss スクリプト　195
作成
　プロキシ・データベース　33
サスペンド・モード　27
　IBM システムでのシャットダウンしたコンパニオンの再起動　96
　定義　207

し

システム・フェールオーバ　1
システム・プロシージャ
　フェールオーバによる動作変化　201
　プロキシ・データベース、発行　35
失敗した prepare_failback からのリカバリ　139
　IBM　99
　Veritas 3.5 システム　192
手動フェールバック
　IBM　94
準備
　HP　46
　IBM　77
条件
　リソース　13

シングルサーバ・モード　26
　定義　207

す

数
　オープン・データベース、sp_configure コマンド　13
　オープン・データベース、条件　13
　デバイス、条件　13
　ユーザ接続、sp_configure コマンド　13
　ユーザ接続、条件　13
ステーブル・フェールオーバ・モード　25
ストアド・プロシージャ
　installhasvss スクリプト　17
スレッショルド、マスタ・ログに追加
　IBM　81
　Sun Cluster アクティブ／アクティブ　111
　Sun Cluster のアクティブ／パッシブ　152
　Veritas システム　176

せ

セカンダリ・コンパニオン　14
　定義　207
接続フェールオーバ　1
設定
　Adaptive Server　12
　Sun Cluster アクティブ／アクティブ　112
　Sun のアクティブ／アクティブ　103–141
　アクティブ／パッシブのクライアント　147
　アクティブ／パッシブ、Sun Cluster　147
　高可用性の Sun Cluster アクティブ／パッシブ　143–167
　コンパニオン・サーバ、監査用　22
　対称型　19–22, 65, 92, 127
　パラメータ、監査用　22
　非対称型　19–22
　リソース・グループの手動設定　163
　リソース・グループ、手動、Sun Cluster　131
設定条件
　HP　45
　IBM　76
設定パラメータ、確認
　IBM　80
　Sun Cluster アクティブ／アクティブ　111
　Sun Cluster のアクティブ／パッシブ　152
　Veritas システム　176
設定、対称型
　Veritas システム　184

た

対称型
　設定　65, 92
対称型設定　19–22, 127
　Veritas システム　184
　コンパニオン　21
　パフォーマンス、Adaptive Server　22
対称型モード
　add role コマンド　199
　alter role コマンド　199
　create database コマンド　199
　create role コマンド　199
　disk init コマンド　200
　disk mirror コマンド　200
　disk remirror コマンド　200
　disk resize コマンド　200
　disk unmirror コマンド　200
　drop database コマンド　200
　drop role コマンド　199
　grant コマンド　200
　revoke コマンド　200
　shutdown with nowait コマンド　200
　shutdown コマンド　200
　定義　208
単一のシステム　16

て

ディスク障害とフェールオーバ　17
ディスク・ミラーリングとコンパニオン・クラスタ　17
データベース
　作成、プロキシ　32
　必要なオープン・データベース数　13
　プロキシ　31–36
データベース ID とフェールオーバ　201
テーブル・ロックとフェールオーバ　201
デバイス、必要数　13
デフォルト・デバイス、新規作成　109
　HP　49
　IBM　81
　Veritas システム　174

と

ドメイン　29
　フェールオーバ中のチェック　201
トラブルシューティング
　dbcc ha_admin オプション　195

の

ノード　14
　定義　207
ノーマル・コンパニオン・モード　26
　IBM の再開　96
　Veritas システムの再開　186
　定義　207
ノーマル・コンパニオン・モード、Sun Cluster アクティブ／アクティブでの再開　129

は

パッケージ
　HP システムでの設定　52
　制御スクリプト、HP システムでの作成　59
パラメータ、設定の確認
　HP　50
　IBM　80
　Sun Cluster アクティブ／アクティブ　111
　Sun Cluster のアクティブ／パッシブ　152
　Veritas システム　176

