

CLUSTERPROシステム構築ガイド

CLUSTERPRO[®] for Linux Ver 2.0

システム設計編(応用)

第3版 2003.03.28

改版履歴

版 数	改版年月日	改版ページ	内 容
第1版	2001.12.14		Ver1.0の第3版をベースに新規作成
第2版	2002.7.23	7	フローティングIPの活性時のリトライ処理の追記
第3版	2003.3.28	10 26,27	スクリプトの環境変数名において、”AEMS_GROUPNAME… グループ名”をフェイルオーバーグループ名に変更 環境変数の記載を修正

はじめに

『CLUSTERPROシステム構築ガイド』は、これからクラスタシステムを設計・導入しようとしているシステムエンジニアや、すでに導入されているクラスタシステムの保守・運用管理を行う管理者や保守員の方を対象にしています。

CLUSTERPRO®は日本電気株式会社の登録商標です。

Linuxは、Linus Torvalds氏の米国およびその他の国における登録商標あるいは商標です。その他のシステム名、社名、製品名等はそれぞれの会社の商標または登録商標です。

CLUSTERPROドキュメント体系

CLUSTERPROのドキュメントは、CLUSTERPROをご利用になる局面や読者に応じて以下の通り分冊しています。初めてクラスタシステムを設計する場合は、システム構築ガイド【入門編】を最初に読んでください。

■ システム構築ガイド

【入門編】

(必須) 設計・構築・運用・保守

クラスタシステムをはじめて設計・構築する方を対象にした入門書です。

【システム設計編(基本/共有ディスク)】

(必須) 設計・構築・運用・保守

クラスタシステムを設計・構築を行う上でほとんどのシステムで必要となる事項をまとめたノウハウ集です。構築前に知っておくべき情報、構築にあたっての注意事項などを説明しています。

【システム設計編(応用)】

(選択) 設計・構築・運用・保守

設計編(基本)で触れなかったCLUSTERPROのより高度な機能を使用する場合に必要な事項をまとめたノウハウ集です。

【クラスタ生成ガイド(共有ディスク)】

(必須) 設計・構築・運用・保守

CLUSTERPROのインストール後に行う環境設定を実際の作業手順に沿って分かりやすく説明しています。

【運用/保守編】

(必須) 設計・構築・運用・保守

クラスタシステムの運用を行う上で必要な知識と、障害発生時の対処方法やエラー一覧をまとめたドキュメントです。

【GUIリファレンス】

(必須) 設計・構築・運用・保守

クラスタシステムの運用を行う上で必要なCLUSTERPROマネージャなどの操作方法をまとめたリファレンスです。

【コマンドリファレンス】

(選択) 設計・構築・運用・保守

CLUSTERPROのスクリプトに記述できるコマンドやサーバから実行できる運用管理コマンドについてのリファレンスです。

【トレッキングツール編】

(選択) 設計・構築・運用・保守

CLUSTERPROトレッキングツールの操作方法を説明したリファレンスです。分冊(GUI、システム構成(共有ディスクシステム))しています。

目次

1	CLUSTERPROの高度な機能.....	6
1.1	リソース.....	6
1.1.1	リソース監視.....	6
1.2	スクリプト.....	8
1.2.1	スクリプトの種類.....	8
1.2.2	スクリプトの環境変数.....	9
1.2.3	スクリプトの構造.....	11
1.2.4	スクリプト作成のヒント.....	32
1.2.5	ディスクアクセス中のプロセス強制停止機能.....	32
2	高度なシステムの設計.....	33
2.1	アプリケーション監視(armload).....	33
2.1.1	監視対象となる障害.....	33
2.1.2	環境設定.....	33
2.2	アプリケーションモニタ.....	34
3	マスタクラスタ.....	35
3.1	スーパークラスタ.....	35
3.2	マスタクラスタ.....	35

1 CLUSTERPROの高度な機能

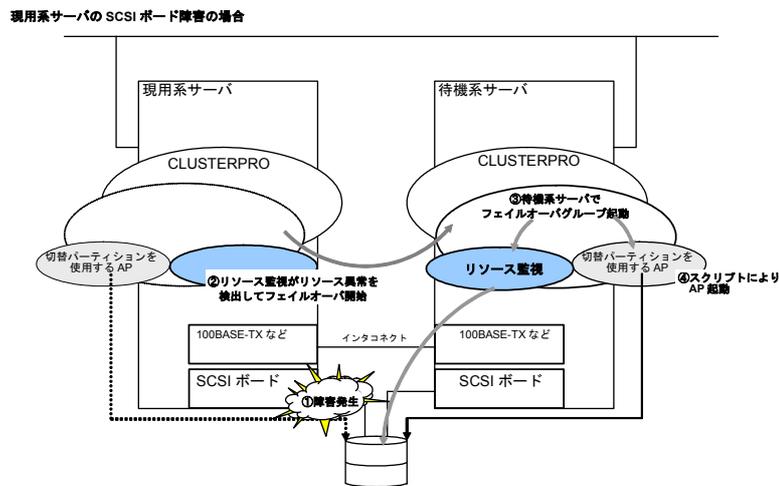
1.1 リソース

1.1.1 リソース監視

CLUSTERPROコマンドのarmrspと同等なリソース監視機能を、フェイルオーバーグループのリソースとして持つことができます。

1.1.1.1 概要

リソース監視は、指定された種類のリソースを監視します。リソースの異常を検出した場合には、フェイルオーバーまたはグループの停止を行います。



フェイルオーバー先については、次の中から選択できます。

- (1) 最高プライオリティサーバ
フェイルオーバーポリシーの設定にしたがって、決定します。
- (2) 安定動作中サーバ
安定稼動中サーバは、次の順序で決定します。
 1. グループ起動後、リソース異常を検出したことのないサーバのうち、フェイルオーバーポリシーが最も高いサーバ
 2. リソース異常を検出したサーバのうち、最も過去に異常を検出したサーバ

1.1.1.2 監視対象となるリソース

監視対象となるリソースは以下のものです。リソースの特性上、監視可能な状態の範囲に以下の2つがあります。

- (A) 起動(活性化)時/運用中(一旦活性化に成功しリソースとして使用中)共に監視可能なもの
 - * 切替パーティション
- (B) 起動(活性化)時のみ監視可能なもの
 - * フローティングIP
 - * スクリプト
 - * Public LAN

1.1.1.3 リソース異常の検出方法

- * CLUSTERパーティション
I/Oエラーが発生したとき、異常とみなします。
- * ディスク
 - (1)共有ディスク(SCSI)
SCSIコマンドのTEST_UNIT_RAEDYにより、異常を判断します。したがって、アダプタやケーブルの異常を検出することはできますが、ディスクのメディアエラーを検出することはできません。
またSCSIコマンドのTEST_UNIT_RAEDYをサポートしていないホストバスアダプタは使用できません。
 - (2)NFSサーバ
NFSサーバへのpingにより、異常を判断します。
- * フローティングIP
リソースの活性化に失敗したとき、異常とみなします。フローティングIPアドレスのネットワークアドレス誤りや、IPアドレスの競合などがこれに該当します。
活性化時にIPアドレスの競合を検出した場合には30秒間隔で最大5回のリトライを行います。¹
- * スクリプト
スクリプトに記述したexitの終了コードが0以外のとき、異常と判断します。
- * パブリックLAN
指定のIPアドレスへpingを発行することにより、異常を検出します。

1.1.1.4 注意事項

- * 障害発生が発生したサーバ以外に正常なサーバがない場合には、フェイルオーバは行いません。
- * 指定された回数(既定値は8回)フェイルオーバが繰り返された後はフェイルオーバをしません。サーバが共通して使用しているリソースの障害によるフェイルオーバの繰り返しを避けるためです。このカウンタは以下の場合にリセットされます。
 - = サーバの起動時
 - = サーバの再起動時
 - = armrspコマンドを起動するフェイルオーバグループの起動時
 - = 監視対象リソースの正常状態が指定以上継続したとき
- * Public LANの監視は、設定したIPアドレスに対してpingを通すことにより行います。したがって、LAN自体は正常でも、pingが通らなければ異常と判断し、フェイルオーバを行います。
Public LANの監視には、デフォルトゲートウェイのIPアドレスを設定することをお勧めします。

