CLUSTERPRO for Linux Ver3.0

リソース詳細編

2006.09.29
第10版

改版履歴

版数	改版日付	内 容
1	2004/07/30	3.x用初版新規作成
2	2004/09/30	全体的に章の構成を変更 1.1 グループの説明を追加 1.2 execリソース追加 1.3 ディスクリソース追加 1.4 フローティングIPリソース追加 1.5 ミラーディスクリソース追加 1.7 VxVMに関する運用保守を「メンテナンス編」へ移動 1.8 NASリソースを追加 2.1 モニタリソースの説明を追加 2.1.1 監視タイミングの説明を追加 2.1.6 監視遅延警告の説明を追加 2.1.7 監視開始待ちの説明を追加 2.2 ディスクモニタリソース追加 2.4 IPモニタリソース追加 2.5 NIC Link Up/Downモニタ追加 5.1 ミラーコネクトのbonding化を追加 4 付録を追加
3	2004/11/30	2.3.2 RAWモニタリソースに関する注意事項に2.6系kernelでの注意事項を追加 2.3.4 2.6系kernelでのRAWモニタリソースの設定例を追加
4	2004/12/17	1, 1.2.1, 1.4.1, 1.6.1, 1.8.1, 2.1.1, 2.2.1, 2.3.1, 2.4.1, 2.5.1.1, 2.8.1, 2.9.1, 5.1.1.1.1 にSXに対応 1.2.1, 1.3.1, 1.4.1, 1.6.1, 1.8.1, 2.1.1, 2.1.6, 2.1.7, 2.2.1, 2.3.1, 2.4.1, 2.5.1.1, 2.6.1, 2.7.1, 2.8.1, 2.9.1, 2.10.1, 2.11.1, 5.1.1.1.1 トレッキングツールのバージョン情報修正 2.6 ミラーディスクコネクトモニタリソースを追加 2.7 ミラーディスクモニタリソースを追加 2.8 PIDモニタリソースを追加 2.9 ユーザ空間モニタリソースを追加 2.10 VxVMデーモンモニタリソースを追加 2.11 VxVMボリュームモニタリソースを追加
5	2005/03/31	1 グループリソースにグループリソース一覧を追加 1.1.5 活性異常、非活性異常検出 を追加 1.1.6 再起動回数制限 を追加 1.4.6 フローティングIPリソースに関する注意事項 を追加 1.5 ミラーディスクリソースを修正 2 モニタリソース にモニタリソース一覧を追加 2.1.3 異常検出 に回復動作を実行する条件を追加 回復動作手順を変更 2.1.4 監視異常からの復帰(正常) を追加 2.1.5 回復動作時の回復対象活性/非活性異常 を追加 2.1.8 再起動回数制限を追加 2.5 NIC Link Up/Downモニタリソース 動作確認済ディスとリビーション及びNICドライバを更新 2.9.3 依存するrpm を追加

		2.9.4 監視方法 を追加 2.9.5 監視の拡張設定を追加 2.9.6 監視ロジックを追加 2.9.9.3 ipmiによる監視の注意事項を追加 2.9.7 ipmi動作可否の確認追加 2.9.8 ipmiコマンド追加 3 ハートビートリソースを追加 4 シャットダウンストール監視を追加
6	2005/04/08	XE 3.1-4に対応
7	2005/06/30	1.5.3 ミラー用のパーティションをOSと同じディスクに確保するための仕様拡張。 2.2.2 監視方法 にTUR(SG_IO)に関する記述を追加
8	2005/10/31	1 グループリソース の全グループリソースの章にグループリソースの依存関係の規定値を追加 1.2.2 1.3.2 1.4.2 1.5.2 1.6.2 1.7.2 1.8.2 1.1.7 再起動回数初期化 を追加 1.3 ディスクリソース を更新 1.4 フローティングIPリソース を更新 1.5.3 ミラーディスク の構成変更 1.5.4 ミラーパラメータ を追加 1.7.4 VxVMディスクグループリソース を追加 1.7.5 VxVMボリュームリソース を追加 2.1 モニタリソース にマルチターゲットモニタリソースに関する記述を追加 2.1.2 監視インターバル を追加 2.1.9 監視プライオリティ を追加 2.9 ユーザ空間モニタリソース にkeepaliveに関する記述を追加 2.12 マルチターゲットモニタリソース に関する記述を追加 3 ハートビートリソースにカーネルモードLANハートビートを追加
9	2006/03/31	1.2.3 スクリプトに関する注意事項を追加 1.3.5 ディスクリソースの注意事項を追加 1.4.4 サーバ別フローティングIPアドレスの接続に関する記述を追加 1.4.5 フローティングIPアドレス非活性待ち合わせ処理に関する項目を追加 1.4.6(3) フローティングIPアドレス 注意事項を追加 1.5.5 ミラーディスクリソースの注意事項を追加 1.7.5 VxVMボリュームリソースの注意事項を追加 1.8.4 NASリソースの注意事項を追加 2.1.6 ユーザ空間モニタリソースに関する記述を追加 2.3.2 RAWモニタリソースの注意事項を追加 2.9.9.1 ユーザ空間モニタリソースの注意事項を追加 2.9.9.2 softdog による監視の注意事項に関する項目を追加 2.9.9.3 ipmi による監視の注意事項を追加 2.9.9.4 keepalive による監視の注意事項に関する項目を追加
10	2006/09/29	1.2.3 スクリプトに関する注意事項を更新

最新の動作確認情報、システム構築ガイド、アップデート、トレッキングツールなどは
以下のURLに掲載されています。
システム構築前に最新版をお取り寄せください。

NECインターネット内でのご利用
　　http://soreike.wsd.mt.nec.co.jp/
　　[クラスタシステム]→[技術情報]→[CLUSTERPROインフォメーション]

NECインターネット外でのご利用
　　http://www.ace.comp.nec.co.jp/CLUSTERPRO/
　　[ダウンロード]→[Linuxに関するもの]→[ツール]

1 グループリソース

各グループに登録することができるグループリソース数はエディションとバージョンにより異なります。登録可能なグループリソース数は以下の表を参照してください。

エディション	バージョン	グループリソース数 （1 グループあたり）
SE	3.0-1 ～ 3.0-3	16
SE	3.0-4 以降	128
LE	3.0-1 以降	16
XE	3.0-1 以降	128
SX	3.1-2 以降	128

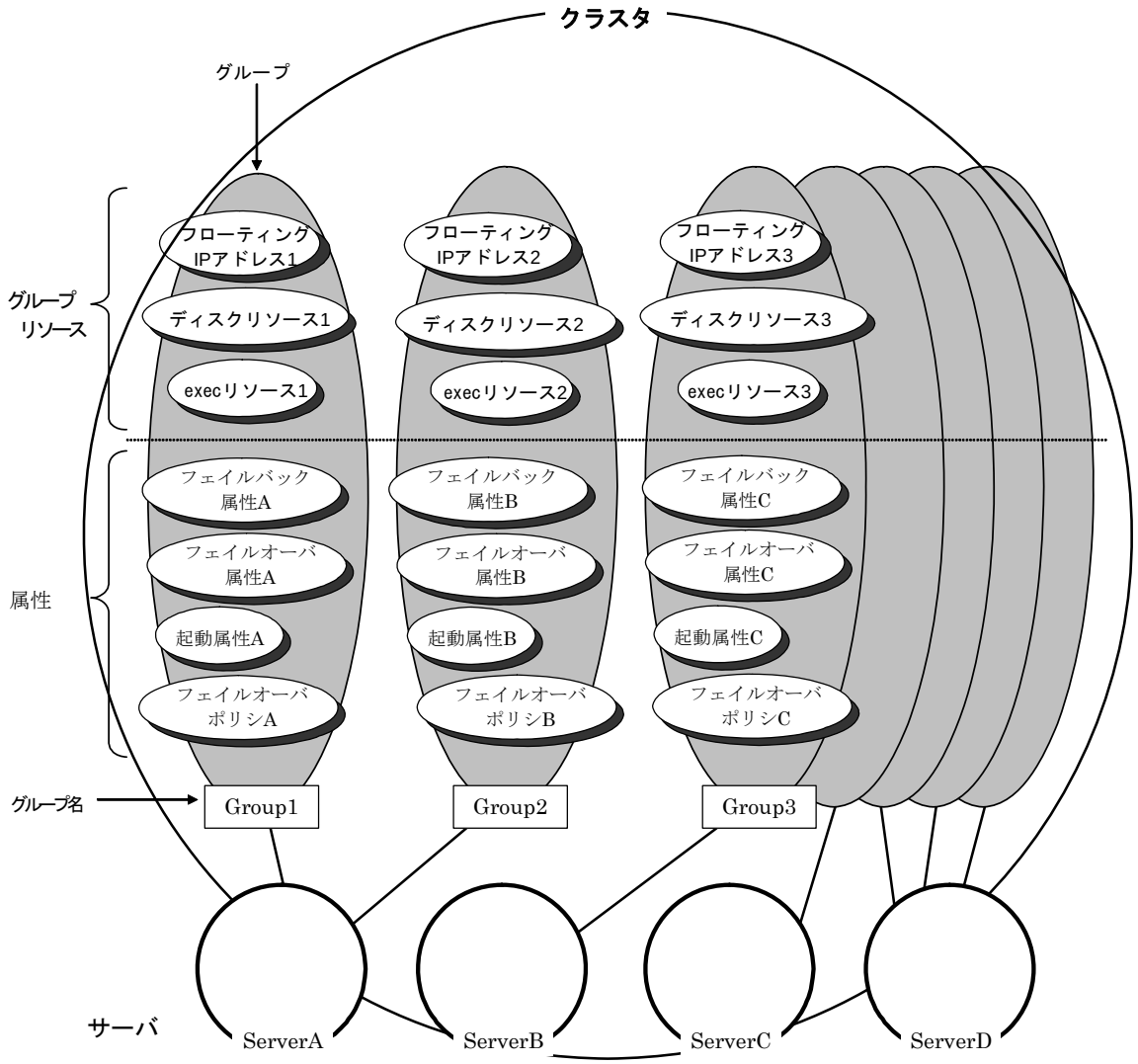
現在サポートされているグループリソースは以下です。

グループリソース名	略称	機能概要
execリソース	exec	1.2 execリソース を参照
ディスクリソース	disk	1.3 ディスクリソース を参照
フローティングIPリソース	fip	1.4 フローティングIPリソース を参照
ミラーディスクリソース	md	1.5 ミラーディスクリソース を参照
RAWリソース	raw	1.6 RAWリソース を参照
VxVMディスクグループリソース	vxdg	1.7 VxVM関連リソース を参照
VxVMボリュームリソース	vxvol	
NASリソース	nas	1.8 NASリソース を参照

1.1 グループ

グループとは、クラスタシステム内のある1つの独立した業務を実行するために必要な資源の集まりのことで、フェイルオーバを行なう単位になります。

グループは、グループ名、グループリソース、属性を持ちます。



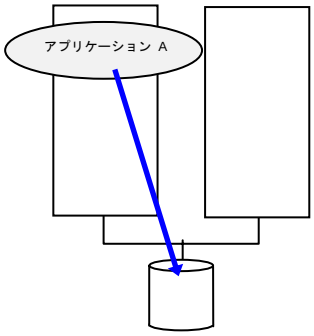
各グループのリソースは、それぞれひとまとまりのグループとして処理されます。すなわち 、ディスクリソース1とフローティングIPアドレス1を持つGroup1においてフェイルオーバが発生した場合、ディスクリソース1とフローティングIPアドレス1がフェイルオーバすることになります(ディスクリソース1のみが、フェイルオーバすることはありません)。

また、ディスクリソース1は、他のグループ(たとえばGroup2)に含まれることはありません。

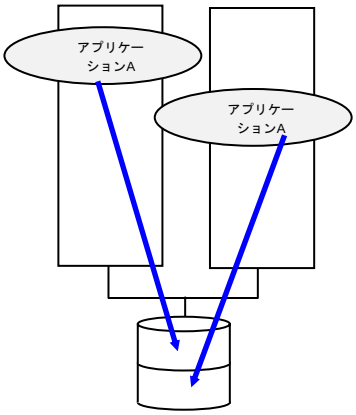
1.1.1 運用形態

CLUSTERPROでは、以下の運用形態をサポートしています。

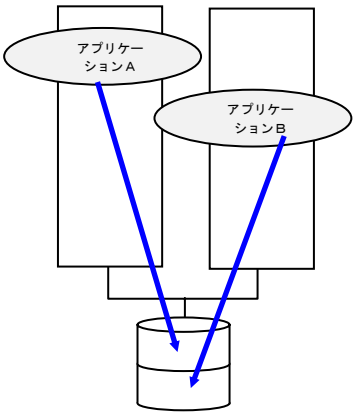
* 片方向スタンバイクラスタ
  クラスタシステム全体で同一の業務アプリケーションが1つしか動作しないシステム形態



* 同一アプリケーション双方向スタンバイクラスタ
  クラスタシステム全体で同一の業務アプリケーションが複数動作するシステム形態

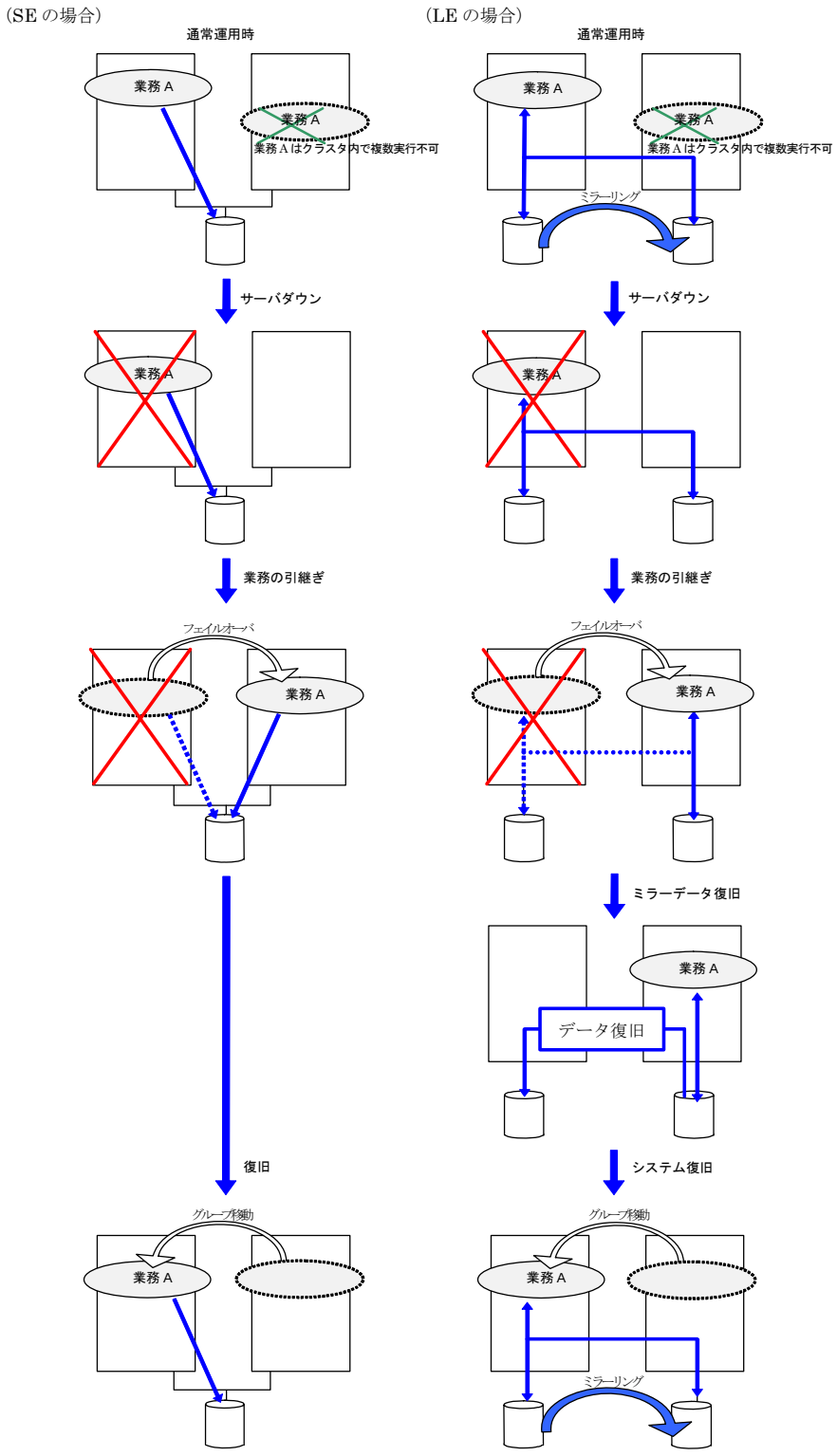


* 異種アプリケーション双方向スタンバイクラスタ
  複数の種類の業務アプリケーションが、それぞれ異なるサーバで稼動し、相互に待機するシステム形態



(1)　片方向スタンバイクラスタ

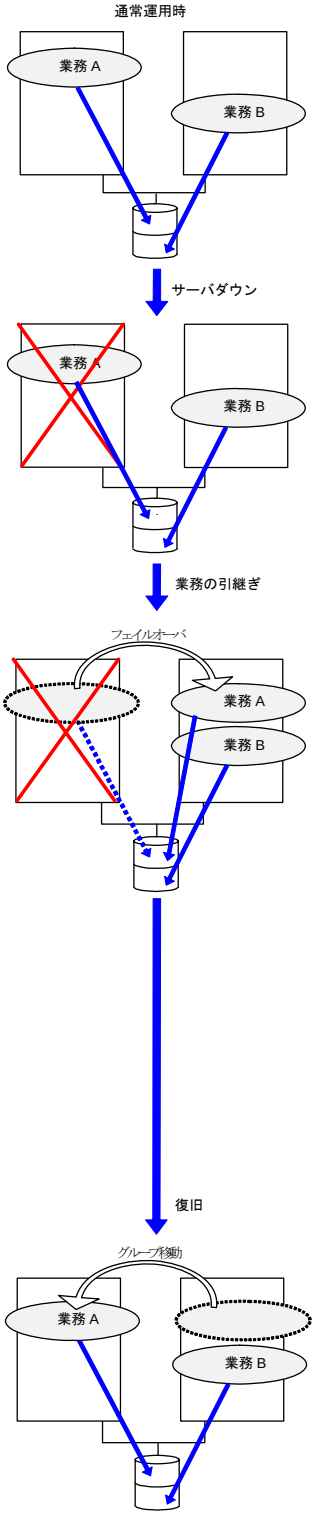
片方向スタンバイクラスタとは、ある業務についてグループを1グループに制限したクラスタシステムです。



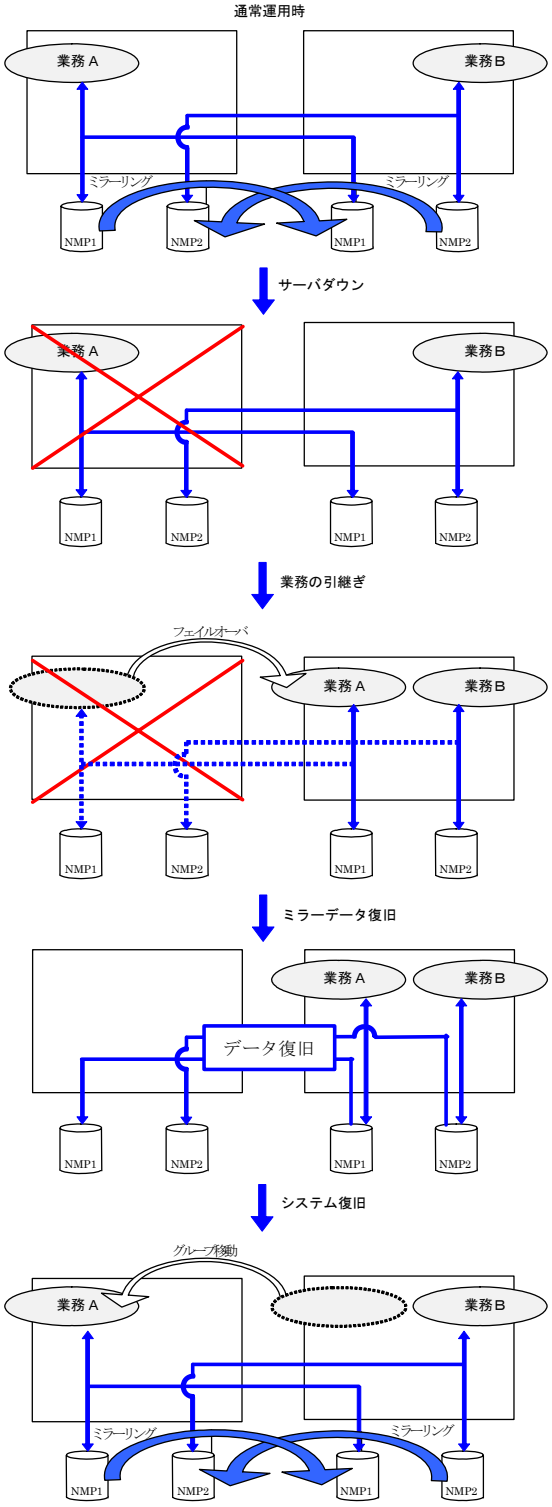
(2)　双方向スタンバイクラスタ

双方向スタンバイクラスタとは、ある業務が複数のサーバ上で同時に動作することが可能なクラスタシステムです。

（SE の場合）



（LE の場合）



1.1.2 フェイルオーバポリシ

フェイルオーバ可能なサーバリストとその中でのフェイルオーバ優先順位です。フェイルオーバ発生時のフェイルオーバポリシによる動作の違いを説明します。

＜図中記号の説明＞

サーバ状態	説明
○	正常状態（クラスタとして正常に動作している）
×	停止状態（クラスタが停止状態）

3ノードの場合

グループ	フェイルオーバポリシ		
	優先度1サーバ	優先度2サーバ	優先度3サーバ
A	サーバ1	サーバ3	サーバ2
B	サーバ2	サーバ3	サーバ1

2ノードの場合

グループ	フェイルオーバポリシ	
	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

AとBはグループ起動属性が自動起動、フェイルバック属性が手動フェイルバックに設定されているものとします。

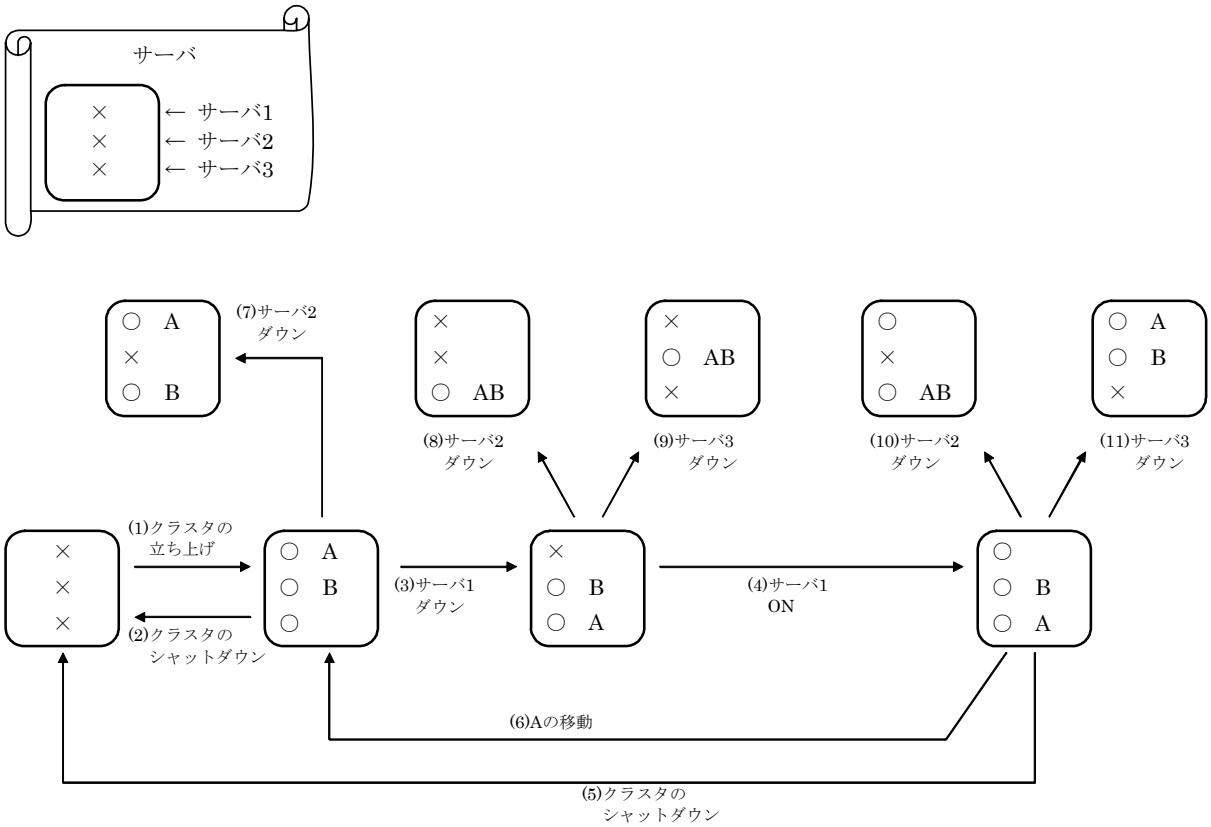
* クラスタ内にフェイルオーバ排他属性の異なるグループが混在した場合、フェイルオーバ排他属性の異なるグループはお互いを干渉しません。例えば、排他なしの属性を持つグループが起動しているサーバで、完全排他の属性を持つグループが起動することはあります。逆に、完全排他の属性を持つグループが起動しているサーバで、排他なしの属性を持つグループが起動することもあります。

* フェイルオーバ排他属性が通常排他あるいは完全排他のグループについて、起動あるいはフェイルオーバするサーバの決定規則としては、そのサーバに対するフェイルオーバ優先順位に基づきます。また優先順位が同じ場合には、グループ名のアルファベット順の若い方を優先とします。

* Webマネージャ用グループのフェイルオーバ優先順位はサーバの優先順位に基づきます。サーバの優先順位はクラスタプロパティのマスタサーバタブで設定します。

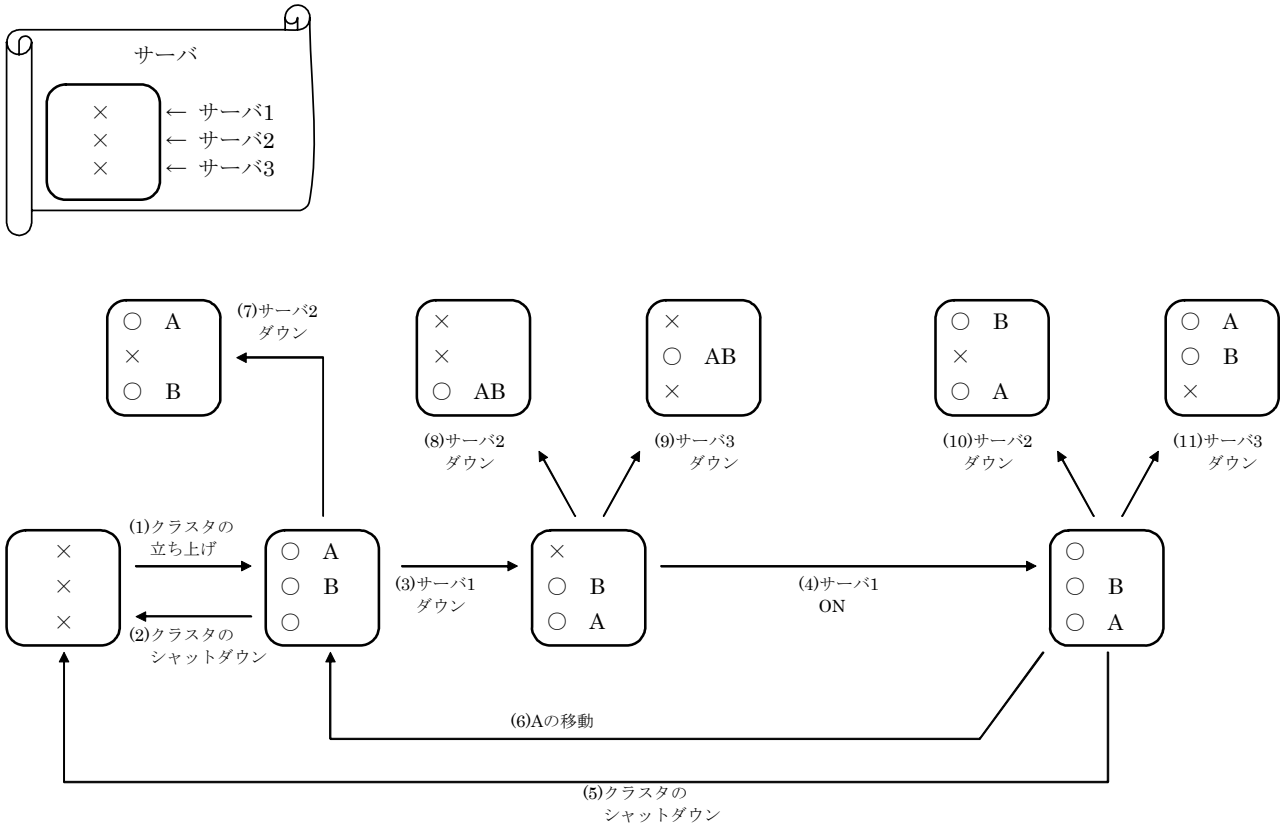
1.1.2.1 SE,XE,SXの場合

(1) グループAとBのフェイルオーバ排他属性が排他なしの場合



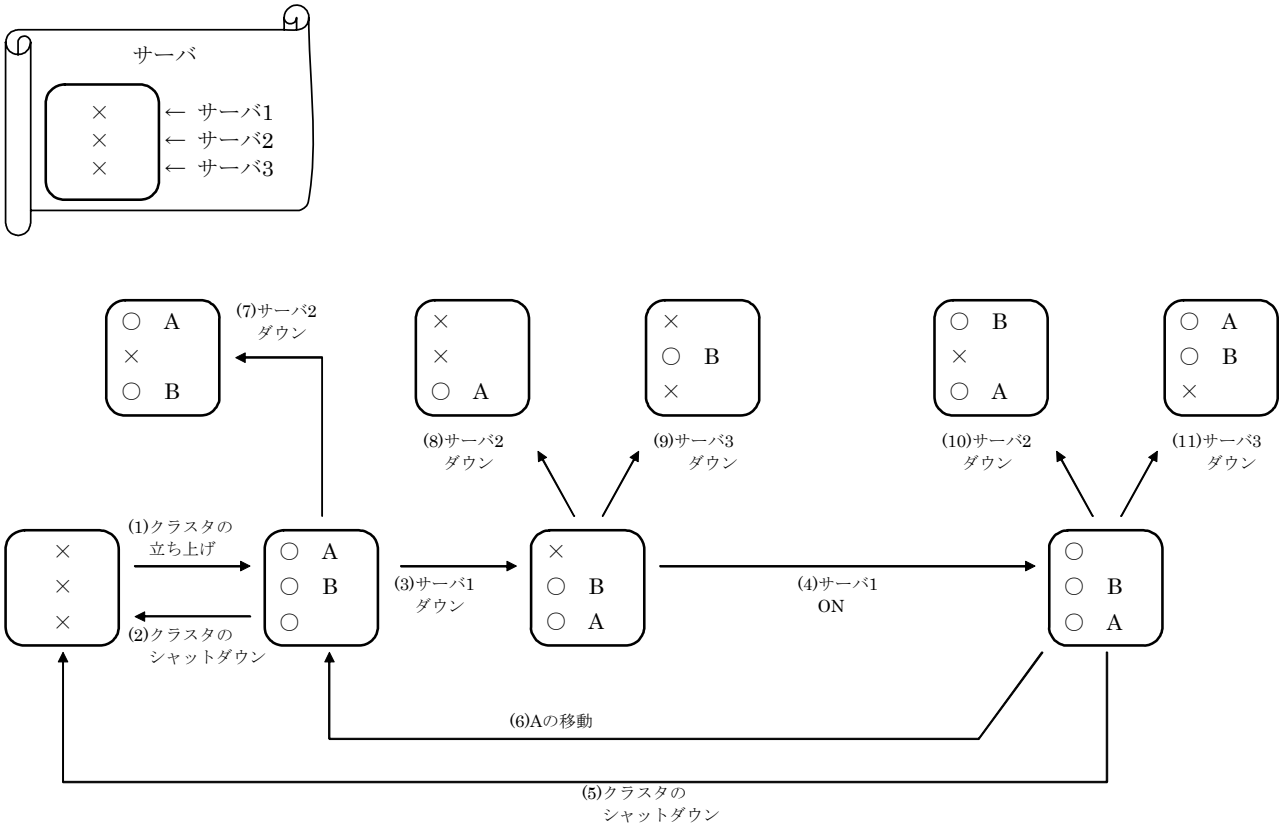
1. クラスタの立ち上げ
2. クラスタのシャットダウン
3. サーバ1ダウン　：　次に優先順位の高いサーバへフェイルオーバする
4. サーバ1の電源ON
5. クラスタのシャットダウン
6. グループAの移動
7. サーバ2ダウン　：　次に優先順位の高いサーバへフェイルオーバする
8. サーバ2ダウン　：　次に優先順位の高いサーバへフェイルオーバする
9. サーバ3ダウン　：　次に優先順位の高いサーバへフェイルオーバする
10. サーバ2ダウン　：　次に優先順位の高いサーバへフェイルオーバする
11. サーバ2ダウン　：　次に優先順位の高いサーバへフェイルオーバする

(2)　グループAとBのフェイルオーバ排他属性が通常排他の場合



1. クラスタの立ち上げ
2. クラスタのシャットダウン
3. サーバ1ダウン　：　通常排他のグループが起動されていないサーバへフェイルオーバする
4. サーバ1の電源ON
5. クラスタのシャットダウン
6. グループAの移動
7. サーバ2ダウン　：　通常排他のグループが起動されていないサーバへフェイルオーバする
8. サーバ2ダウン　：　通常排他のグループが起動されていないサーバは存在しないが、起動可能なサーバが存在するのでフェイルオーバする
9. サーバ3ダウン　：　通常排他のグループが起動されていないサーバは存在しないが、起動可能なサーバが存在するのでフェイルオーバする
10. サーバ2ダウン　：　通常排他のグループが起動されていないサーバへフェイルオーバする
11. サーバ3ダウン　：　通常排他のグループが起動されていないサーバへフェイルオーバする

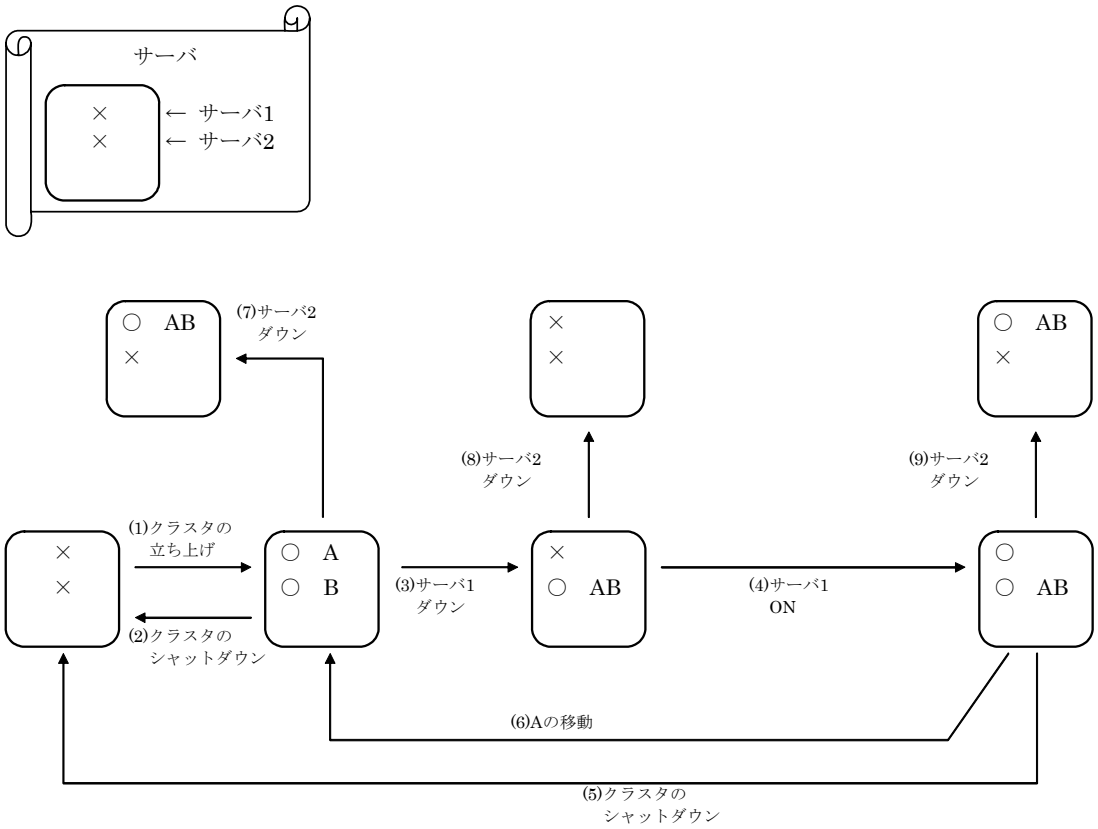
(3)　グループAとBのフェイルオーバ排他属性が完全排他の場合



1. クラスタの立ち上げ
2. クラスタのシャットダウン
3. サーバ1ダウン　：　完全排他のグループが起動されていないサーバへフェイルオーバする
4. サーバ1の電源ON
5. クラスタのシャットダウン
6. グループAの移動
7. サーバ2ダウン　：　完全排他のグループが起動されていないサーバへフェイルオーバする
8. サーバ2ダウン　：　フェイルオーバしない（グループBは停止する）
9. サーバ3ダウン　：　フェイルオーバしない（グループAは停止する）
10. サーバ2ダウン　：　完全排他のグループが起動されていないサーバへフェイルオーバする
11. サーバ3ダウン　：　完全排他のグループが起動されていないサーバへフェイルオーバする

1.1.2.2 LEの場合（サーバ2台のSE,XE,SXの場合も含む）

(1)　グループAとBのフェイルオーバ排他属性が排他なしの場合



1. クラスタの立ち上げ
2. クラスタのシャットダウン
3. サーバ1ダウン　：　グループAの待機系サーバへフェイルオーバする
4. サーバ1の電源ON
5. クラスタのシャットダウン
6. グループAの移動
7. サーバ2ダウン　：　グループBの待機系サーバへフェイルオーバする
8. サーバ2ダウン
9. サーバ2ダウン　：　待機系サーバへフェイルオーバする

1.1.3 アプリケーション

クラスタに対応したアプリケーションは、“フェイルオーバ”または、“グループの移動”が発生した場合に、スクリプトにより相手サーバで再起動されます。よって、同一レビジョンのアプリケーションがフェイルオーバポリシで設定してある全サーバに存在していることが必須です。また、引き継ぐべきデータを共有ディスクまたはミラーディスク上に集められるような性質のものでなくてはなりません。

CLUSTERPRO環境下で動作するアプリケーションは、この他にもいくつかの前提条件をクリアしたものでなければなりません。詳細については、「6.2 CLUSTERPRO環境下でのアプリケーション」を参照してください。

1.1.4 フェイルオーバ要因

フェイルオーバを引き起こす要因としては、以下のものがあります。

* サーバのシャットダウン
* 電源ダウン
* OSのパニック
* OSのストール
* CLUSTERPROサーバの異常
* グループリソースの活性失敗または非活性失敗
* モニタリソースによる異常検出

1.1.5 活性異常、非活性異常検出

活性異常、非活性異常検出時には以下の制御が行われます。

* グループリソース活性異常検出時の流れ
  + グループリソースの活性時に異常を検出した場合、活性リトライを行います。
  + 「活性リトライしきい値」の活性リトライに失敗した場合、フェイルオーバを行います。
  + 「フェイルオーバしきい値」のフェイルオーバを行っても活性出来ない場合、最終動作を行います。

* グループリソース非活性異常検出時の流れ
  + 非活性時に異常を検出した場合、非活性リトライを行います。
  + 「非活性リトライしきい値」の非活性リトライに失敗した場合、最終動作を行います。

活性リトライ回数とフェイルオーバ回数はサーバごとに記録されるため、「活性リトライしきい値」「フェイルオーバしきい値」はサーバごとの活性リトライ回数とフェイルオーバ回数の上限値になります。