ひ

非高可用性接続
　Sun Cluster のアクティブ／パッシブ　151
非対称型設定　19–22
　コンパニオン　64
　コンパニオンの設定　91, 125, 183
　説明　19
　パフォーマンス、Adaptive Server　20
非対称型モード
　add role コマンド　199
　alter role コマンド　199
　create database コマンド　199
　create role コマンド　199
　disk init コマンド　200
　disk mirror コマンド　200
　disk remirror コマンド　200
　disk resize コマンド　200
　disk unmirror コマンド　200
　drop database コマンド　200
　drop role コマンド　199
　grant コマンド　200
　revoke コマンド　200
　shutdown with nowait コマンド　200
　shutdown コマンド　200
　定義　208
非対称型、定義　208

ふ

フェールオーバ
　Adaptive Server interfaces ファイルへのエントリの追加、Sun アクティブ／アクティブのフェールオーバ中　107
　dbcc ha_admin オプション　205
　HP システムでの設定　51
　IBM システムでの管理　93
　IBM システムでのログのロケーション　99
　interfaces ファイルの hafailover ラベル　3
　interfaces ファイルへのエントリの追加、フェールオーバ中、Veritas 3.5 システム　172
　Sun アクティブ／アクティブの管理　128
　sybsecurity　23
　sysdevices、マッピング　2
　Veritas システムでの管理　185
　移行モード　25
　一連のステップ　1, 9
　稼働条件　12
　監査証跡　23
　クライアント接続　3
　高可用性　14
　コンパニオン・フェールオーバ　1
　サーバワイドな変更を行うシステム・プロシージャ　203
　システム・フェールオーバ　1
　システム・プロシージャの動作変化の原因　201
　使用するドメイン　29
　図　3
　ステーブル・モード　25
　接続フェールオーバ　1
　設定に関する注意事項　16
　説明　1, 9

定義　208
ディスク障害　17
ディスク・ミラーリング　17
データ型の設定　18
データベース ID　201
データベース全体の変更を行うシステム・プロシージャ　204
テーブル・ロック　201
ドメイン・チェック　201
フェールオーバとともに実行するアプリケーション　13
変更、CTLIB API 呼び出し　193
モード　25–27
ユーザのログイン　3
フェールオーバのトラブルシューティング
IBM システムの HACMP for AIX　98
Sun Cluster アクティブ／アクティブ　139
Veritas クラスタ　191
フェールバック　27
do_advisory オプション、実行　5
IBM システムにおける手動での方法　94
sp_companion、構文　5
sp_companion、発行　5
アクティブ／パッシブ Sun Cluster　147
実行　5
説明　4, 9
定義　208
プライマリ・コンパニオン、Sun アクティブ／アクティブ　128
プライマリ・ノード IBM　94
プライマリ・コンパニオン　14
IBM システムでのモニタ対象リソース　93
定義　208
プローブ用のログインの追加　111, 152, 207
プロキシ・データベース　31–36
sp_dboption コマンド　35
サイズ　33
作成　32
作成のタイミング　33
システム・プロシージャ、発行　35
手動更新　35
使用できないコマンド　34
設定、フェールオーバ　31
定義　208
プロパティ、SY.ase ファイル　117, 179

ま

マスタ・ログ、スレッショルドの追加
IBM　81
Sun Cluster アクティブ／アクティブ　111
Sun Cluster のアクティブ／パッシブ　152
Veritas システム　176

も

モード
@@cmpstate、モードの確認　28
コンパニオン・サーバ　26
サスペンド　27
ノーマル・コンパニオン　26
ノーマル・コンパニオンからサスペンドへの再開　27
フェールバック　27
モニタリング・テーブル、高可用性でのインストール　16

ゆ

ユーザ
接続、必要数　13
フェールオーバにおけるログイン　3
ユーザの追加　111, 152, 207

り

リソース・グループ
IBM システムでの HACMP による設定　87
リソースの条件　13

ろ

ローカル・サーバ、sysservers による追加
HP　49
IBM　82
Sun アクティブ／アクティブ　109
Veritas システム　174

ログ
IBM システムでのフェールオーバ・ログのロケーション 99
Sun Cluster のアクティブ／アクティブでスレッショルドをマスタ・ログに追加 111
Sun Cluster のアクティブ／パッシブでスレッショルドをマスタ・ログに追加 152
Veritas システムでのロケーション 192
スレッショルドをマスタ・ログに追加、Veritas 176
ロケーション、Sun Cluster のアクティブ／パッシブ 167
ログイン
条件 13
フェールオーバ 3