¹ リトライ処理はSE2.1/LE2.1以降の機能です。

1.2 スクリプト

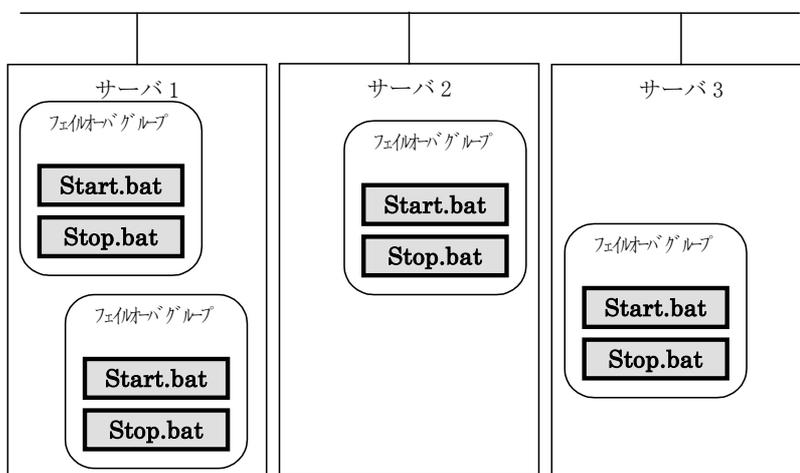
CLUSTERPROの配下で、プログラムプロダクトやアプリケーションを動作させるためには、スクリプトに起動/終了/復旧の手順を記述する必要があります。本章では、その記述方法を説明します。

スクリプトはLinuxのシェルに記述して作成します。また、スクリプト内ではLinuxのコマンド以外にも、CLUSTERPROコマンドを使用することができます。

1.2.1 スクリプトの種類

フェイルオーバーグループには、それぞれ開始スクリプトと終了スクリプトが用意されています。

CLUSTERPROは、クラスタの状態遷移が必要な場面において、フェイルオーバーグループごとのスクリプトを実行します。クラスタ環境下で動作させたいプログラムプロダクトやアプリケーションの起動/終了、もしくは復旧の手順を、これらのスクリプトに記述する必要があります。



Start.bat : 開始スクリプト
Stop.bat : 終了スクリプト

フェイルオーバーポリシーで設定されている、本来スクリプトが実行されるべきサーバを、そのスクリプトの「プライマリサーバ」と呼びます。

1.2.2 スクリプトの環境変数

CLUSTERPROは、スクリプトを実行する際に、どの状態で実行したか(スクリプト実行要因)等の情報を、環境変数にセットします。

スクリプト内で下図の環境変数を分岐条件として、システム運用にあった処理内容を記述できます。

終了スクリプトの環境変数は、直前に実行された開始スクリプトの内容を、値として返します。

ただし、ARMS_EVENTの環境変数RECOVERに対応した終了スクリプト、およびARMS_FACTORの各環境変数に対応した開始スクリプトはありません。

環境変数	環境変数の値	意味	フェイルオーバー型
ARMS_EVENT …スクリプト実行要因	START	前回、クラスタシャットダウンした時に動作していたサーバと、同一あるいはより高いプライオリティのサーバで実行された。フェイルオーバーグループの移動により、移動先のサーバで実行された。 armloadコマンドがアプリケーション異常を検出し、同じサーバでスクリプトが再起動された。	使用可
	RECOVER	プライマリサーバとしてフェイルオーバーグループが設定してあるサーバの、クラスタ復帰を行った。 armloadコマンドがアプリケーション異常を検出し、同じサーバでスクリプトが起動/再起動された。	使用可
	FAILOVER	フェイルオーバーにより、フェイルオーバー先のサーバで実行された。 armloadコマンドがアプリケーションの異常を検出し、フェイルオーバー先のサーバでスクリプトが起動された。 前回シャットダウンした時に動作していたサーバより、低いプライオリティのサーバで実行された。	使用可
ARMS_FACTOR …グループ停止要因	CLUSTERSHUTDOWN	クラスタシャットダウンにより、サーバのシャットダウンが実行された場合	使用可
	SERVERSHUTDOWN	サーバシャットダウンにより、サーバのシャットダウンが実行された場合	使用可
	CLUSTERLEAVE	サーバ切り離しにより、サーバの切り離しが実行された場合	使用可
	GROUPSTOP	グループ停止により、グループの停止が実行された場合	使用可
	GROUPMOVE	グループ移動により、グループの移動が実行された場合	使用可
	GROUPFAILOVER	armloadコマンド及びarmrspコマンドにより、グループのフェイルオーバーが実行された場合	使用可
	GROUPRESTART	armloadコマンドより、スクリプトの再起動が実行された場合	使用可
ARMS_SERVER …スクリプトの実行サーバ	HOME	フェイルオーバーグループの、プライマリサーバで実行された。	使用可
	OTHER	フェイルオーバーグループの、プライマリサーバ以外で実行された。	使用可

環境変数	環境変数の値	意 味	フェイルオーバー型
ARMS_DISK … 共有ディスク上のパーティション接続情報、またはNFSリソース接続情報	SUCCESS	接続に失敗しているパーティションはない。	使用可
	FAILURE	接続に失敗しているパーティションがある。	使用可
ARMS_PRIORITY … スクリプトが実行したサーバのプライオリティ	1~16	実行されているサーバの、プライオリティを示す。1から始まる数字で、小さいほどプライオリティが高いサーバ。(とりうる値は、1からサブクラスタ内のサーバ数となる) ARMS_PRIORITYが1の場合、プライマリサーバで実行されたことを示す。	使用可
ARMS_GROUPNAME … フェイルオーバーグループ名	フェイルオーバーグループ名	スクリプトが属している、フェイルオーバーグループ名を示す。	使用可

1.2.3 スクリプトの構造

1.2.3.1 スクリプトの実行タイミング

開始/終了スクリプトの実行タイミングと環境変数の関連を、クラスタ状態遷移図にあわせて説明します。

- * 説明を簡略にするため、2台構成のクラスタで説明します。
3台以上の構成の場合に、発生する可能性のある実行タイミングと環境変数の関連は、補足という形で説明します。
- * 図中の○や×はサーバの状態を表しています。

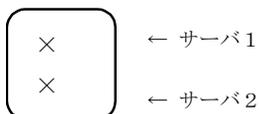
サーバ	サーバ状態
○	正常状態（クラスタとして正常に動作している）
×	停止状態（クラスタシャットダウンによる停止状態）
× d	ダウン状態（サーバダウンによる停止状態）
○ d	ダウン後再起動状態（クラスタから切り離されている）

(例) ○A : 正常状態にあるサーバにおいてフェイルオーバーグループAが動作している。

- * 各フェイルオーバーグループは、起動したサーバの中で、最もプライオリティの高いサーバ上で起動されます。
- * クラスタに定義されているフェイルオーバーグループはA、B、Cの3つで、それぞれ以下のようなフェイルオーバーポリシーを持っています。