グループ活性に成功したサーバでは、活性リトライ回数とフェイルオーバ回数はリセットされます。

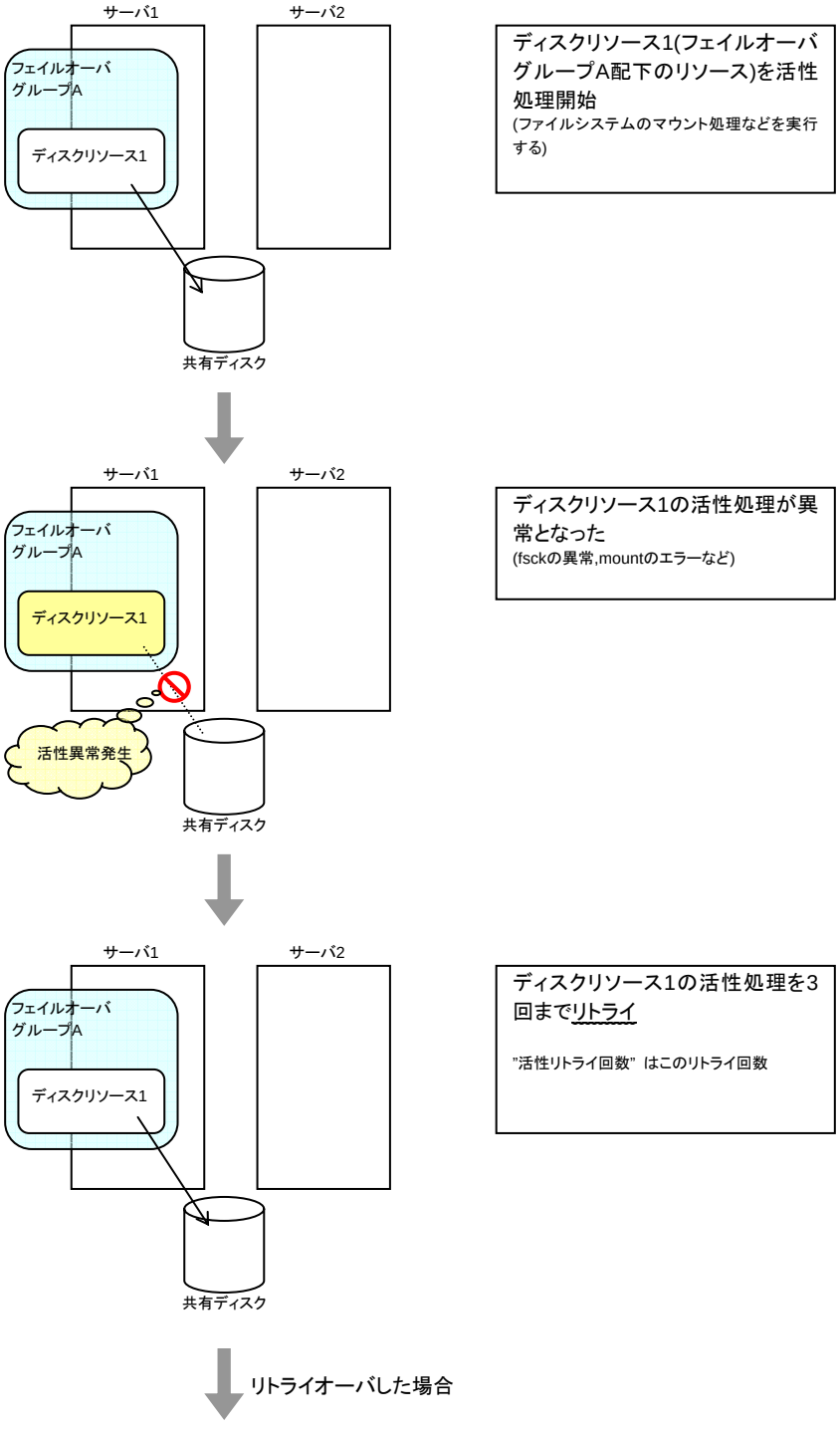
回復動作の活性リトライ回数及びフェイルオーバ回数は回復動作に失敗した場合でも1回としてカウントされることに注意してください。

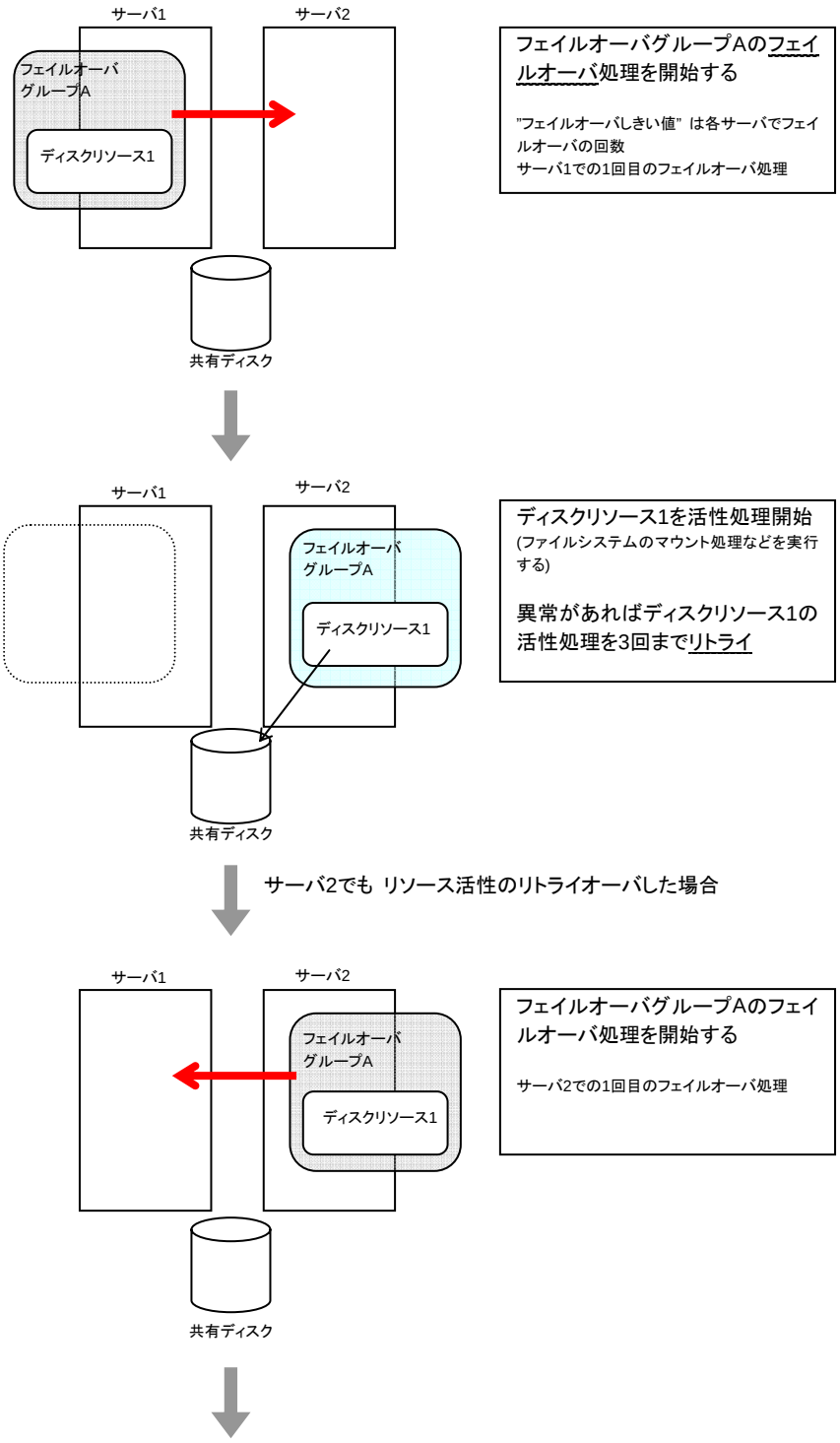
以下の設定例でグループリソース活性異常検出時の流れを説明します。

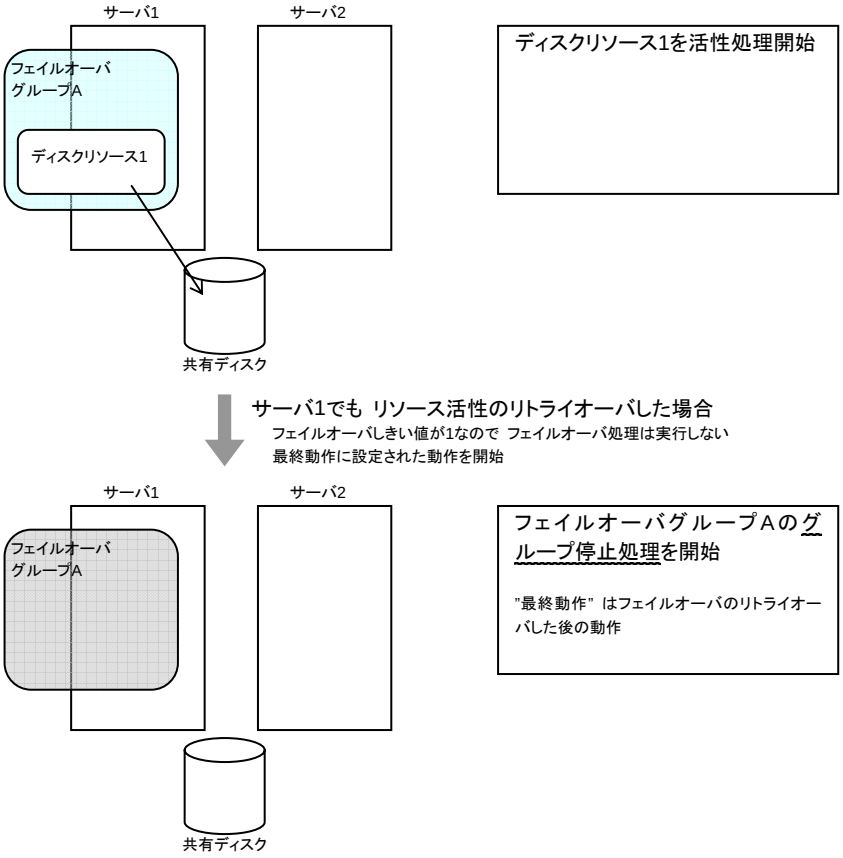
[設定例]

活性リトライしきい値	3回
フェイルオーバしきい値	1回
最終動作	グループ停止

を指定している場合の挙動の例


サーバ1
サーバ2
フェイルオーバ
グループA
ディスクリソース1
共有ディスク
ディスクリソース1(フェイルオーバグループA配下のリソース)を活性処理開始
(ファイルシステムのマウント処理などを実行する)
サーバ1
サーバ2
フェイルオーバ
グループA
ディスクリソース1
活性異常発生
共有ディスク
ディスクリソース1の活性処理が異常となった
(fsckの異常,mountのエラーなど)
サーバ1
サーバ2
フェイルオーバ
グループA
ディスクリソース1
共有ディスク
ディスクリソース1の活性処理を3回までリトライ
”活性リトライ回数” はこのリトライ回数
リトライオーバした場合


サーバ1
サーバ2
フェイルオーバ
グループA
ディスクリソース1
共有ディスク
フェイルオーバグループAのフェイルオーバ処理を開始する
"フェイルオーバしきい値" は各サーバでフェイルオーバの回数
サーバ1での1回目のフェイルオーバ処理
サーバ1
サーバ2
フェイルオーバ
グループA
ディスクリソース1
共有ディスク
ディスクリソース1を活性処理開始
(ファイルシステムのマウント処理などを実行する)
異常があればディスクリソース1の活性処理を3回までリトライ
サーバ2でも リソース活性のリトライオーバした場合
サーバ1
サーバ2
フェイルオーバ
グループA
ディスクリソース1
共有ディスク
フェイルオーバグループAのフェイルオーバ処理を開始する
サーバ2での1回目のフェイルオーバ処理


サーバ1
サーバ2
フェイルオーバ
グループA
ディスクリソース1
共有ディスク
ディスクリソース1を活性処理開始
サーバ1でも リソース活性のリトライオーバした場合
フェイルオーバしきい値が1なので フェイルオーバ処理は実行しない
最終動作に設定された動作を開始
サーバ1
サーバ2
フェイルオーバ
グループA
共有ディスク
フェイルオーバグループAのグループ停止処理を開始
"最終動作" はフェイルオーバのリトライオーバした後の動作

1.1.6 再起動回数制限

活性異常、非活性異常検出時の最終動作として「クラスタデーモンの停止とOSシャットダウン」、または「クラスタデーモンの停止とOS再起動」を設定している場合に、活性異常、非活性異常の検出によるシャットダウン回数、または再起動回数を制限することができます。

この最大再起動回数はサーバごとの再起動回数の上限になります。

再起動回数はサーバごとに記録されるため、最大再起動回数はサーバごとの再起動回数の上限になります。

また、グループ活性、非活性異常検出時の最終動作による再起動回数とモニタリソース異常の最終動作による再起動回数も別々に記録されます。

最大再起動回数をリセットする時間に0を設定した場合には、再起動回数はリセットされません。リセットする場合はclpregctrlコマンドを使用する必要があります。clpregctrlコマンドに関しては「コマンド編」を参照してください。

以下の設定例で再起動回数制限の流れを説明します。

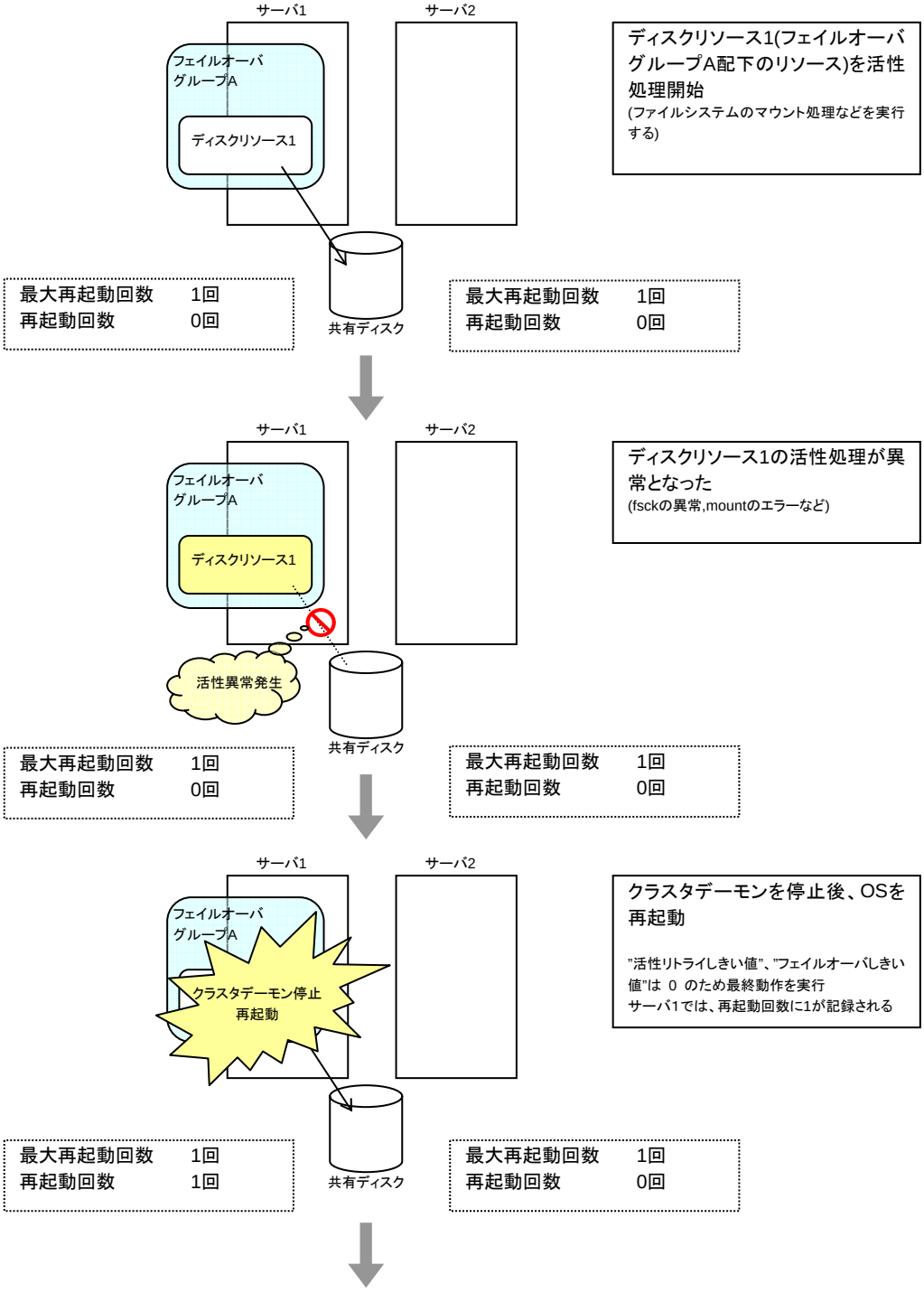
最大再起動回数が1回に設定されているため、一度だけ最終動作である「クラスタデーモンの停止とOS再起動」が実行されます。

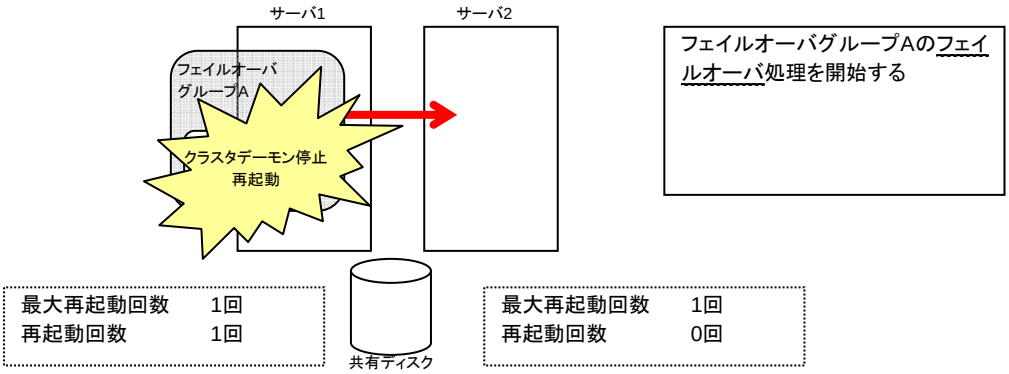
また、最大再起動回数をリセットする時間が10分に設定されているため、クラスタシャットダウン後再起動時にグループの活性に成功した場合には、10分経過すると再起動回数はリセットされます。

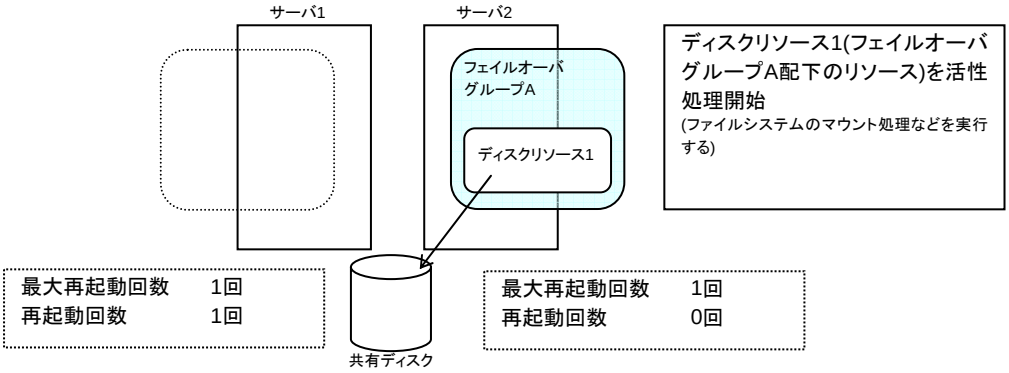
[設定例]

活性リトライしきい値	0回
フェイルオーバしきい値	0回
最終動作	クラスタデーモンの停止とOS再起動
最大再起動回数	1回
最大再起動回数をリセットする時間	10分

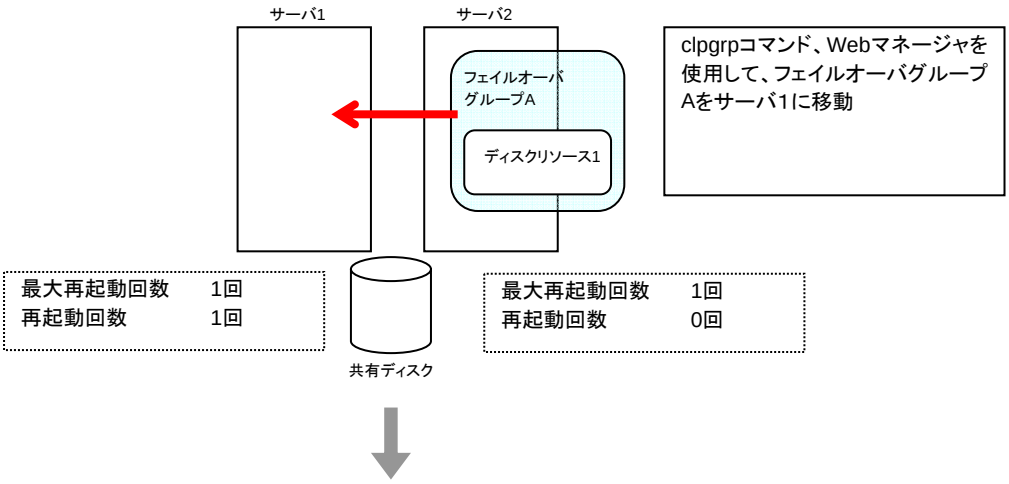
を指定している場合の挙動の例

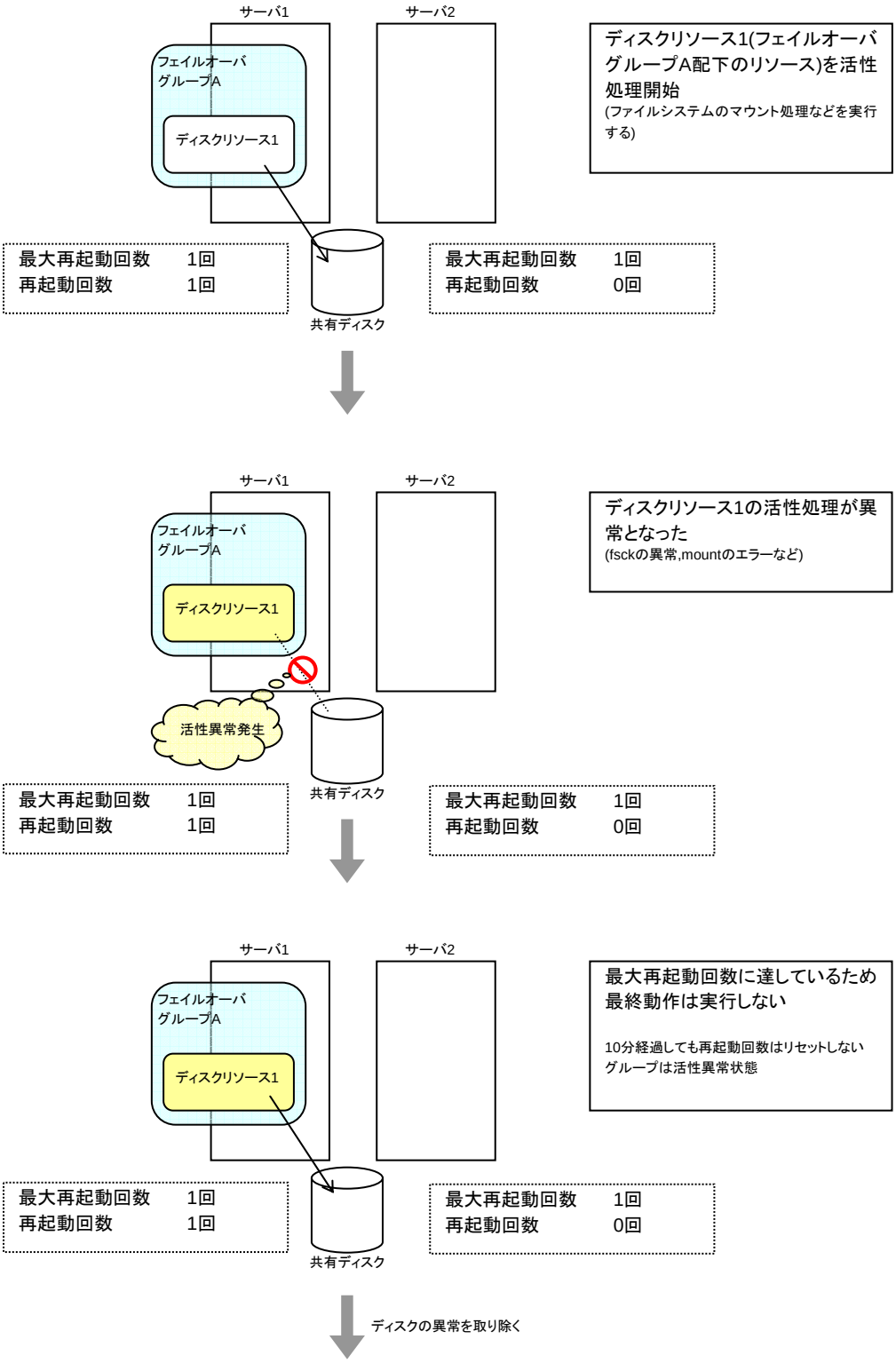

サーバ1
サーバ2
フェイルオーバ
グループA
ディスクリソース1
共有ディスク
最大再起動回数 1回
再起動回数 0回
最大再起動回数 1回
再起動回数 0回
ディスクリソース1(フェイルオーバグループA配下のリソース)を活性処理開始
(ファイルシステムのマウント処理などを実行する)
サーバ1
サーバ2
フェイルオーバ
グループA
ディスクリソース1
活性異常発生
共有ディスク
最大再起動回数 1回
再起動回数 0回
最大再起動回数 1回
再起動回数 0回
ディスクリソース1の活性処理が異常となった
(fsckの異常,mountのエラーなど)
サーバ1
サーバ2
フェイルオーバ
グループA
クラスタデーモン停止
再起動
共有ディスク
最大再起動回数 1回
再起動回数 1回
最大再起動回数 1回
再起動回数 0回
クラスタデーモンを停止後、OSを再起動
”活性リトライしきい値”、”フェイルオーバしきい値”は 0 のため最終動作を実行
サーバ1では、再起動回数に1が記録される

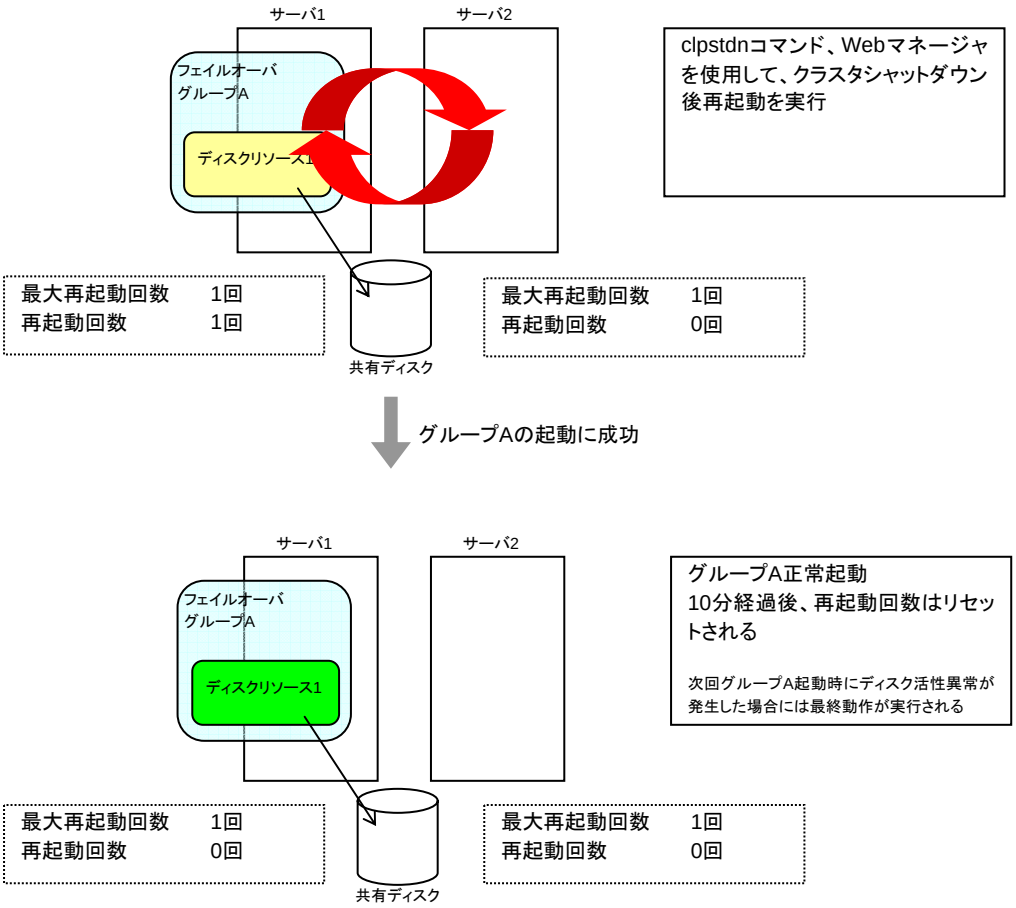




サーバ2では リソース活性成功
サーバ1は再起動完了




サーバ1
サーバ2
フェイルオーバ
グループA
ディスクリソース1
ディスクリソース1(フェイルオーバグループA配下のリソース)を活性処理開始
(ファイルシステムのマウント処理などを実行する)
最大再起動回数 1回
再起動回数 1回
最大再起動回数 1回
再起動回数 0回
共有ディスク
ディスクリソース1の活性処理が異常となった
(fsckの異常,mountのエラーなど)
活性異常発生
最大再起動回数に達しているため最終動作は実行しない
10分経過しても再起動回数はリセットしない
グループは活性異常状態
ディスクの異常を取り除く



1.1.7 再起動回数初期化

再起動回数を初期化する場合、clpregctrlコマンドを使用してください。clpregctrlコマンドに関しては「コマンド編」を参照してください。

1.2 execリソース

CLUSTERPROでは、CLUSTERPROによって管理され、グループの起動時、終了時、フェイルオーバ発生時、及び移動の場合に実行されるアプリケーションやシェルスクリプトを登録できます。

execリソースには、ユーザ独自のプログラムやシェルスクリプトなども登録できます。

シェルスクリプトは、shのシェルスクリプトと同じ書式なので、それぞれのアプリケーションの事情にあわせた処理を記述できます。

1.2.1 CLUSTERPROのバージョン

以下のCLUSTERPROのバージョンでサポートします。

CLUSTERPRO	Version
サーバ	SE3.0-1 以降、LE3.0-1 以降、XE3.0-1 以降、SX3.1-2 以降
トレッキングツール	対応するバージョンについては動作環境編 トレッキングツールの動作環境を参照してください。

1.2.2 依存関係

規定値では、以下のグループリソースタイプに依存します。

グループリソースタイプ	Edition
フローティングIPリソース	SE、LE、XE、SX
ディスクリソース	SE、XE、SX
ミラーディスクリソース	LE
RAWリソース	SE、XE、SX
VxVMディスクグループリソース	SE
VxVMボリュームリソース	SE
NASリソース	SE、LE、XE、SX

1.2.3 execリソースに関する注意事項

* execリソースで実行されるアプリケーションの同一レビジョンのものが、フェイルオーバポリシに設定されている全サーバに存在していることが必須です。

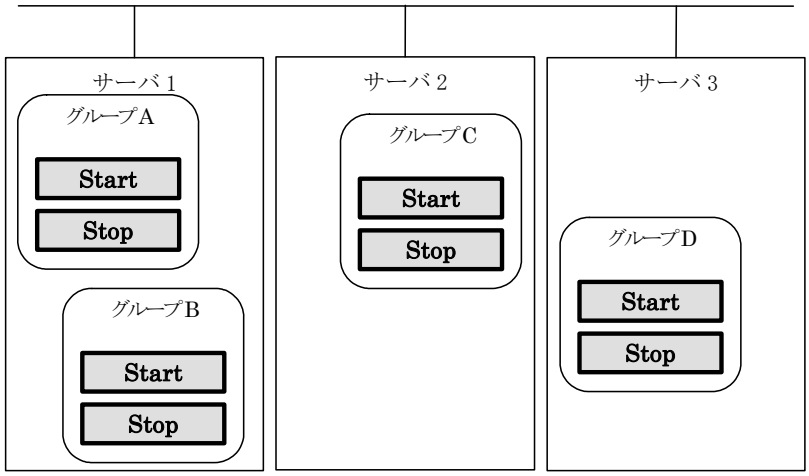
* CLUSTERPROのバージョンが3.0-1～3.1-8の場合、execリソースは下記のシグナル以外はマスクされます。
  + SIGTERM
  + SIGKILL
  + SIGCLD
  + SIGABRT
  + SIGSEGV
  + SIGSTOP
  + SIGILL
  + SIGFPE
  + SIGUSR1

1.2.4 スクリプト

1.2.4.1 スクリプトの種類

execリソースには、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、execリソース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　：　開始スクリプト
Stop　：　終了スクリプト

1.2.4.2 スクリプトの環境変数

CLUSTERPROは、スクリプトを実行する場合に、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。開始スクリプトではCLP_FACTOR及びCLP_PIDの環境変数はセットされません。

CLP_LASTACTION の環境変数は、CLP_FACTOR の環境変数が CLUSTERSHUTDOWNまたはSERVERSHUTDOWNの場合にのみセットされます。

環境変数	環境変数の値	意　味
CLP_EVENT …スクリプト実行要因	START	クラスタの起動により、実行された場合。 グループの起動により、実行された場合。 グループの移動により、移動先のサーバで実行された場合。 モニタリソースの異常検出によるグループの再起動により、同じサーバで実行された場合。 モニタリソースの異常検出によるグループリソースの再起動により、同じサーバで実行された場合。
	FAILOVER	サーバダウンにより、フェイルオーバ先のサーバで実行された場合。 モニタリソースの異常検出により、フェイルオーバ先のサーバで実行された場合。 グループリソースの活性失敗により、フェイルオーバ先のサーバで実行された場合。
CLP_FACTOR …グループ停止要因	CLUSTERSHUTDOWN	クラスタ停止により、グループの停止が実行された場合。
	SERVERSHUTDOWN	サーバ停止により、グループの停止が実行された場合。
	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。
	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。
	GROUPFAILOVER	モニタリソースの異常検出により、グループのフェイルオーバが実行された場合。 グループリソースの活性失敗により、グループのフェイルオーバが実行された場合。
	GROUPRESTART	モニタリソースの異常検出により、グループの再起動が実行された場合。
	RESOURCERESTART	モニタリソースの異常検出により、グループリソースの再起動が実行された場合。
CLP_LASTACTION …クラスタ停止後処理	REBOOT	OSをrebootする場合。
	HALT	OSをhaltする場合。
	NONE	何もしない。
CLP_SERVER …スクリプトの実行サーバ	HOME	グループの、プライマリサーバで実行された。
	OTHER	グループの、プライマリサーバ以外で実行された。

環境変数	環境変数の値	意　味
CLP_DISK …共有ディスクまたはミラーディスク上のパーティション接続情報	SUCCESS	接続に失敗しているパーティションはない。
	FAILURE	接続に失敗しているパーティションがある。
CLP_PRIORITY …スクリプトが実行されたサーバのフェイルオーバポリシの順位	1～クラスタ内のサーバ数	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。 CLP_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。
CLP_GROUPNAME …グループ名	グループ名	スクリプトが属している、グループ名を示す。
CLP_RESOURCENAME …リソース名	リソース名	スクリプトが属している、リソース名を示す。
CLP_PID …プロセスID	プロセスID	プロパティとして開始スクリプトが非同期に設定されている場合、開始スクリプトのプロセスIDを示す。開始スクリプトが同期に設定されている場合、本環境変数は値を持たない。

1.2.4.3 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。

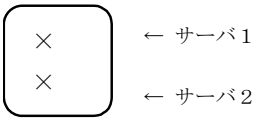
サーバ	サーバ状態
○	正常状態（クラスタとして正常に動作している）
×	停止状態（クラスタが停止状態）

（例）○A ： 正常状態にあるサーバにおいてグループAが動作している。

* 各グループは、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ポリシを持っています。

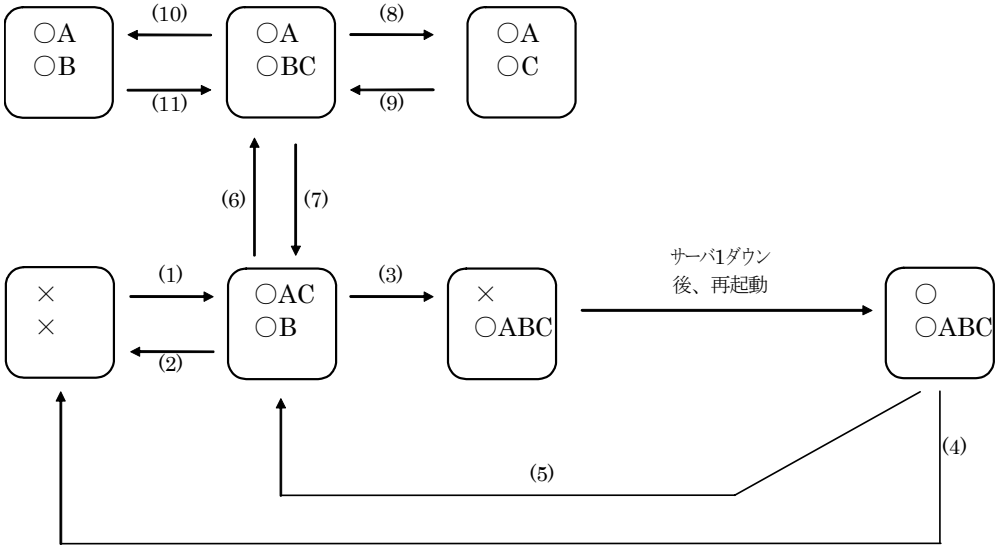
グループ	優先度1サーバ	優先度2サーバ
A	サーバ1	サーバ2
B	サーバ2	サーバ1
C	サーバ1	サーバ2

* 上のサーバをサーバ1、下のサーバをサーバ2とします。



【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。

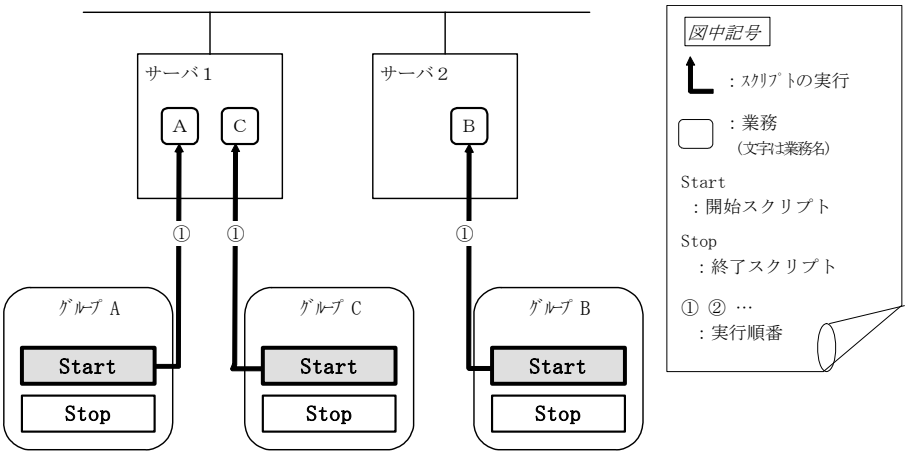


図中の(1)～(11)は、次ページからの説明に対応しています。

(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た時を指します。

各グループは、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。

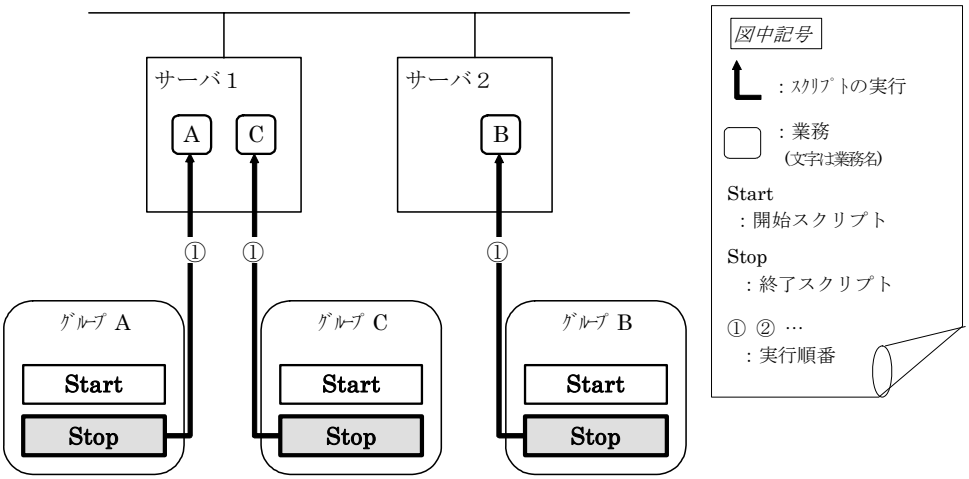


Startに対する環境変数

グループ	環境変数	値
A	CLP_EVENT	START
	CLP_SERVER	HOME
B	CLP_EVENT	START
	CLP_SERVER	HOME
C	CLP_EVENT	START
	CLP_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。



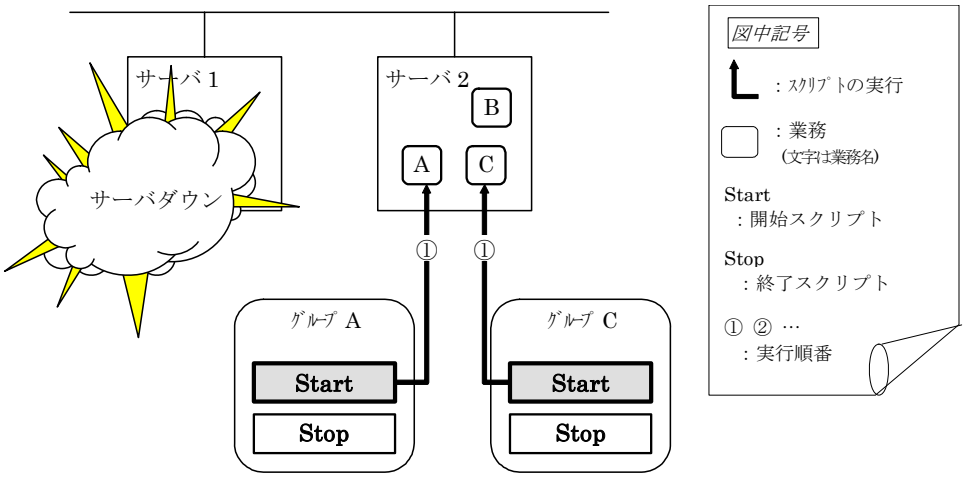
Stopに対する環境変数

グループ	環境変数	値
A	CLP_EVENT	START
	CLP_SERVER	HOME
B	CLP_EVENT	START
	CLP_SERVER	HOME
C	CLP_EVENT	START
	CLP_SERVER	HOME

(3) サーバ1ダウンによるフェイルオーバ

サーバ1をプライマリサーバとする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、CLP_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、CLP_SERVER(=OTHER)を分岐条件にして記述しておく必要があります。