フェイルオーバーグループ	優先度1サーバ	優先度2サーバ
A	サーバ1	サーバ2
B	サーバ2	サーバ1
C	サーバ1	サーバ2

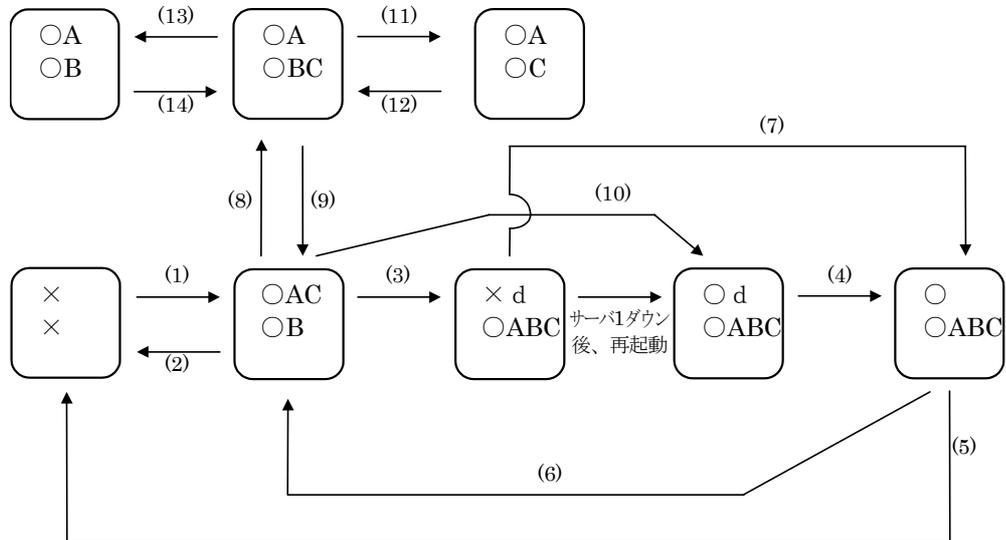
- * 上のサーバをサーバ1、下のサーバをサーバ2とします。



図中の(1)～(14)は、次ページからの説明に対応しています。

【クラスタ状態遷移図】

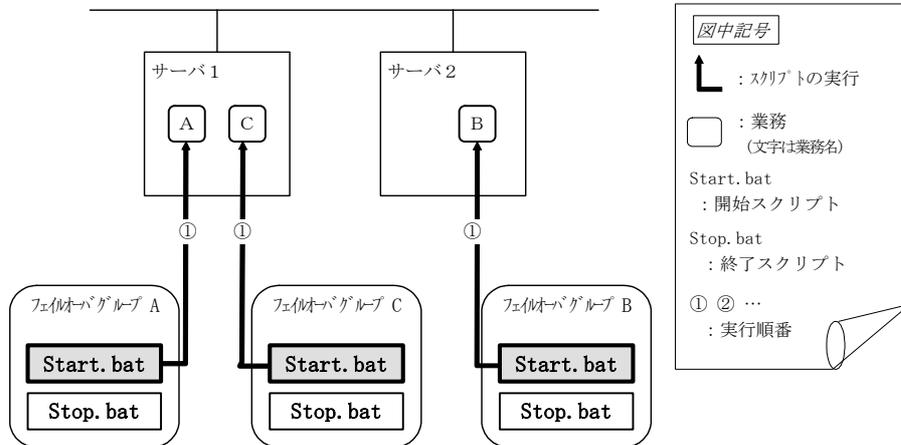
代表的なクラスタ状態遷移について説明します。



(1) 通常立ち上げ

ここで言う通常立ち上げとは、開始スクリプトがプライマリサーバで正常に実行され、直前の終了スクリプトもプライマリサーバで実行されていた時を指します。

各フェイルオーバーグループは、起動したサーバの中で、最もプライオリティの高いサーバ上で起動されます。

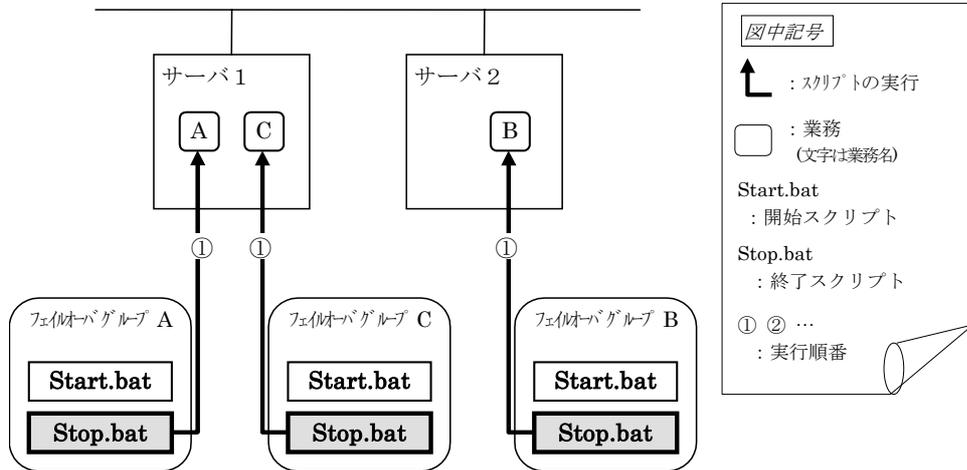


Start.batに対する環境変数

フェイルオーバーグループ	環境変数	値
A	ARMS_EVENT	START
	ARMS_SERVER	HOME
B	ARMS_EVENT	START
	ARMS_SERVER	HOME
C	ARMS_EVENT	START
	ARMS_SERVER	HOME

(2) 通常シャットダウン

ここでいう通常シャットダウンとは、終了スクリプトに対応する開始スクリプトが、通常立ち上げにより実行された、もしくはグループの移動（オンラインフェイルバック）により実行された直後の、クラスタシャットダウンを指します。



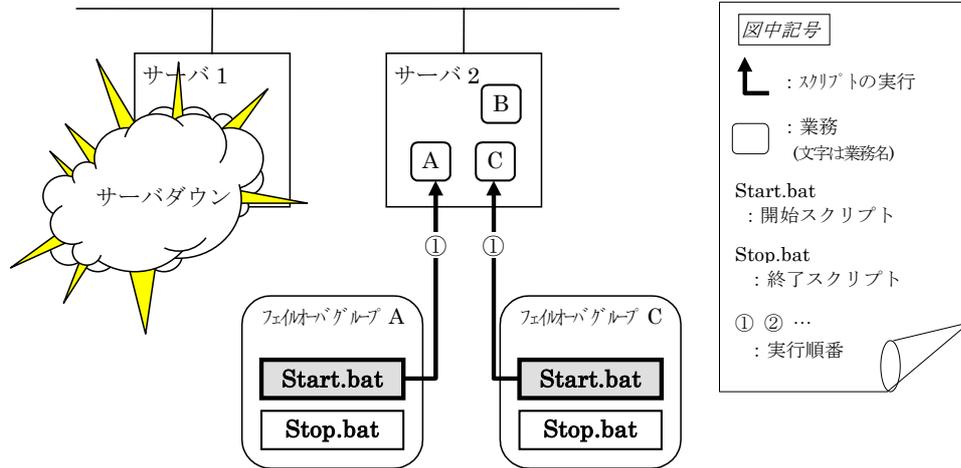
Stop.batに対する環境変数

フェイルオーバーグループ	環境変数	値
A	ARMS_EVENT	START
	ARMS_SERVER	HOME
B	ARMS_EVENT	START
	ARMS_SERVER	HOME
C	ARMS_EVENT	START
	ARMS_SERVER	HOME

(3) サーバ1ダウンによるフェイルオーバー

サーバ1をプライマリサーバとするフェイルオーバーグループの開始スクリプトが、障害発生により下位のプライオリティサーバ(サーバ2)で実行されます。開始スクリプトには、ARMS_EVENT(=FAILOVER)を分岐条件にして、業務の起動/復旧処理(たとえばデータベースのロールバック処理など)を記述しておく必要があります。

プライマリサーバ以外でのみ実行したい処理がある場合は、ARMS_SERVER(=OTHER)を分岐条件にして記述しておく必要があります。



Start.batに対する環境変数

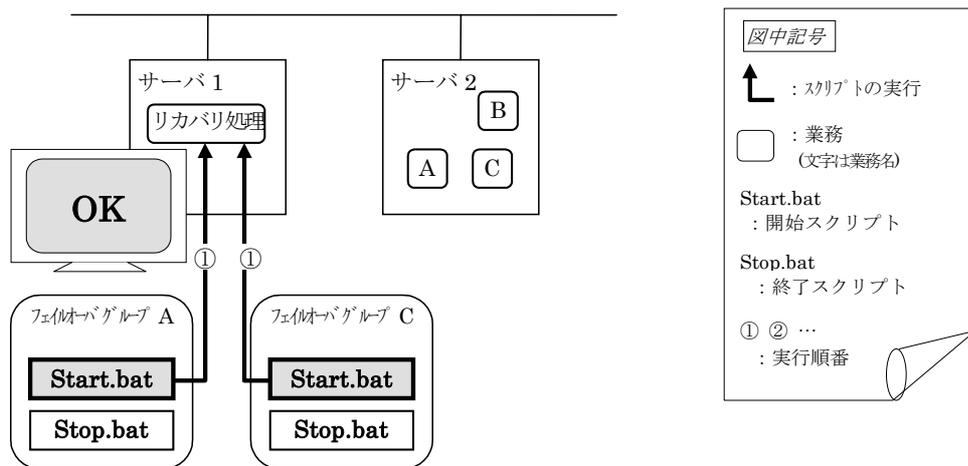
フェイルオーバーグループ	環境変数	値
A	ARMS_EVENT	FAILOVER
	ARMS_SERVER	OTHER
C	ARMS_EVENT	FAILOVER
	ARMS_SERVER	OTHER

(4) サーバ1のクラスタ復帰

ダウン後再起動状態(非クラスタとして動作中)のサーバ1をクラスタに復帰させる場合、サーバ1でフェイルオーバ発生時に動作していたフェイルオーバグループの、開始スクリプトが実行されます(フェイルオーバが発生したサーバでリカバリ処理が実行されます)。

リカバリ処理 (たとえば、ローカルディスクにあるデータベース情報などの修復) を行うために、ARMS_EVENT(=RECOVER)を分岐条件にして、処理を記述しておく必要があります(特にリカバリ処理が必要無い場合でも、業務の起動処理を実行しないように、スクリプトを記述してください)。