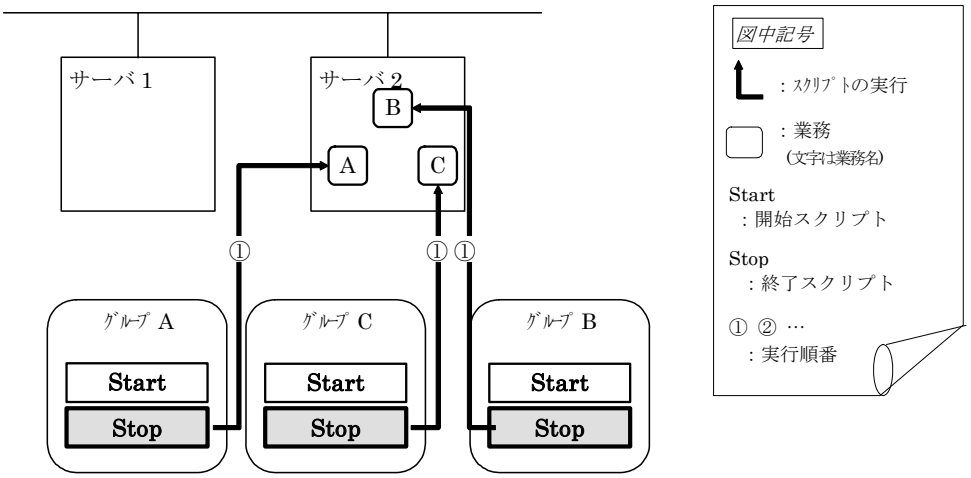


Startに対する環境変数

グループ	環境変数	値
A	CLP_EVENT	FAILOVER
	CLP_SERVER	OTHER
C	CLP_EVENT	FAILOVER
	CLP_SERVER	OTHER

(4) サーバ1フェイルオーバ後クラスタシャットダウン

グループAとCの終了スクリプトが、フェイルオーバ先のサーバ2で実行されます(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)。

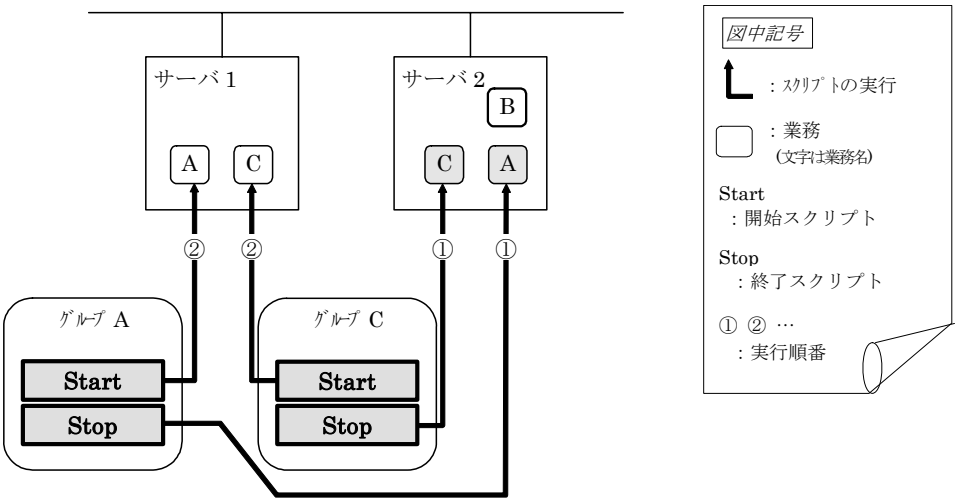


Stopに対する環境変数

グループ	環境変数	値
A	CLP_EVENT	FAILOVER
	CLP_SERVER	OTHER
B	CLP_EVENT	START
	CLP_SERVER	HOME
C	CLP_EVENT	FAILOVER
	CLP_SERVER	OTHER

(5) グループAとCの移動

グループAとCの終了スクリプトが、フェイルオーバ先のサーバ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に対する環境変数

グループ	環境変数	値
A	CLP_EVENT	FAILOVER [1]
	CLP_SERVER	OTHER
C	CLP_EVENT	FAILOVER
	CLP_SERVER	OTHER

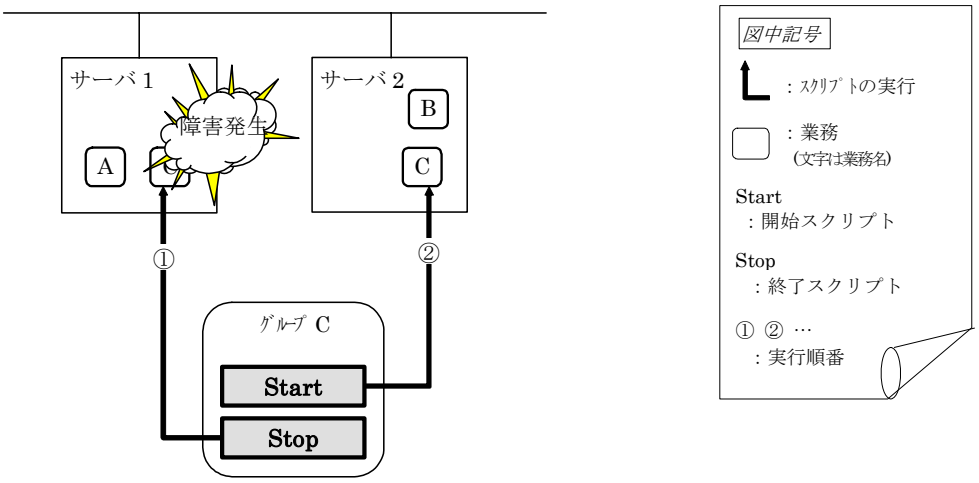
Startに対する環境変数

グループ	環境変数	値
A	CLP_EVENT	START
	CLP_SERVER	HOME
C	CLP_EVENT	START
	CLP_SERVER	HOME

[1] 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となる。
「1.2.4.3(5) グループAとCの移動」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になるが、「1.2.4.3(5) グループAとCの移動」の前にクラスタシャットダウンが行われていると、STARTとなる。

(6) グループCの障害、フェイルオーバ

グループCに障害が発生すると、サーバ1でグループCの終了スクリプトが実行され、サーバ2でグループCの開始スクリプトがで実行されます。



サーバ1のStop

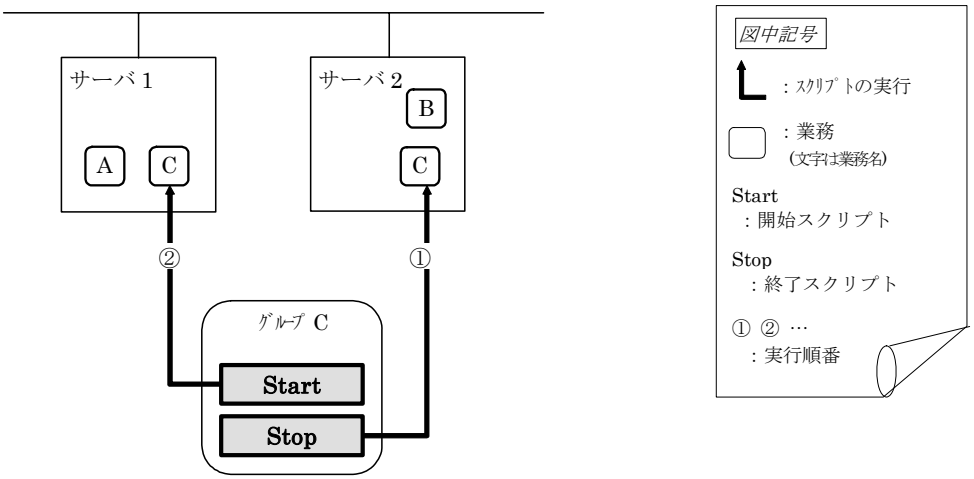
グループ	環境変数	値
C	CLP_EVENT	START
	CLP_SERVER	HOME

サーバ2のStart

グループ	環境変数	値
C	CLP_EVENT	FAILOVER
	CLP_SERVER	OTHER

(7) グループCの移動

(6)でサーバ2にフェイルオーバしてきたグループ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((6)よりフェイルオーバしてきたため)

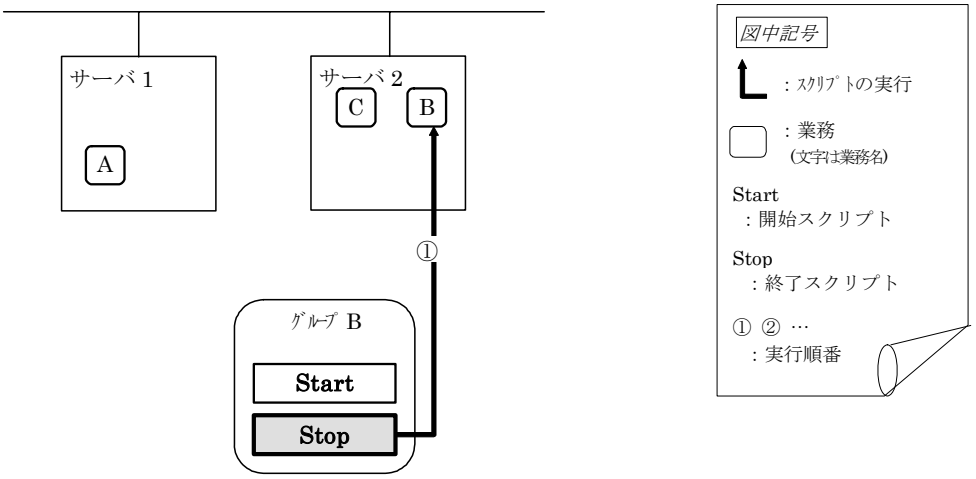
グループ	環境変数	値
C	CLP_EVENT	FAILOVER
	CLP_SERVER	OTHER

Start

グループ	環境変数	値
C	CLP_EVENT	START
	CLP_SERVER	HOME

(8) グループBの停止

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。

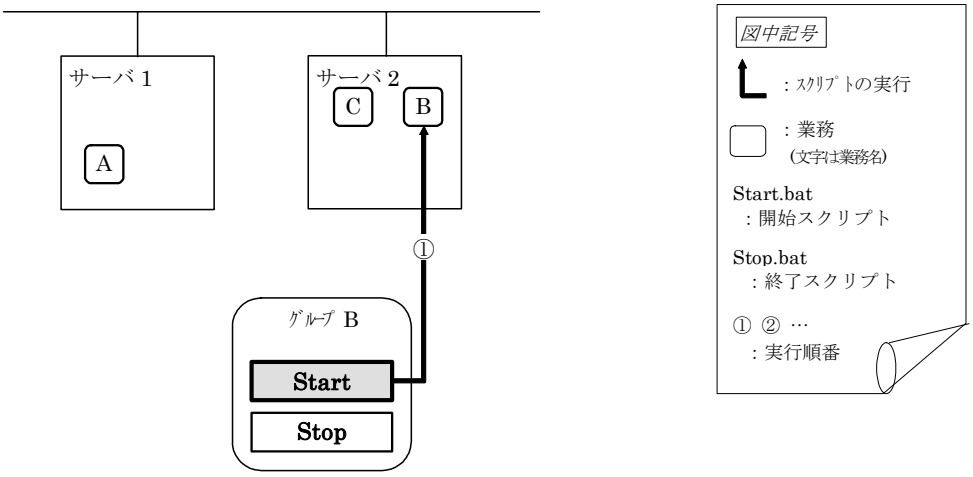


Stop

グループ	環境変数	値
B	CLP_EVENT	START
	CLP_SERVER	HOME

(9) グループBの起動

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。

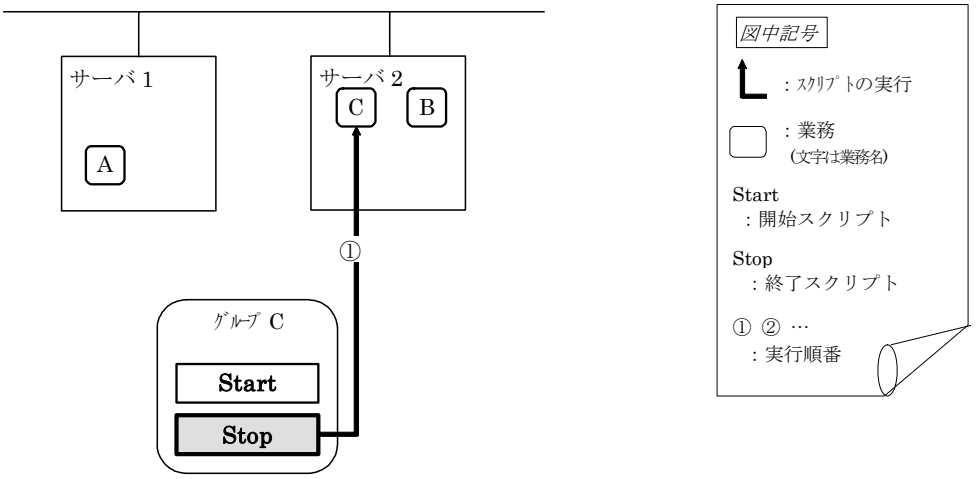


Start

グループ	環境変数	値
B	CLP_EVENT	START
	CLP_SERVER	HOME

(10) グループCの停止

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。

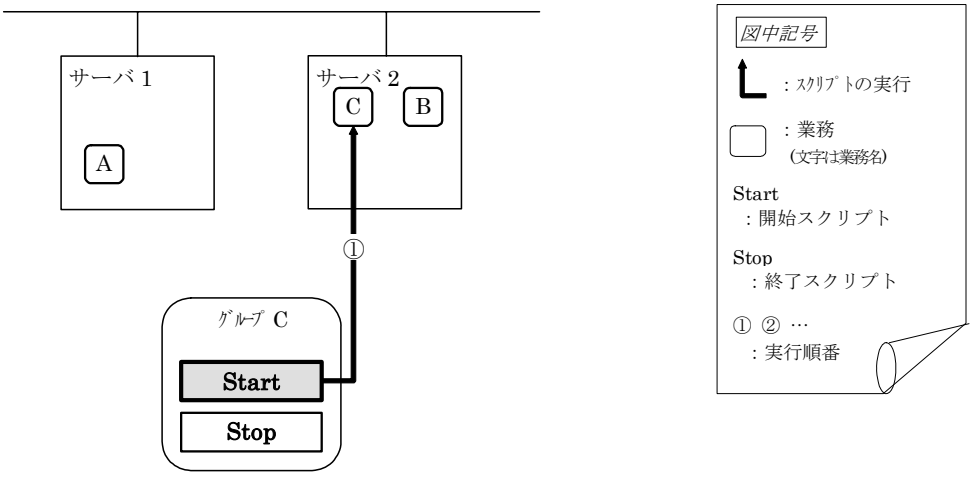


Stop

グループ	環境変数	値
C	CLP_EVENT	FAILOVER
	CLP_SERVER	OTHER

(11) グループCの起動

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。

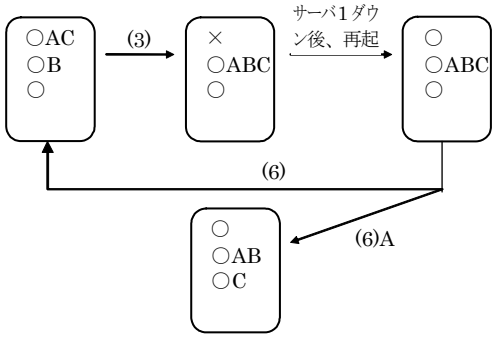


Start

グループ	環境変数	値
C	CLP_EVENT	START
	CLP_SERVER	OTHER

【補足1】

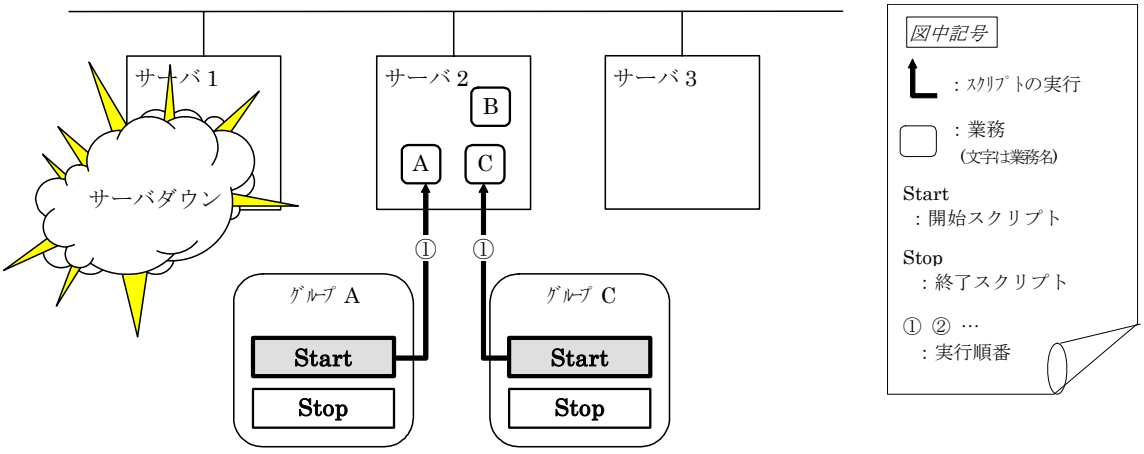
フェイルオーバポリシに設定されているサーバを3つ以上持つグループにおいて、プライマリサーバ以外のサーバで、異なった動作を行なう場合 CLP_SERVER(HOME/OTHER)の代わりに、CLP_PRIORITYを使用する



(例1) クラスタ状態遷移図「(3) サーバ1ダウンによるフェイルオーバ」の場合

サーバ1をプライマリサーバとする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ポリシを持つサーバ2で実行されます。開始スクリプトには、CLP_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。

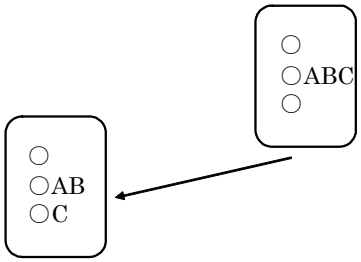
2番目に高いフェイルオーバポリシを持つサーバのみで実行したい処理がある場合は、CLP_PRIORITY(=2)を分岐条件にして記述しておく必要があります。



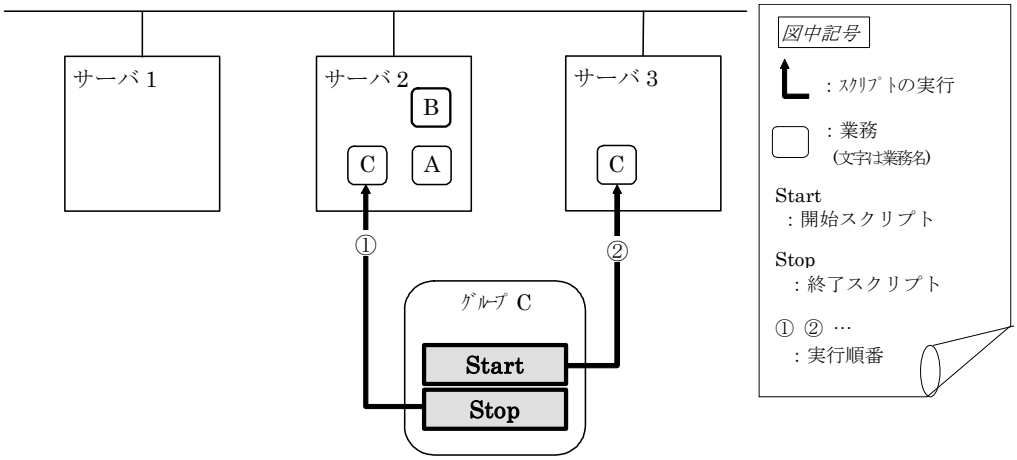
Startに対する環境変数

グループ	環境変数	値
A	CLP_EVENT	FAILOVER
	CLP_SERVER	OTHER
	CLP_PRIORITY	2
C	CLP_EVENT	FAILOVER
	CLP_SERVER	OTHER
	CLP_PRIORITY	2

(例2) クラスタ状態遷移図「(7) グループCの移動」の場合



グループCの終了スクリプトが、フェイルオーバ元のサーバ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に対する環境変数

グループ	環境変数	値
C	CLP_EVENT	FAILOVER
	CLP_SERVER	OTHER
	CLP_PRIORITY	2

Startに対する環境変数

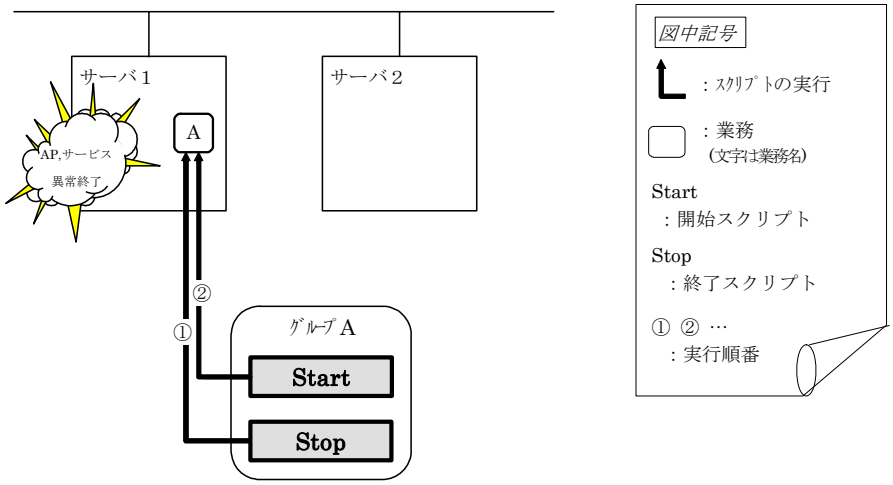
グループ	環境変数	値
C	CLP_EVENT	START
	CLP_SERVER	OTHER
	CLP_PRIORITY	3

【補足2】

リソースモニタがスクリプトを(再)起動する場合

リソースモニタ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。

(例1) リソースモニタがサーバ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でグループAの再起動を行う場合



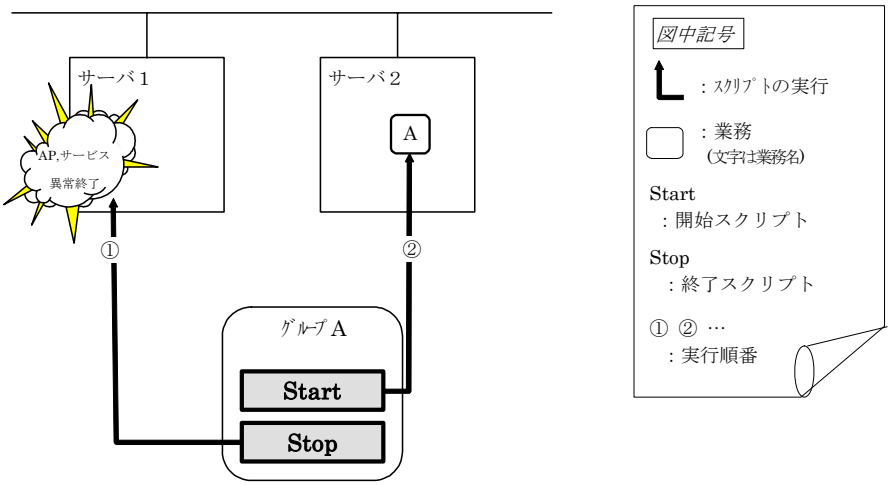
Stopに対する環境変数

グループ		環境変数	値
A	①	CLP_EVENT	Start実行時と同一の値

Startに対する環境変数

グループ		環境変数	値
A	②	CLP_EVENT	START

(例2) リソースモニタがサーバ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へフェイルオーバをしてサーバ2でグループAの起動を行う場合



Stopに対する環境変数

グループ		環境変数	値
A	①	CLP_EVENT	Start実行時と同一の値

Startに対する環境変数

ｸﾞﾙｰﾌﾟ		環境変数	値
A	②	CLP_EVENT	FAILOVER

1.2.4.4 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。
文中の（数字）は「1.2.4.3 スクリプトの実行タイミング」の各動作をさします。

A. グループA開始スクリプト：start.shの一例

```
#! /bin/sh
# ****************************************
# *                start.sh                *
# ****************************************

if [ "$CLP_EVENT" = "START" ]
then
        if [ "$CLP_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
                        (5) グループ A と C の移動

                if [ "$CLP_SERVER" = "HOME" ]
                then
                        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ないたい処理
                        この処理を行う実行タイミング：
                                (1) 通常立ち上げ
                                (5) グループ A と C の移動

                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ないたい処理
                        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 [ "$CLP_EVENT" = "FAILOVER" ]
then
```

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。

実行サーバ環境変数を参照して、処理の振り分けを行う。

```
        if [ "$CLP_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：
                        (3) サーバ 1 ダウンによるフェイルオーバ

                if [ "$CLP_SERVER" = "HOME" ]
                then
                        処理概要：
                                フェイルオーバ後、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

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

        fi
else
        #NO_CLP

fi
#EXIT

exit 0
```

DISK 接続情報環境変数を参照して、エラー処理を行う。

実行サーバ環境変数を参照して、処理の振り分けを行う。

CLUSTERPROは動作していない

B.　グループA終了スクリプト：stop.shの一例

```
#! /bin/sh
# ***************************************
# *                stop.sh                *
# ***************************************

if [ "$CLP_EVENT" = "START" ]
then
        if [ "$CLP_DISK" = "SUCCESS" ]
        then
                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

                if [ "$CLP_SERVER" = "HOME" ]
                then
                        処理概要：
                                プライマリサーバで、業務が通常処理される場合のみ行ないたい処理
                        この処理を行う実行タイミング：
                                (2) 通常シャットダウン

                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ないたい処理
                        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 [ "$CLP_EVENT" = "FAILOVER" ]
then
```

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。

実行サーバ環境変数を参照して、処理の振り分けを行う。

```
        if [ "$CLP_DISK" = "SUCCESS" ]
        then
                処理概要：
                    フェイルオーバ後、通常終了処理
                この処理を行う実行タイミング：
                    (4) サーバ 1 フェイルオーバ後クラスタシャットダウン
                    (5) グループ A と C の移動

                if [ "$CLP_SERVER" = "HOME" ]      実行サーバ環境変数を参照して、処理の振り分けを行う。
                then
                        処理概要：
                            フェイルオーバ後、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：
                            (4) サーバ 1 フェイルオーバ後クラスタシャットダウン
                            (5) グループ A と C の移動
                fi
        else
            ディスク関連エラー処理
        fi
else
        #NO_CLP                                   CLUSTERPROは動作していない

fi
#EXIT
exit 0
```

1.2.4.5 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、実行に時間を必要とするコマンドを実行する場合には、コマンドの実行が完了したことを示すメッセージを標準出力するようにしてください。メッセージはechoコマンドにて標準出力することができます。その上で、スクリプトが属しているリソースのプロパティでログ出力先を設定します。

  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。

但し、デフォルトではログ出力されません。ログ出力先の設定については「トレッキングツール編」を参照してください。

(例：スクリプト中のイメージ)

```
echo “appstart..”
appstart
echo “OK”
```

* ログ出力先に設定されたファイルには、サイズが無制限に出力されますのでファイルシステムの空き容量に注意してください。

1.3 ディスクリソース

1.3.1 CLUSTERPROのバージョン

以下のCLUSTERPROのバージョンでサポートします。

CLUSTERPRO	Version
サーバ	SE3.0-1 以降、XE3.0-1 以降、SX3.1-2 以降
トレッキングツール	対応するバージョンについては動作環境編 トレッキングツールの動作環境を参照してください。

1.3.2 依存関係

規定値では、以下のグループリソースタイプに依存します。

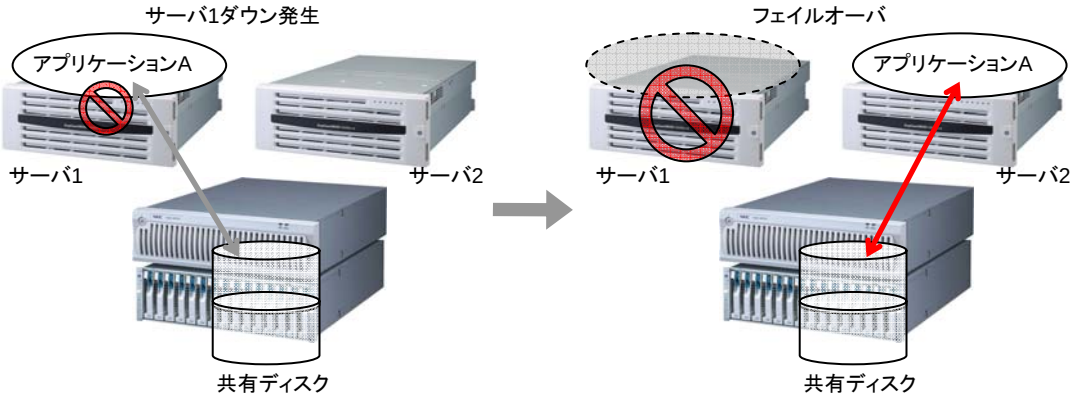
グループリソースタイプ	Edition
フローティングIPリソース	SE、XE、SX

1.3.3 切替パーティション

* 切替パーティションとは、クラスタを構成する複数台のサーバに接続された共有ディスク上のパーティションをいいます。

* 切替はフェイルオーバグループ毎に、フェイルオーバポリシにしたがって行われます。業務に必要なデータは、切替パーティション上に格納しておくことで、フェイルオーバ時、フェイルオーバグループの移動時等に、自動的に引き継がれます。

* 切替パーティションは全サーバで、同一領域に同じデバイス名でアクセスできるようにしてください。



1.3.4 fsck実行タイミング

以下のバージョンでfsck実行タイミングを調節することが可能です。

CLUSTERPRO	Version
サーバ	SE3.1-6 以降、XE3.1-6 以降、SX3.1-6 以降
トレッキングツール	3.1-6 以降

(1) Mount実行前のfsckアクション（fsckタイミング）

切替パーティションのマウントを実行する前のfsckの挙動を選択できます。

+ 必ず実行する
  fsckを必ず実行します。

+ 指定回数に達したら実行する
  切替パーティションをマウントした回数が、指定した回数（fsckインターバル）に達したらfsckを実行します。

  例）指定回数が5回の場合

  サーバ1：ディスクリソース活性 1回目
  　　fsck：実行しない(マウントカウント0回）
  　　マウント：成功(マウントカウント1回）
  　　：
  サーバ1：ディスクリソース活性 5回目
  　　fsck：実行しない(マウントカウント4回）
  　　マウント：成功(マウントカウント5回）
  　　：
  サーバ1：ディスクリソース活性 6回目
  　　fsck：実行(マウントカウント5回)
  　　→成功(マウントカウントリセット 0回）
  　　マウント：成功(マウントカウント1回）
  　　：

+ 実行しない
  fsckを実行しません。

(2) Mount失敗時のfsckアクション

切替パーティションのマウントに失敗した場合のfsckの挙動を選択できます。
マウントリトライ回数に 0 を設定した場合は、この設定にかかわらずfsckを実行しません。

+ 実行する
  fsckを実行します。ただし、以下のMount実行前のfsckアクションによってはfsckを実行されない場合があります。
  = 「必ず実行する」設定の場合
  = 「指定回数に達したら実行する」の設定で、設定した回数に達しfsckを実行した場合

+ 実行しない
  fsckを実行しません。
  Mount実行前のfsckアクションが「実行しない」の場合との組み合わせは推奨しません。
  この設定では、ディスクリソースはfsckを実行しないため、切替パーティションにfsckで修復可能な異常があった場合、ディスクリソースをフェイルオーバできません。

1.3.5 共有ディスクリソースに関する注意事項

* 全サーバで同一パーティションに対して、同一デバイス名でアクセスできるように設定してください。

* 共有ディスクに対してLinuxのmdによるストライプセット、ボリュームセット、ミラーリング、パリティ付ストライプセットの機能はサポートしていません。

* ファイルシステムのアクセス制御（mount/umount）は、CLUSTERPROが行いますので、OS側でmount/umountする設定を行わないでください。

* ディスクリソースに設定されたパーティションデバイス名はクラスタ内の全サーバでリードオンリーの状態になります。グループを活性するサーバで、活性時にリードオンリーは解除されます。

* CLUSTERPROのバージョンによってfsckの実行タイミングが異なります。
  = mount実行前に必ずfsckを実行
    - SE3.0-1～3.0-3

  = mount失敗時にのみfsckを実行
    - SE3.0-4以降
    - XE3.0-1以降
    - SX3.1-2以降

  = fsckの実行タイミングを調節可能
    - SE3.1-6以降
    - XE3.1-6以降
    - SX3.1-6以降

* CLUSTERPROのバージョンが3.0-1～3.1-7の場合、ディスクリソースのマウントポイントにシンボリックリンクを指定しないでください。パスの一部に含まれる場合も同様です。

* CLUSTERPROのバージョンが3.1-8以降の場合、ディスクリソース、VxVMボリュームリソース、NASリソースのmount/umountは同一サーバ内で排他的に動作するため、ディスクリソースの活性/非活性に時間がかかることがあります。

1.4 フローティングIPリソース

1.4.1 CLUSTERPROのバージョン

以下のCLUSTERPROのバージョンでサポートします。

CLUSTERPRO	Version
サーバ	SE3.0-1 以降、LE3.0-1 以降、XE3.0-1 以降、SX3.1-2 以降
トレッキングツール	対応するバージョンについては動作環境編 トレッキングツールの動作環境を参照してください。

1.4.2 依存関係

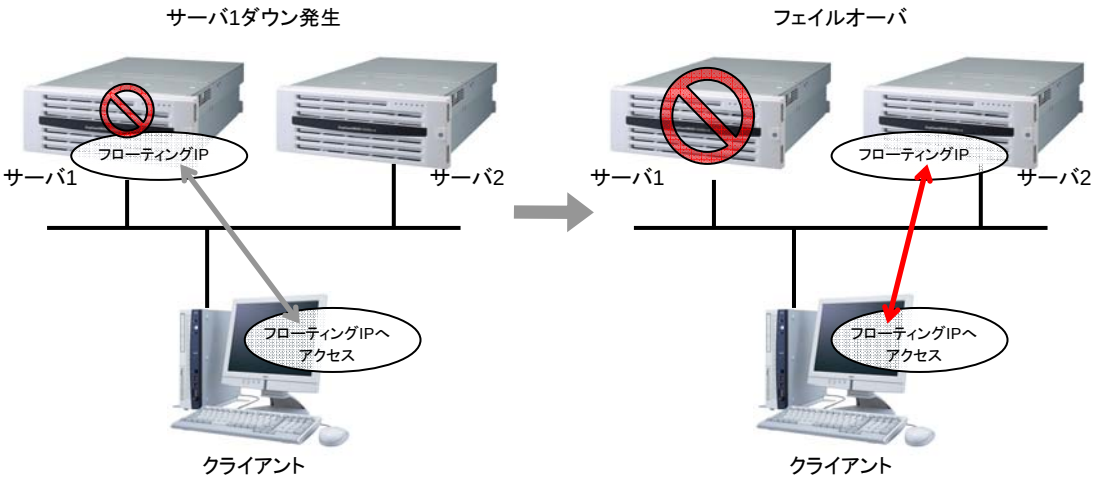
規定値では、依存するグループリソースタイプはありません。

グループリソースタイプ	Edition
-	-

1.4.3 フローティングIP

クライアントアプリケーションは、フローティングIPアドレスを使用してクラスタサーバに接続することができます。フローティングIPアドレスを使用することにより、“フェイルオーバ”または、“グループの移動”が発生しても、クライアントは、接続先サーバの切り替えを意識する必要がありません。

フローティングIPアドレスは、同一LAN上でもリモートLANからでも使用可能です。



(1) アドレスの割り当て

フローティングIPアドレスに割り当てるIPアドレスは、以下の条件を満たす必要があります。

クラスタサーバが所属するLANと同じネットワークアドレス内で かつ 使用していないホストアドレス

この条件内で必要な数(一般的にはフェイルオーバグループ数分)のIPアドレスを確保してください。

このIPアドレスは一般のホストアドレスと変わらないため、インタネットなどのグローバルIPアドレスから割り当てることも可能です。

(2) 切替方式

サーバからのARPブロードキャストにより、ARPテーブル上のMACアドレスが切り替わります。

ARPブロードキャストパケットの内容は以下になります。

0	1	2	3
ff	ff	ff	ff
ff	ff	MACアドレス	
(6byte)			
08	06	00	01
08	00	06	04
00	02		
MACアドレス(6byte)			
FIPアドレス(4byte)			
MACアドレス(6byte)			
		FIPアドレス	
(4byte)		00	00
00	00	00	00
00	00	00	00
00	00	00	00
00	00	00	00

(3) 経路制御

ルーティングテーブルの設定は不要です。

(4) 使用条件

以下のマシンからフローティングIPアドレスにアクセスできます。

* クラスタサーバ自身
* 同一クラスタ内の他のサーバ、他のクラスタシステム内のサーバ
* クラスタサーバと同一LAN内 及び リモートLANのクライアント

さらに以下の条件であれば上記以外のマシンからでもフローティングIPアドレスが使用できます。但し、すべてのマシン、アーキテクチャの接続を保障できません。事前に充分に評価をしてください。

* 通信プロトコルがTCP/IPであること
* ARPプロトコルをサポートしていること

スイッチングHUBにより構成されたLANであっても、フローティングIPアドレスのメカニズムは問題なく動作します。

サーバダウン時には、接続していたTCP/IPコネクションは切断されます。

1.4.4 サーバ別フローティングIPアドレス

以下のバージョンの場合、サーバごとにフローティングIPアドレスを設定することが可能です。

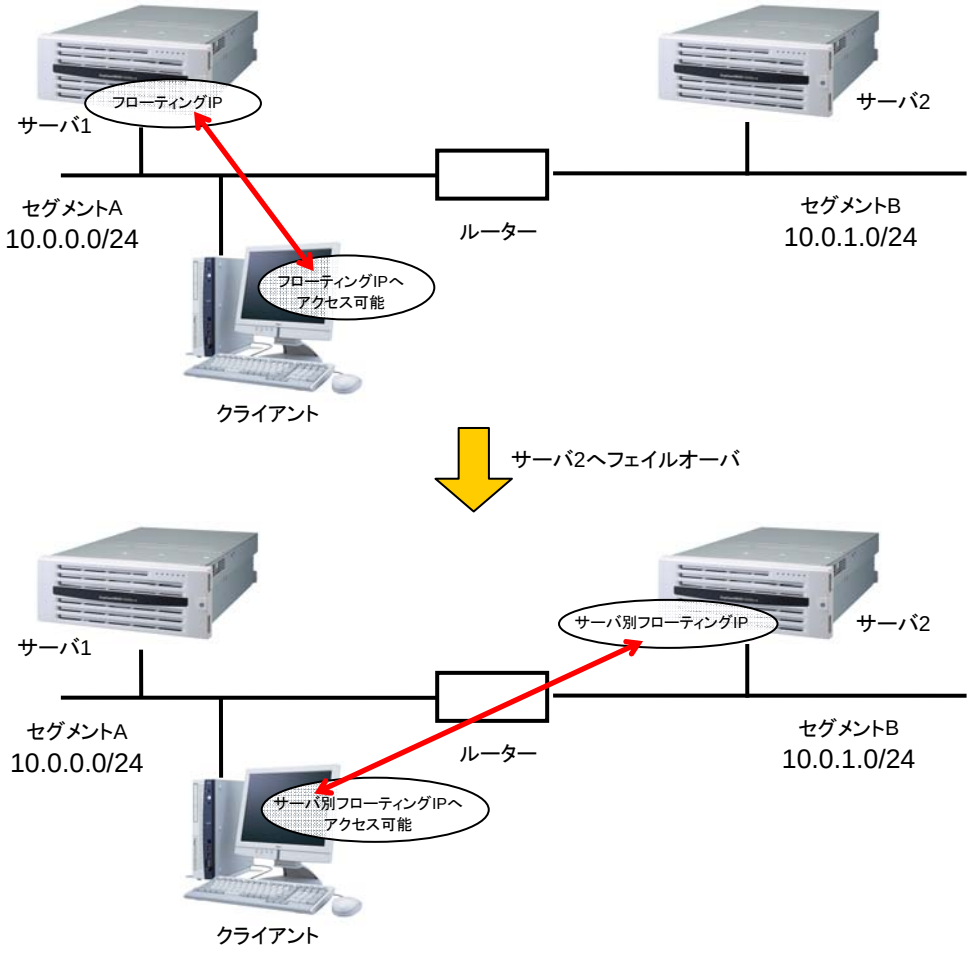
CLUSTERPRO	Version
サーバ	SE3.1-6 以降、LE3.1-6 以降、XE3.1-6 以降、SX3.1-6 以降
トレッキングツール	3.1-6 以降

サーバ別フローティングIPアドレスの設定をすることで異なるネットワークセグメントでのクラスタ化を実現できます。

例）フローティングIPリソース fip1 の設定が以下の場合

* セグメントA（10.0.0.0/24）にあるサーバ1の設定
  フローティングIPアドレス ：10.0.0.1/24

* セグメントB（10.0.1.0/24）にあるサーバ2の設定
  サーバ別フローティングIPアドレス ：10.0.1.1/24



* クライアントからの接続にサーバ別フローティングIPを使用する場合には、フェイルオーバが発生した後、フェイルオーバー先のサーバのサーバ別フローティングIPアドレスで再接続する必要があります。

1.4.5 フローティングIPリソース非活性待ち合わせ処理

以下のバージョンでフローティングIPリソースを待ち合わせる処理をおこないます。

CLUSTERPRO	Version
サーバ	SE3.1-8 以降、LE3.1-8 以降、SX3.1-8 以降
トレッキングツール	-

ifconfigコマンドによるフローティングIPアドレスの非活性化を実行した後に以下の処理を行います。

* この設定はトレッキングツールで変更できません。

(1) ifconfig コマンドによる待ち合わせ処理

+ ifconfigコマンドを実行し、OSに付加されているIPアドレスの一覧を取得します。IPアドレスの一覧にフローティングIPアドレスが存在しなければ非活性と判断します。
+ IPアドレスの一覧にフローティングIPアドレスが存在する場合は、1秒間待ち合わせます。
+ 上記の処理を最大4回繰り返します。