データミラーリングの運用の場合、クラスタ復帰にて、データの復旧(ミラーセットの再構築)を行います。

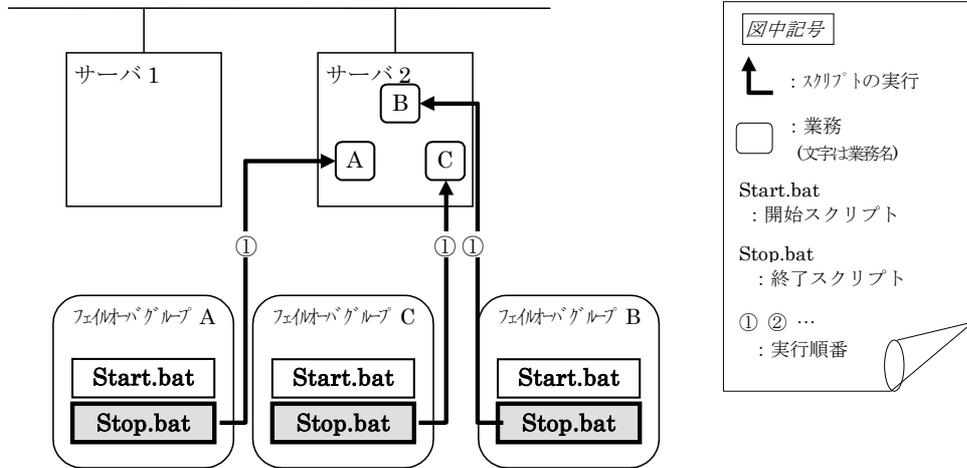


Start.batに対する環境変数

フェイルオーバーグループ	環境変数	値
A	ARMS_EVENT	RECOVER
	ARMS_SERVER	HOME
C	ARMS_EVENT	RECOVER
	ARMS_SERVER	HOME

(5) サーバ1フェイルオーバー後クラスタシャットダウン

フェイルオーバーグループAとCの終了スクリプトが、フェイルオーバー先のサーバ2で実行されます(フェイルオーバーグループBの終了スクリプトは、通常シャットダウンでの実行です)。

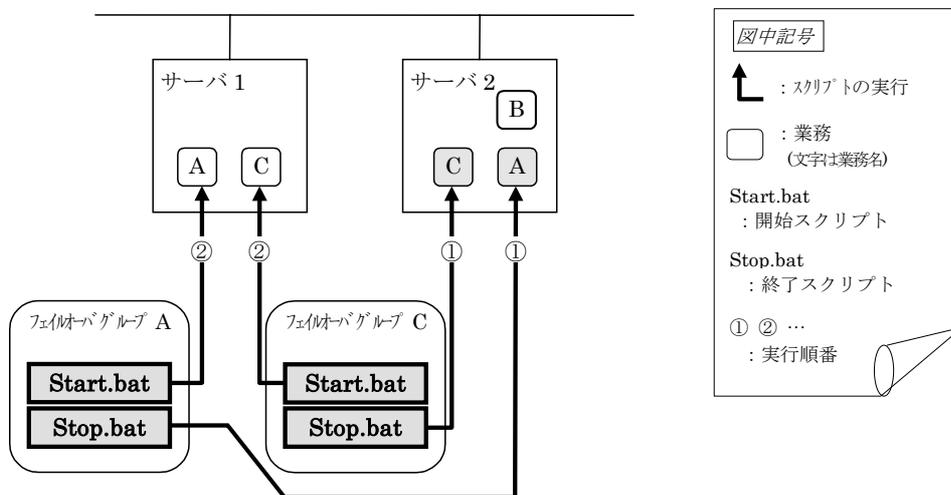


Stop.batに対する環境変数

フェイルオーバーグループ	環境変数	値
A	ARMS_EVENT	FAILOVER
	ARMS_SERVER	OTHER
B	ARMS_EVENT	START
	ARMS_SERVER	HOME
C	ARMS_EVENT	FAILOVER
	ARMS_SERVER	OTHER

(6) フェイルオーバーグループAとCの移動

フェイルオーバーグループAとCの終了スクリプトが、フェイルオーバー先のサーバ2で実行された後、サーバ1で開始スクリプトが実行されます。



Stop.batに対する環境変数

フェイルオーバーグループ	環境変数	値
A	ARMS_EVENT	FAILOVER2
	ARMS_SERVER	OTHER
C	ARMS_EVENT	FAILOVER
	ARMS_SERVER	OTHER

Start.batに対する環境変数

フェイルオーバーグループ	環境変数	値
A	ARMS_EVENT	START
	ARMS_SERVER	HOME
C	ARMS_EVENT	START
	ARMS_SERVER	HOME

² 終了スクリプトの環境変数の値は、直前に実行された開始スクリプトの環境変数の値になります。

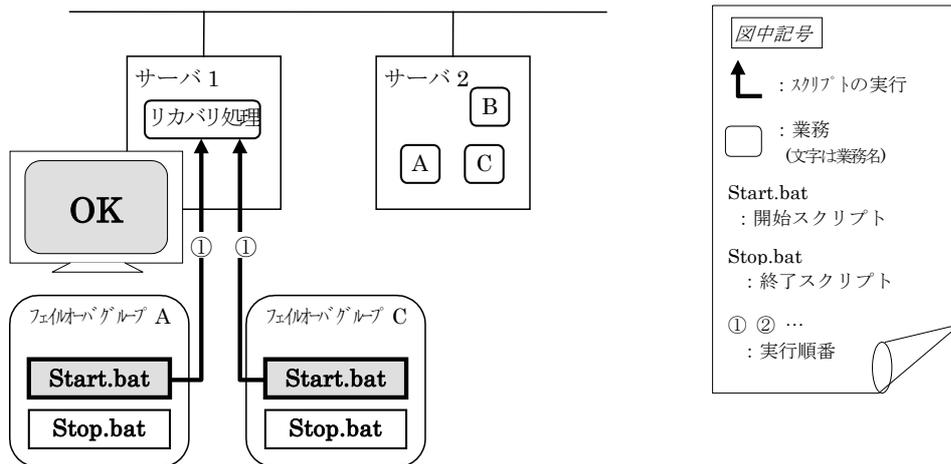
(6)の遷移の場合、直前にクラスタシャットダウンがないのでFAILOVERになりますが、(6)の前にクラスタシャットダウンが行われていると、STARTになります。

(7) サーバ1の起動(自動復帰モード)

サーバ1の自動復帰を実行します。サーバ1でフェイルオーバ発生時に動作していたフェイルオーバグループの、開始スクリプトが実行されます(フェイルオーバが発生したサーバでリカバリ処理が実行されます)。

「(4) サーバ1のクラスタ復帰」と同様の注意が必要です。

データミラーリング運用の場合、クラスタ復帰にて、データの復旧(ミラーセットの再構築)を行います。

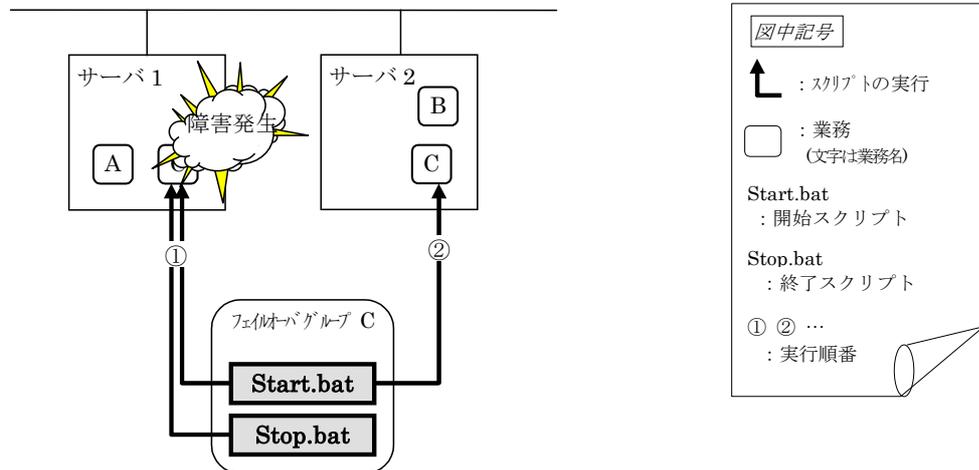


Start.batに対する環境変数

フェイルオーバーグループ	環境変数	値
A	ARMS_EVENT	RECOVER
	ARMS_SERVER	HOME
C	ARMS_EVENT	RECOVER
	ARMS_SERVER	HOME

(8) フェイルオーバーグループCの障害、フェイルオーバー

フェイルオーバーグループCに障害が発生すると、サーバ1でフェイルオーバーグループCの終了スクリプト、開始スクリプト(リカバリ処理)が実行され、サーバ2でフェイルオーバーグループCの開始スクリプトがで実行されます。フェイルオーバーグループCの開始スクリプトには、ARMS_EVENT(=RECOVER)とARMS_EVENT(=FAILOVER)を分岐条件にして、それぞれ業務の起動/復旧処理を記述しておく必要があります。