(2) ping コマンドによる非活性確認処理

+ pingコマンドを実行し、フローティングIPアドレスからの応答有無を確認します。フローティングIPアドレスから応答がなければ非活性と判断します。
+ フローティングIPアドレスから応答がある場合は、1秒間待ち合わせます。
+ 上記の処理を最大4回繰り返します
+ ping コマンドはタイムアウト1秒で実行します。

1.4.6 フローティングIPリソースに関する注意事項

(1) IPアドレス重複についての注意事項 1

本注意事項は、以下のバージョンの場合参照してください。

CLUSTERPRO	Version
サーバ	SE3.1-6 以降、LE3.1-6 以降、XE3.1-6 以降、SX3.1-6 以降
トレッキングツール	3.1-6 以降

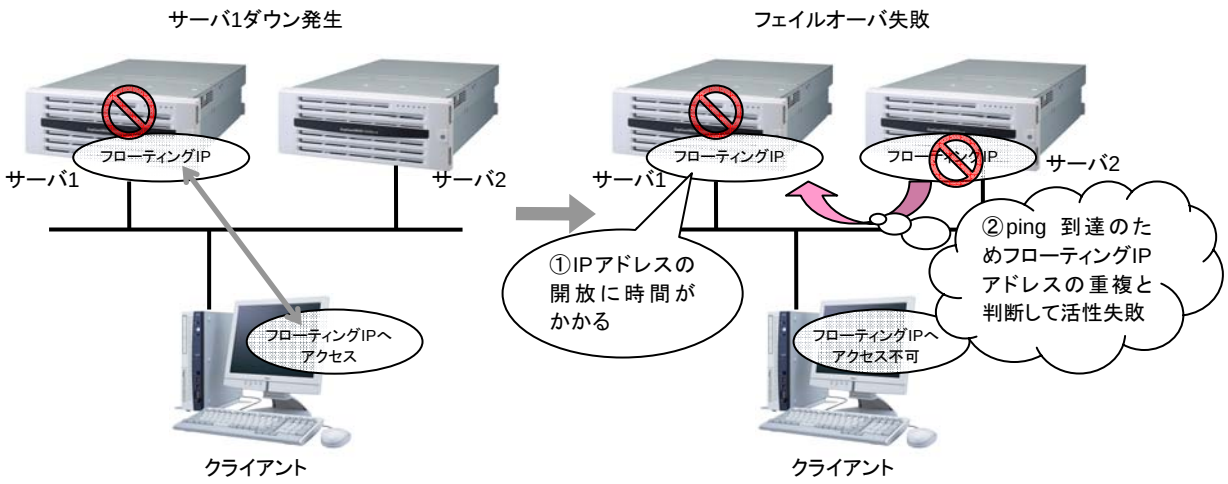
A. フローティングIPリソースで以下の設定の場合、リソースのフェイルオーバに失敗することがあります。
- 「活性リトライしきい値」に規定値より小さい値を設定している場合
- 「Pingリトライ回数」、「Pingインターバル」を設定していない場合

この現象は以下の原因で発生します。

1. フェイルオーバ元となるサーバでフローティングIPアドレスを非活性後、ifconfigコマンドの仕様によりIPアドレスの開放に時間がかかることがある

2. フェイルオーバ先のサーバでフローティングIPアドレスの活性時に、二重活性防止のために活性予定のフローティングIPアドレスに対してpingコマンド実行を実行すると、上記1 のためにpingが到達し、リソース活性異常となる

この現象は以下の設定をすることで回避することができます。
- リソースの「活性リトライしきい値」を大きくする（規定値 5回）
- 「Pingリトライ回数」、「Pingインターバル」を大きくする



B. フローティングIPアドレスを活性した状態でOSのストールが発生した場合、以下の設定の場合にリソースのフェイルオーバに失敗することがあります。
- 「Pingタイムアウト」に 0 以外を指定した場合
- 「FIP強制活性」が Off の場合

この現象は以下の原因で発生します。
1. フローティングIPアドレスを活性した状態で以下のようなOSの部分的なストールが発生
   - ネットワークモジュールは動作し、他ノードからのpingに反応する状態
   - ユーザ空間モニタリソースでストール検出不可の状態

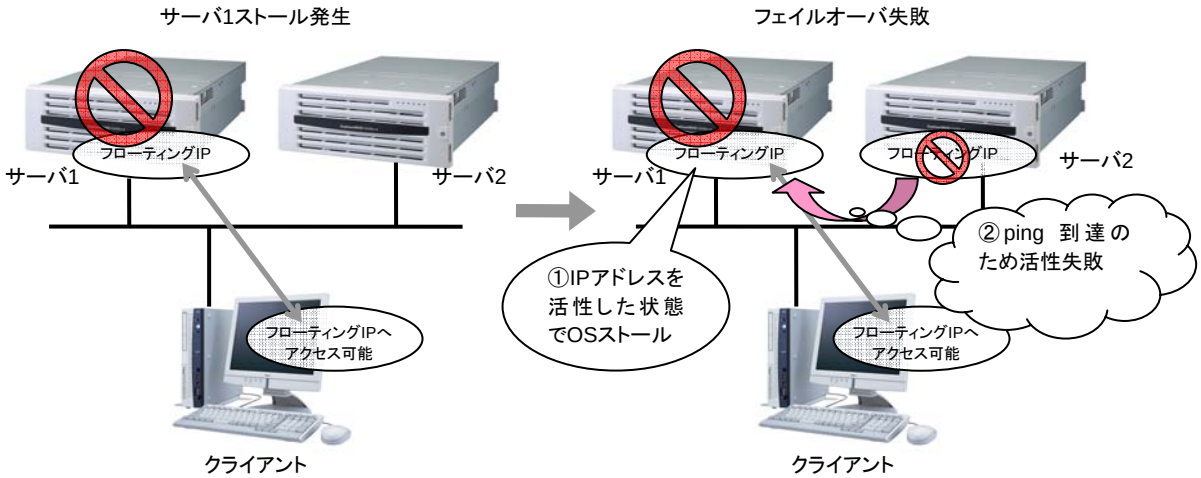
2. フェイルオーバ先のサーバでフローティングIPアドレスの活性時に、二重活性防止のために活性予定のフローティングIPアドレスに対してpingコマンド実行を実行すると、上記1 のためにpingが到達し、リソース活性異常となる

この現象が多発するマシン環境では以下の設定をすることで回避することが出来ます。ただし、フェイルオーバ後、ストールの状況によってはグループの両系活性が発生することがあり、タイミングによってサーバシャットダウンが起こるので注意してください。両系活性の詳細は「メンテナンス編」を参照してください。

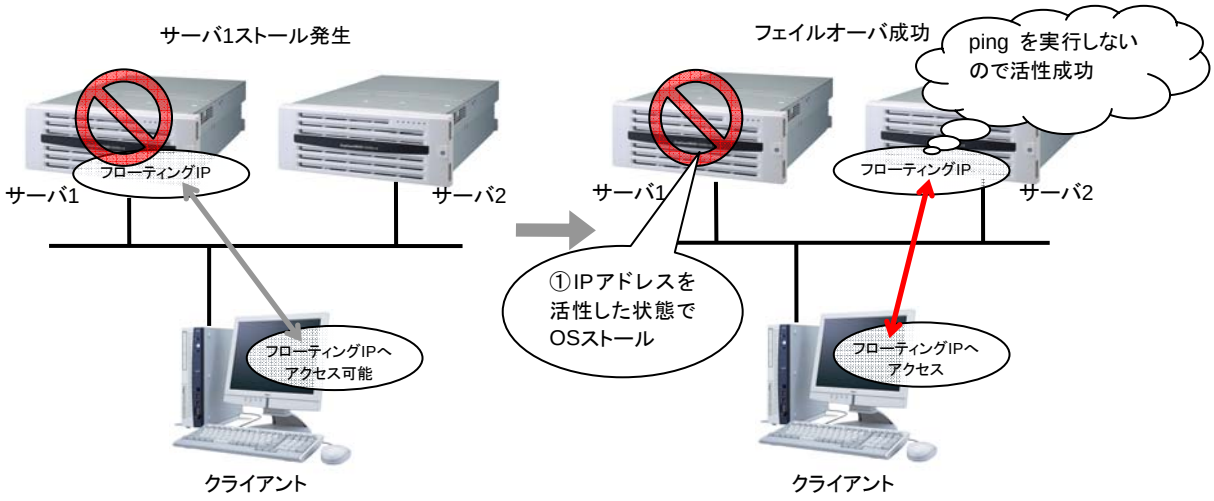
- 「Pingタイムアウト」に 0 を設定する
  フローティングIPアドレスに対して重複確認を行いません。

- 「FIP強制活性」を On に設定する
  フローティングIPアドレスが他のサーバで使用されている場合でも、強制的にフローティングIPアドレスを活性します。

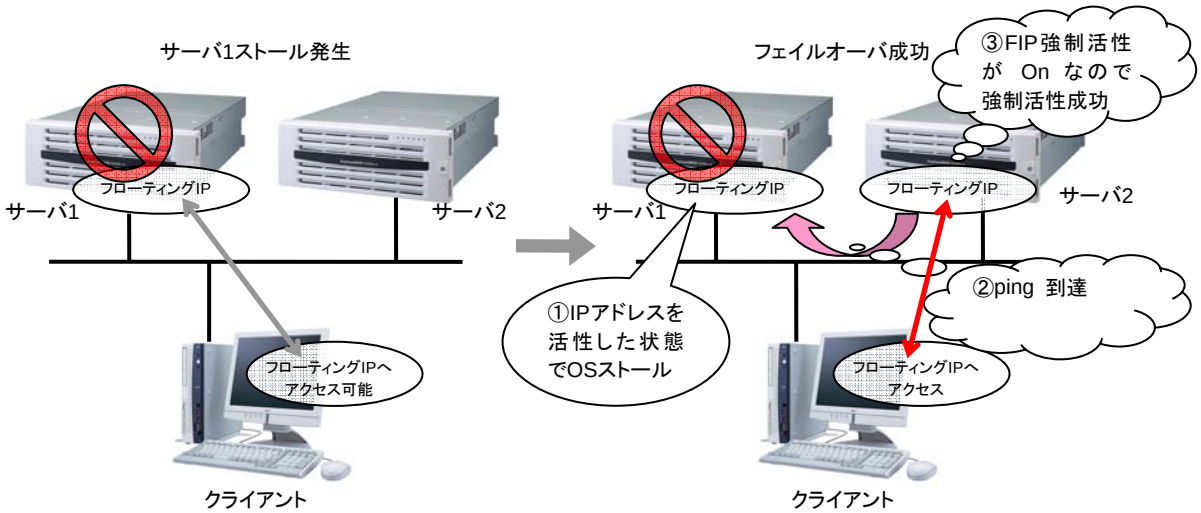
* Pingタイムアウトの設定が 0 以外かつ FIP強制活性が Off の場合



* Pingタイムアウトの設定が 0 の場合



* Pingタイムアウトの設定が 0 以外かつ FIP強制活性が On の場合



(2) IPアドレス重複についての注意事項 2

本注意事項は、以下のバージョンの場合参照してください。

CLUSTERPRO	Version
サーバ	SE3.1-5 以前、LE3.1-5 以前、XE3.1-5 以前、SX3.1-5 以前
トレッキングツール	3.1-5 以前

A. フローティングIPリソースの「活性リトライしきい値」の設定値に小さい値を設定している場合、リソースのフェイルオーバに失敗する場合があります。

この現象の原因は「1.4.6(1)A」を参照してください。

この現象を防ぐために、フローティングIPリソースの「活性リトライしきい値」の規定値は 5回となっています。(推奨)
「活性リトライしきい値」の設定を変更する場合は十分注意してください。

B. フローティングIPアドレスを活性した状態でOSのストールが発生した場合、フローティングIPアドレスの「Pingタイムアウト」の設定によって、リソースのフェイルオーバに失敗する場合があります。

この現象の原因は「1.4.6(1)B」を参照してください。

「Pingタイムアウト」の規定値はリソース活性時にpingを実行する設定になっています。
「Pingタイムアウト」の設定を 0 にすることで、活性予定のフローティングIPアドレスに対してpingを実行しません。詳細は「トレッキングツール編」を参照してください。
この現象が多発するマシン環境ではこの設定を行うことで回避することが出来ます。ただし、フェイルオーバ後、ストールの状況によってはグループの両系活性が発生することがあり、タイミングによってサーバシャットダウンが起こるので注意してください。両系活性の詳細は「メンテナンス編」を参照してください。

(3) その他注意事項

* CLUSTERPROのバージョンが3.1-8以降の場合、ifconfig コマンドによるIPアドレス一覧の取得、およびフローティングIPリソースの活性/非活性の各処理は60秒(固定)でタイムアウトします。

1.5 ミラーディスクリソース

1.5.1 CLUSTERPROのバージョン

以下のCLUSTERPROのバージョンでサポートします。

CLUSTERPRO	Version
サーバ	LE3.0-1 以降
トレッキングツール	対応するバージョンについては動作環境編 トレッキングツールの動作環境を参照してください。

1.5.2 依存関係

規定値では、以下のグループリソースタイプに依存します。

グループリソースタイプ	Edition
フローティングIPリソース	LE

1.5.3 ミラーディスク

(1) ミラーディスク

+ ミラーディスクは、クラスタを構成する2台のサーバ間でディスクデータのミラーリングを行うディスクのペアのことです。
+ ミラーディスクとして使用するにはディスクの緒元をサーバ間で同じにする必要があります。

* ディスクのタイプ

両サーバのミラーディスクまたは、ミラー用のパーティションを確保するディスクは、ディスクのタイプを同じにしてください。

動作確認済みのディスクのタイプについては「動作環境編」を参照してください。

例）

組み合わせ	サーバ1	サーバ2
OK	SCSI	SCSI
OK	IDE	IDE
NG	IDE	SCSI

* ディスクのジオメトリ

両サーバのミラーディスクまたは、ミラー用のパーティションを確保するディスクは、ディスクのジオメトリを同じにしてください。

両サーバで同じモデルのディスクを使用することを推奨します。

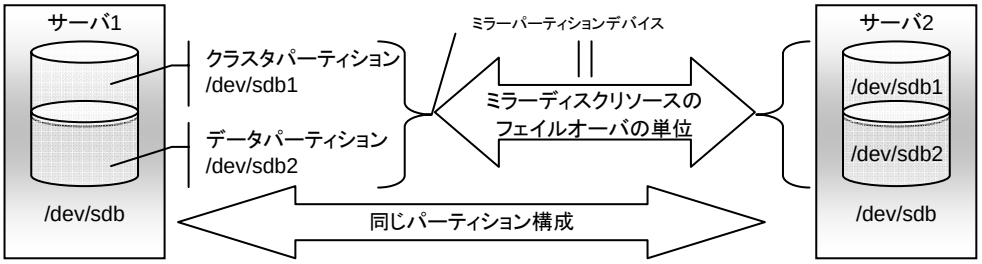
例）

組み合わせ		ヘッド	セクタ	シリンダ
OK	サーバ1	240	63	15881
	サーバ2	240	63	15881
NG	サーバ1	240	63	15881
	サーバ2	120	63	31762

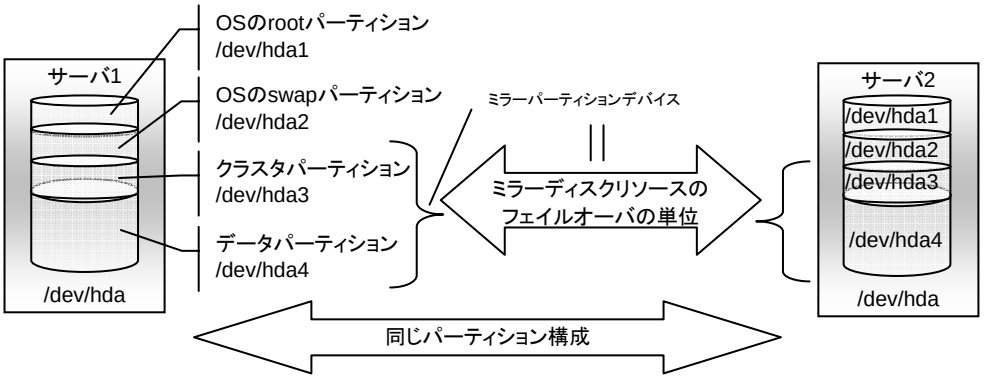
*　ディスクのパーティション

両サーバで同一パーティションに対して、同一デバイス名でアクセスできるように設定してください。

例）両サーバに1つのSCSIディスクを増設して1つのミラーディスクのペアにする場合



例）両サーバのOSが格納されているIDEディスクの空き領域を使用して
ミラーディスクのペアにする場合



+ ミラーパーティションデバイスはCLUSTERPROのミラーリングドライバが上位に提供するデバイスです。

+ フェイルオーバはミラーパーティションデバイス単位に実行されます。

+ ミラーパーティションデバイスのマウント、アンマウントはCLUSTERPROが行います。OSのfstabにエントリする必要はありません。ユーザが直接ミラーパーティションデバイスを操作することは避けてください。

+ ミラーパーティションのスペシャルデバイス名は /dev/NMPx(xは数字の1～8) を使用します。他のデバイスドライバでは、/dev/NMPxを使用しないでください。

+ ミラーパーティションのメジャー番号の218 を使用します。他のデバイスドライバでは、メジャー番号の218を使用しないでください。

+ クラスタパーティションとデータパーティションの2つのパーティションをペアで確保してください。

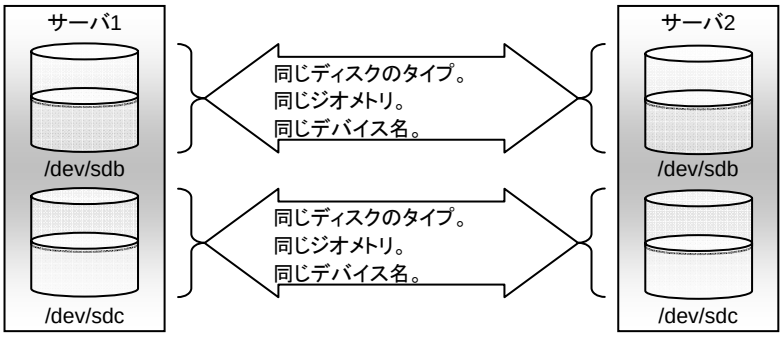
+ OS(rootパーティションやswapパーティション)と同じディスク上にミラーパーティション(クラスタパーティション、データパーティション)を確保することも可能です。

  = 障害時の保守性を重視する場合
    - OS(rootパーティションやswapパーティション)と別にミラー用のディスクを用意することを推奨します。

  = H/W Raidの仕様の制限でLUNの追加ができない場合
    H/W Raidのプリインストールモデルで LUN構成変更が困難な場合
    - OS(rootパーティションやswapパーティション)と同じディスクに
      ミラーパーティション(クラスタパーティション、データパーティション)を
      確保することも可能です。

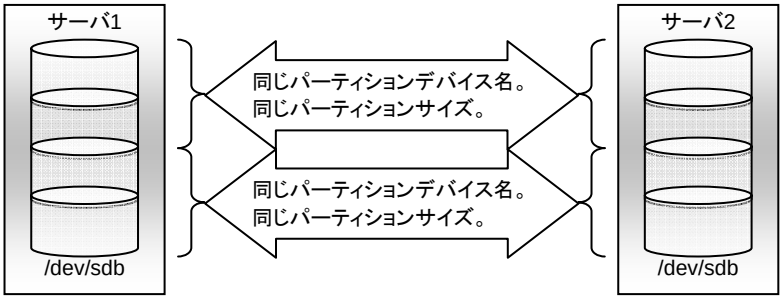
* ディスクの配置
  ミラーディスクとして複数のディスクを使用することができます。
  また1つのディスクに複数のミラーパーティションデバイスを割り当てて使用することができます。

  例）両サーバに2つのSCSIディスクを増設して2つのミラーディスクのペアにする場合。



  + 1つのディスク上にクラスタパーティションとデータパーティションをペアで確保してください。
  + データパーティションを1つ目のディスク、クラスタパーティションを2つ目のディスクとするような使い方はできません。

  例）両サーバに1つのSCSIディスクを増設して2つのミラーパーティションにする場合



* ディスクに対してLinuxのmdやLVMによるストライプセット、ボリュームセット、ミラーリング、パリティ付きストライプセットの機能はサポートしていません。

(2) データパーティション

CLUSTERPROサーバがミラーパーティションのミラーリングしたデータ（業務データ等）を格納するパーティションのことを、データパーティションといいます。

データパーティションは以下のように割り当ててください。

* データパーティションのサイズ
  1GB以上のパーティションを確保してください。
  パーティションサイズは4096バイトの倍数にしてください。ブロック数では4の倍数 [2]です。

* パーティションID
  83(Linux)

* ファイルシステムは ミラーリソースのmkfsの設定が「する」の場合にはクラスタ生成時に自動的に構築されます。

* ファイルシステムのアクセス制御(mount/umount)は、CLUSTERPROサーバがおこないますので、OS側でデータパーティションをmount/umountする設定をおこなわないでください。

(3) クラスタパーティション

CLUSTERPROサーバがミラーパーティション制御のために使用する専用パーティションを、クラスタパーティションといいます。

クラスタパーティションは以下のように割り当ててください。

* クラスタパーティションのサイズ
  最低10MB確保してください。
  ジオメトリによって10MB以上になる場合がありますが、10MB以上でも問題ありません。

* パーティションID
  83(Linux)

* クラスタパーティションは、データミラーリング用のデータパーティションとペアで割り当てる必要があります。

* クラスタパーティションにファイルシステムを構築する必要はありません。

* ファイルシステムのアクセス制御(mount/umount)は、CLUSTERPROサーバがミラーパーティションデバイスをマウントするデバイスとして行いますので、OS側でクラスタパーティションをmount/umountする設定を行わないでください。

[2] Linuxの場合、1ブロックのサイズはデフォルトでは1024バイトです。

(4) ミラーパーティションデバイス

1つのミラーディスクリソースで1つのミラーパーティションデバイスを上位に提供します。
ミラーディスクリソースとして登録すると、1台のサーバ(通常はリソースグループのプライマリサーバ)からのみアクセス可能になります。

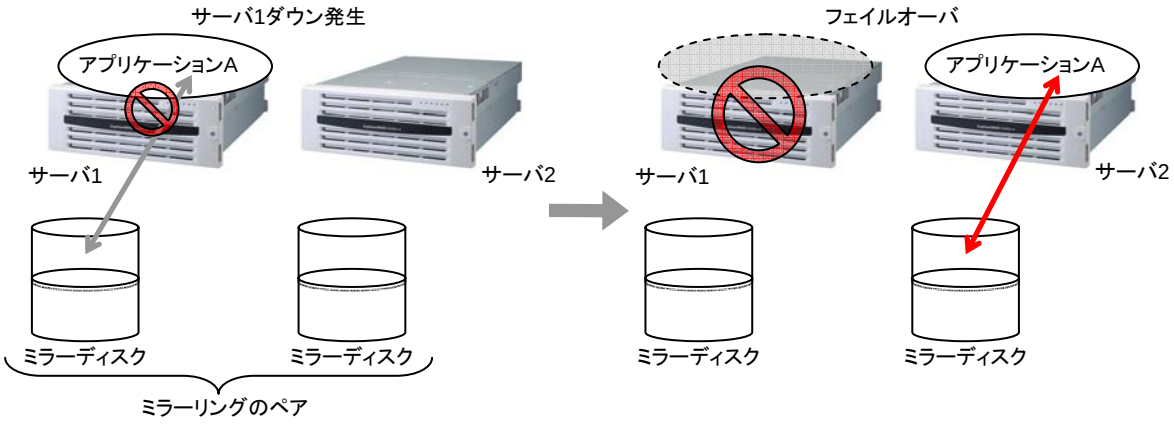
通常、ユーザ(AP)はファイルシステムを経由してI/Oを行うため、ミラーパーティションデバイス(dev/NMPx)を意識する必要はありません。トレッキングツールで情報を作成するときに自動的にデバイス名を割り当てます。

* ファイルシステムのアクセス制御(mount/umount)は、CLUSTERPROサーバがおこないますので、OS側でミラーパーティションデバイスをmount/umountする設定を行わないでください。

業務アプリケーション等からミラーパーティション（ミラーディスクリソース）へのアクセス可否の考え方は、共有ディスクを使用した切替パーティション（ディスクリソース）と同じです。

* ミラーパーティションの切り替えはフェイルオーバグループ毎に、フェイルオーバポリシにしたがって行われます。

* 業務に必要なデータは、ミラーパーティション上に格納しておくことで、フェイルオーバ時、フェイルオーバグループの移動時等に、自動的に引き継がれます。



1.5.4 ミラーパラメータ

(1) リクエストキューの最大数

ミラーディスクドライバが上位からのI/O要求をキューイングするためのキューの個数を設定します。

大きくするとパフォーマンスが向上しますが、物理メモリを多く消費します。

小さくすると物理メモリの使用量が少なくなりますが、パフォーマンスが低下する可能性があります。

以下を目安に設定してください。

= 以下のような条件では大きくすると性能向上が期待できます。
- サーバに物理メモリが多く搭載されていて空きメモリサイズが十分ある。
- ディスクのI/O性能が良い。

= 以下のような条件では小さくすることを推奨します。
- サーバに搭載されている物理メモリが少ない。
- ディスクのI/O性能が悪い。
- OSのsyslogに alloc_pages: 0-order allocation failed (gfp=0x20/0) がエントリされる。

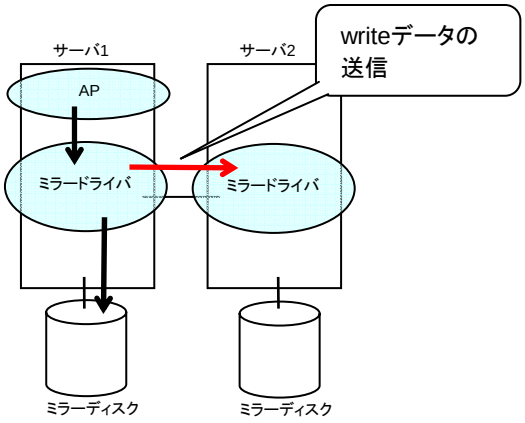
(2) 接続タイムアウト

ミラー復帰やデータ同期時に、サーバ間通信の接続成功を待つタイムアウト。

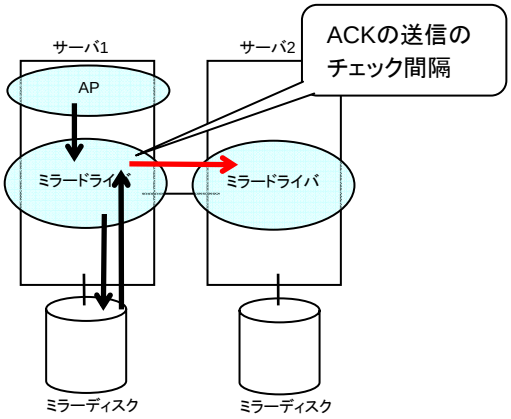
(3)　送信タイムアウト

このタイムアウトは以下で使用します。

① ミラー復帰やデータ同期時に、現用系サーバが待機系サーバにwriteデータを送信開始してから送信完了を待つまでのタイムアウト



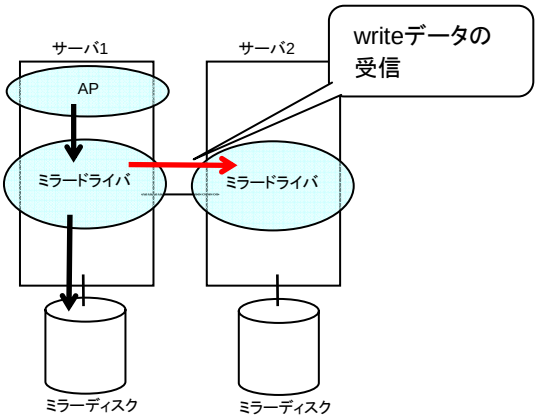
② 現用系サーバが待機系サーバへwrite完了通知のACKを送信する要否を確認する時間間隔



(4)　受信タイムアウト

このタイムアウトは下記に使用します。

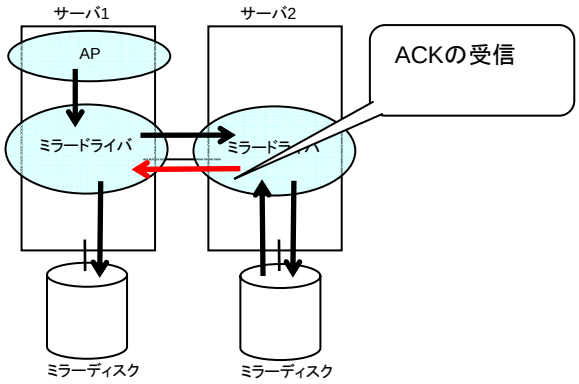
① 待機系サーバが現用系サーバからのwriteデータを受信開始してから受信完了を待つまでのタイムアウト



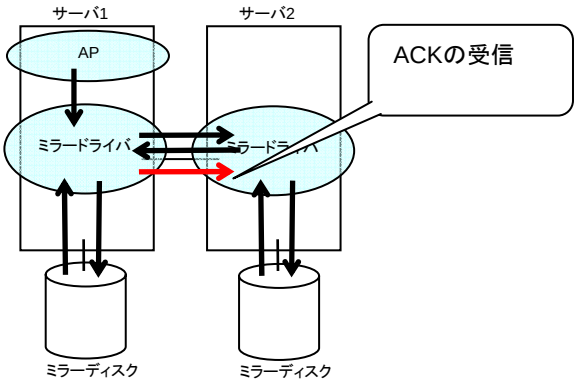
(5)　Ackタイムアウト(LE3.1-6以降)

このタイムアウトは下記に使用します。

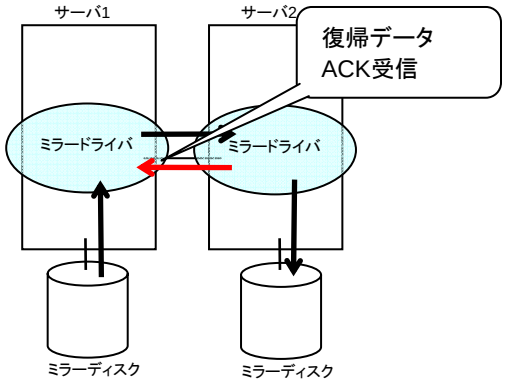
① 現用系サーバが待機系サーバへwriteデータを送信完了してから待機系サーバからのwrite完了通知のACKの受信を待つまでのタイムアウト
タイムアウト以内に、ACKを受信できないと、現用系サーバ側の差分ビットマップを蓄積します。



② 待機系サーバが現用系サーバへのwrite完了通知のACKを送信完了してから、現用系サーバのACKを受信を待つまでのタイムアウト。
タイムアウト以内に、現用系サーバのACKを受信できないと、待機系サーバ側の差分ビットマップを蓄積します。



③ ミラー復帰時に、コピー元サーバが復帰データ送信を開始してから、コピー先サーバからの受信完了通知のACKを待つまでのタイムアウト



(6) Bitmap更新間隔(LE3.1-6以降)

待機系サーバで差分ビットマップへ書き込むデータのキューをチェックする時間間隔

(7)　フラッシュスリープ時間（LE3.1-1以降）

待機系(ミラー先)側でバッファに溜まったwriteデータを定期的にディスクへ書き出すためのスレッドの待ち時間の設定をします。

* 値を大きくする
  + 待機系(ミラー先)側のOSへの負荷が低くなります。
  + write性能は低下します。

* 値を小さくする
  + 待機系(ミラー先)側のOSへの負荷が高くなります。
  + write性能は向上します。

上記の傾向はあくまで挙動の目安であり 以下の条件や環境により本パラメータを変更したときの効果が出ないことがあります。デフォルトのまま使用することを推奨します。

  + OSの種別
  + メモリサイズ
  + ファイルシステムのチューニング
  + ディスクインタフェースの種別
  + ディスクやディスクインタフェースボードの性能(キャッシュサイズ、シークタイムなど)
  + アプリケーションのwriteロジック

(8)　フラッシュカウント(LE3.1-6以降)

待機系(ミラー先)側でバッファに溜まったwriteデータのバッファブロックが指定個数になったら、ディスクへ書き出します。

* 値を大きくする
  + writeデータのサイズが小さい場合はwrite性能が低下します。
  + writeデータのサイズが大きい場合はwrite性能が向上します。

* 値を小さくする
  + writeデータのサイズが小さい場合はwrite性能が向上します。
    例えば、小さいデータをファイルシステムが行うflush動作に委ねないで
    頻繁にwriteするようなAPの場合にはwrite性能が向上する可能性があります。
  + writeデータのサイズが大きい場合はwrite性能が低下します。

上記の傾向はあくまで挙動の目安であり 以下の条件や環境により本パラメータを変更したときの効果が出ないことがあります。デフォルトのまま使用することを推奨します。

  + メモリサイズ
  + ファイルシステムのチューニング
  + アプリケーションのwriteロジック

(9) 初期ミラー構築（LE3.1-1以降）

クラスタ構築後の初回起動時に初期ミラー構築[3] を行う挙動を設定します。

A. 初期ミラー構築を行う
LE3.0-4までの動作と同様に クラスタ構築後の初回起動時に初期ミラー構築を行います。

B. 初期ミラー構築を行わない
クラスタ構築後の初回起動時に初期ミラー構築を行いません。クラスタ構築前にCLUSTERPRO以外の手段でミラーディスクの内容を同一にしておく必要があります。

(10) 初期mkfs（LE3.1-1以降）

クラスタ構築後の初回起動時にミラーディスクのデータパーティションへのmkfsの挙動を設定します。

A. 初期mkfsを行う
LE3.0-4までの動作と同様に クラスタ構築直後の初回起動時にミラーディスクのデータパーティションへのmkfsを行います。

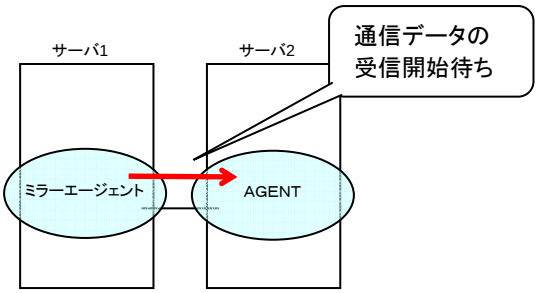
B. 初期mkfsを行わない
クラスタ構築直後の初回起動時にミラーディスクのデータパーティションへのmkfsを行いません。
ミラーディスクのデータパーティションにすでにファイルシステムが構築されていて二重化するデータがあり mkfsが不要な場合に設定します。
ミラーディスクのパーティション構成は ミラーディスクリソースの条件を満たしている必要があります。[4]
シングルサーバをクラスタ化する場合には ご注意ください。

「初期ミラー構築を行わない」を選択した場合には、「初期mkfsを行う」は選択できません。（mkfsしたパーティションはmkfsした直後でも パーティションイメージでは差分が発生しているためです）

[3] FastSync Optionの有無に関わらずデータパーティションの全容量のコピーを実行します。
[4] ミラーディスクにクラスタパーティションが必ず必要です。シングルサーバのディスクをミラーの対象とする時にクラスタパーティションを確保できない場合には、一旦バックアップを採ってパーティションを再確保してください。

(11)　ミラーエージェント受信タイムアウト(LE3.1-6以降)

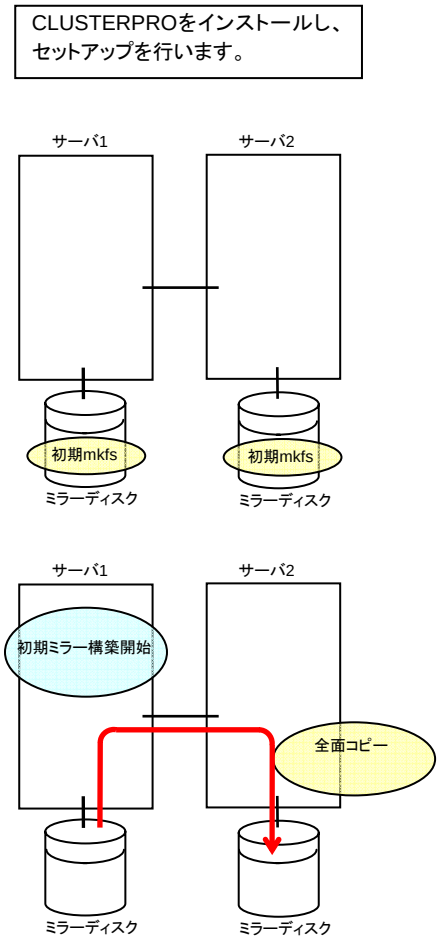
ミラーエージェントが相手サーバとの通信socketを作成してから受信開始を待つまでのタイムアウト



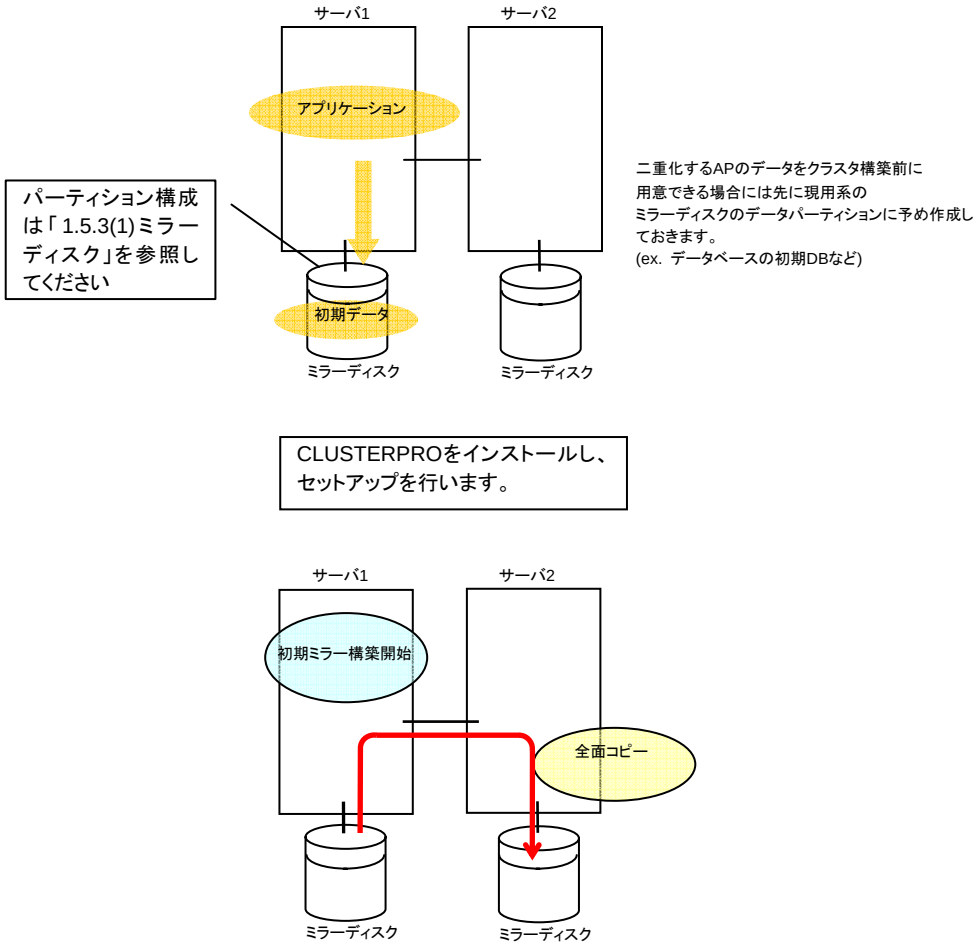
(12) 構築例

過去にミラーディスクとして使用していたディスクを流用する場合は、クラスタパーティションに以前のデータが残っているので初期化が必要です。クラスタパーティションの初期化については「メンテナンス編 ミラーディスクの流用」を参照してください。

A. 初期ミラー構築を行う
初期mkfsを行う



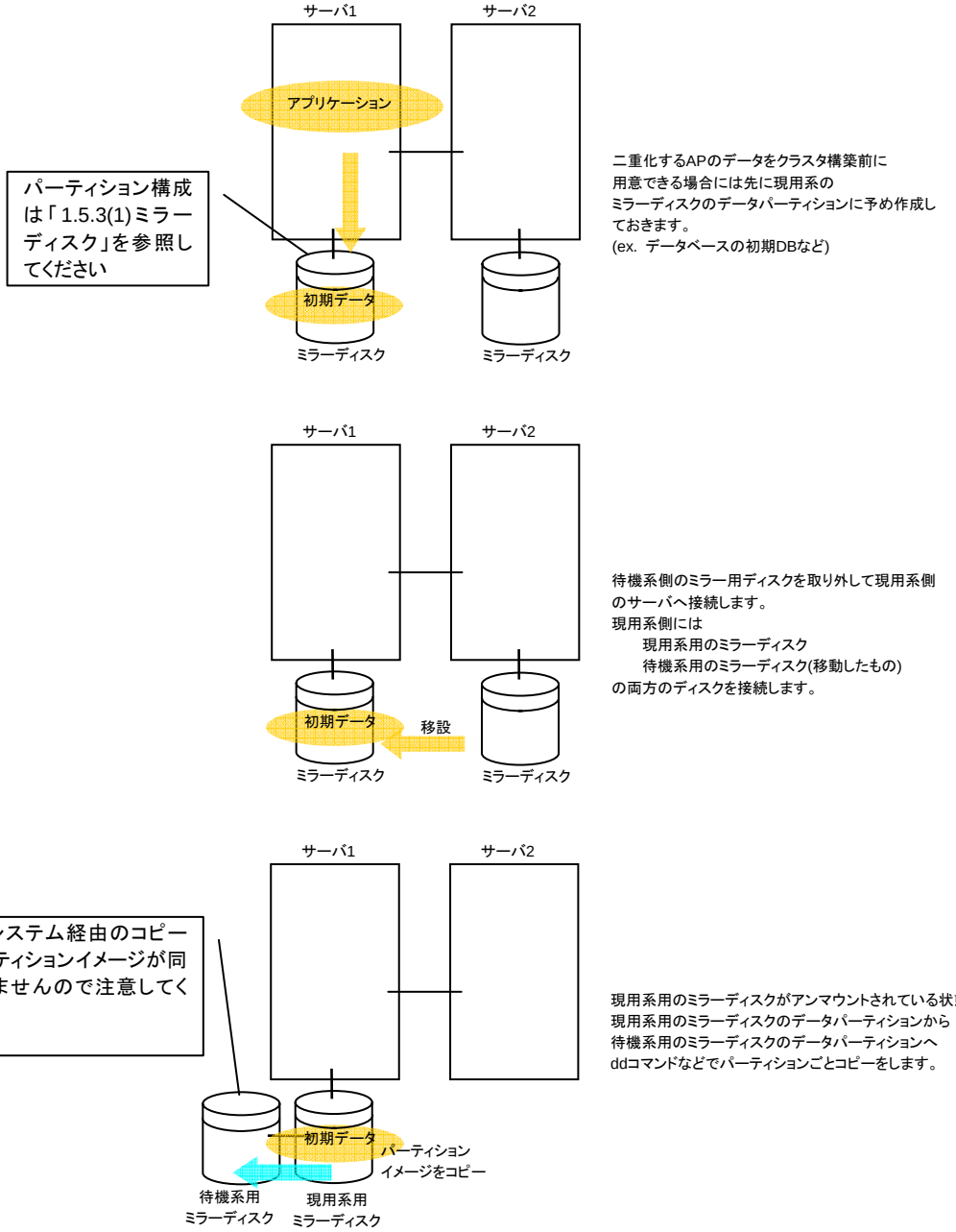
B.　初期ミラー構築を行う
　　初期mkfsを行わない

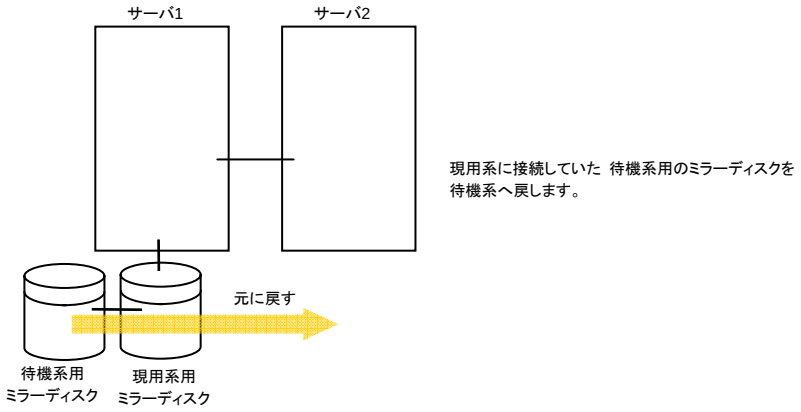


C.　初期ミラー構築を行わない
初期mkfsを行わない

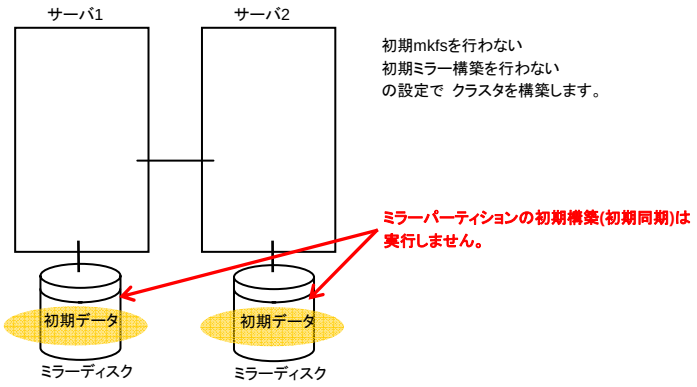
例えば以下のような方法で両サーバのミラーディスクの内容を同一にすることができます。(クラスタ構築後にはできません。必ずクラスタ構築前に実施してください。)

例1
ディスクのパーティションイメージをコピーする方法

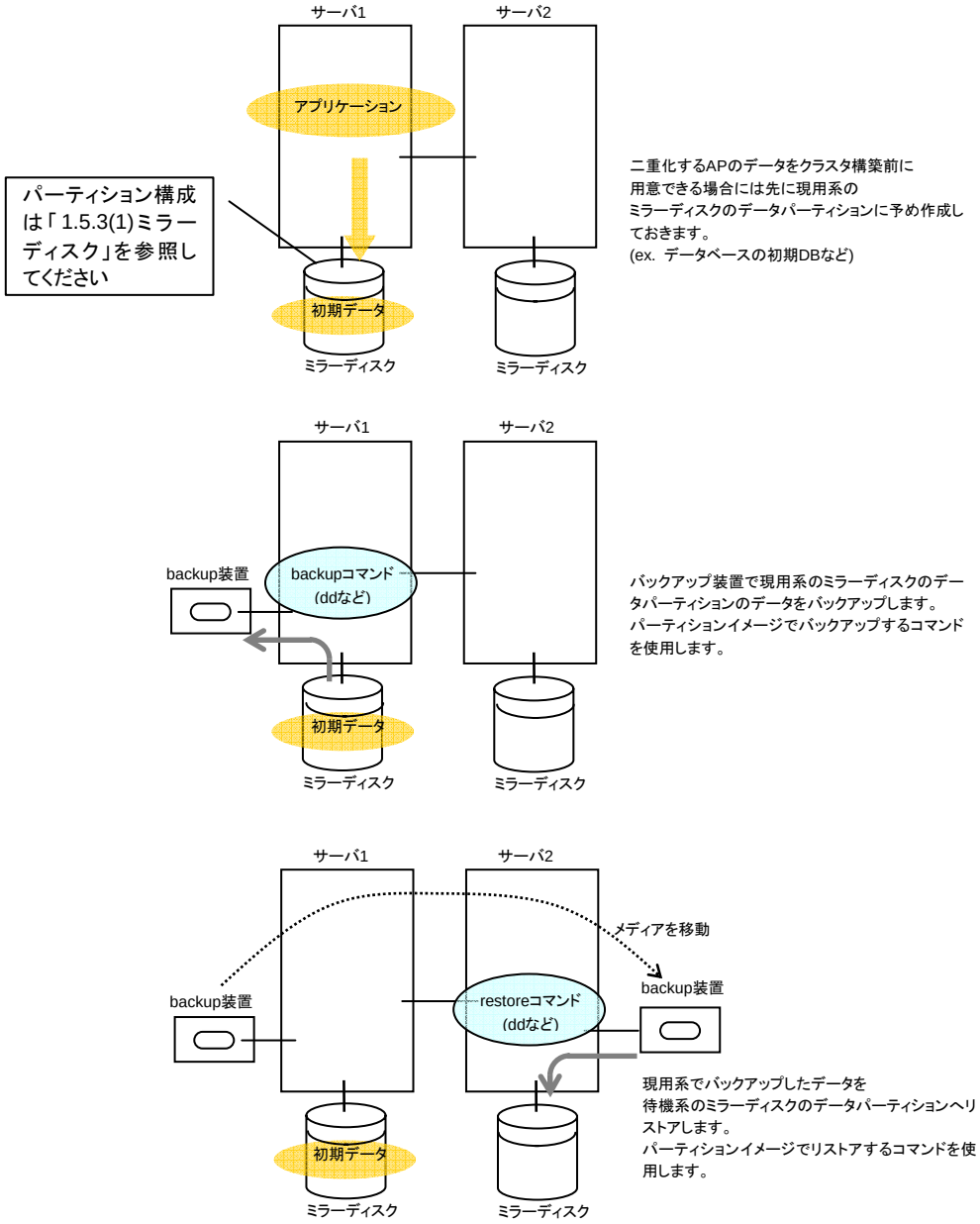




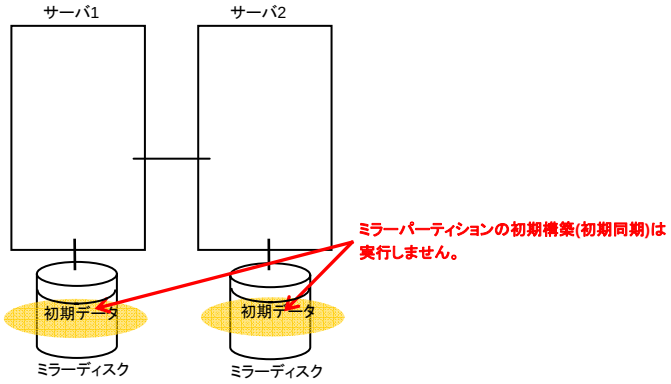
CLUSTERPROをインストールし、
セットアップを行います。



例2
バックアップ装置でコピーをする方法



CLUSTERPROをインストールし、
セットアップを行います。



1.5.5 ミラーディスクリソースに関する注意事項

* 両サーバで同一パーティションに対して、同一デバイス名でアクセスできるように設定してください。

* CLUSTERPROのバージョンによってfsckの実行タイミングが異なります。
  = mount実行前に必ずfsckを実行
    - LE3.0-1～3.0-4

  = mount失敗時にのみfsckを実行
    - LE3.1-1以降

* CLUSTERPROのバージョンが3.0-1～3.1-7の場合、ミラーディスクリソースのマウントポイントにシンボリックリンクを指定しないでください。パスの一部に含まれる場合も同様です。

1.6 RAWリソース

RAWリソースとはrawデバイスのリソースです。

rawデバイスはLinux上のデバイスで、ファイルシステムを使用しないでパーティションデバイスを直接I/Oします。一般的にファイルシステムの代わりにアプリケーションが独自のデータ構造を構築します。

1.6.1 CLUSTERPROのバージョン

以下のCLUSTERPROのバージョンでサポートします。

CLUSTERPRO	Version
サーバ	SE3.0-4 以降、XE3.1-4 以降、SX3.1-2 以降
トレッキングツール	対応するバージョンについては動作環境編 トレッキングツールの動作環境を参照してください。

1.6.2 依存関係

規定値では、以下のグループリソースタイプに依存します。

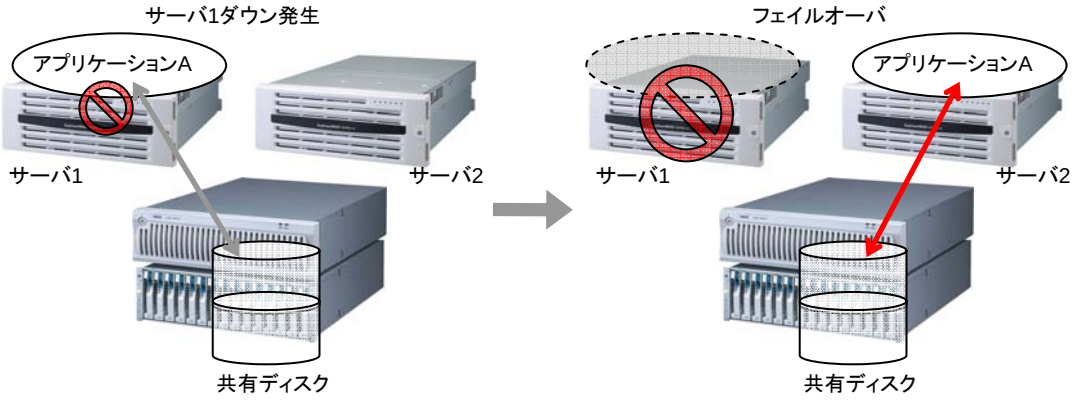
グループリソースタイプ	Edition
フローティングIPリソース	SE、XE、SX

1.6.3 切替パーティション

* 切替パーティションとは、クラスタを構成する複数台のサーバに接続された共有ディスク上のパーティションをいいます。

* 切替はフェイルオーバグループ毎に、フェイルオーバポリシにしたがって行われます。業務に必要なデータは、切替パーティション上に格納しておくことで、フェイルオーバ時、フェイルオーバグループの移動時等に、自動的に引き継がれます。

* 切替パーティションは全サーバで、同一領域に同じデバイス名でアクセスできるようにしてください。



1.6.4 RAWリソースに関する注意事項

* 同一パーティションに対して、同一デバイス名でアクセスできるように設定してください。

* 共有ディスクに対してLinuxのmdによるストライプセット、ボリュームセット、ミラーリング、パリティ付ストライプセットの機能はサポートしていません。

* RAWデバイスのアクセス制御（bind）は、CLUSTERPROが行いますので、OS側でbindする設定を行わないでください。

* グループが活性されていないサーバではパーティションはリードオンリーの状態になっています。

* 既にサーバプロパティの「ディスク I/F一覧」、「RAWモニタリソース」または「VxVMボリュームリソース」に登録されているRAWデバイスは登録しないでください。VxVMボリュームリソースのRAWデバイスについては「1.7.6 CLUSTERPROで制御する際の注意」を参照してください。

1.7 VxVM関連リソース

1.7.1 動作確認情報

1.7.1.1 CLUSTERPROのバージョン

以下のCLUSTERPROのバージョンでサポートします。

CLUSTERPRO	Version
サーバ	SE3.0-4 以降
トレッキングツール	3.0-4 以降

1.7.1.2 ディストリビューション

以下のバージョンで動作確認しています。

Distribution	kernel
Red Hat Enterprise Linux AS release 3 (Taroon)	2.4.21-4.EL
Red Hat Enterprise Linux ES release 3 (Taroon)	2.4.21-4.Elsmp

1.7.1.3 VERITAS Volume Manager のバージョン

以下のバージョンで動作確認しています。

rpm	Version	Release
VRTSvlic	3.00	009
VRTSvxvm	3.2	update5_RH3
VRTSvxfs	3.4.4	RHEL3

1.7.1.4 ボリューム上のファイルシステムについて

現在動作確認を完了しているファイルシステムは下記の通りです。

- vxfs

1.7.2 依存関係

規定値では、以下のグループリソースタイプに依存します。

* VxVMディスクグループリソース

グループリソースタイプ	Edition
フローティングIPリソース	SE

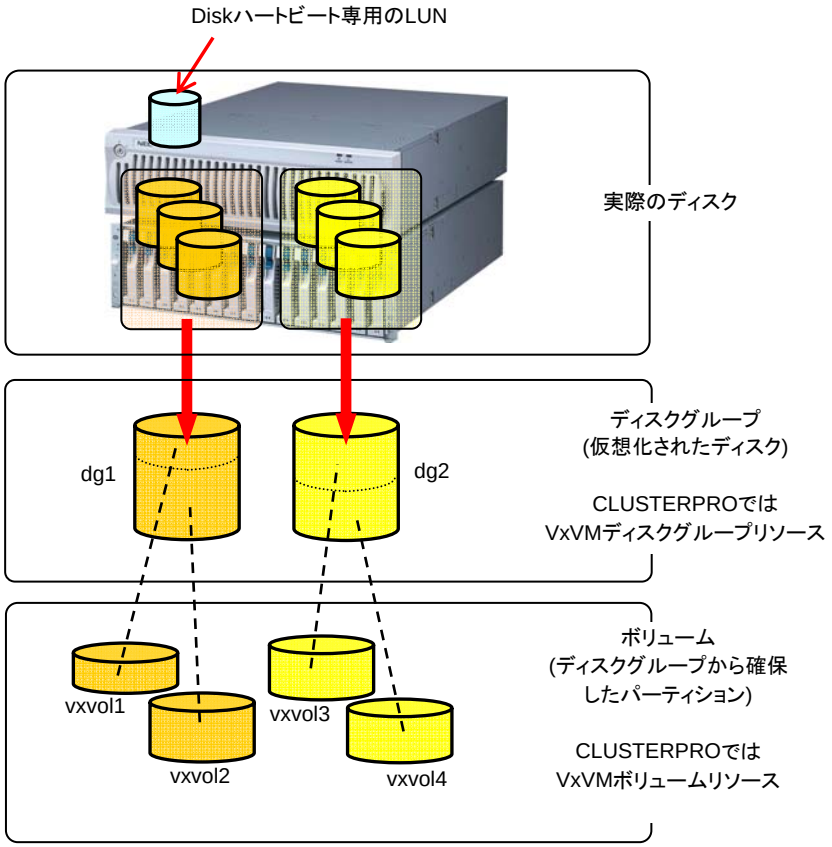
* VxVMボリュームリソース

グループリソースタイプ	Edition
フローティングIPリソース	SE
VxVMディスクグループリソース	SE

1.7.3 CLUSTERPROで制御するリソース

VERITAS Volume Manager **ディスクグループ**（以降 ディスクグループ）とは物理ディスクを仮想的にグループ化したものです。このディスクグループから確保した論理的なパーティションを**ボリューム**といいます。

CLUSTERPROは、ディスクグループとボリュームをそれぞれVxVMディスクグループリソース、VxVMボリュームリソースとして制御することができます。



1.7.4 VxVMディスクグループリソース

(1) ディスクグループについて

* ディスクグループの定義はCLUSTERPRO側で行いません。
* ディスクグループの活性(インポート)/非活性(デポート)処理はCLUSTERPROのVxVMディスクグループリソースで行います。
* CLUSTERPROの設定情報に含まれるディスクグループはOS起動時に自動的にデポート処理を行います。
* CLUSTERPROの設定情報に含まれていないディスクグループはデポートしません。

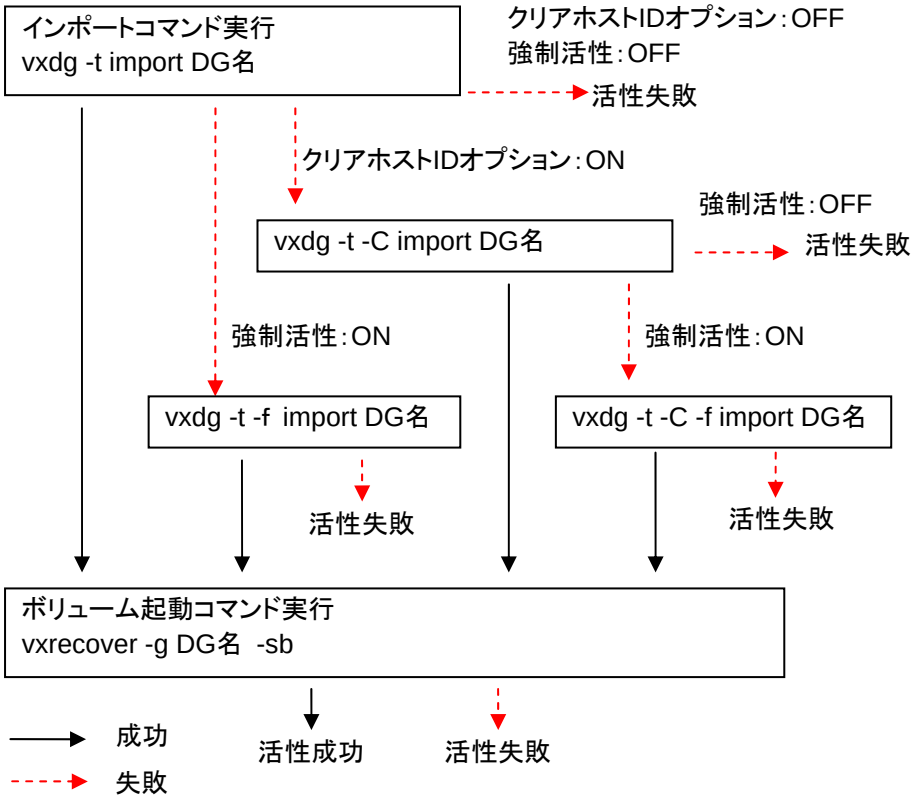
(2) 活性時に実行するコマンド

VxVMディスクグループ活性時に以下のコマンドを実行します。

コマンド	オプション	使用するタイミング
vxdg	import	ディスクグループインポート時
	-t	ディスクグループインポート時
	-C	ディスクグループインポートに失敗し、クリアホストIDオプションが ON の場合
	-f	ディスクグループインポートに失敗し、強制活性オプションが ON の場合

コマンド	オプション	使用するタイミング
vxrecover	-g	指定したディスクグループのボリューム起動時
	-sb	指定したディスクグループのボリューム起動時

活性時のシーケンス



* フェイルオーバ元サーバでディスクグループを正常にデポートできなかった場合、フェイルオーバ先サーバでは、VxVMの仕様により、クリアホストIDオプションが OFF の場合はディスクグループをインポートできません。

* インポートタイムアウトが発生した場合、実際にはインポートが成功していることがあります。インポートオプションに、ホストIDクリアもしくは強制インポートオプションを設定している場合、インポートリトライを行い、この現象を回避することができます。(SE3.1-6以降)

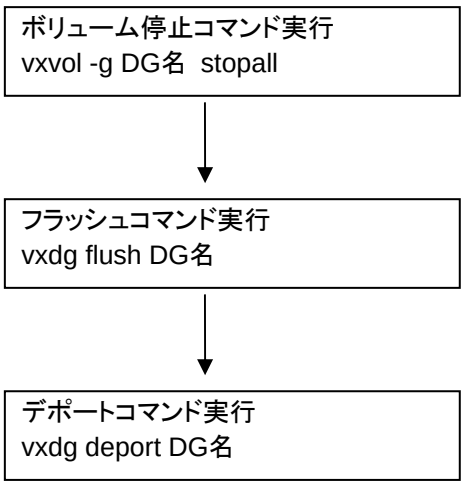
(3) 非活性時に実行するコマンド

VxVMディスクグループ非活性時に以下のコマンドを実行します。

コマンド	オプション	使用するタイミング
vxdg	deport	ディスクグループデポート時
	flush	フラッシュ時

コマンド	オプション	使用するタイミング
vxvol	-g	指定したディスクグループのボリューム停止時
	stopall	指定したディスクグループのボリューム停止時

非活性時のシーケンス



1.7.5 VxVMボリュームリソース

* ボリュームについて
  + ボリュームの定義はCLUSTERPRO側で行いません。
  + ボリューム上のファイルシステムのマウント/アンマウントはCLUSTERPROのVxVMボリュームリソースで行います。
  + ディスクグループをインポートしボリュームが起動された状態でアクセス可能なrawデバイス(/dev/vx/rdsk/[ディスクグループ名]/[ボリューム名]) のみを使用する場合 (=ボリューム上にファイルシステムを構築しないでrawアクセスを行う場合) には VxVMボリュームリソースは不要です。

* 依存関係について
  + デフォルトの依存関係は「VxVMボリュームリソースはVxVMディスクグループリソースに依存する」設定になっています。

* CLUSTERPROのバージョンによってfsckの実行タイミングが異なります。
  = mount失敗時にのみfsckを実行
    - SE3.0-4以降
    - XE3.0-1以降
    - SX3.1-2以降

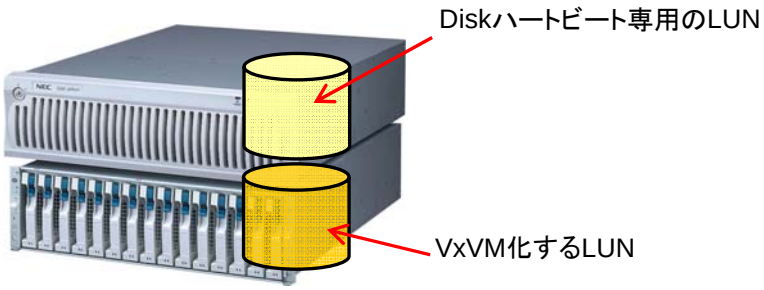
  = fsckの実行タイミングを調節可能
    （詳細は「1.3.4 fsck実行タイミング」を参照してください）
    - SE3.1-6以降
    - XE3.1-6以降
    - SX3.1-6以降

* CLUSTERPROのバージョンが3.0-1～3.1-7の場合、VxVMボリュームリソースのマウントポイントにシンボリックリンクを指定しないでください。パスの一部に含まれる場合も同様です。

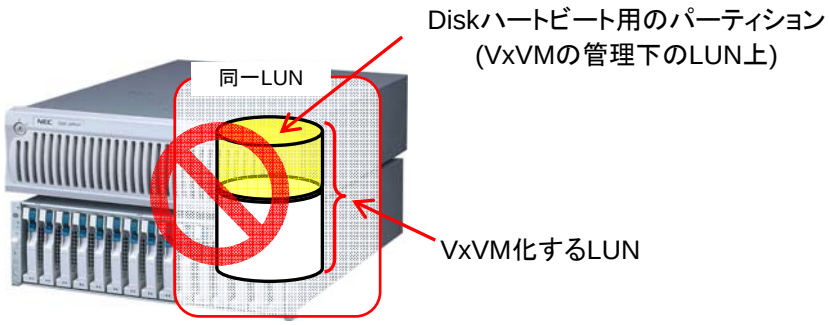
* CLUSTERPROのバージョンが3.1-8以降の場合、ディスクリソース、VxVMボリュームリソース、NASリソースのmount/umountは同一サーバ内で排他的に動作するため、VxVMボリュームリソースの活性/非活性に時間がかかることがあります。

1.7.6 CLUSTERPROで制御する際の注意事項

(1) Diskハートビート専用のLUNを確保してください。



ディスクグループに追加するディスクは物理ディスク単位で追加します。ディスクグループはどちらか片方のサーバでのみインポートされます。したがって、両サーバから同時にアクセスが必要なDiskハートビート用のパーティションは、ディスクグループに追加するディスクと同一LUNに持つことはできません。



(2)　ボリュームRAWデバイスの実RAWデバイスについて事前に調べておいてください。

CLUSTERPROをインストールする前に、片サーバで活性しうる全てのディスクグループをインポートし、全てのボリュームを起動した状態にします。

以下のコマンドを実行します。

```
# raw -qa
/dev/raw/raw2    bound to major 199, minor 2
/dev/raw/raw3    bound to major 199, minor 3
```

①　②

例）ディスクグループ名、ボリューム名がそれぞれ以下の場合

＋　ディスクグループ名　dg1

＋　dg1配下のボリューム名　vol1、vol2

以下のコマンドを実行します。

```
# ls -l /dev/vx/dsk/dg1/
brw-------    1 root     root     199,     2 5月  15 22:13 vol1
brw-------    1 root     root     199,     3 5月  15 22:13 vol2
```

③

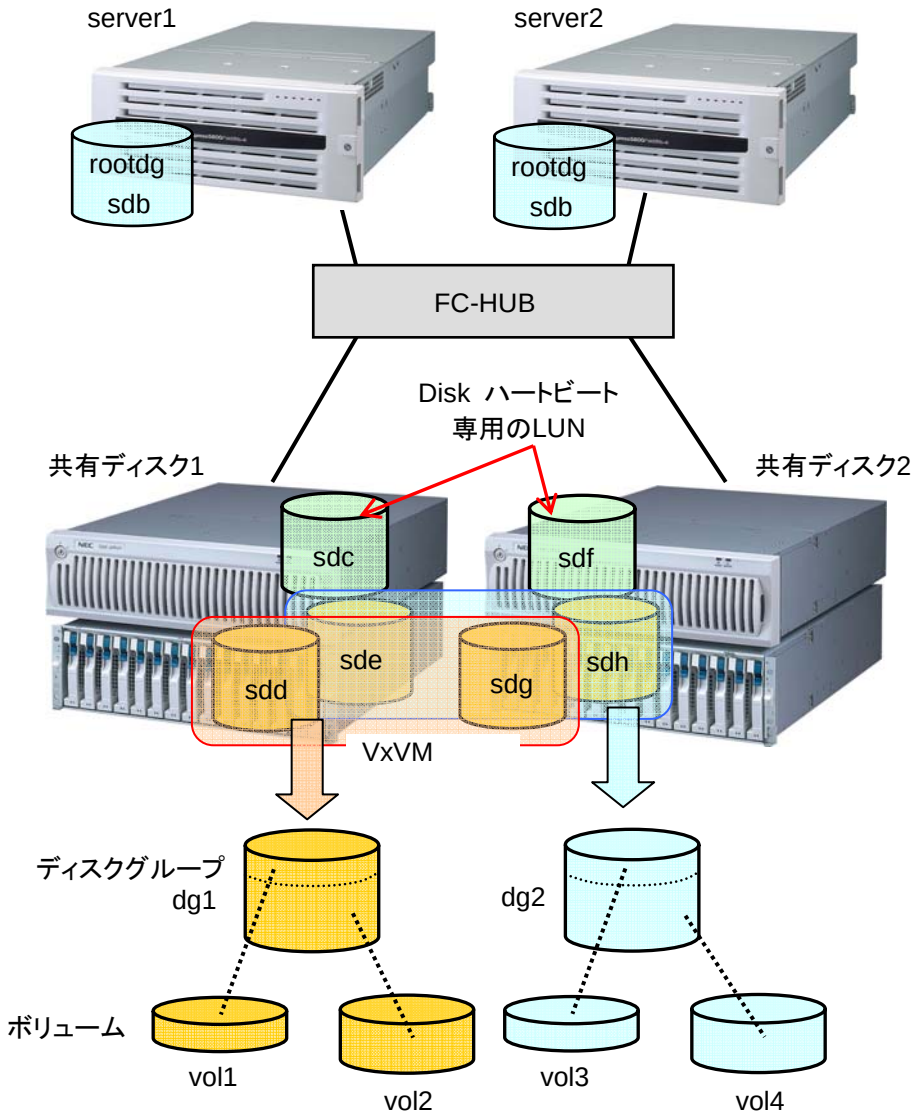
②と③のメジャー/マイナー番号が等しいことを確認します。

これにより確認されたRAWデバイス①はCLUSTERPROのDiskハートビートリソース、RAWリソース、RAWモニタリソースでは絶対に設定しないでください。

1.7.7 VERITAS Volume Manager を用いたクラスタ構築

1.7.7.1 VERITAS Volume Manager の構成

CLUSTERPROで動作確認済みのVERITAS Volume Manager の構成は以下のようになります。



前頁のVxVMの設定例は以下のようになります。

ディスクグループ1			
dg1	物理ディスク1	/dev/sdd	
	物理ディスク2	/dev/sdg	
	ボリューム		
	vol1 *1	ボリュームデバイス名	/dev/vx/dsk/dg1/vol1
		ボリュームRAWデバイス名	/dev/vx/rdsk/dg1/vol1
		ファイルシステム	vxfs
	vol2 *1	ボリュームデバイス名	/dev/vx/dsk/dg1/vol2
		ボリュームRAWデバイス名	/dev/vx/rdsk/dg1/vol2
		ファイルシステム	vxfs
ディスクグループ2			
dg2	物理ディスク1	/dev/sde	
	物理ディスク2	/dev/sdh	
	ボリューム		
	vol3 *1	ボリュームデバイス名	/dev/vx/dsk/dg2/vol3
		ボリュームRAWデバイス名	/dev/vx/rdsk/dg2/vol3
		ファイルシステム	vxfs
	vol4 *1	ボリュームデバイス名	/dev/vx/dsk/dg2/vol4
		ボリュームRAWデバイス名	/dev/vx/rdsk/dg2/vol4
		ファイルシステム	vxfs
rootdg用ディスク			
server1側	/dev/sdbのパーティション		
server2側	/dev/sdbのパーティション		
Diskハートビートリソース用LUN			
共有ディスク1	/dev/sdcのパーティション		
共有ディスク2	/dev/sdfのパーティション		

*1 動作確認した環境では、ディスクグループに物理ディスクを複数登録し、ボリュームを共有ディスクの筐体間でミラーリングしました。

1.7.7.2 CLUSTERPRO環境のサンプル

リソースの各設定パラメータの詳細については「トレッキングツール編」を参照してください。ここで設定するVxVMのパラメータは「1.7.7.1 VERITAS Volume Manager の構成」のVxVMの設定例をもとに設定します。

	設定パラメータ	設定値
クラスタ構成	クラスタ名	cluster
	サーバ数	2
	フェイルオーバグループ数	3
	モニタリソース数	8
ハートビートリソース	LANハートビート数	2
	COMハートビート数	1
	DISKハートビート数	2
1台目のサーバの情報（マスタサーバ）	サーバ名	server1
	インタコネクトのIPアドレス(専用)	192.168.0.1
	インタコネクトのIPアドレス（バックアップ）	10.0.0.1
	パブリックのIPアドレス	10.0.0.1
	COMハートビートデバイス	/dev/ttyS0
	DISKハートビートデバイス	/dev/sdc1 /dev/raw/raw10
		/dev/sdf1 /dev/raw/raw11
2台目のサーバの情報	サーバ名	server2
	インタコネクトのIPアドレス(専用)	192.168.0.2
	インタコネクトのIPアドレス（バックアップ）	10.0.0.2
	パブリックのIPアドレス	10.0.0.2
	COMハートビートデバイス	/dev/ttyS0
	DISKハートビートデバイス	/dev/sdc1 /dev/raw/raw10
		/dev/sdf1 /dev/raw/raw11
1つ目のグループ(Webマネージャ用)	タイプ	フェイルオーバ
	グループ名	WebManager
	起動サーバ	server1→server2
	グループリソース数	1
1つ目のグループリソース*1	タイプ	floating ip resource
	グループリソース名	WebManagerFIP1
	IPアドレス	10.0.0.11
2つ目のグループ(業務用)	タイプ	フェイルオーバ
	グループ名	failover1
	起動サーバ	server1→server2
	グループリソース数	4
1つ目のグループリソース	タイプ	floating ip resource
	グループリソース名	fip1
	IPアドレス	10.0.0.12
2つ目のグループリソース	タイプ	VxVM disk group resource

		設定パラメータ	設定値
		グループリソース名	vxdg1
		ディスクグループ名	dg1
		ホストIDクリア	ON
		強制インポート	OFF
	3つ目のグループリソース	タイプ	VxVM volume resource
		グループリソース名	vxvol1
		ボリュームデバイス名	/dev/vx/dsk/dg1/vol1
		ボリュームRAWデバイス名	/dev/vx/rdsk/dg1/vol1
		マウントポイント	/mnt/vol1
		ファイルシステム	vxfs
	4つ目のグループリソース	タイプ	VxVM volume resource
		グループリソース名	vxvol2
		ボリュームデバイス名	/dev/vx/dsk/dg1/vol2
		ボリュームRAWデバイス名	/dev/vx/rdsk/dg1/vol2
		マウントポイント	/mnt/vol2
		ファイルシステム	vxfs
3つ目のグループ (業務用)		タイプ	フェイルオーバ
		グループ名	failover2
		起動サーバ	server2→server1
		グループリソース数	4
	1つ目のグループリソース	タイプ	floating ip resource
		グループリソース名	fip2
		IPアドレス	10.0.0.13
	2つ目のグループリソース	タイプ	VxVM disk group resource
		グループリソース名	vxdg2
		ディスクグループ名	dg2
		ホストIDクリア	ON
		強制インポート	OFF
	3つ目のグループリソース	タイプ	VxVM volume resource
		グループリソース名	vxvol3
		ボリュームデバイス名	/dev/vx/dsk/dg2/vol3
		ボリュームRAWデバイス名	/dev/vx/rdsk/dg2/vol3
		マウントポイント	/mnt/vol3
		ファイルシステム	vxfs
	4つ目のグループリソース	タイプ	VxVM volume resource
		グループリソース名	vxvol4
		ボリュームデバイス名	/dev/vx/dsk/dg2/vol4
		ボリュームRAWデバイス名	/dev/vx/rdsk/dg2/vol4
		マウントポイント	/mnt/vol4
		ファイルシステム	vxfs
1つ目のモニタリソース （デフォルト作成）		タイプ	user mode monitor
		モニタリソース名	userw
2つ目のモニタリソース （VxVMディスクグループリソース追加時自動作成）		タイプ	VxVM daemon monitor
		モニタリソース名	vxdw
3つ目のモニタリソース （vxvol1の監視）		タイプ	VxVM volume monitor
		モニタリソース名	vxvolw1
		監視デバイス	/dev/vx/rdsk/dg1/vol1
		VxVMボリュームリソース	vxvol1
		異常検出時	クラスタデーモン停止とOS

	設定パラメータ	設定値
		シャットダウン
4つ目のモニタリソース（vxvol2の監視）	タイプ	VxVM volume monitor
	モニタリソース名	vxvolw2
	監視デバイス	/dev/vx/rdsk/dg1/vol2
	VxVMボリュームリソース	vxvol2
	異常検出時	クラスタデーモン停止とOSシャットダウン
5つ目のモニタリソース（vxvol3の監視）	タイプ	VxVM volume monitor
	モニタリソース名	vxvolw3
	監視デバイス	/dev/vx/rdsk/dg2/vol3
	VxVMボリュームリソース	vxvol3
	異常検出時	クラスタデーモン停止とOSシャットダウン
6つ目のモニタリソース（vxvol4の監視）	タイプ	VxVM volume monitor
	モニタリソース名	vxvolw4
	監視デバイス	/dev/vx/rdsk/dg2/vol4
	VxVMボリュームリソース	vxvol4
	異常検出時	クラスタデーモン停止とOSシャットダウン
7つ目のモニタリソース（rootdgの監視）	タイプ	raw monitor
	モニタリソース名	raww1
	監視対象RAWデバイス名	/dev/raw/raw20
	デバイス名	/dev/sdb
	異常検出時	クラスタデーモン停止とOSシャットダウン
8つ目のモニタリソース	タイプ	ip monitor
	モニタリソース名	ipw1
	監視IPアドレス	10.0.0.254（ゲートウェイ）
	異常検出時	“WebManager” グループのフェイルオーバ

= *1: Webマネージャを接続するフローティングIPを用意して専用のグループに入れます。Webマネージャ専用のグループが停止しない限り、Webブラウザからはサーバの実IPを意識することなくアクセスできます。

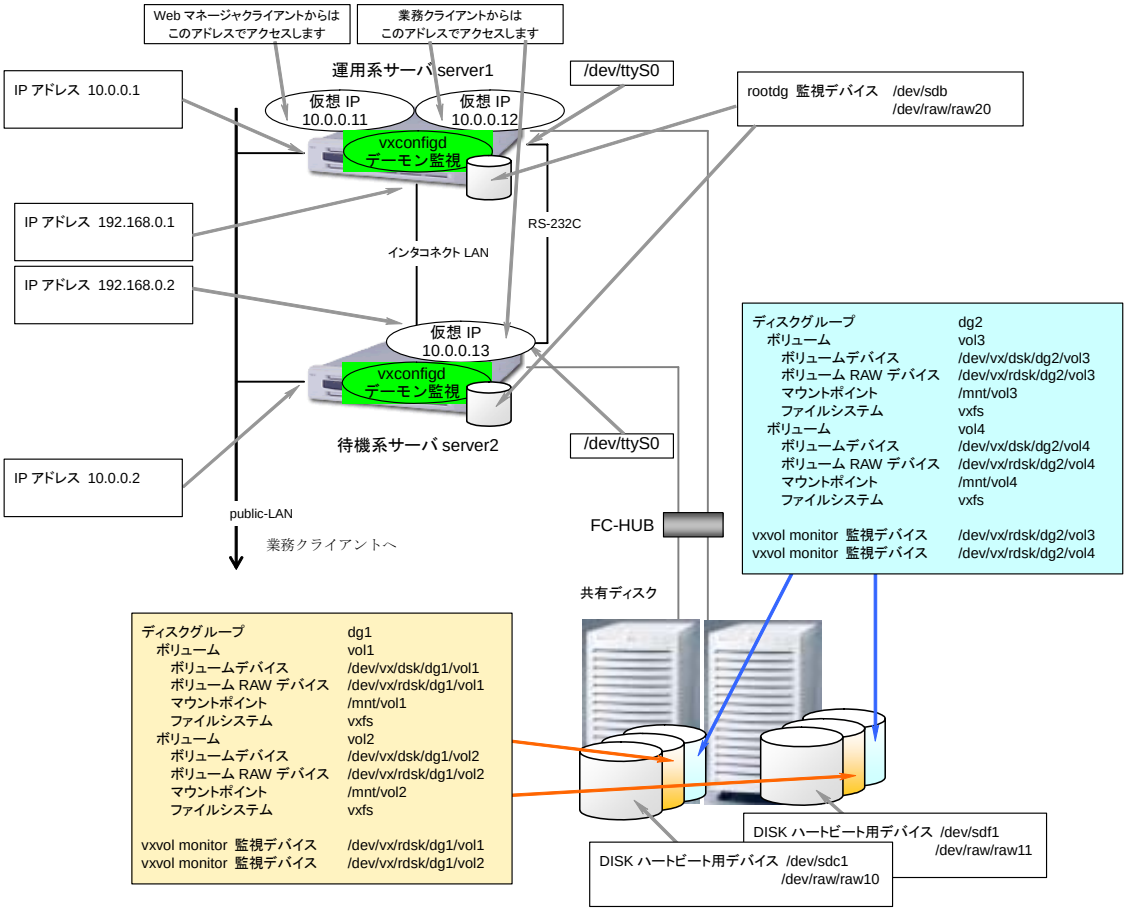
* VxVMボリュームモニタリソースは、監視したいVxVMボリュームリソースとそのボリュームRAWデバイスを正しく設定してください。

* rootdgの監視はRAWモニタリソースで監視してください。

* VxVMデーモンリソースはVxVMのvxconfigdデーモンを監視します。1つ目のVxVMディスクグループリソース設定時に自動的に追加されます。

* 以下のリソースで設定するRAWデバイスは絶対に重複しないようにしてください。
  + ディスクハートビートリソースのRAWデバイス
  + VxVMボリュームリソースのボリュームRAWデバイスの実RAWデバイス
  + RAWリソースのRAWデバイス
  + RAWモニタリソースの監視対象RAWデバイス

このクラスタの構成イメージを下図に示します。



1.7.7.3 クラスタ生成手順概要

以下の手順でクラスタを生成します。

(1) VERITAS Volume Manager のセットアップ
VERITAS Volume Manager をサーバにセットアップします。

(2) ボリュームRAWデバイスの確認
ボリュームRAWデバイスの実RAWデバイスを確認してください。「1.7.6 CLUSTERPROで制御する際の注意」を参照してください。

(3) トレッキングツールのセットアップ
トレッキングツールをセットアップします。

(4) CLUSTERPROサーバのセットアップ
クラスタを構成する全サーバでCLUSTERPROサーバをセットアップします。

(5) クラスタ構成情報の生成
トレッキングツールを使用してクラスタ構成情報を作成してFDに保存します。
「1.7.7.4 クラスタ構成情報の作成手順」を参照してください。