サーバ1のStop.bat

フェイルオーバーグループ	環境変数	値
C	ARMS_EVENT	START
	ARMS_SERVER	HOME

サーバ1のStart.bat

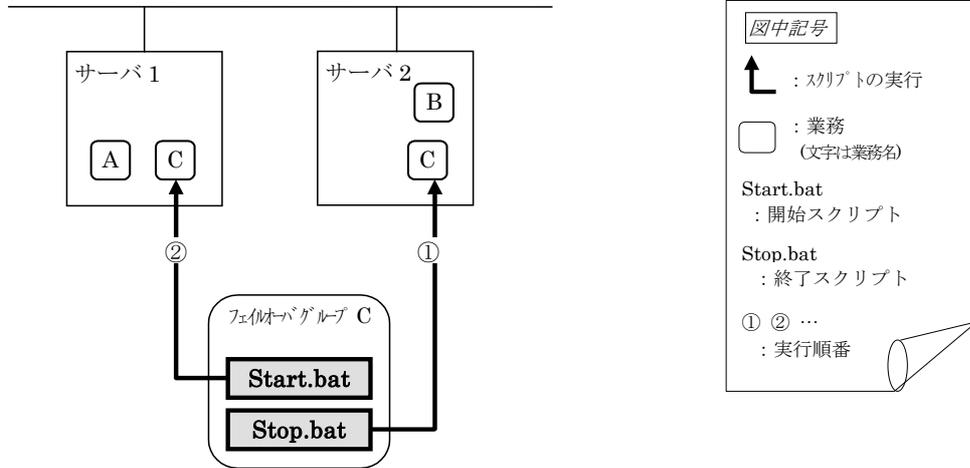
フェイルオーバーグループ	環境変数	値
C	ARMS_EVENT	RECOVER
	ARMS_SERVER	HOME

サーバ2のStart.bat

フェイルオーバーグループ	環境変数	値
C	ARMS_EVENT	FAILOVER
	ARMS_SERVER	OTHER

(9) フェイルオーバーグループCの移動

(8)でサーバ2にフェイルオーバーしてきたフェイルオーバーグループCを、サーバ2よりサーバ1へ移動します。サーバ2で終了スクリプトを実行した後、サーバ1で開始スクリプトを実行します。



Stop.bat((8)よりフェイルオーバーしてきたため)

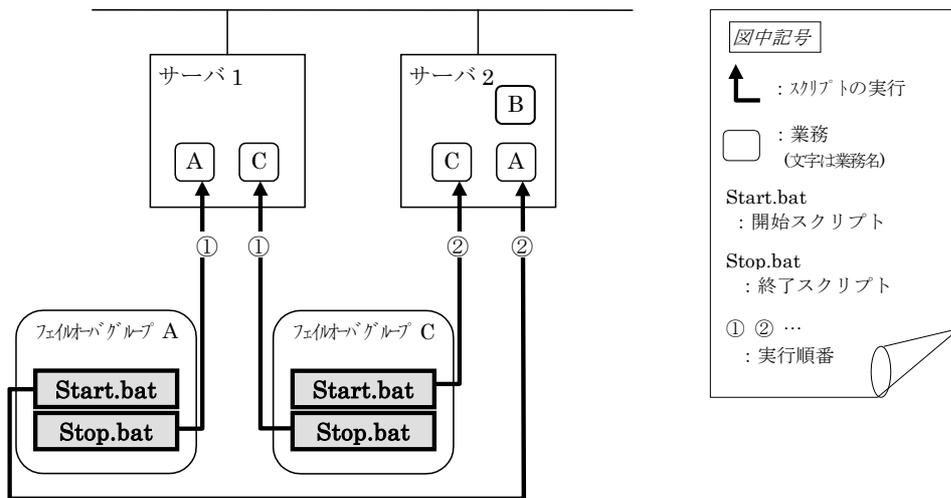
フェイルオーバーグループ	環境変数	値
C	ARMS_EVENT	FAILOVER
	ARMS_SERVER	OTHER

Start.bat

フェイルオーバーグループ	環境変数	値
C	ARMS_EVENT	START
	ARMS_SERVER	HOME

(10) サーバ1のサーバ切り離し

サーバ1をクラスタシステムより切り離します。



Stop.bat

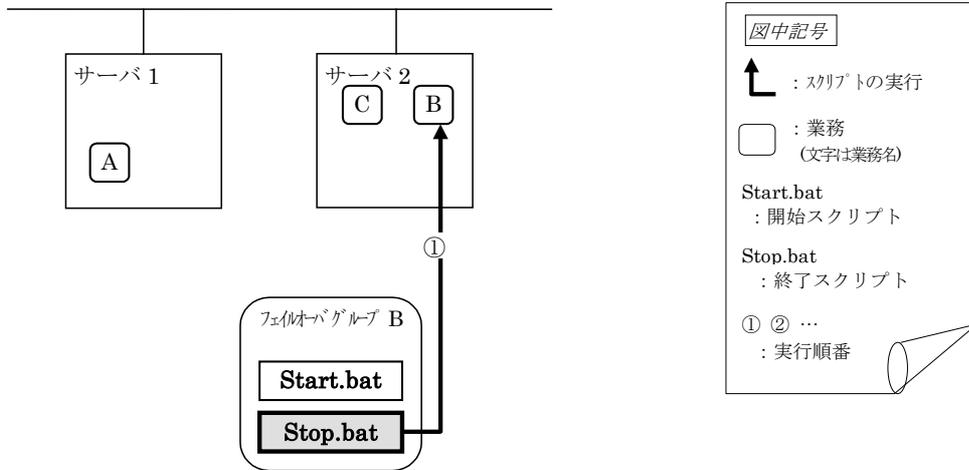
フェイルオーバーグループ	環境変数	値
A	ARMS_EVENT	START
	ARMS_SERVER	HOME
C	ARMS_EVENT	START
	ARMS_SERVER	HOME

Start.bat

フェイルオーバーグループ	環境変数	値
A	ARMS_EVENT	FAILOVER
	ARMS_SERVER	OTHER
C	ARMS_EVENT	FAILOVER
	ARMS_SERVER	OTHER