(6) FDのハンドキャリー
トレッキングツールで作成したFDをマスタサーバに挿入します。

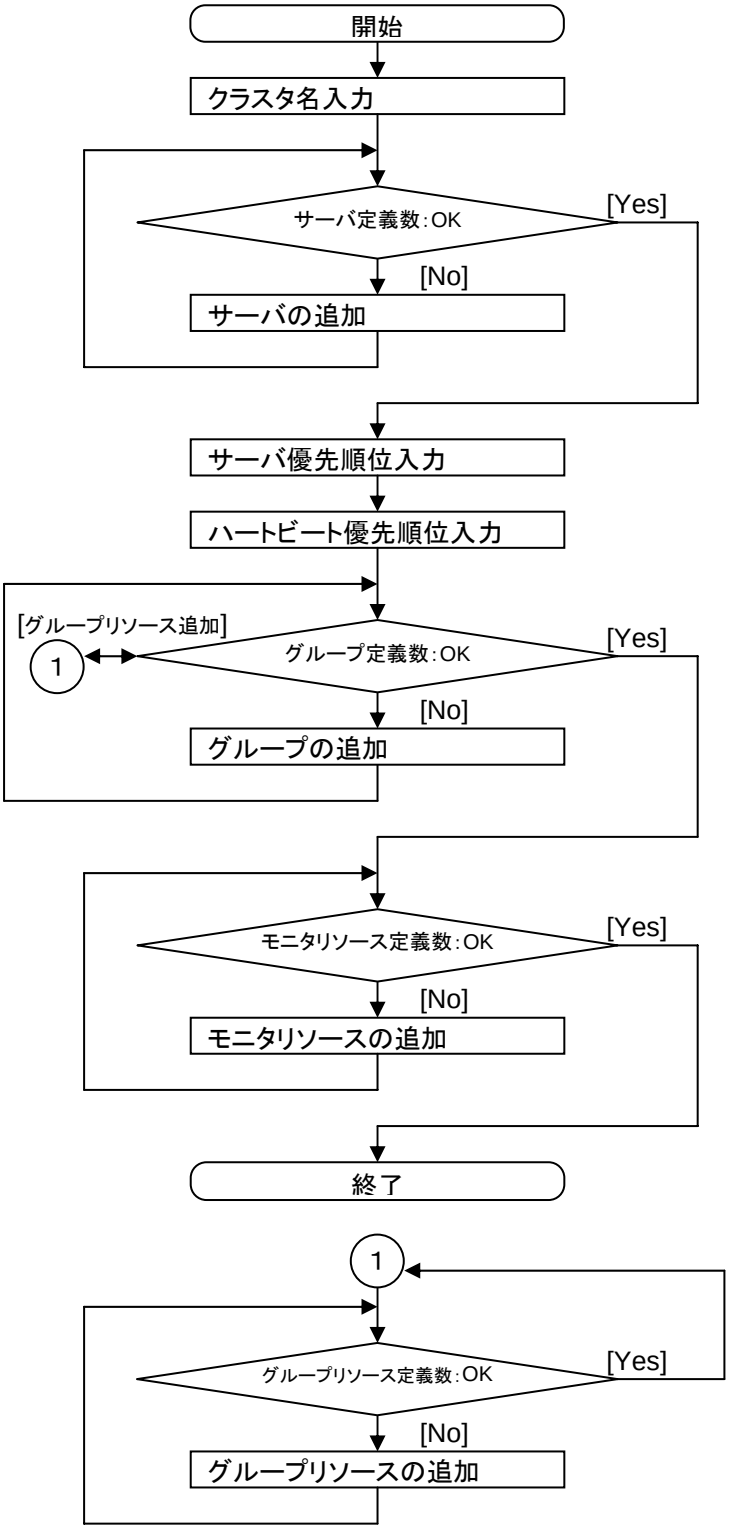
(7) クラスタ生成コマンドの実行
FDを挿入したサーバでクラスタ生成コマンドを実行します。

(8) サーバの再起動
クラスタを構成するサーバを再起動します。

(9) CLUSTERPRO Webマネージャの接続
ブラウザを使用してCLUSTERPROサーバに接続します。

1.7.7.4 クラスタ構成情報の作成手順

クラスタ構成情報の作成手順を以下の流れで説明します。

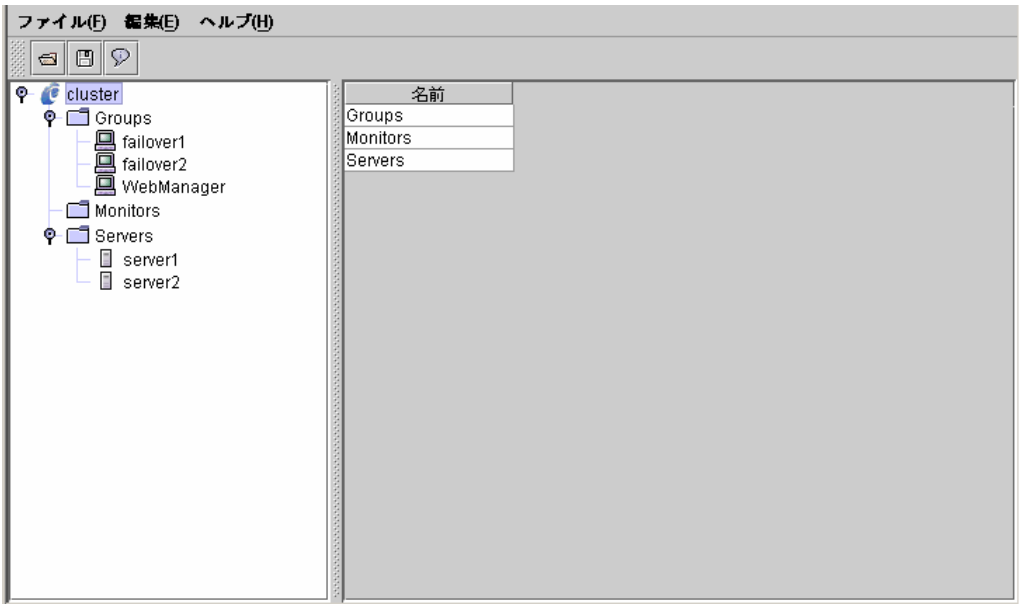


(7)-(11) 参照

(2)-(5) 参照

(1) トレッキングツールを起動します。
サーバ、ハートビートリソース、グループを追加します。追加の手順は「クラスタ生成編（共有ディスク）」を参照してください。

ツリービューは以下のようになります。



(2) 1つ目のグループリソース情報を入力します。

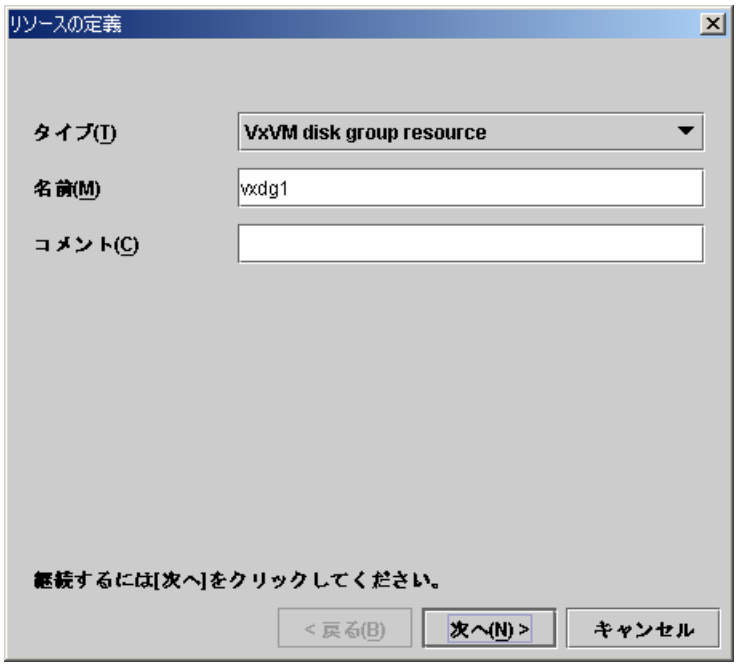
タイプ	floating ip resource
グループリソース名	fip1
IPアドレス	10.0.0.12

「クラスタ生成編（共有ディスク）」を参照してください。

(3)　ツリービューのfailover1にフォーカスを合わせて、メニューバー[編集]→[追加]を選択します。
2つ目のグループリソース情報を入力します。

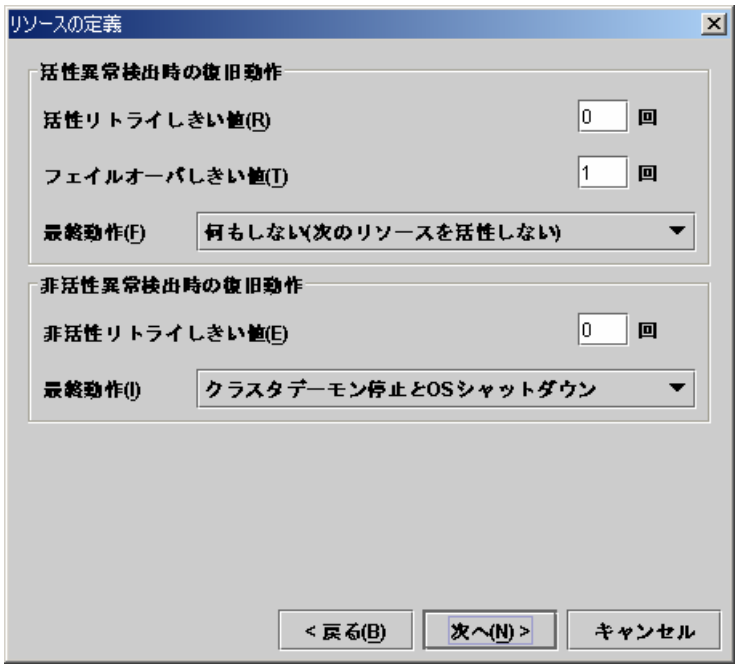
タイプ	VxVM disk group resource
グループリソース名	vxdg1
ディスクグループ名	dg1

A.　以下の画面でタイプ及びグループリソース名を入力して[次へ]ボタンを選択します。

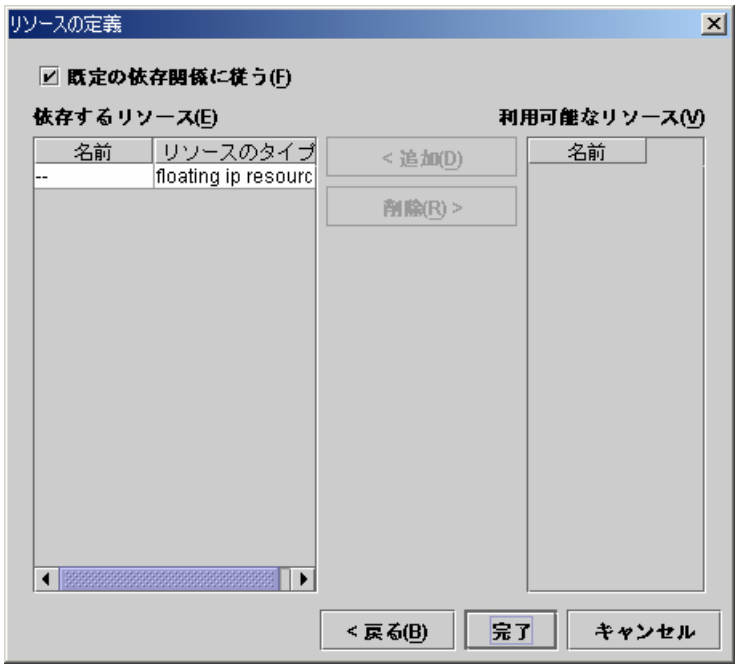


B.　以下の画面でディスクグループ名を入力して[次へ]ボタンを選択します。

C.　以下の画面で[次へ]ボタンを選択します。



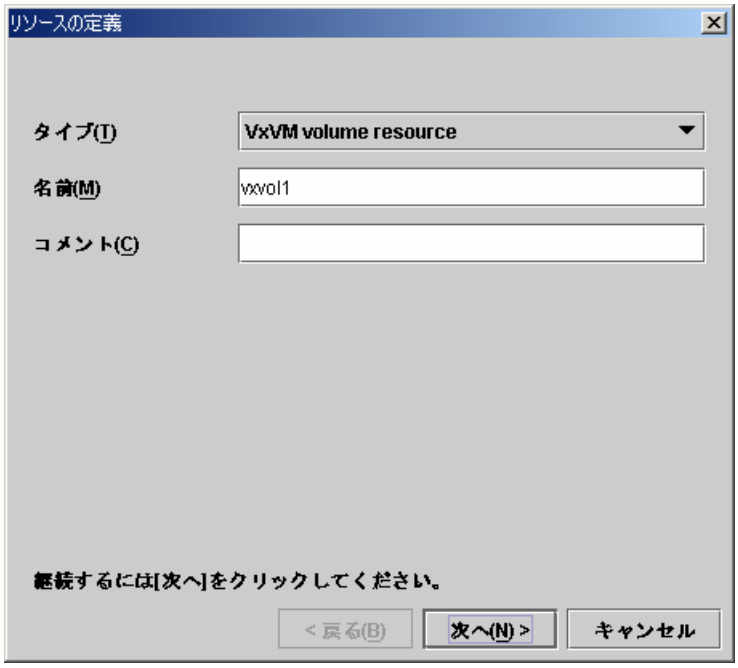
D.　以下の画面で[完了]ボタンを選択します。



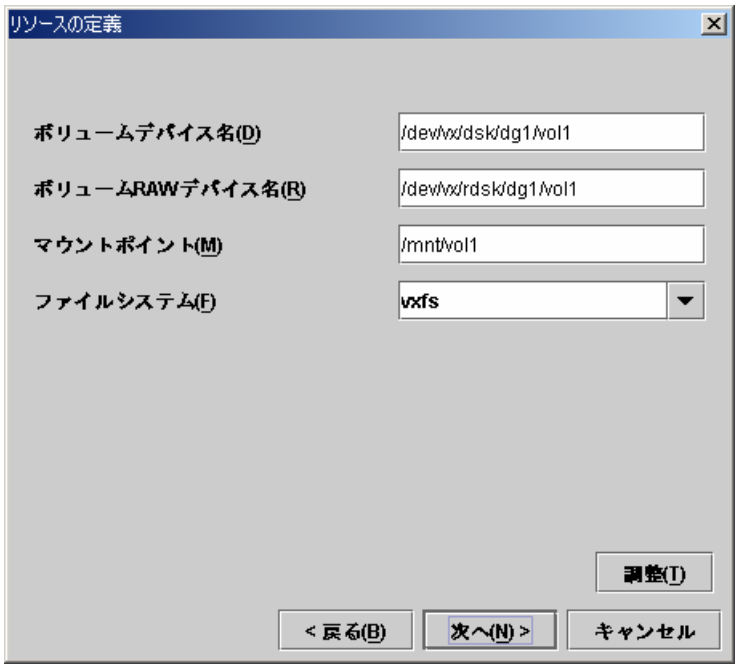
(4) ツリービューのfailover1にフォーカスを合わせて、メニューバー[編集]→[追加]を選択します。
3つ目のグループリソース情報を入力します。

タイプ	VxVM volume resource
グループリソース名	vxvol1
ボリュームデバイス名	/dev/vx/dsk/dg1/vol1
ボリュームRAWデバイス名	/dev/vx/rdsk/dg1/vol1
マウントポイント	/mnt/vol1
ファイルシステム	vxfs

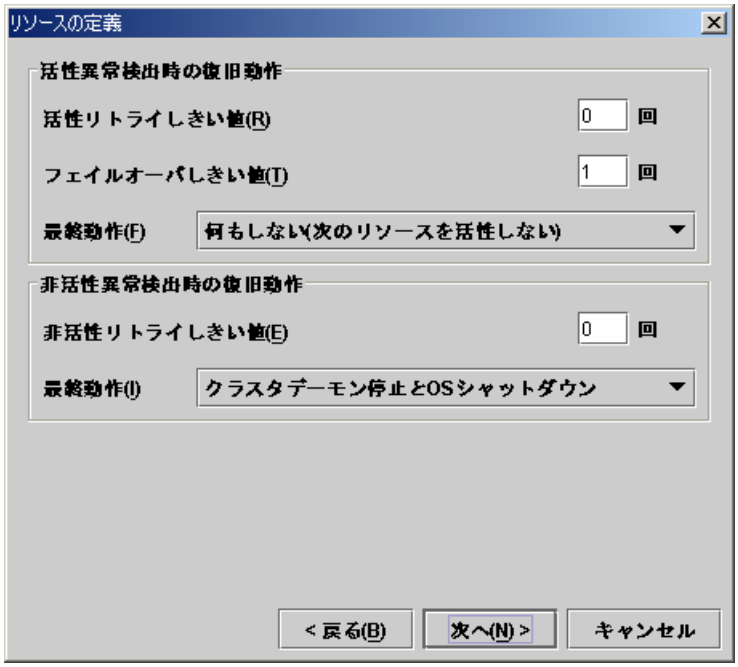
A. 以下の画面でタイプ及びグループリソース名を入力して[次へ]ボタンを選択します。



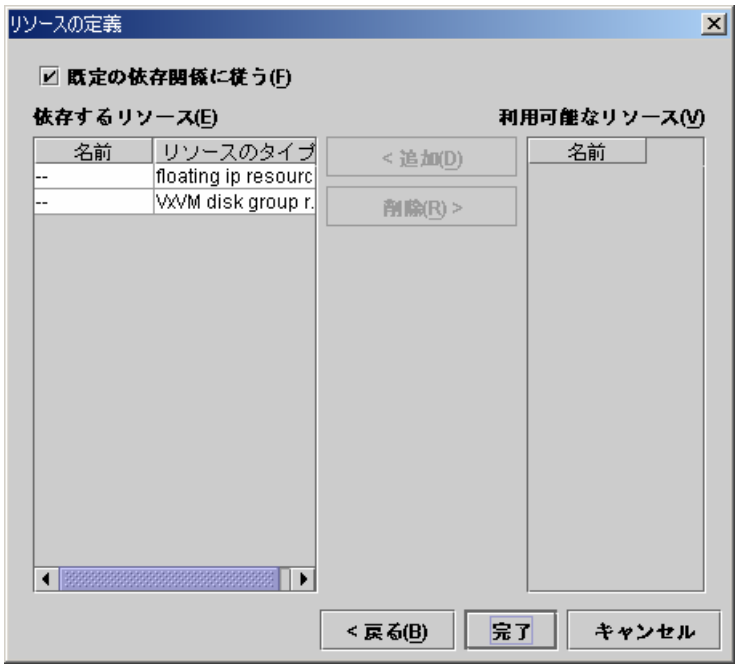
B. 以下の画面でボリュームデバイス名、ボリュームRAWデバイス名、マウントポイント及びファイルシステムを入力して[次へ]ボタンを選択します。



C. 以下の画面で[次へ]ボタンを選択します。



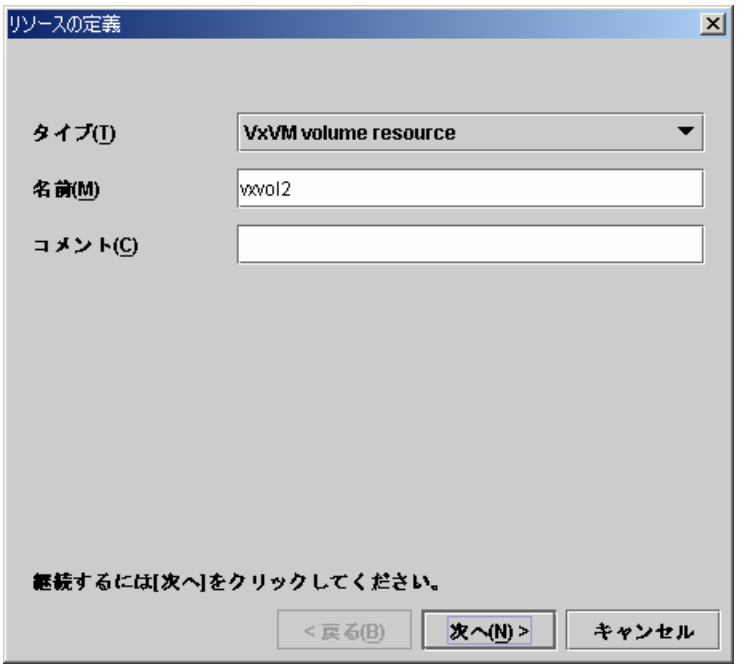
D.　以下の画面で[完了]ボタンを選択します。



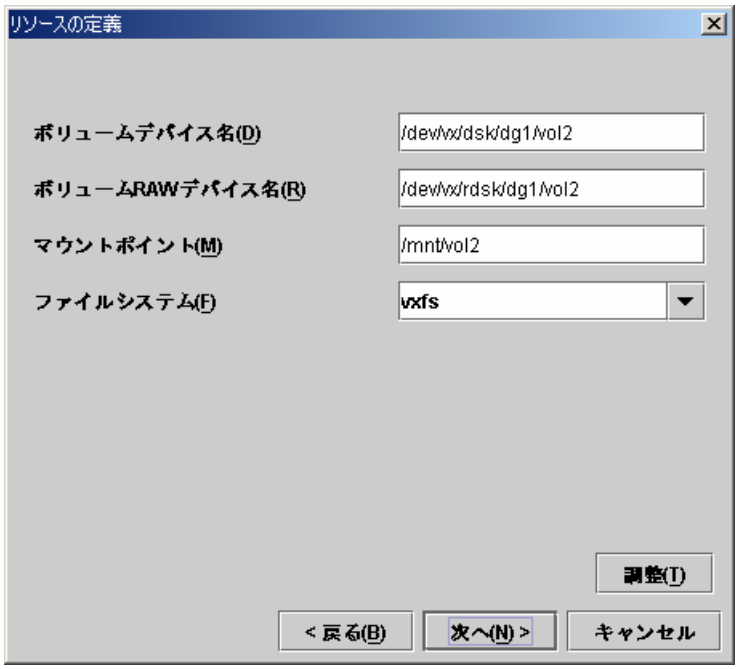
(5) ツリービューのfailover1にフォーカスを合わせて、メニューバー[編集]→[追加]を選択します。
4つ目のグループリソース情報を入力します。

タイプ	VxVM volume resource
グループリソース名	vxvol2
ボリュームデバイス名	/dev/vx/dsk/dg1/vol2
ボリュームRAWデバイス名	/dev/vx/rdsk/dg1/vol2
マウントポイント	/mnt/vol2
ファイルシステム	vxfs

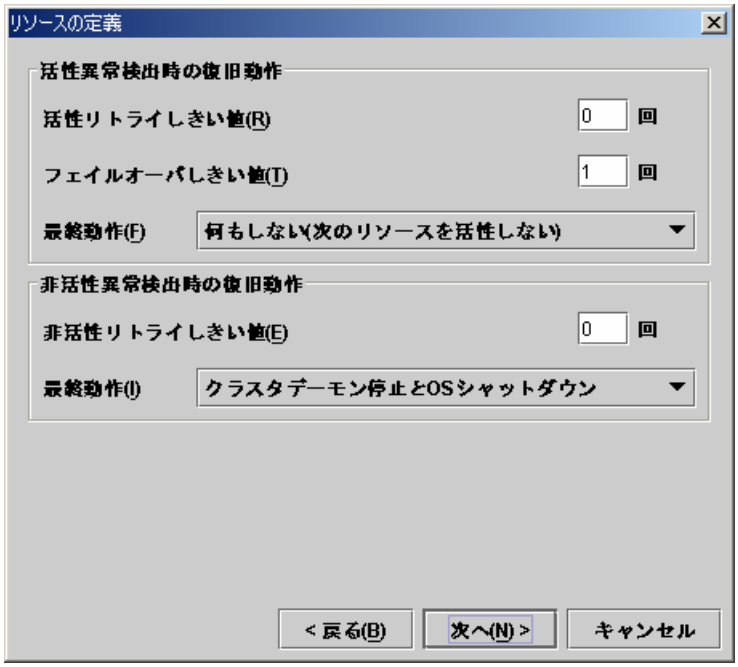
A. 以下の画面でタイプ及びグループリソース名を入力して[次へ]ボタンを選択します。



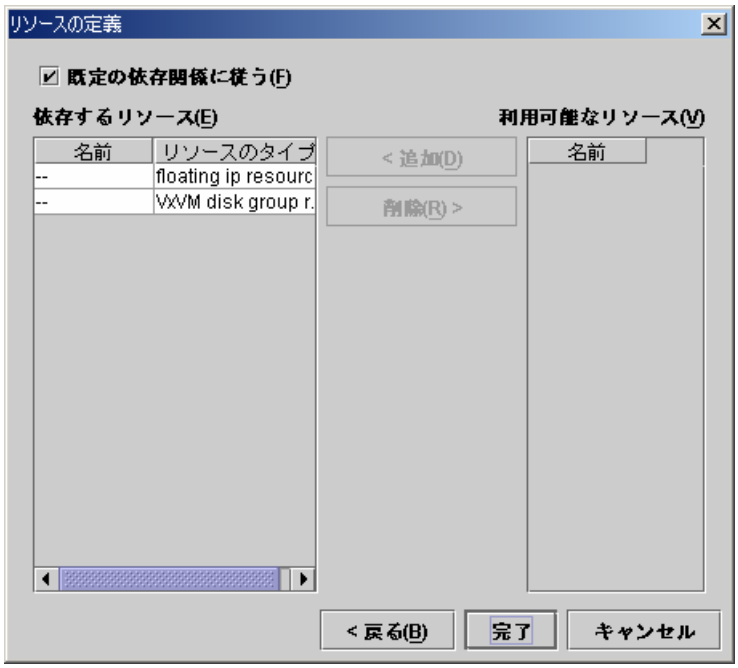
B. 以下の画面でボリュームデバイス名、ボリュームRAWデバイス名、マウントポイント及びファイルシステムを入力して[次へ]ボタンを選択します。



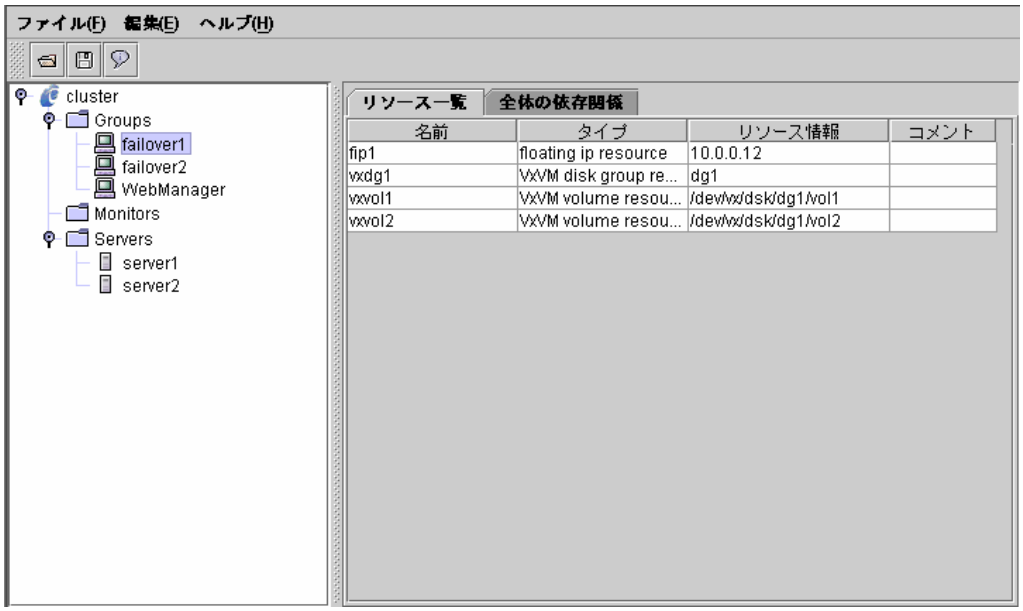
C. 以下の画面で[次へ]ボタンを選択します。



D.　以下の画面で[完了]ボタンを選択します。



failover1のテーブルビューは以下のようになります。

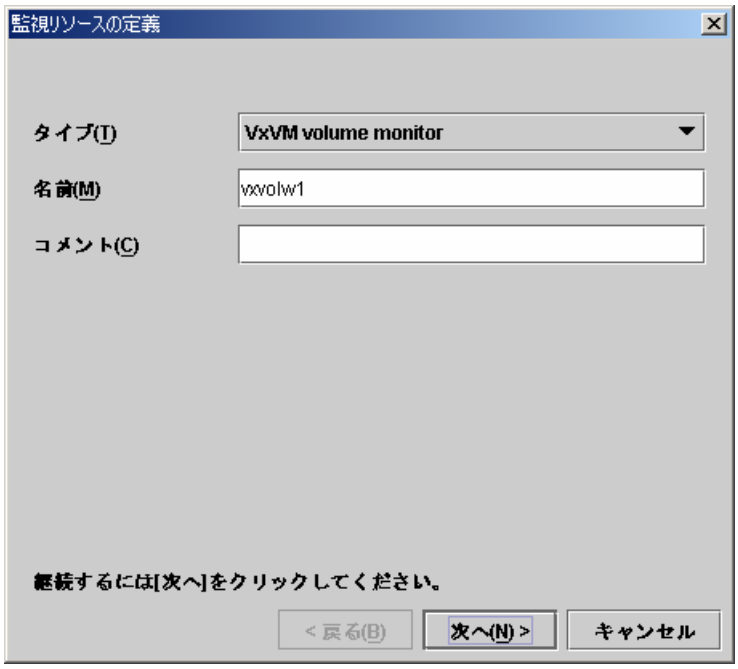


(6)　2つ目のグループについても1つ目のグループと同様にリソースを追加します。
failover2のテーブルビューは以下のようになります。

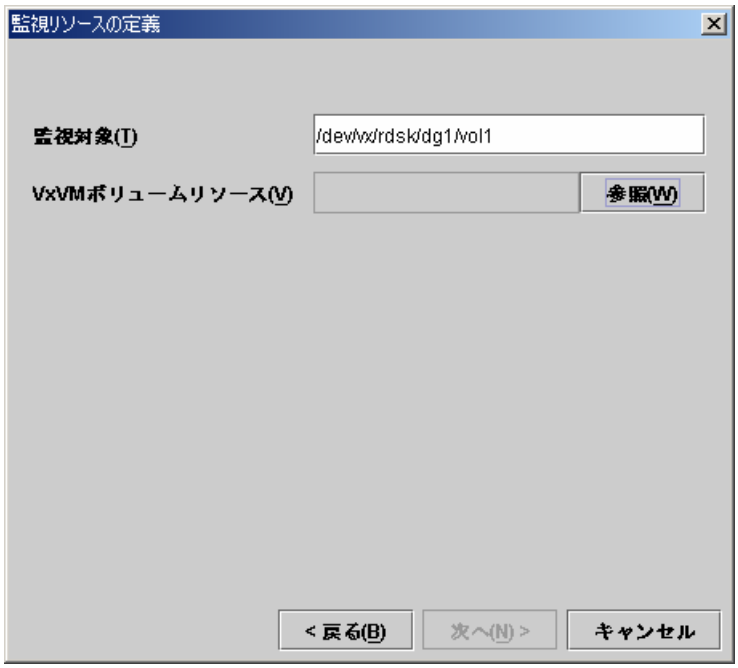
(7) ツリービューのMonitorsにフォーカスを合わせて、メニューバー[編集]→[追加]を選択します。
3つ目のモニタリソース情報を入力します。1つ目のモニタリソース（ユーザ空間モニタ）はクラスタ名を定義したときにデフォルトで作成されています。2つ目のモニタリソース（VxVMデーモンモニタ）はVxVMディスクグループリソースを追加したときに自動的に作成されています。

タイプ	VxVM volume monitor
モニタリソース名	vxvolw1
監視デバイス	/dev/vx/rdsk/dg1/vol1
VxVMボリュームリソース	vxvol1
異常検出時	クラスタデーモン停止とOSシャットダウン

A. 以下の画面でタイプ及びモニタリソース名を入力して[次へ]ボタンを選択します。



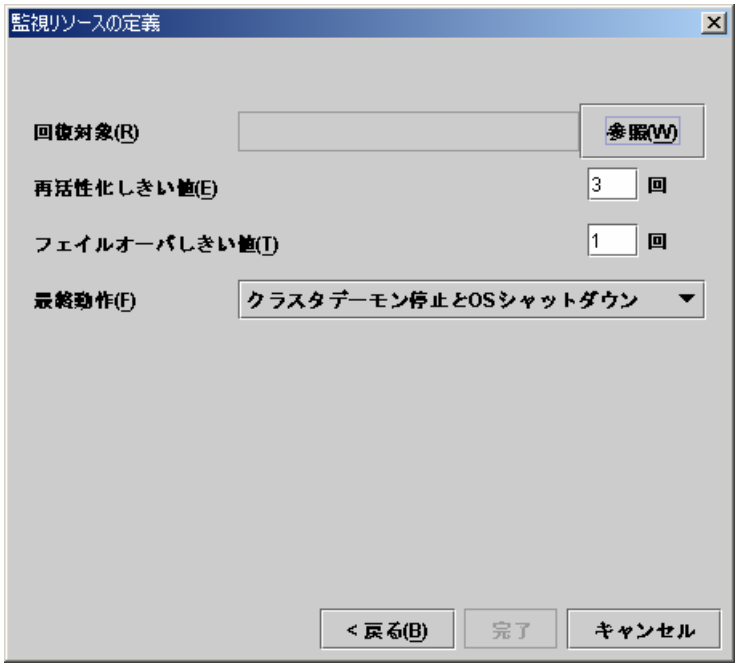
B.　以下の画面で監視デバイスを入力して[参照]ボタンを選択します。



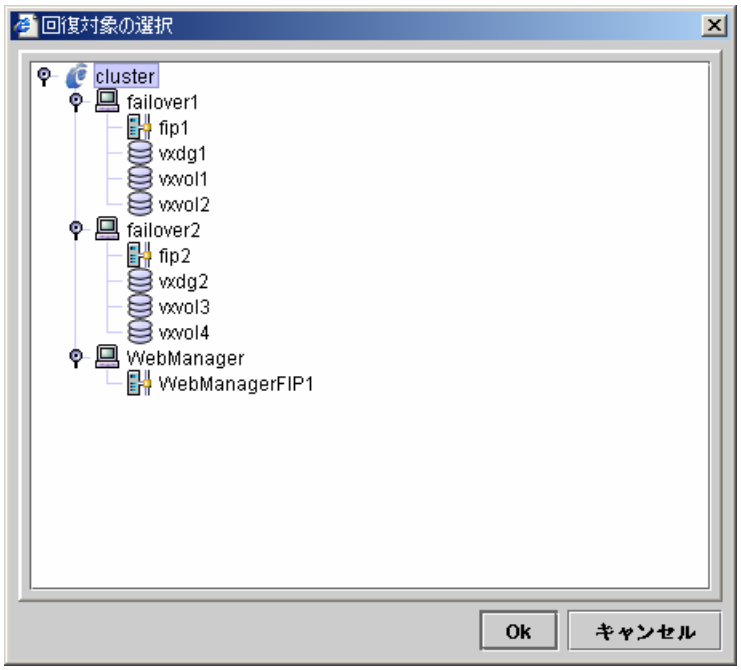
以下のダイアログでvxvol1を選択して、[OK]ボタンを選択します。

C. VxVMボリュームリソースにvxvol1が設定されたのを確認して、 [次へ]ボタンを選択します。

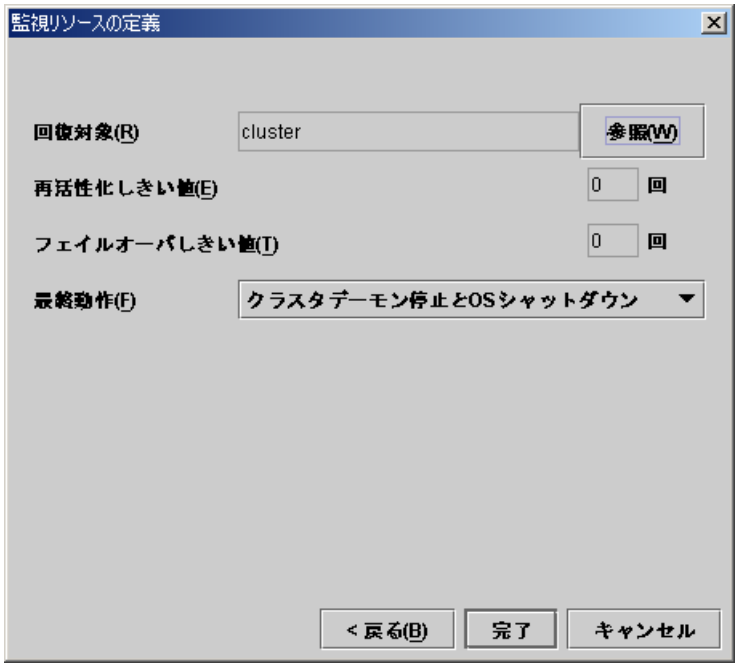
D. 以下の画面で異常検出時の動作を入力します。[参照]ボタンを選択します。



以下のダイアログでclusterを選択して、[Ok]ボタンを選択します。



E. 回復対象にclusterが設定されたのを確認して、最終動作に「クラスタデーモン停止とOSシャットダウン」を設定します。[完了]ボタンを選択します。



(8) 以下のモニタリソースは (7) と同様の手順で設定してください。

4つ目のモニタリソース

タイプ	VxVM volume monitor
モニタリソース名	vxvolw2
監視デバイス	/dev/vx/rdsk/dg1/vol2
VxVMボリュームリソース	vxvol2
異常検出時	クラスタデーモン停止とOSシャットダウン

5つ目のモニタリソース

タイプ	VxVM volume monitor
モニタリソース名	vxvolw3
監視デバイス	/dev/vx/rdsk/dg2/vol3
VxVMボリュームリソース	vxvol3
異常検出時	クラスタデーモン停止とOSシャットダウン

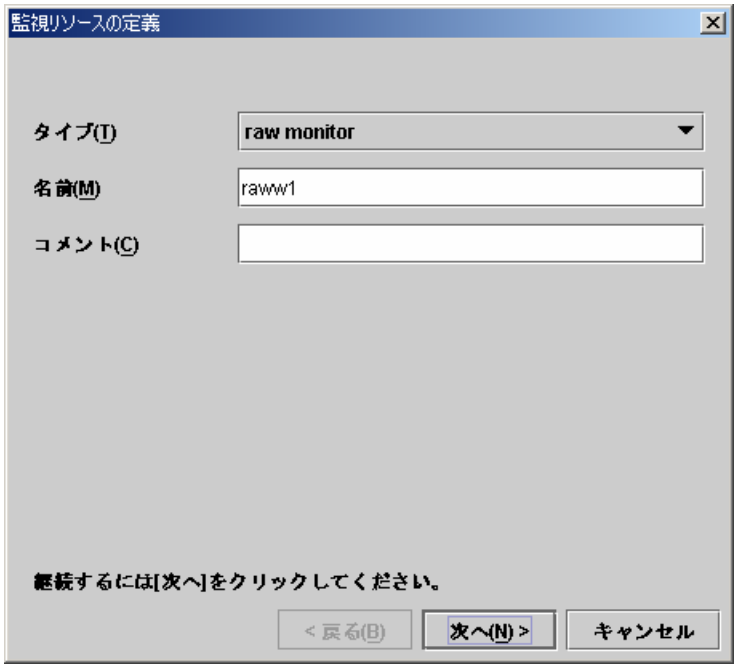
6つ目のモニタリソース

タイプ	VxVM volume monitor
モニタリソース名	vxvolw4
監視デバイス	/dev/vx/rdsk/dg2/vol4
VxVMボリュームリソース	vxvol4
異常検出時	クラスタデーモン停止とOSシャットダウン

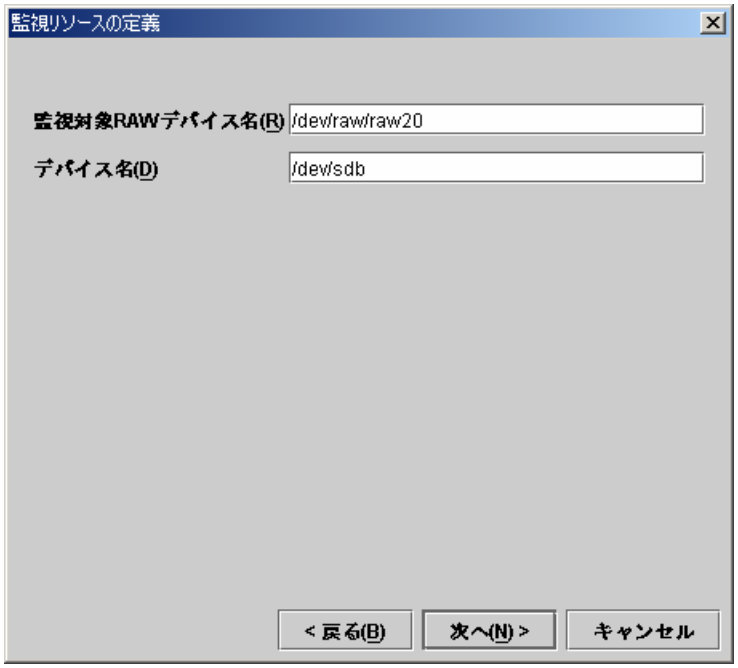
(9) ツリービューのMonitorsにフォーカスを合わせて、メニューバー[編集]→[追加]を選択します。
7つ目のモニタリソース情報を入力します。

タイプ	raw monitor
モニタリソース名	raww1
監視対象RAWデバイス名	/dev/raw/raw20
デバイス名	/dev/sdb
異常検出時	クラスタデーモン停止とOSシャットダウン

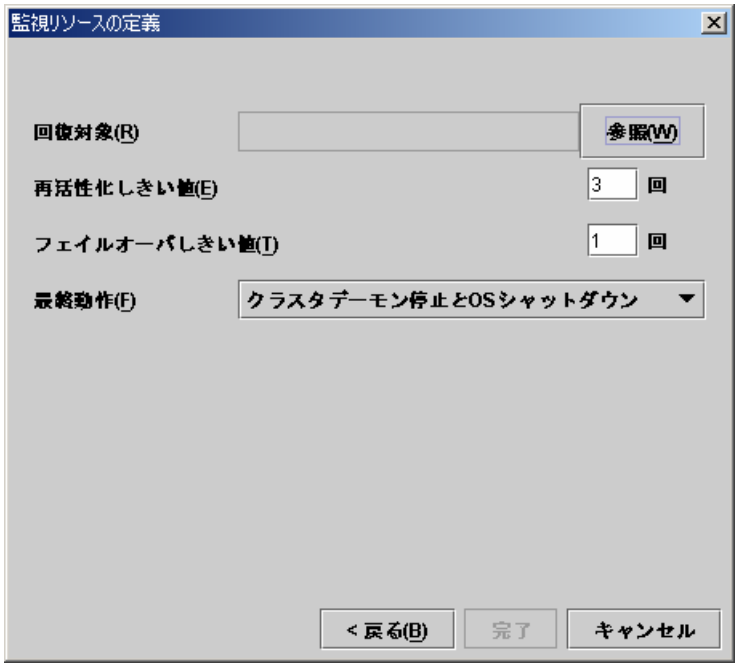
A. 以下の画面でタイプ及びモニタリソース名を入力して[次へ]ボタンを選択します。



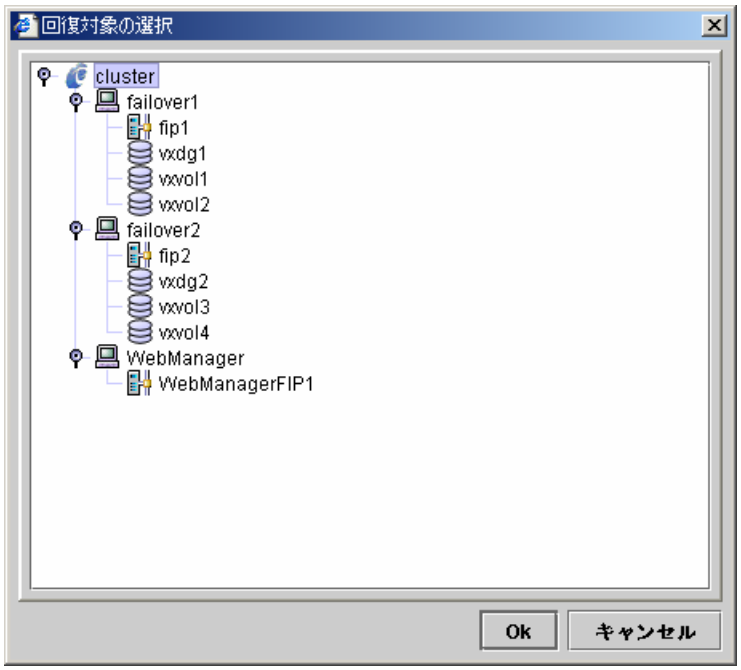
B. 以下の画面で監視対象RAWデバイス名及びデバイス名を入力して[次へ]ボタンを選択します。



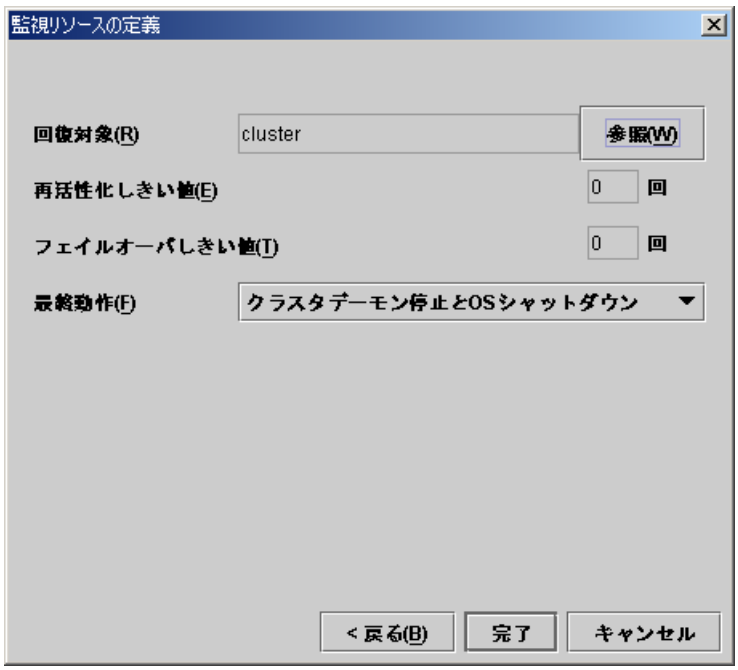
C.　以下の画面で異常検出時の動作を入力します。[参照]ボタンを選択します。



以下のダイアログでclusterを選択して、[Ok]ボタンを選択します。



D. 回復対象にclusterが設定されたのを確認して、最終動作に「クラスタデーモン停止とOSシャットダウン」を設定します。[完了]ボタンを選択します。

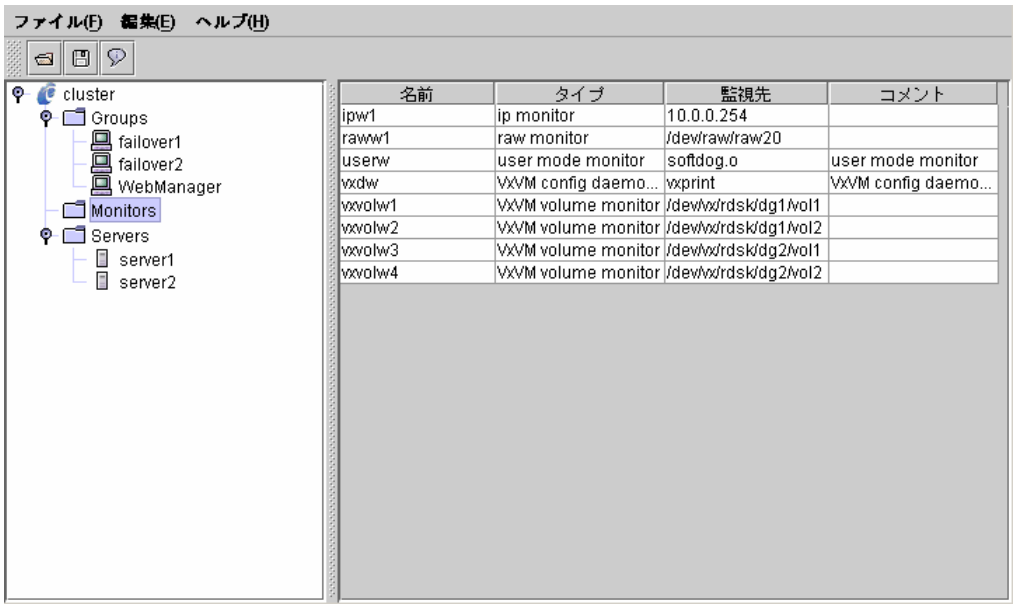


(10) 8つ目のモニタリソース情報を入力します。

タイプ	ip monitor
モニタリソース名	ipw1
監視IPアドレス	10.0.0.254 （ゲートウェイ）
異常検出時	“WebManager” グループのフェイルオーバ

「クラスタ生成編（共有ディスク）」を参照してください。

(11) Monitorsのテーブルビューは以下のようになります。



以上でクラスタ構成情報の生成は終了です。以降の手順は「クラスタ生成編（共有ディスク）」を参照してください。

1.8 NASリソース

1.8.1 CLUSTERPROのバージョン

以下のCLUSTERPROのバージョンでサポートします。

CLUSTERPRO	Version
サーバ	SE3.1-1 以降、LE3.1-1 以降、XE3.1-4 以降、SX3.1-2 以降
トレッキングツール	対応するバージョンについては動作環境編 トレッキングツールの動作環境を参照してください。

1.8.2 依存関係

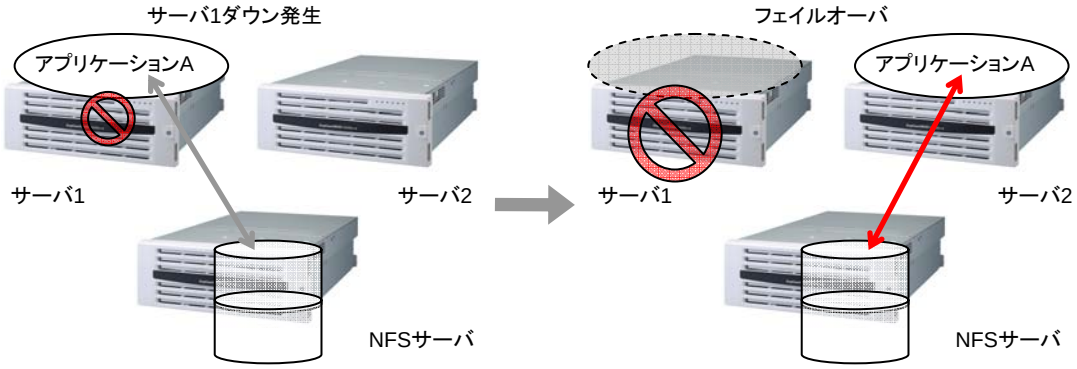
規定値では、以下のグループリソースタイプに依存します。

グループリソースタイプ	Edition
フローティングIPリソース	SE、LE、XE、SX

1.8.3 NAS リソース

* NASリソースは、NFSサーバ上の資源を制御します。

* 業務に必要なデータは、NFSサーバ上に格納しておくことで、フェイルオーバ時、フェイルオーバグループの移動時等に、自動的に引き継がれます。



1.8.4 NAS リソースに関する注意事項

* ファイルシステムのアクセス制御（mount/umount）は、CLUSTERPROが行いますので、OS側でmount/umountする設定を行わないでください。

* NFSサーバ上で、クラスタを構成しているサーバへNFS資源のアクセス許可を設定する必要があります。

* CLUSTERPROサーバ側でportmapサービスを起動する設定を行ってください。

* NASサーバ名にホスト名を指定する場合は名前解決できるように設定を行ってください。

* XE版でGFSサーバ機能を二重化する場合には、CLUSTERPROのホームページからGFS編をダウンロードして参照してください。

* CLUSTERPROのバージョンが3.0-1～3.1-7の場合、NASリソースのマウントポイントにシンボリックリンクを指定しないでください。パスの一部に含まれる場合も同様です。

* CLUSTERPROのバージョンが3.1-8以降の場合、ディスクリソース、VxVMボリュームリソース、NASリソースのmount/umountは同一サーバ内で排他的に動作するため、NASリソースの活性/非活性に時間がかかることがあります。

2 モニタリソース

現在サポートされているモニタリソースは以下です。

モニタリソース名	略称	機能概要
ディスクモニタリソース	diskw	2.2 ディスクモニタリソース を参照
RAWモニタリソース	raww	2.3 RAWモニタリソース を参照
IPモニタリソース	ipw	2.4 IPモニタリソース を参照
NIC Link Up/Downモニタリソース	miiw	2.5 NIC Link Up/Downモニタリソース を参照
ミラーディスクコネクトモニタリソース	mdnw	2.6 ミラーディスクコネクトモニタリソース を参照
ミラーディスクモニタリソース	mdw	2.7 ミラーディスクモニタリソース を参照
PIDモニタリソース	pidw	2.8 PIDモニタリソース を参照
ユーザ空間モニタリソース	userw	2.9 ユーザ空間モニタリソース を参照
VxVMデーモンモニタリソース	vxdg	2.10 VxVMデーモンモニタリソース を参照
VxVMボリュームモニタリソース	vxvol	2.11 VxVMボリュームモニタリソース を参照
マルチターゲットモニタリソース	mtw	2.12 マルチターゲットモニタリソース を参照

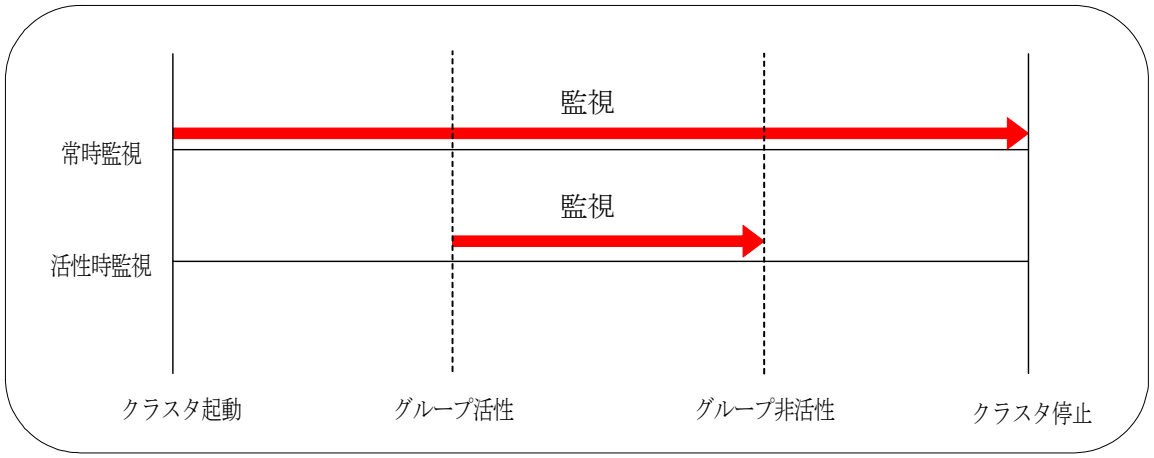
2.1 モニタリソース

モニタリソースは、指定された監視対象を監視します。監視対象の異常を検出した場合には、グループリソースの再起動やフェイルオーバなどを行います。

モニタリソースの監視するタイミングは以下があり、監視可能な状態の範囲が2つあります。
各モニタリソースの監視タイミングは、バージョンが3.0-4までは変更不可、3.1-1以降では変更可能ですが初期設定では以下の設定になります。

+ 常時監視（クラスタ起動時～クラスタ停止時）
  = ディスクモニタリソース
  = IPモニタリソース
  = ユーザ空間モニタリソース
  = ミラーディスクモニタリソース
  = ミラーディスクコネクトモニタリソース
  = RAWモニタリソース
  = VxVMデーモンモニタリソース
  = NIC Link Up/Downモニタリソース
  = マルチターゲットモニタリソース

+ 活性時監視（グループ活性時～グループ非活性時）
  = pidモニタリソース
  = VxVMボリュームモニタリソース



2.1.1 監視タイミング

バージョンにより機能範囲が異なります。

CLUSTERPRO	Version
サーバ	SE3.0-1～3.0-4、LE3.0-1～3.0-4、XE3.0-1
トレッキングツール	対応するバージョンについては動作環境編 トレッキングツールの動作環境を参照してください。

モニタリソース	監視タイミング	対象リソース選択範囲
ディスクモニタリソース	常時固定	-
IPモニタリソース	常時固定	-
ユーザ空間モニタリソース	常時固定	-
ミラーディスクモニタリソース[5]	常時固定	-
ミラーディスクコネクトモニタリソース[5]	常時固定	-
RAWモニタリソース[6]	常時固定	-
VxVMデーモンモニタリソース[7]	常時固定	-
pidモニタリソース	活性時固定	exec
VxVMボリュームモニタリソース[7]	活性時固定	vxvol

3.1-1以降のバージョンは、一部の監視リソースを除き監視タイミングの選択が可能です。

CLUSTERPRO	Version
サーバ	SE3.1-1 以降、LE3.1-1 以降、XE3.1-4 以降、SX3.1-2 以降 (マルチターゲットモニタリソースはVersion3.1-6以降)
トレッキングツール	対応するバージョンについては動作環境編 トレッキングツールの動作環境を参照してください。

モニタリソース	監視タイミング	対象リソース選択範囲
ディスクモニタリソース	常時/活性時から選択可能	全て可能
IPモニタリソース	常時/活性時から選択可能	全て可能
ユーザ空間モニタリソース	常時固定	-
ミラーディスクモニタリソース[5]	常時固定	-
ミラーディスクコネクトモニタリソース[5]	常時固定	-
RAWモニタリソース[6]	常時/活性時から選択可能	全て可能
VxVMデーモンモニタリソース[7]	常時/活性時から選択可能	全て可能
NIC Link Up/Downモニタリソース	常時/活性時から選択可能	全て可能
pidモニタリソース	活性時固定	exec
VxVMボリュームモニタリソース[7]	活性時固定	vxvol
マルチターゲットモニタリソース	常時/活性時から選択可能	全て可能

[5] LEの場合のみです。
[6] SE,LEの場合のみです。
[7] SEの場合のみです。

2.1.2 監視インターバル

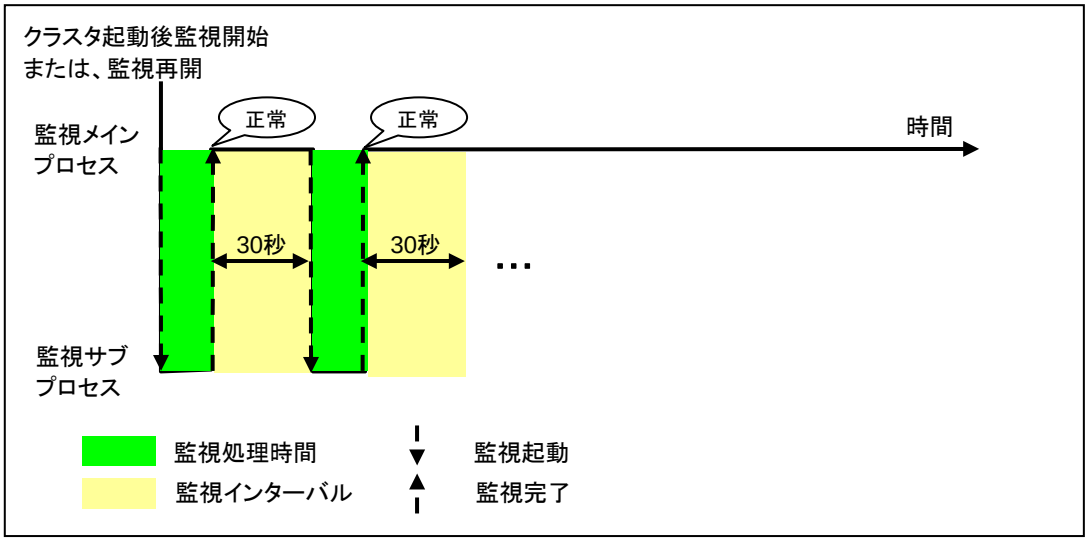
ユーザ空間監視リソースを除く全てのモニタリソースは、監視インターバル毎に監視が行われます。

以下は、この監視インターバルの設定による正常または、異常時におけるモニタリソースへの監視の流れを時系列で表した説明です。

* 監視正常検出時

[設定例]
<監視>
監視インターバル 30秒
監視タイムアウト 60秒
監視リトライ回数 0回

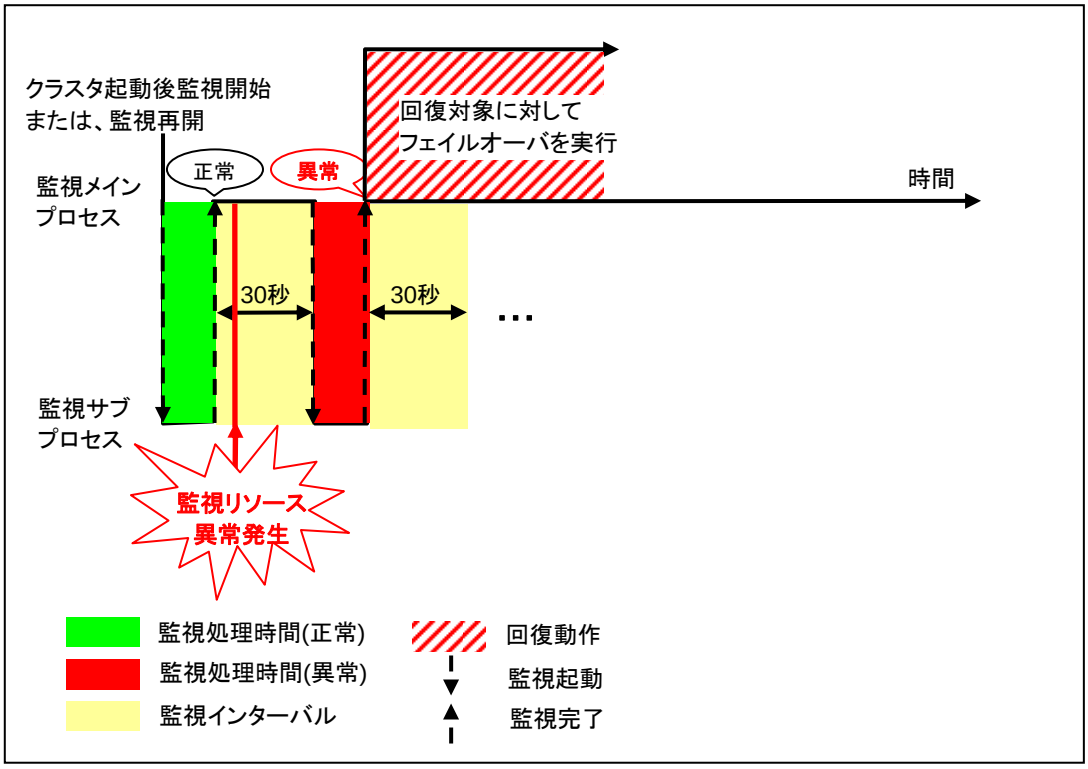
を指定している場合の挙動の例



*　監視異常検出時(監視リトライ設定なし)

[設定例]
　　<監視>
　　監視インターバル　　30秒
　　監視タイムアウト　　60秒
　　監視リトライ回数　　0回
　<異常検出>
　　回復対象　　グループ
　　再活性化しきい値　　0回
　　フェイルオーバしきい値　　1回
　　最終動作　　何もしない

を指定している場合の挙動の例

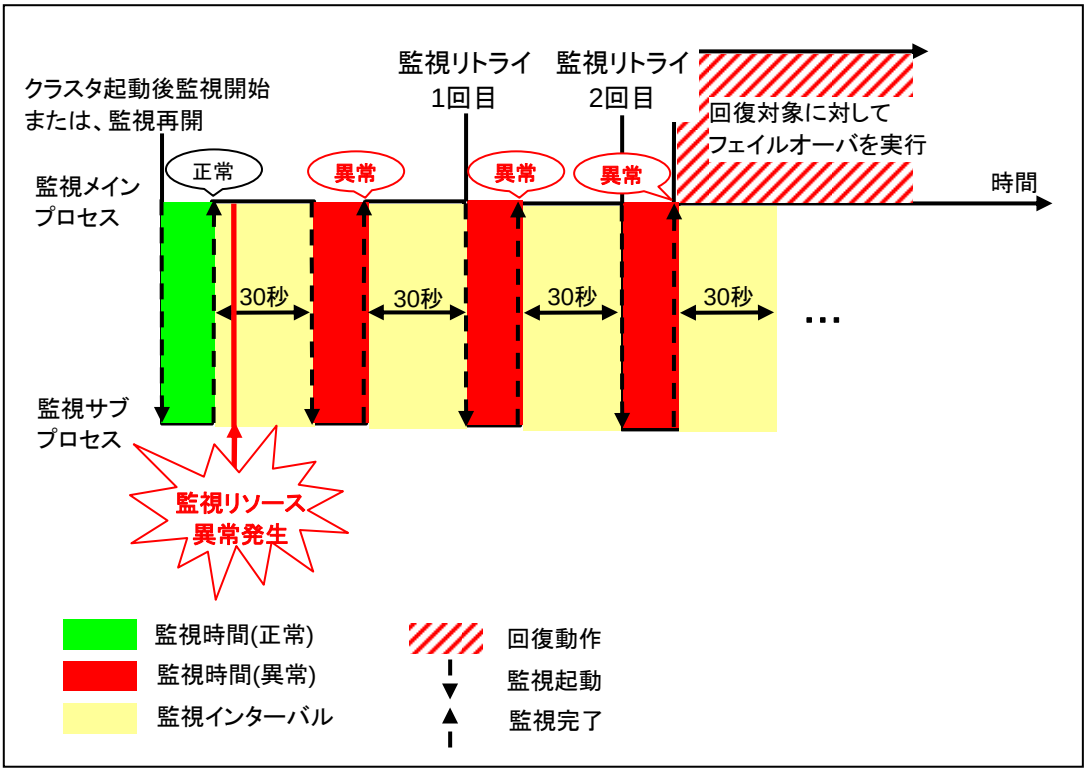


　監視異常発生後、次回監視で監視異常を検出し回復対象に対してフェイルオーバが行われます。

*　監視異常検出時(監視リトライ設定あり)

[設定例]
　　<監視>
　　監視インターバル　　30秒
　　監視タイムアウト　　60秒
　　監視リトライ回数　　2回
　<異常検出>
　　回復対象　　グループ
　　再活性化しきい値　　0回
　　フェイルオーバしきい値　　1回
　　最終動作　　何もしない

を指定している場合の挙動の例

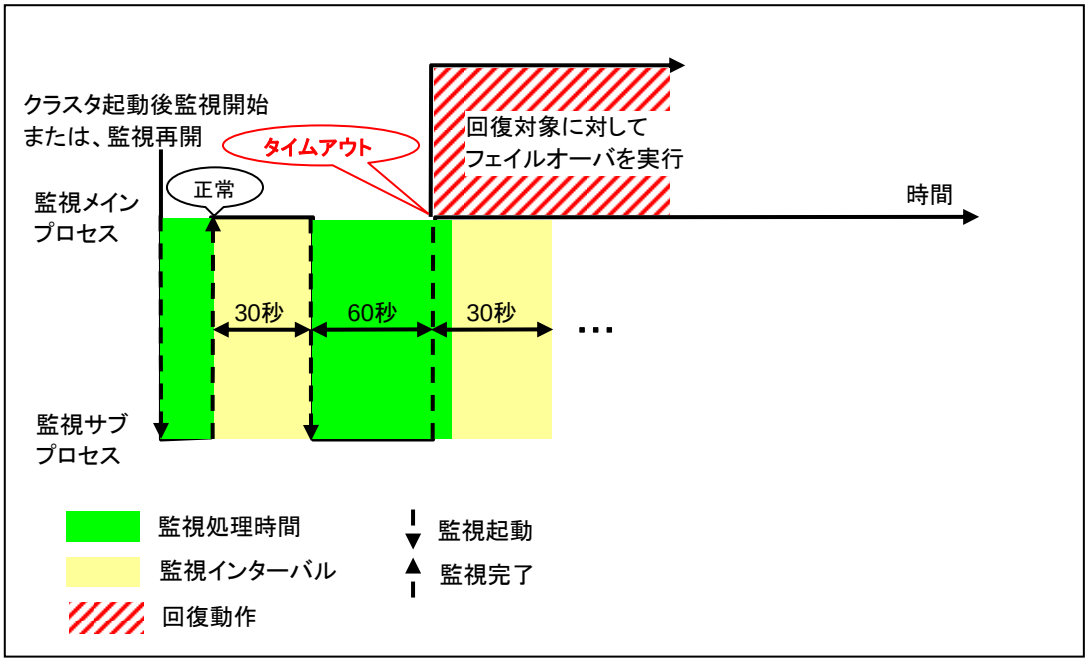


　監視異常発生後、次回監視で監視異常を検出し監視リトライ以内で回復しなければ、回復対象に対してフェイルオーバが行われます。

*　監視タイムアウト検出時(監視リトライ設定なし)

[設定例]
　　<監視>
　　監視インターバル　　30秒
　　監視タイムアウト　　60秒
　　監視リトライ回数　　0回
　<異常検出>
　　回復対象　　グループ
　　再活性化しきい値　　0回
　　フェイルオーバしきい値　　1回
　　最終動作　　何もしない

を指定している場合の挙動の例

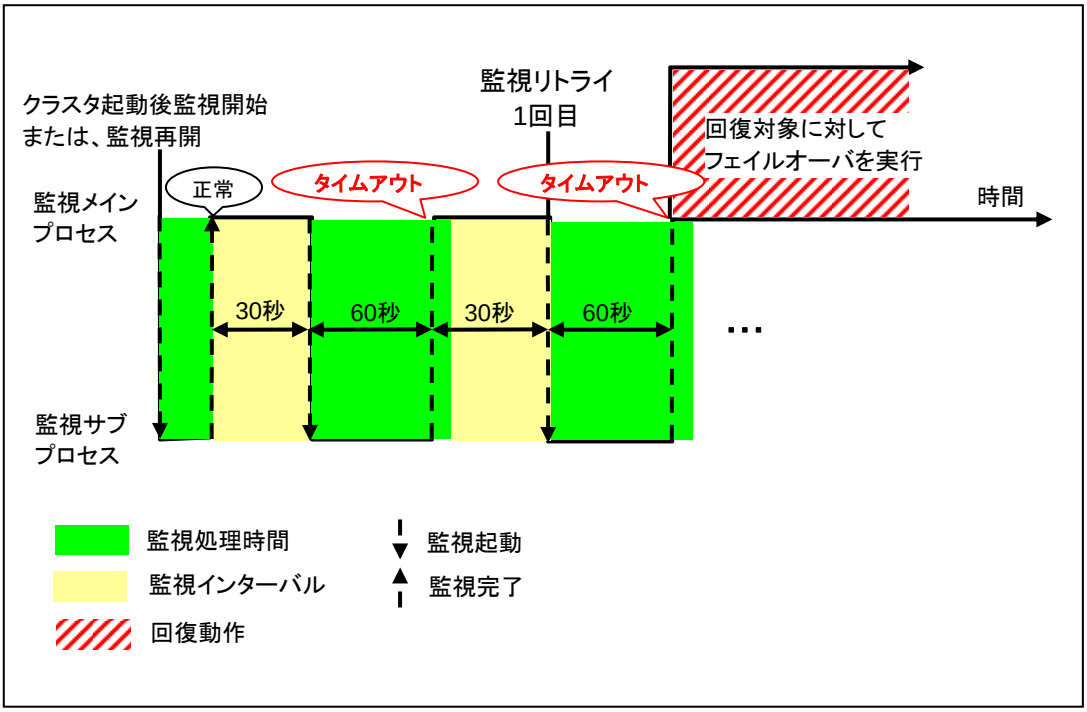


監視タイムアウト発生後、直ぐに回復対象への回復動作に対してフェイルオーバが行われます。

*　監視タイムアウト検出時(監視リトライ設定あり)

[設定例]
　　<監視>
　　監視インターバル　　30秒
　　監視タイムアウト　　60秒
　　監視リトライ回数　　1回
　<異常検出>
　　回復対象　　グループ
　　再活性化しきい値　　0回
　　フェイルオーバしきい値　　1回
　　最終動作　　何もしない

を指定している場合の挙動の例



監視タイムアウト発生後、監視リトライを行い回復対象に対してフェイルオーバが行われます。

2.1.3 異常検出

異常検出時には回復対象に対して以下の回復動作が行われます。

+ 監視対象の異常を検出すると再活性化を行います。
+ 再活性化しきい値の再活性化に失敗した場合、フェイルオーバを行います。
+ フェイルオーバしきい値のフェイルオーバを行っても異常を検出する場合、最終動作を行います。

回復動作は、回復対象が以下の状態であれば行われません。

回復対象	状態	再活性化[8]	フェイルオーバ[9]	最終動作[10]
グループリソース/フェイルオーバグループ	停止済	×	×	×
	起動/停止中	×	×	×
	起動済	○	○	○
	起動/停止失敗	○	○	○
クラスタ	-	-	-	○

モニタリソースの異常検出時の設定で回復対象にグループリソース（ディスクリソース、execリソース、...）を指定し、モニタリソースが異常を検出した場合の回復動作遷移中（再活性化 → フェイルオーバ → 最終動作）には、以下のコマンドまたは、Webマネージャからのクラスタ及びグループへの制御は行わないでください。

+ クラスタの停止 / サスペンド
+ グループの開始 / 停止 / 移動

モニタリソース異常による回復動作遷移中に上記の制御を行うと、そのグループの他のグループリソースが停止しないことがあります。
また、モニタリソース異常状態であっても最終動作実行後であれば上記制御を行うことが可能です。

監視リソースの状態が異常から復帰(正常)した場合は、再活性化回数、フェイルオーバ回数、最終動作の実行要否はリセットされます。

回復動作の再活性化回数及びフェイルオーバ回数は回復動作に失敗した場合でも1回としてカウントされることに注意してください。

[8] 再活性化しきい値に1以上が設定されている場合のみ有効になります。
[9] フェイルオーバしきい値に1以上が設定されている場合のみ有効になります。
[10] 最終動作に"何もしない"以外が設定されている場合のみ有効になります。

以下は、IPモニタリソースのIPリソースとしてゲートウェイを指定した場合で片サーバのみ異常を検出する時の流れを説明します。

[設定例]

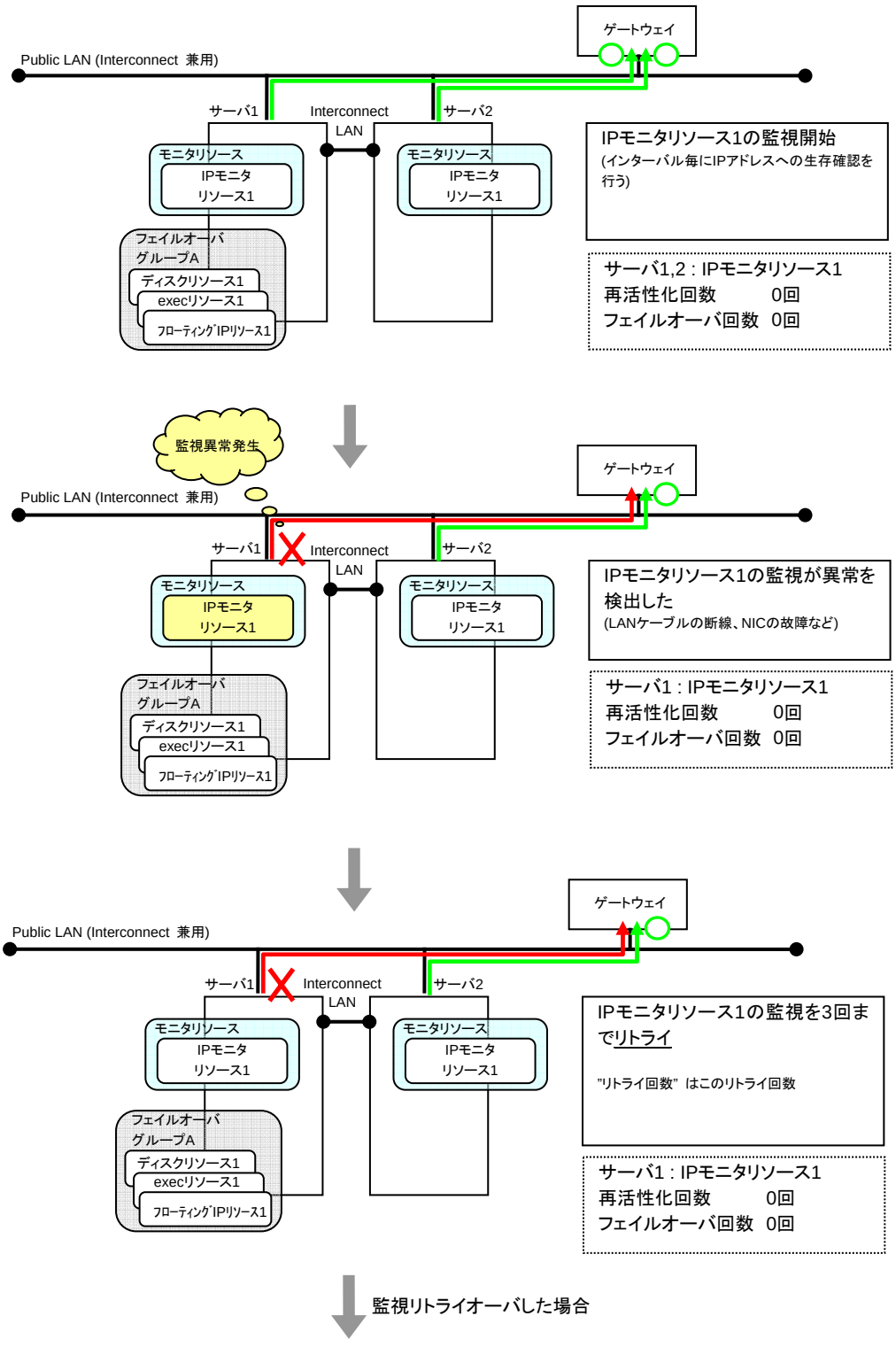
<監視>

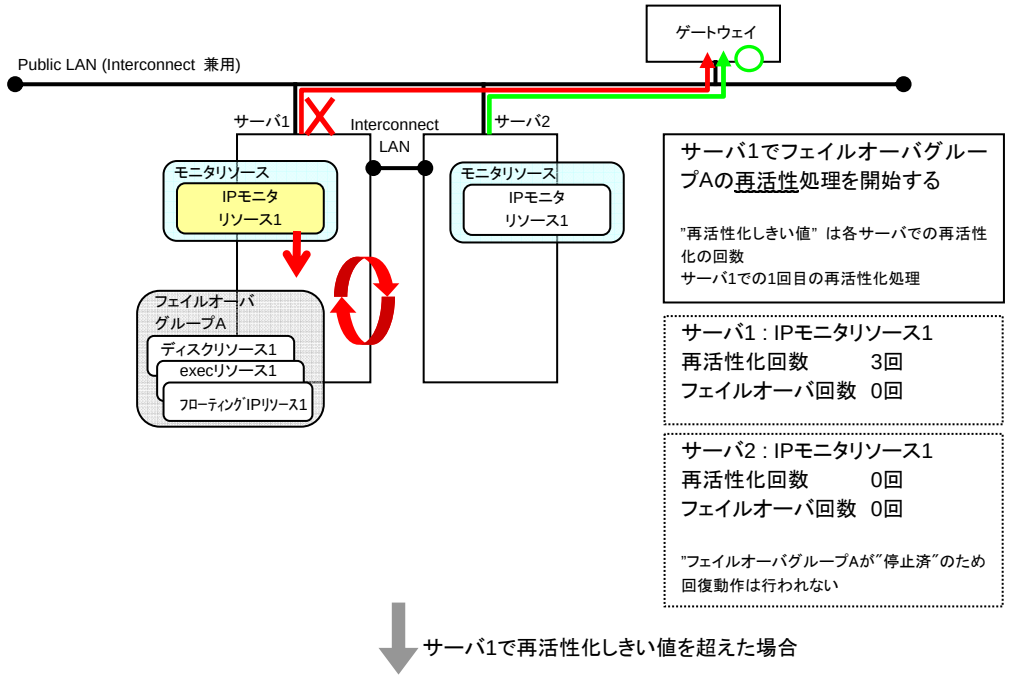
インターバル	30秒
タイムアウト	30秒
リトライ回数	3回

<異常検出>

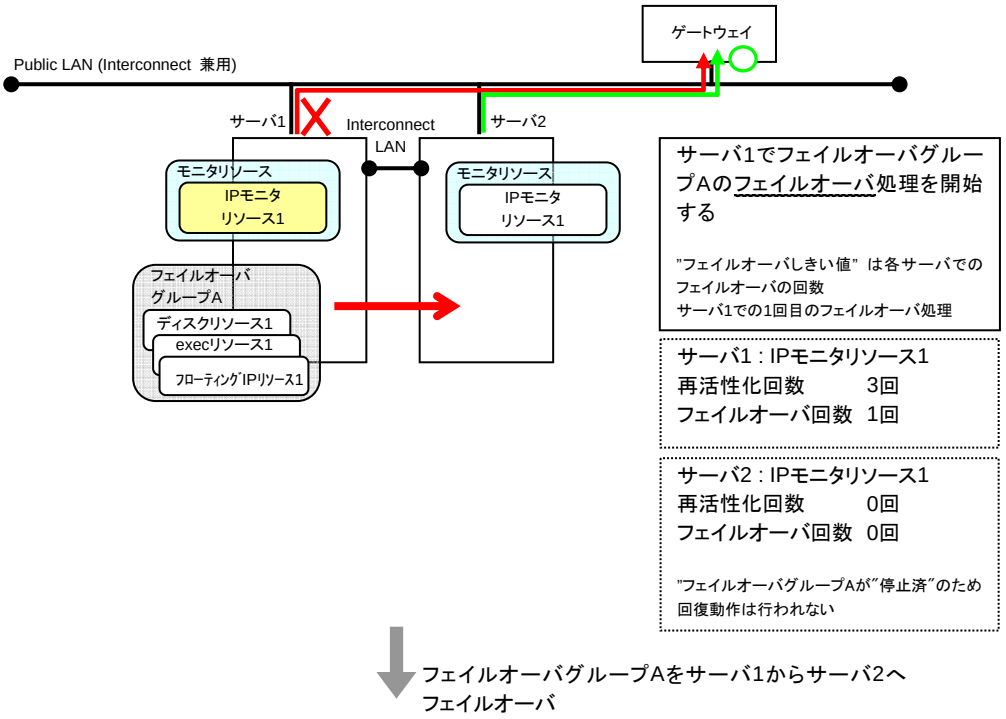
回復対象	グループA
再活性化しきい値	3回
フェイルオーバしきい値	1回
最終動作	何もしない

を指定している場合の挙動の例

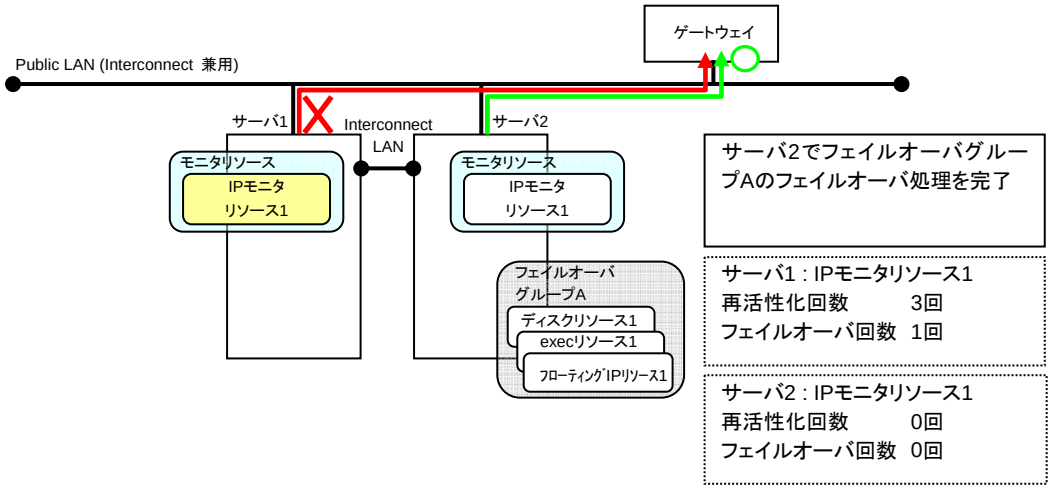

ゲートウェイ
Public LAN (Interconnect 兼用)
サーバ1
Interconnect
LAN
サーバ2
モニタリソース
IPモニタ
リソース1
フェイルオーバ
グループA
ディスクリソース1
execリソース1
ﾌﾛｰﾃｨﾝｸﾞIPﾘｿｰｽ1
IPモニタリソース1の監視開始
(インターバル毎にIPアドレスへの生存確認を行う)
サーバ1,2 : IPモニタリソース1
再活性化回数 0回
フェイルオーバ回数 0回
監視異常発生
IPモニタリソース1の監視が異常を検出した
(LANケーブルの断線、NICの故障など)
サーバ1 : IPモニタリソース1
再活性化回数 0回
フェイルオーバ回数 0回
IPモニタリソース1の監視を3回までリトライ
"リトライ回数" はこのリトライ回数
サーバ1 : IPモニタリソース1
再活性化回数 0回
フェイルオーバ回数 0回
監視リトライオーバした場合