(11) フェイルオーバーグループBの停止

フェイルオーバーグループBの終了スクリプトがサーバ2で実行されます。

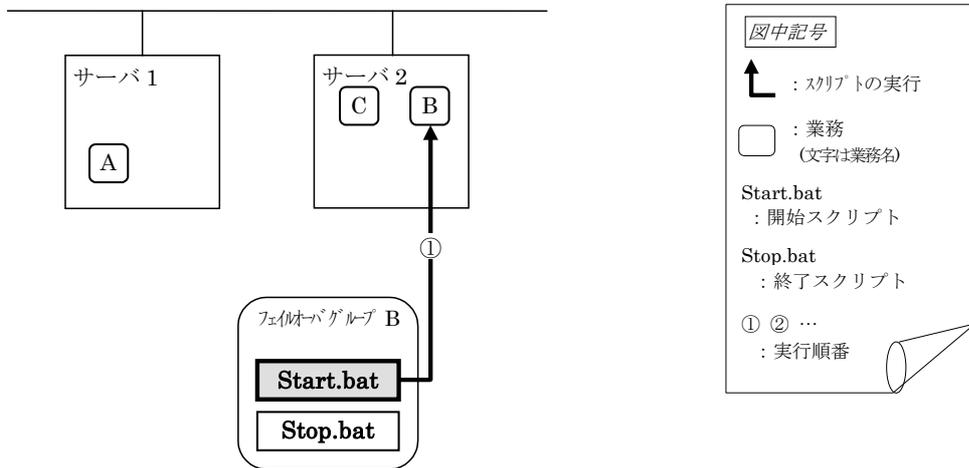


Stop.bat

フェイルオーバーグループ	環境変数	値
B	ARMS_EVENT	START
	ARMS_SERVER	HOME

(12) フェイルオーバーグループBの起動

フェイルオーバーグループBの開始スクリプトがサーバ2で実行されます。

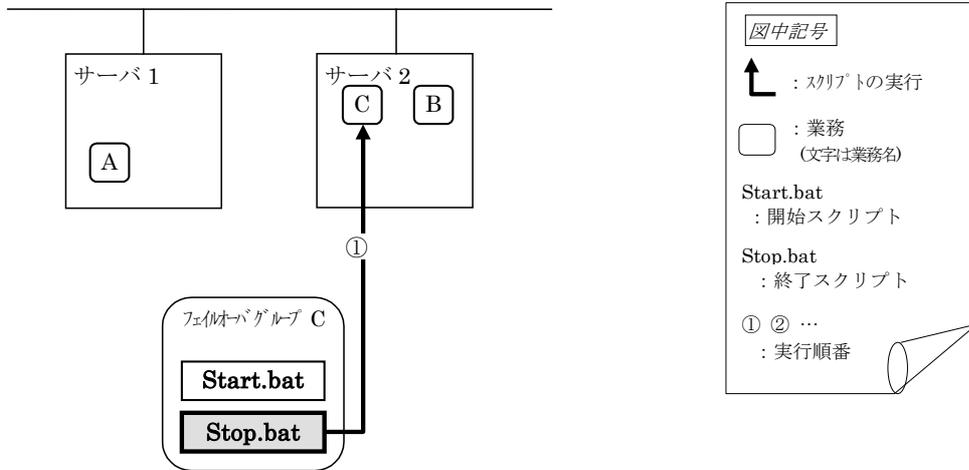


Start.bat

フェイルオーバーグループ	環境変数	値
B	ARMS_EVENT	START
	ARMS_SERVER	HOME

(13) フェイルオーバーグループCの停止

フェイルオーバーグループCの終了スクリプトが、サーバ2で実行されます。



図中記号

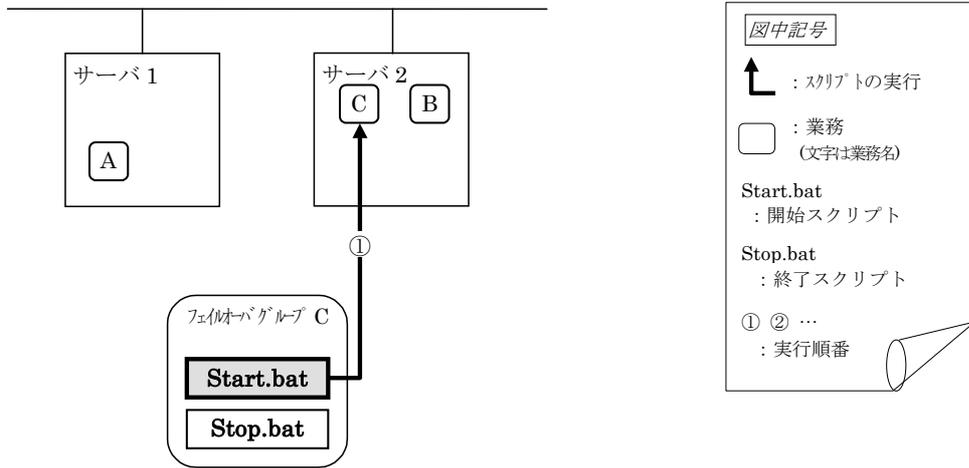
- ↑ : スクリプトの実行
- : 業務 (文字は業務名)
- Start.bat : 開始スクリプト
- Stop.bat : 終了スクリプト
- ① ② ... : 実行順番

Stop.bat

フェイルオーバーグループ	環境変数	値
C	ARMS_EVENT	FAILOVER
	ARMS_SERVER	OTHER

(14) フェイルオーバーグループCの起動

フェイルオーバーグループCの開始スクリプトがサーバ2で実行されます。



図中記号

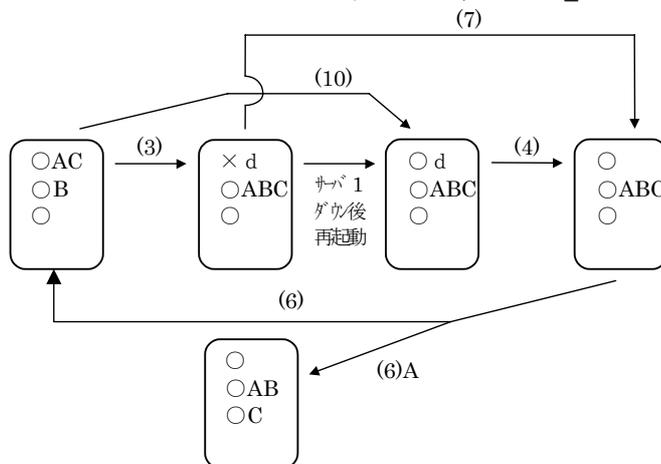
- ↑ : スクリプトの実行
- : 業務 (文字は業務名)
- Start.bat : 開始スクリプト
- Stop.bat : 終了スクリプト
- ① ② ... : 実行順番

Start.bat

フェイルオーバーグループ	環境変数	値
C	ARMS_EVENT	START
	ARMS_SERVER	OTHER

【補足1】

* フェイルオーバーポリシーに設定されているサーバを3つ以上持つフェイルオーバーグループにおいて、プライマリサーバ以外のサーバで、異なった動作を行なう場合 ARMS_SERVER(HOME/OTHER)の代わりに、ARMS_PRIORITY(1~16)を使用する

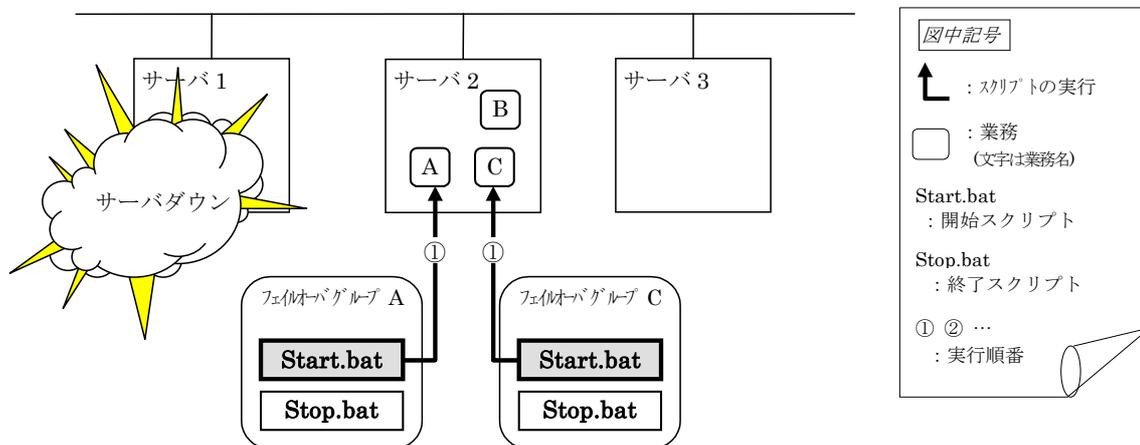


(例1)クラスタ状態遷移図(3)の場合

(3)サーバ1ダウンによるフェイルオーバー

サーバ1をプライマリサーバとするフェイルオーバーグループの開始スクリプトが、障害発生により次に高いフェイルオーバーポリシーを持つサーバ2で実行されます。開始スクリプトには、ARMS_EVENT(=FAILOVER)を分岐条件にして、業務の起動/復旧処理(たとえばデータベースのロールバック処理など)を記述しておく必要があります。

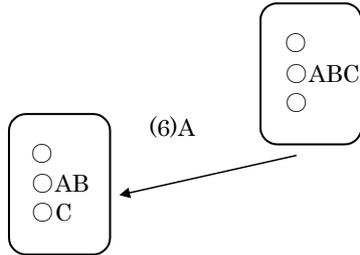
2番目に高いフェイルオーバーポリシーを持つサーバのみで実行したい処理がある場合は、ARMS_PRIORITY(=2)を分岐条件にして記述しておく必要があります。



Start.batに対する環境変数

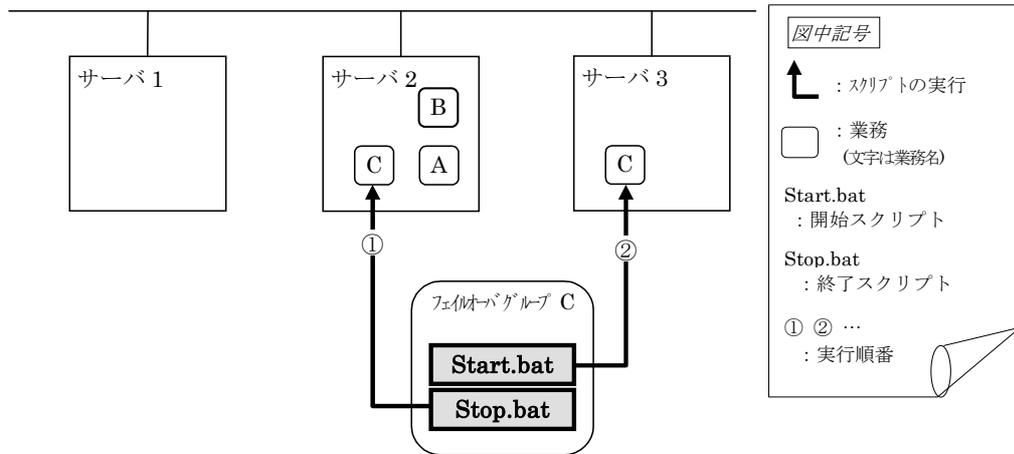
フェイルオーバーグループ	環境変数	値
A	ARMS_EVENT	FAILOVER
	ARMS_SERVER	OTHER
	ARMS_PRIORITY	2
C	ARMS_EVENT	FAILOVER
	ARMS_SERVER	OTHER
	ARMS_PRIORITY	2

(例2)クラスタ状態遷移図(6)Aの場合



(6)AフェイルオーバーグループCの移動

フェイルオーバーグループCの終了スクリプトが、フェイルオーバー先のサーバ2で実行された後、サーバ3で開始スクリプトが実行されます。



Stop.batに対する環境変数

フェイルオーバーグループ	環境変数	値
C	ARMS_EVENT	FAILOVER
	ARMS_SERVER	OTHER
	ARMS_PRIORITY	2

Start.batに対する環境変数

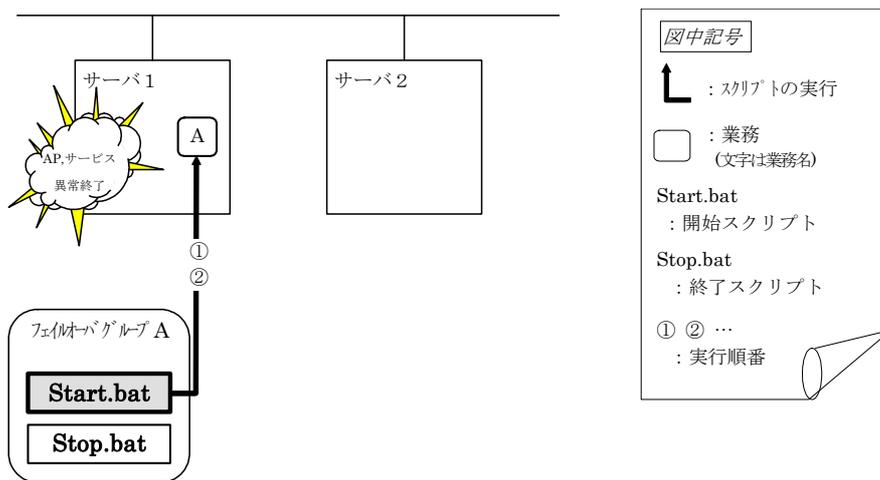
フェイルオーバーグループ	環境変数	値
C	ARMS_EVENT	START
	ARMS_SERVER	OTHER
	ARMS_PRIORITY	3

【補足2】

armloadコマンドがスクリプトを(再)起動する場合

armloadコマンドがアプリケーションの異常を検出し開始スクリプトを(再)起動する場合の環境変数は以下のようになります。

armloadコマンドがサーバ1で起動していたアプリケーションの異常終了を検出してサーバ1でフェイルオーバーグループAの再起動を行う場合



Stop.batに対する環境変数

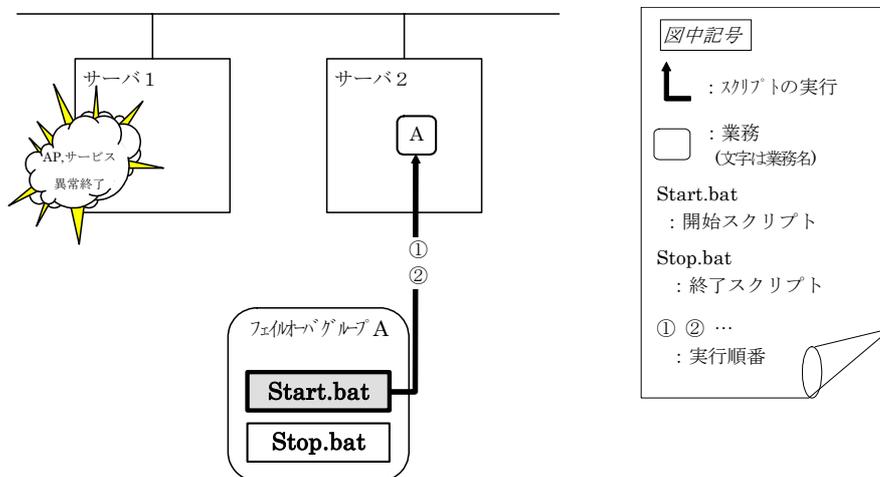
フェイルオーバーグループ	環境変数	値
A	①	ARMS_EVENT Start.bat実行時と同一の値

Start.batに対する環境変数

フェイルオーバーグループ	環境変数	値
A	②	ARMS_EVENT RECOVER
	③	ARMS_EVENT START

Start.batは 2回実行されます。

armloadコマンドがサーバ1で起動していたアプリケーションの異常終了を検出してサーバ2へフェイルオーバーをしてサーバ2でフェイルオーバーグループAの起動を行う場合



Stop.batに対する環境変数

フェイルオーバーグループ		環境変数	値
A	①	ARMS_EVENT	Start.bat実行時と同一の値

Start.batに対する環境変数

フェイルオーバーグループ		環境変数	値
A	②	ARMS_EVENT	RECOVER
	③	ARMS_EVENT	FAILOVER

Start.batは 2回実行されます。

1.2.3.2 スクリプト記述の流れ

前節の、スクリプトの実行タイミングと実際のスクリプト記述を関連付けて説明します。

(1) フェイルオーバーグループA開始スクリプト： start.batの一例

```
#!/bin/sh
# *****
# *          START.BAT          *
# *****

if [ "$ARMS_EVENT" = "START" ]
then
    if [ "$ARMS_DISK" = "SUCCESS" ]
    then
        処理概要：
        業務の通常起動処理
        この処理を行う実行タイミング：
        (1) 通常立ち上げ
        (6) フェイルオーバーグループの移動(オンラインフェイルバック)

    if [ "$ARMS_SERVER" = "HOME" ]
    then
        実行サーバ環境変数を参照して、処理の振り分けを行う。

        処理概要：
        プライマリサーバで、業務が通常起動される場合のみ行わない処理
        この処理を行う実行タイミング：
        (1) 通常立ち上げ
        (6) フェイルオーバーグループの移動(オンラインフェイルバック)

    else

        処理概要：
        プライマリサーバ以外で、業務が通常起動される場合のみ行わない処理
        この処理を行う実行タイミング：

        fi

    else
        ディスク関連エラー処理

    fi

elif [ "$ARMS_EVENT" = "RECOVER" ]
then
    処理概要：
    クラスタ復帰後のリカバリ処理
    この処理を行う実行タイミング：
    (4) クラスタ復帰

elif [ "$ARMS_EVENT" = "FAILOVER" ]
then
```

```
if [ "$ARMS_DISK" = "SUCCESS" ]  
then
```

DISK 接続情報環境変数を参照して、エラー処理を行う。

処理概要：
業務の通常起動処理
この処理を行う実行タイミング：
(3) サーバ1 ダウンによるフェイルオーバー
(10)サーバ1 のサーバ切り離し

```
if [ "$ARMS_SERVER" = "HOME" ]  
then
```

実行サーバ環境変数を参照して、処理の振り分けを行う。

処理概要：
フェイルオーバー後、プライマリサーバで業務が終了される場合のみ行いたい処理
この処理を行う実行タイミング：

```
else
```

処理概要：
フェイルオーバー後、非プライマリサーバで業務が起動される場合のみ行いたい処理
この処理を行う実行タイミング：
(3) サーバ1 ダウンによるフェイルオーバー
(10)サーバ1 のサーバ切り離し

```
fi
```

```
else
```

ディスク関連エラー処理

```
fi
```

```
else
```

```
#NO_ARM
```

CLUSTERPROは動作していない

```
fi
```

```
#EXIT
```

```
exit 0
```

(2) ファイルオーバグループA終了スクリプト： Stop.batの一例

```
#!/bin/sh
# *****
# *          STOP.BAT          *
# *****
arm_rel_path() {
    while [ "$1" != "" ]
    do
        armrelpath $1 > /dev/null 2>&1
        relret=$?

        if [ "$relret" = "0" ]
        then
            armlog "KILL NO PROCESS"
        elif [ "$relret" = "1" ]
        then
            armlog "KILL SOME PROCESS"
        else
            armlog "ARMRELPATH ERROR"
        fi

        shift
    done
}

arm_rel_mntpoint() {
    grp=$ARMS_GROUPNAME
    mntpoint=`armlsmnt $grp 2>&`
    mntret=$?

    if [ "$mntret" = "0" ]
    then
        if [ "$mntpoint" != "" ]
        then
            arm_rel_path $mntpoint
        else
            armlog "NO MOUNT POINT"
        fi
    else
        armlog "ARMLSMNT ERROR"
    fi
}

if [ "$ARMS_EVENT" = "START" ]
then
    if [ "$ARMS_DISK" = "SUCCESS" ]
    then
        if [ "$ARMS_SERVER" = "HOME" ]
        then
            armlog "ARMRELPATH ERROR"
        fi
    fi
fi
```

ディスクアクセスプロセス強制停止処理

スクリプト実行要因の環境変数を参照して、処理の振り分けを行う。

DISK 接続情報環境変数を参照して、エラー後処理を行う。

処理概要：
 業務の通常終了処理
 この処理を行う実行タイミング：
 (2) 通常シャットダウン
 (10)サーバ の切り離し

実行サーバ環境変数を参照して、処理の振り分けを行う。

処理概要：
 プライソリサーバ上で業務が通常処理される場合のみ行わない処理
 この処理を行う実行タイミング：
 (2) 通常シャットダウン
 (10)サーバ1 の切り離し

```

else
    処理概要：
    プライマリサーバ以外で、業務が通常終了される場合のみ行いたい処理
    この処理を行う実行タイミング：

fi
arm_rel_mntpoint
else
    ディスク関連エラー処理

fi
elif [ "$ARMS_EVENT" = "FAILOVER" ]
then
    if [ "$ARMS_DISK" = "SUCCESS" ]
    then
        処理概要：
        フェイルオーバー後、通常終了処理
        この処理を行う実行タイミング：
        (5) サーバ1 フェイルオーバー後クラスタシャットダウン
        (6) ファイルオーバーグループ AC の移動

        実行サーバ環境変数を参照して、処理の振り分けを行う。

    fi
    if [ "$ARMS_SERVER" = "HOME" ]
    then
        処理概要：
        フェイルオーバー後、プライマリサーバで業務が終了される場合のみ行いたい処理
        この処理を行う実行タイミング：

    else
        処理概要：
        フェイルオーバー後、非プライマリサーバで業務が終了される場合のみ行いたい処理
        この処理を行う実行タイミング：
        (5) サーバ1 フェイルオーバー後クラスタシャットダウン
        (6) ファイルオーバーグループ AC の移動

    fi
    arm_rel_mntpoint
else
    ディスク関連エラー処理

fi
else
    #NO_ARM
    CLUSTERPROは動作していない

fi
#EXIT
exit 0

```

1.2.4 スクリプト作成のヒント

以下の点に注意して、スクリプトを作成してください。

- * スクリプト中にて、プログラムプロダクトの起動/終了を行う場合には、PPの起動/終了が完了したことを示すログを出力するようにしてください。ログはarmlogコマンドにて出力することができます。
この情報は、問題発生時、障害の切り分けを行う場合に使用することができます。

1.2.5 ディスクアクセス中のプロセス強制停止機能

デフォルトのstopスクリプト(stop.bat)³は、フェイルオーバーグループのディスクリソースのマウントポイントにアクセスしているプロセスを強制的にkillする機能が動作するような記述となっています。

この機能が不要ない場合にはデフォルトのstopスクリプト中の“arm_rel_mntpoint”をcallしている部分を削除するか“#”などでコメントアウトしてください。

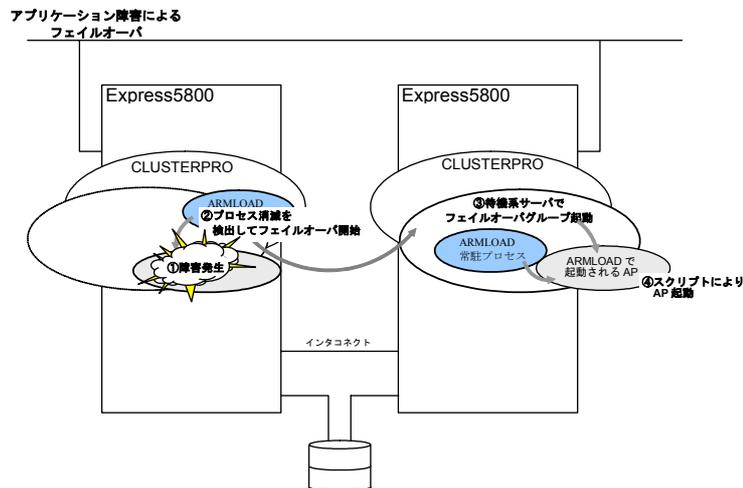
stopスクリプトを編集する場合には、“arm_rel_mntpoint”をcallする前にカレントディレクトリが共有ディスクのマウントポイントとならないようにしてください。

³ CPRO-LX010-06以降のアップデートを適用した後にグループを作成した場合、Ver.1.0.2.0以降のトレッキングツールを使用した場合に有効です。

2 高度なシステムの設計

2.1 アプリケーション監視(armload)

フェイルオーバーグループのスタートアップから業務アプリケーションを起動するとき、armload コマンドを使用して起動することにより、起動したアプリケーションの死活監視を行うことが可能です。



- * 通常運用時
armloadによって起動したアプリケーションのプロセスの存在により死活監視を行います。
- * アプリケーション異常発生時
アプリケーションに異常が発生するとarmloadのパラメータ指定に従って、armload常駐プロセスが異常を検出し、フェイルオーバーグループを次のプライオリティのサーバへ移動します。

2.1.1 監視対象となる障害

監視対象となる障害は以下のものです。

- * armloadが直接起動したプロセスの消滅

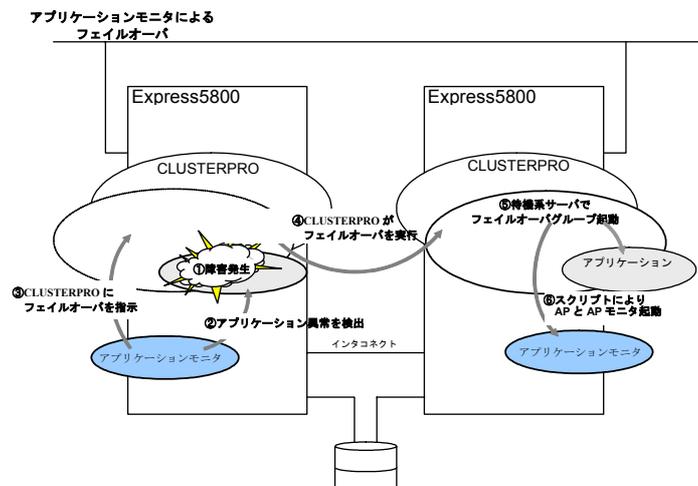
なお、アプリケーションの挙動（ストール等）は、監視できません。（「2.2 アプリケーションモニタ」を参照）

2.1.2 環境設定

フェイルオーバーグループの開始スクリプトにarmloadコマンドで業務APを起動するように記述します。また、停止スクリプトにarmkillコマンドで業務APを終了するように記述します。

armloadコマンドの詳細は、「システム構築ガイド コマンドリファレンス」を参照してください。

2.2 アプリケーションモニタ



armloadはarmloadが直接起動したプロセスの死活監視のみを行います。アプリケーションごとの特性に応じたアプリケーションモニタを作成することで、アプリケーションのストールなどを監視することができます。

- * アプリケーションモニタによるフェイルオーバー
次のCLUSTERPROコマンドを利用して、アプリケーションモニタはグループのフェイルオーバーを実行することができます。
- + armdown
サーバをシャットダウンさせることでフェイルオーバーを実行します。
- + armfover
グループ名を指定することで、フェイルオーバーやグループの移動を実行します。
グループ名はスクリプトの環境変数ARMS_GROUPNAMEで引き取ることができます。

コマンドの詳細は、「システム構築ガイド コマンドリファレンス」を参照してください。

- * アプリケーション監視
監視方法はアプリケーションごとに異なりますが、例えば次のような方法で監視を行います。
- + ストール監視
監視対象がデータベース(DBMS)の場合、定期的にデータベース上に存在するテーブルを照会し、応答がない場合に異常と判断し、フェイルオーバーを実行します。
Webサーバの場合は、定期的にWebアクセスすることでストール監視を行います。
- + その他の監視
アプリケーションが異常状態になる時のシステム内の異常要素(例えばメモリ不足など)が分かる場合、システム内の異常要素を監視することで、フェイルオーバーを実行します。

3 マスタクラスタ

3.1 スーパークラスタ

スーパークラスタは、CLUSTERPROで管理を行う最上位の単位です。スーパークラスタは、フェイルオーバー型クラスタを束ねた構成となります。

スーパークラスタの配下には、次のものを作ることができます。

- * サブクラスタ
フェイルオーバー型クラスタを構成します。

3.2 マスタクラスタ

スーパークラスタを構成するには、必ず一つのマスタクラスタが必要となります。

マスタクラスタを構成するサーバのうちの1台が、スーパークラスタ管理サーバとなり、スーパークラスタ内の各クラスタの構成情報を一元管理します。

マスタクラスタ以外のサーバは、起動時にスーパークラスタ管理サーバにアクセスし、自クラスタ(または自サーバ)の構成情報をダウンロードしたあとで、クラスタとしての動作を開始します。

マスタクラスタを構成する全サーバにスーパークラスタの各クラスタの構成情報のコピーが存在しているので、スーパークラスタ管理サーバがダウンした場合、マスタクラスタ内で正常に動作しているサーバが自動的にスーパークラスタ管理サーバとなります。

マスタクラスタ内の全サーバがダウンした場合は、クラスタ内のサーバを、強制復帰・復帰で正常状態に戻してください。

強制復帰により、正常状態に戻らない場合には、次の方法により復旧して下さい。

CLUSTERPROマネージャから手動で別のフェイルオーバー型クラスタを、新しいマスタクラスタとして指定し、スーパークラスタとしての運用を継続します。

詳細は、「システム構築ガイド 運用/保守編」または「システム構築ガイド GUIリファレンス」を参照してください。

スーパークラスタ管理サーバがダウンまたは未起動状態のとき、スーパークラスタは動作しますが、構成変更を行うことはできません。