サーバ1で再活性化しきい値を超えた場合



フェイルオーバグループAをサーバ1からサーバ2へフェイルオーバ



サーバ2では、IPモニタリソース1が正常なのでフェイルオーバグループAがフェイルオーバすることにより運用を継続することができます。

以下は、IPモニタリソースのIPリソースとしてゲートウェイを指定した場合で両サーバが異常を検出する時の流れを説明します。

[設定例]

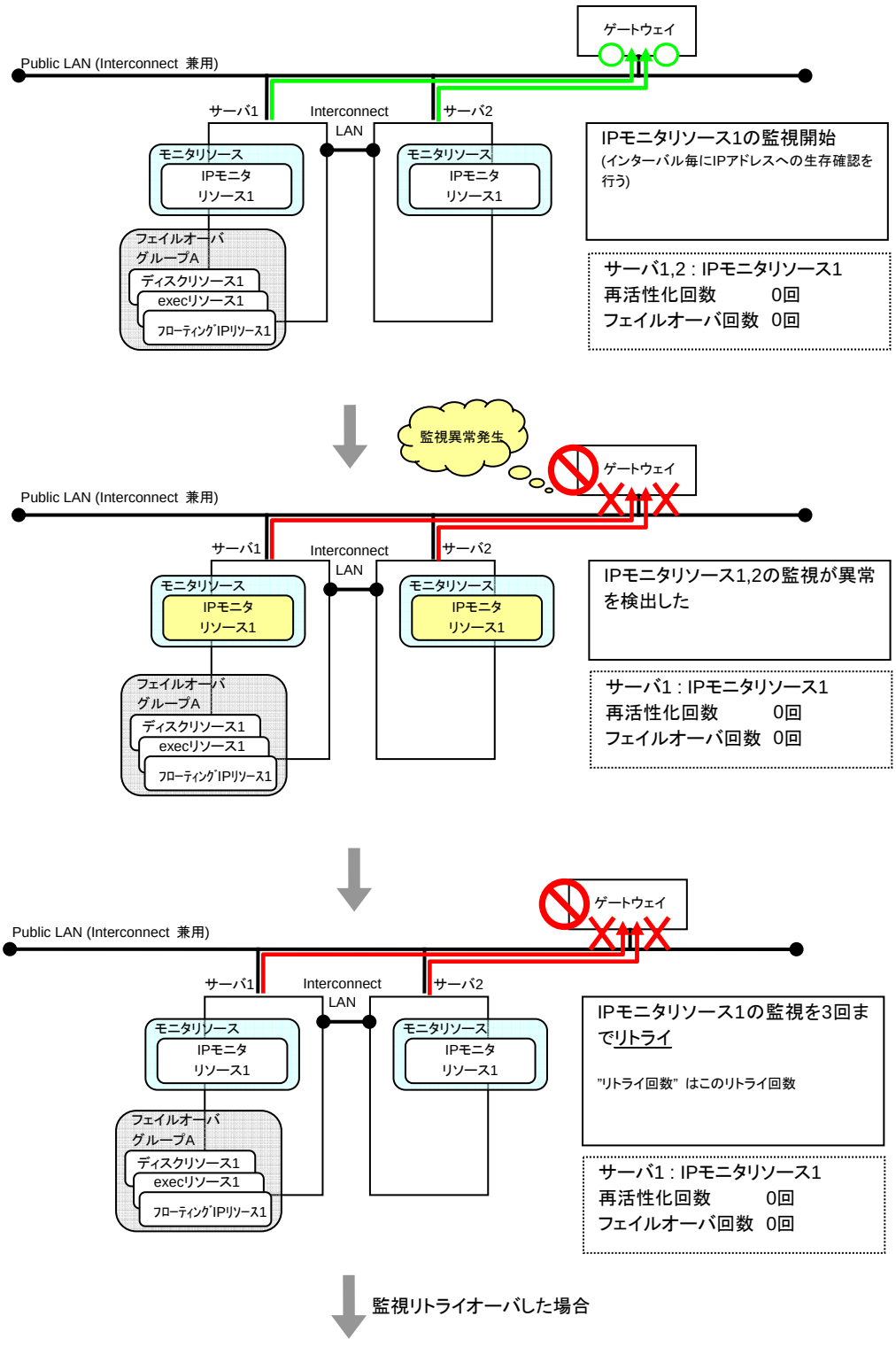
<監視>

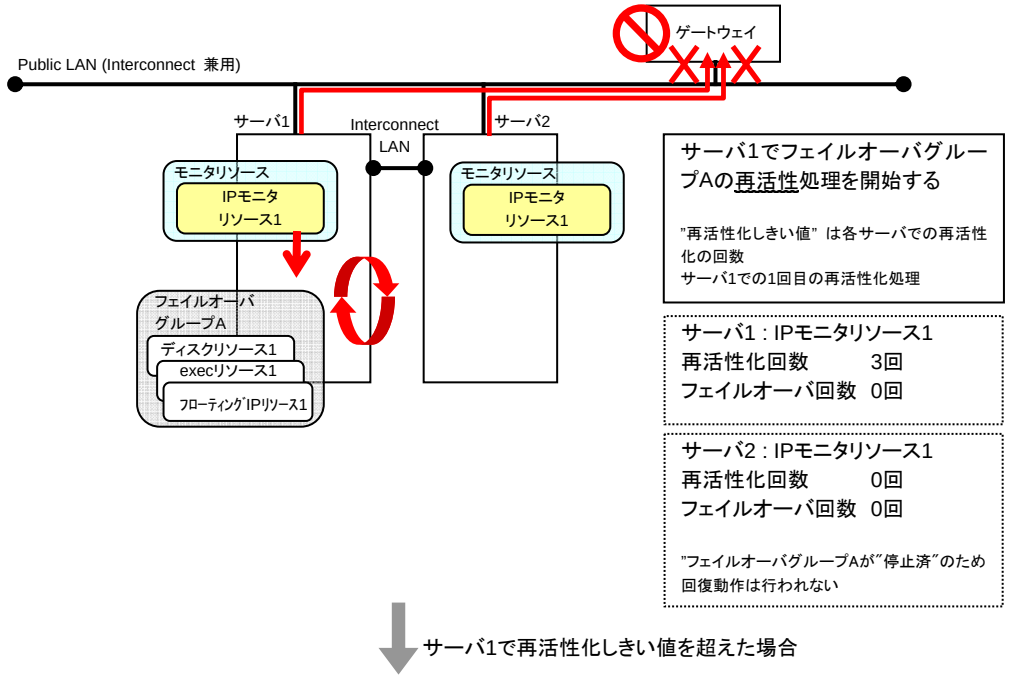
インターバル	30秒
タイムアウト	30秒
リトライ回数	3回

<異常検出>

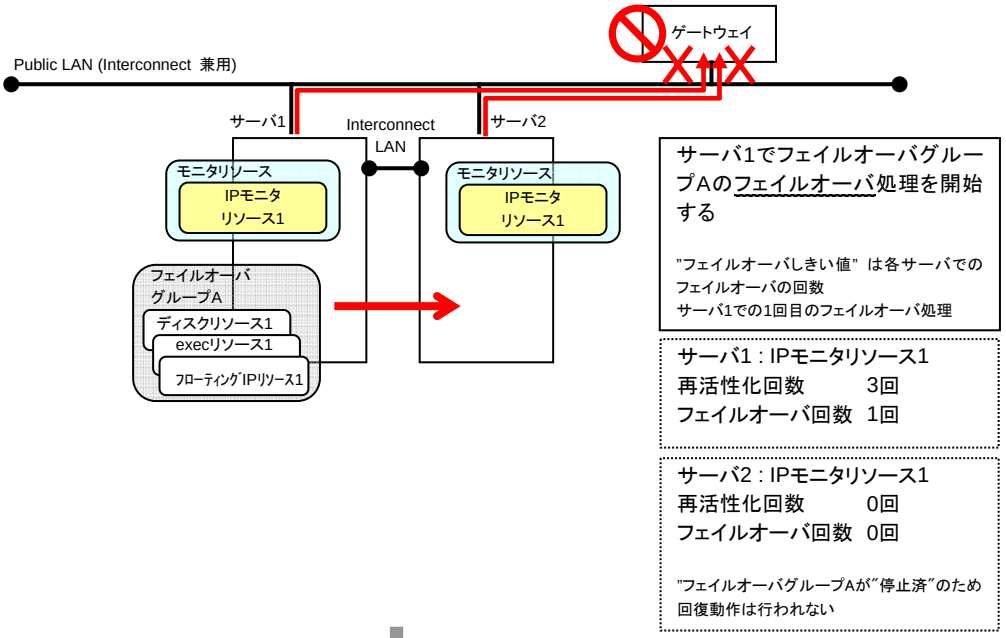
回復対象	グループA
再活性化しきい値	3回
フェイルオーバしきい値	1回
最終動作	何もしない

を指定している場合の挙動の例

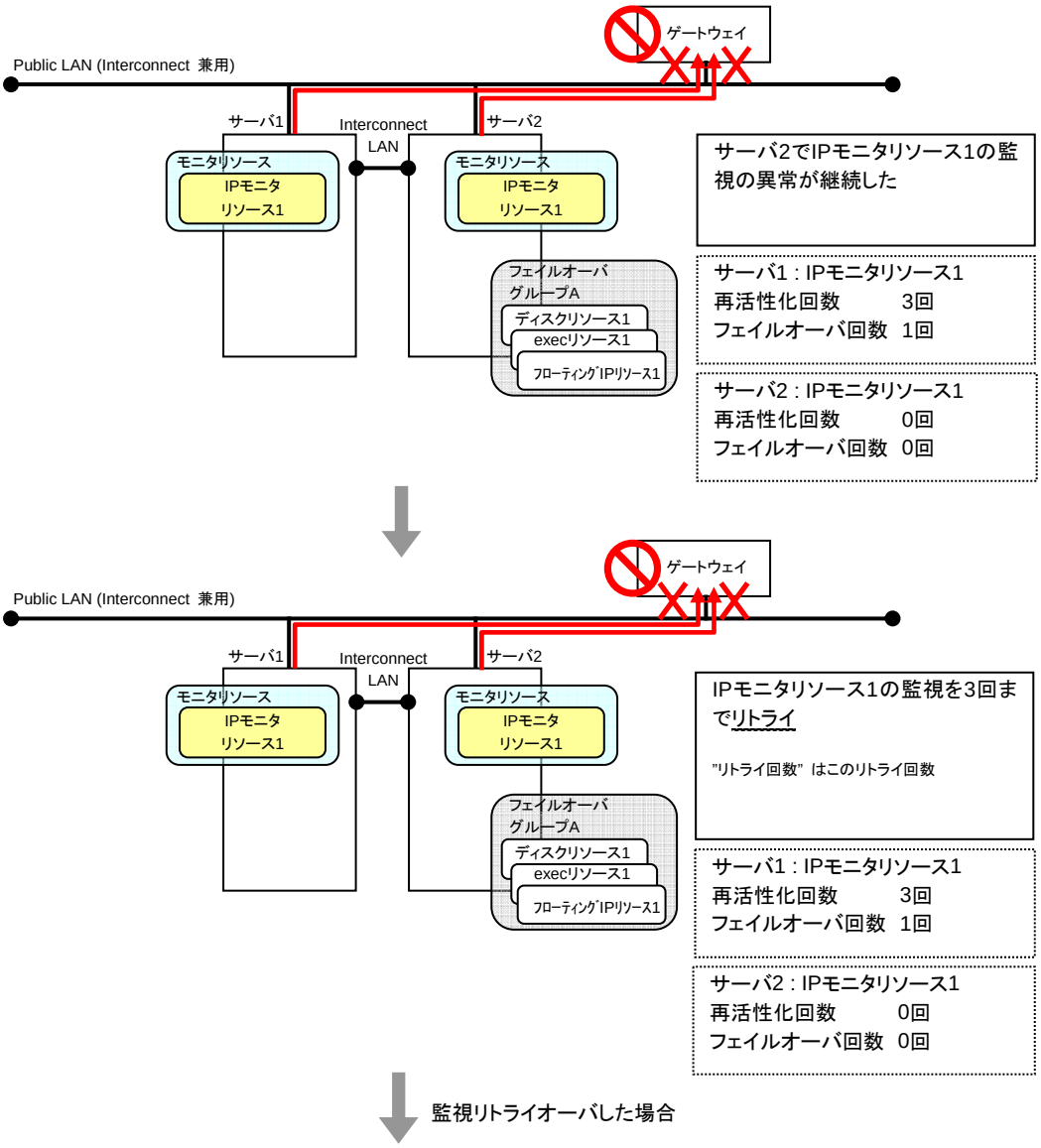

ゲートウェイ
Public LAN (Interconnect 兼用)
サーバ1
Interconnect
LAN
サーバ2
モニタリソース
IPモニタ
リソース1
フェイルオーバ
グループA
ディスクリソース1
execリソース1
ﾌﾛｰﾃｨﾝｸﾞIPﾘｿｰｽ1
IPモニタリソース1の監視開始
(インターバル毎にIPアドレスへの生存確認を行う)
サーバ1,2 : IPモニタリソース1
再活性化回数 0回
フェイルオーバ回数 0回
監視異常発生
IPモニタリソース1,2の監視が異常を検出した
サーバ1 : IPモニタリソース1
再活性化回数 0回
フェイルオーバ回数 0回
IPモニタリソース1の監視を3回までリトライ
"リトライ回数" はこのリトライ回数
監視リトライオーバした場合

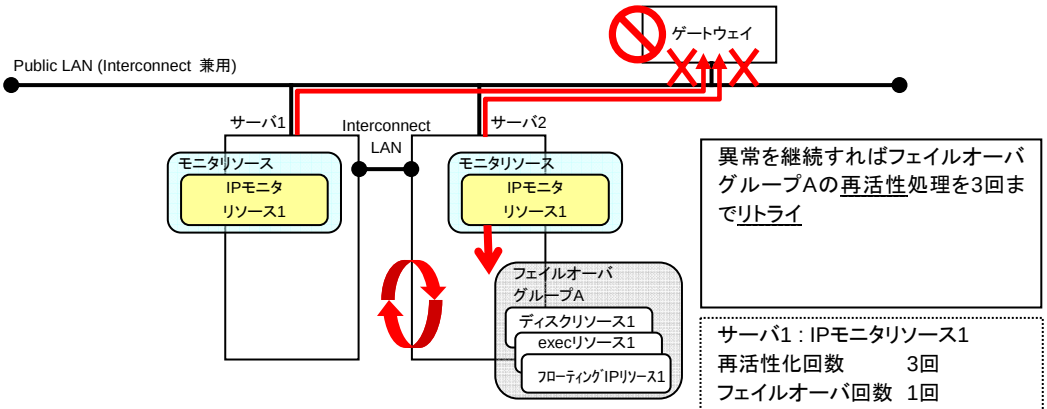


サーバ1で再活性化しきい値を超えた場合

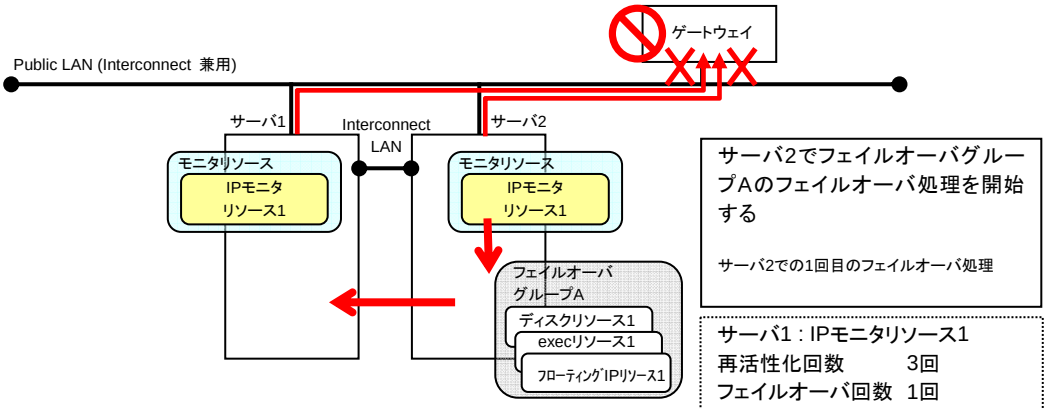


フェイルオーバグループAをサーバ1からサーバ2へフェイルオーバ

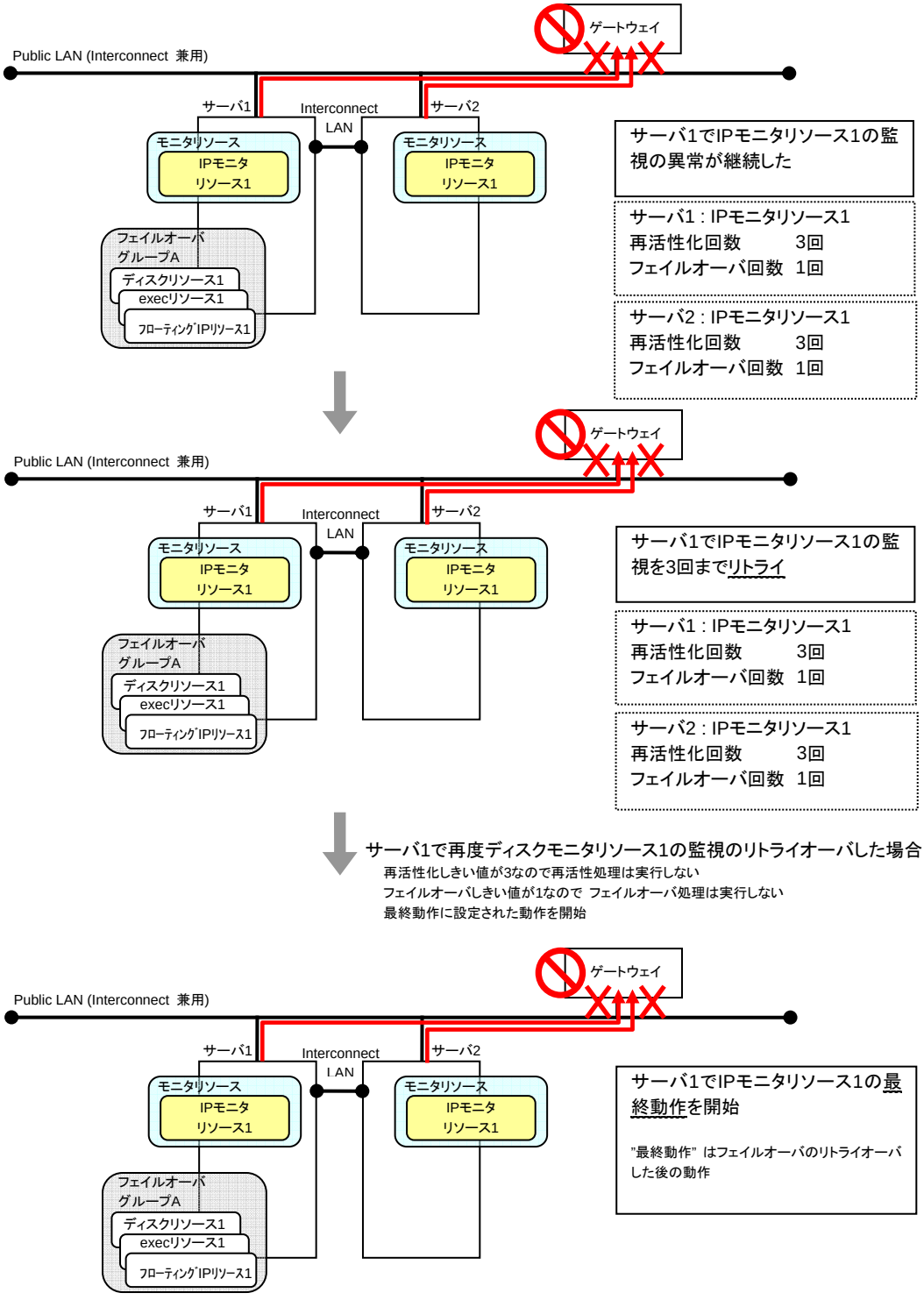

ゲートウェイ
Public LAN (Interconnect 兼用)
サーバ1
Interconnect
LAN
サーバ2
モニタリソース
IPモニタ
リソース1
モニタリソース
IPモニタ
リソース1
フェイルオーバ
グループA
ディスクリソース1
execリソース1
ﾌﾛｰﾃｨﾝｸﾞIPﾘｿｰｽ1
サーバ2でIPモニタリソース1の監視の異常が継続した
サーバ1：IPモニタリソース1
再活性化回数　3回
フェイルオーバ回数　1回
サーバ2：IPモニタリソース1
再活性化回数　0回
フェイルオーバ回数　0回
ゲートウェイ
Public LAN (Interconnect 兼用)
サーバ1
Interconnect
LAN
サーバ2
モニタリソース
IPモニタ
リソース1
モニタリソース
IPモニタ
リソース1
フェイルオーバ
グループA
ディスクリソース1
execリソース1
ﾌﾛｰﾃｨﾝｸﾞIPﾘｿｰｽ1
IPモニタリソース1の監視を3回までリトライ
"リトライ回数" はこのリトライ回数
サーバ1：IPモニタリソース1
再活性化回数　3回
フェイルオーバ回数　1回
サーバ2：IPモニタリソース1
再活性化回数　0回
フェイルオーバ回数　0回
監視リトライオーバした場合



サーバ2でも再活性処理でリトライオーバした場合



フェイルオーバグループAをサーバ2からサーバ1へフェイルオーバ



【補足】

監視しているサーバでモニタリソースが異常から正常に状態変化すると、再活性化回数とフェイルオーバ回数は0にリセットされ、次回監視異常時には同様に回復動作を行います。

以上の流れは、インタコネクトLANが健全であることが前提となります。
全てのインタコネクトLANが切断された状態では、他サーバとの内部通信が不可能なため、監視対象の異常を検出してもグループのフェイルオーバ処理が失敗します。
全てのインタコネクトLANの断線を想定してグループのフェイルオーバを可能にする方法として、異常を検出したサーバをシャットダウンさせることができます。これにより他サーバがサーバダウンを検出してグループのフェイルオーバを開始します。

以下の設定例で、全インタコネクトLANが断線状態での異常検出の流れを説明します。

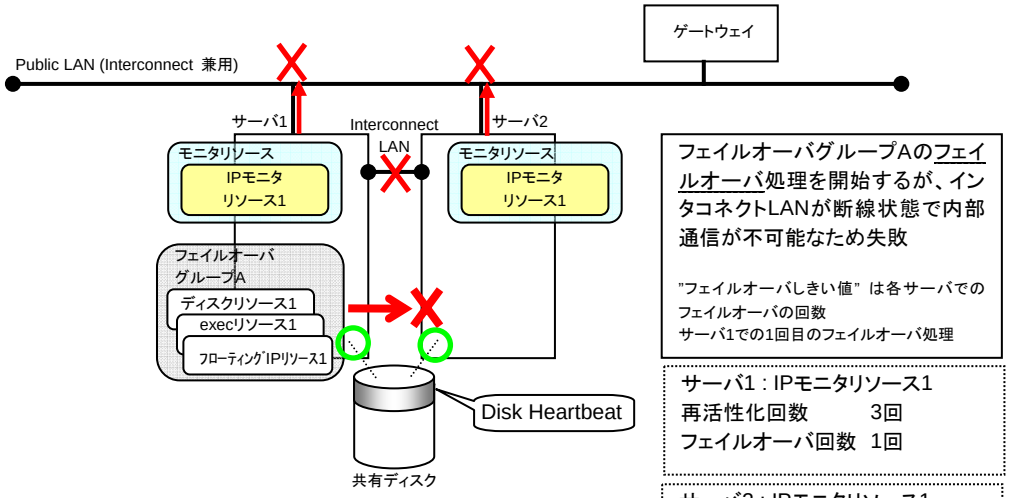
[設定例]
<監視>
インターバル 30秒
タイムアウト 30秒
リトライ回数 3回
<異常検出>
回復対象 フェイルオーバグループA
再活性化しきい値 3回
フェイルオーバしきい値 1回
最終動作 クラスタデーモン停止及びOSシャットダウン

を指定している場合の挙動の例

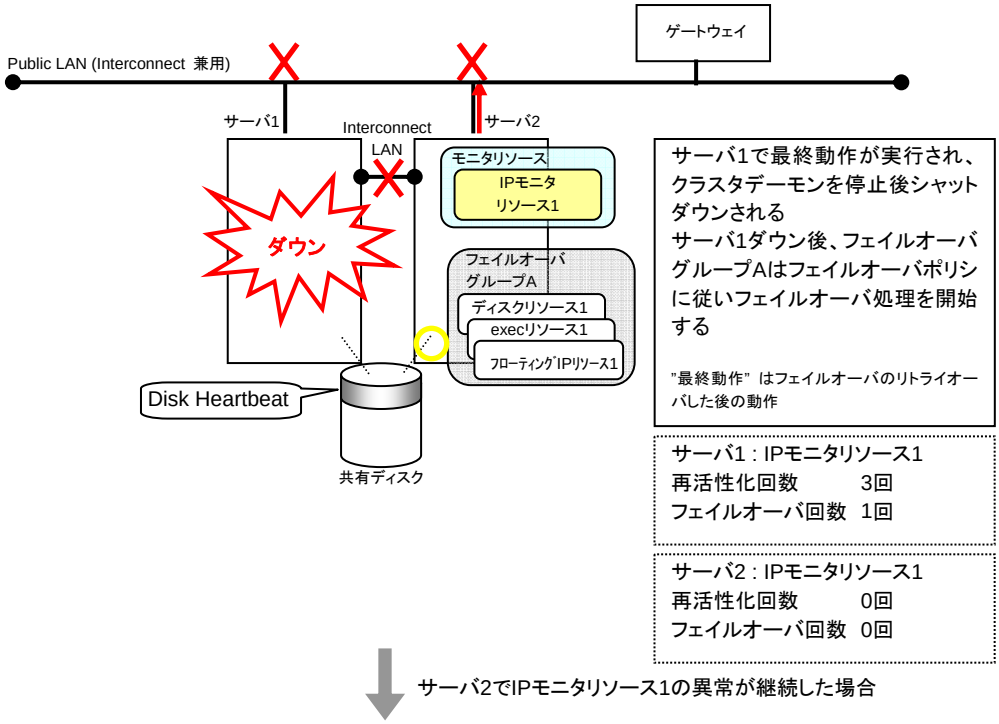
回復対象への再活性化処理は、インタコネクトLANが健全な場合と同じです。
インタコネクトLANが必要となる、サーバ1でのフェイルオーバ処理から説明します。

サーバ1：IPモニタリソース1
再活性化回数 3回
フェイルオーバ回数 0回

再活性化しきい値を超えた場合



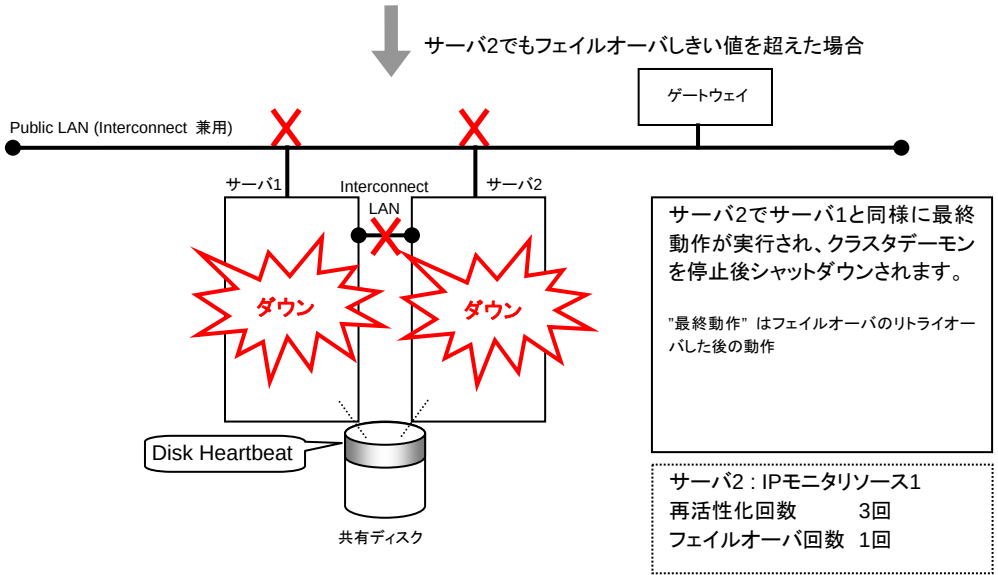
サーバ1でフェイルオーバしきい値を超えた場合



サーバ2においてサーバ1と同様にグループAの再活性化を実行します。

サーバ2でもグループAの再活性化で異常が発生するとフェイルオーバを試みます。しかし、フェイルオーバに関しては、フェイルオーバ先が無いのでフェイルオーバできません。

フェイルオーバしきい値を超えた場合、サーバ1と同様にサーバ2で最終動作が実行されます。



2.1.4 監視異常からの復帰(正常)

監視異常を検出し回復動作遷移中または、全ての回復動作を完了後に監視リソースの復帰を検出すると、その監視リソースが保持している以下のしきい値に対する回数カウンタはリセットされます。

+ 再活性化しきい値
+ フェイルオーバしきい値

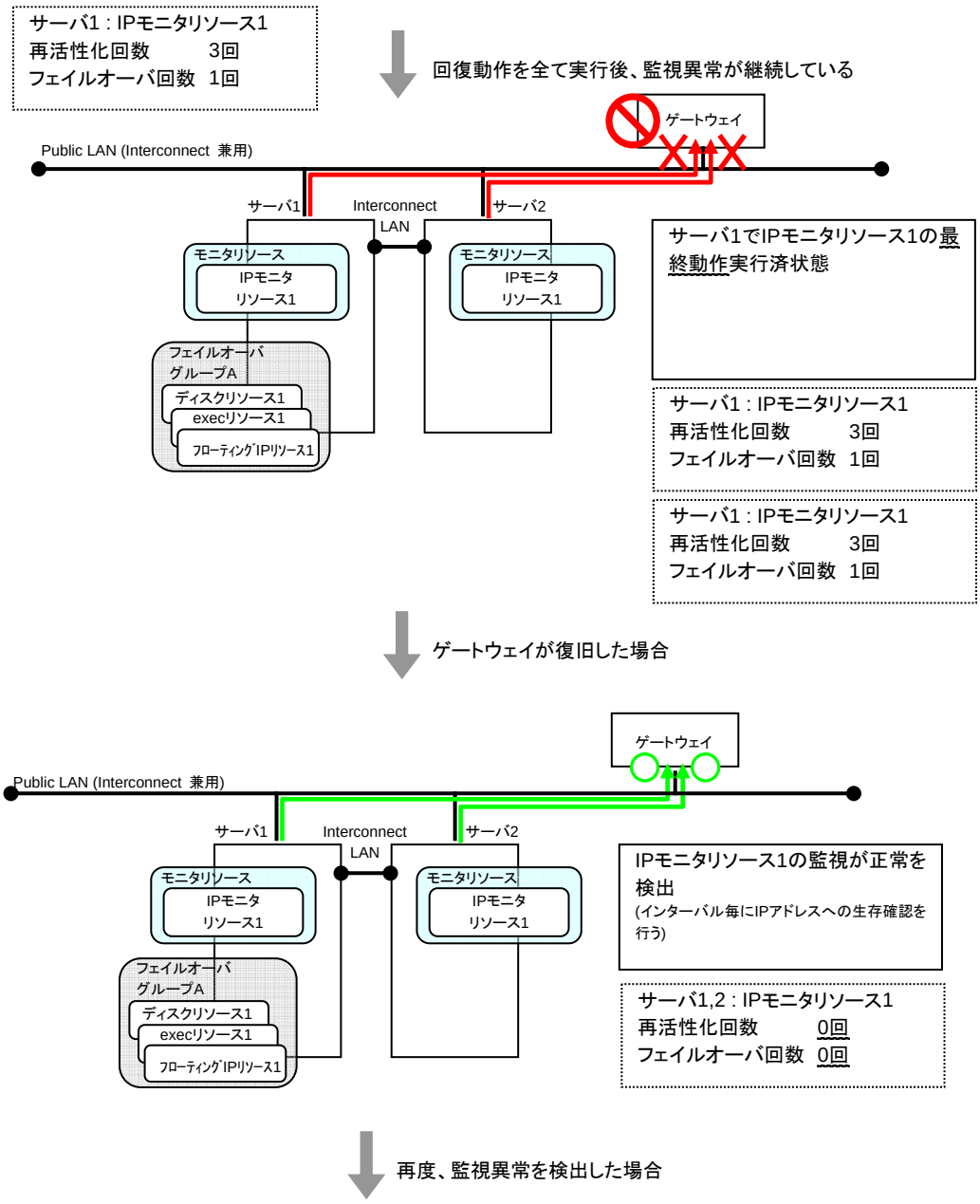
最終動作については、実行要否がリセット(実行要に)されます。

以下は「2.1.3 異常検出」の最終動作実行後から監視が正常に復帰し、再度監視が異常になる流れを説明します。

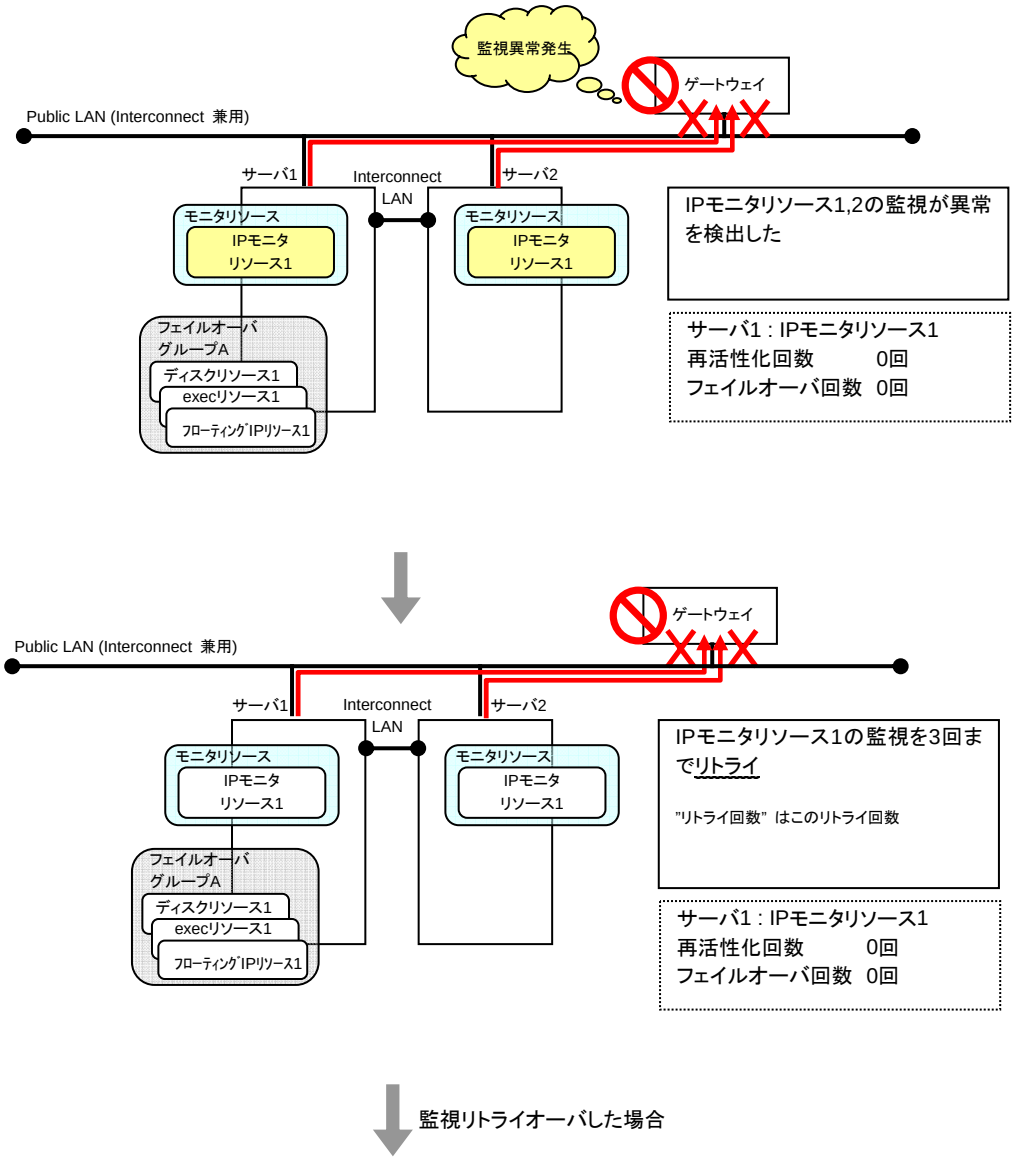
[設定例]

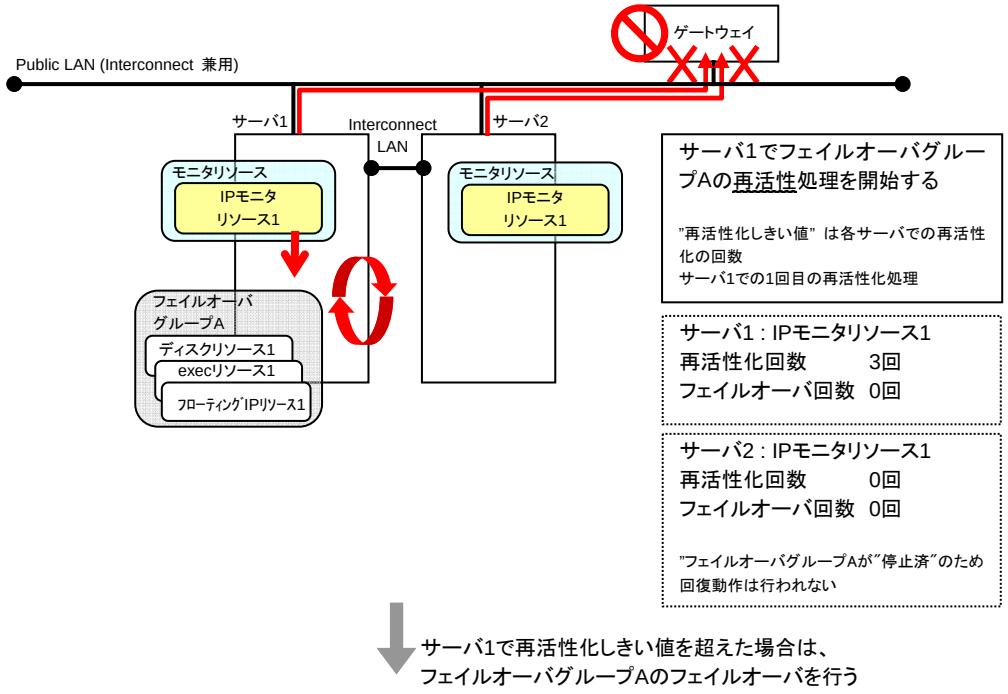
<監視>	
インターバル	30秒
タイムアウト	30秒
リトライ回数	3回
<異常検出>	
回復対象	グループA
再活性化しきい値	3回
フェイルオーバしきい値	1回
最終動作	グループ停止

を指定している場合の挙動の例



監視リソースの復帰を検出されたため再活性化回数及びフェイルオーバ回数はリセットされます。


監視異常発生
ゲートウェイ
Public LAN (Interconnect 兼用)
サーバ1
Interconnect
LAN
サーバ2
モニタリソース
IPモニタ
リソース1
モニタリソース
IPモニタ
リソース1
フェイルオーバ
グループA
ディスクリソース1
execリソース1
ﾌﾛｰﾃｨﾝｸﾞIPﾘｿｰｽ1
IPモニタリソース1,2の監視が異常を検出した
サーバ1：IPモニタリソース1
再活性化回数 0回
フェイルオーバ回数 0回
ゲートウェイ
Public LAN (Interconnect 兼用)
サーバ1
Interconnect
LAN
サーバ2
モニタリソース
IPモニタ
リソース1
モニタリソース
IPモニタ
リソース1
フェイルオーバ
グループA
ディスクリソース1
execリソース1
ﾌﾛｰﾃｨﾝｸﾞIPﾘｿｰｽ1
IPモニタリソース1の監視を3回までリトライ
"リトライ回数" はこのリトライ回数
サーバ1：IPモニタリソース1
再活性化回数 0回
フェイルオーバ回数 0回
監視リトライオーバした場合



監視異常を継続していれば、フェイルオーバグループAの再活性化処理ではなくサーバ2へのフェイルオーバが行われますが、以前に監視リソースの復帰が検出され再活性化回数がリセットされているため再度、再活性化処理を行います。

2.1.5 回復動作時の回復対象活性/非活性異常

監視リソースの監視先と回復対象のグループリソースが同一のデバイス場合で監視異常を検出すると、回復動作中にグループリソースの活性/非活性異常を検出する場合があります。

以下はディスクモニタリソースの監視先とフェイルオーバグループAのディスクリソースを同一デバイスに指定した場合の回復動作の流れを説明します。

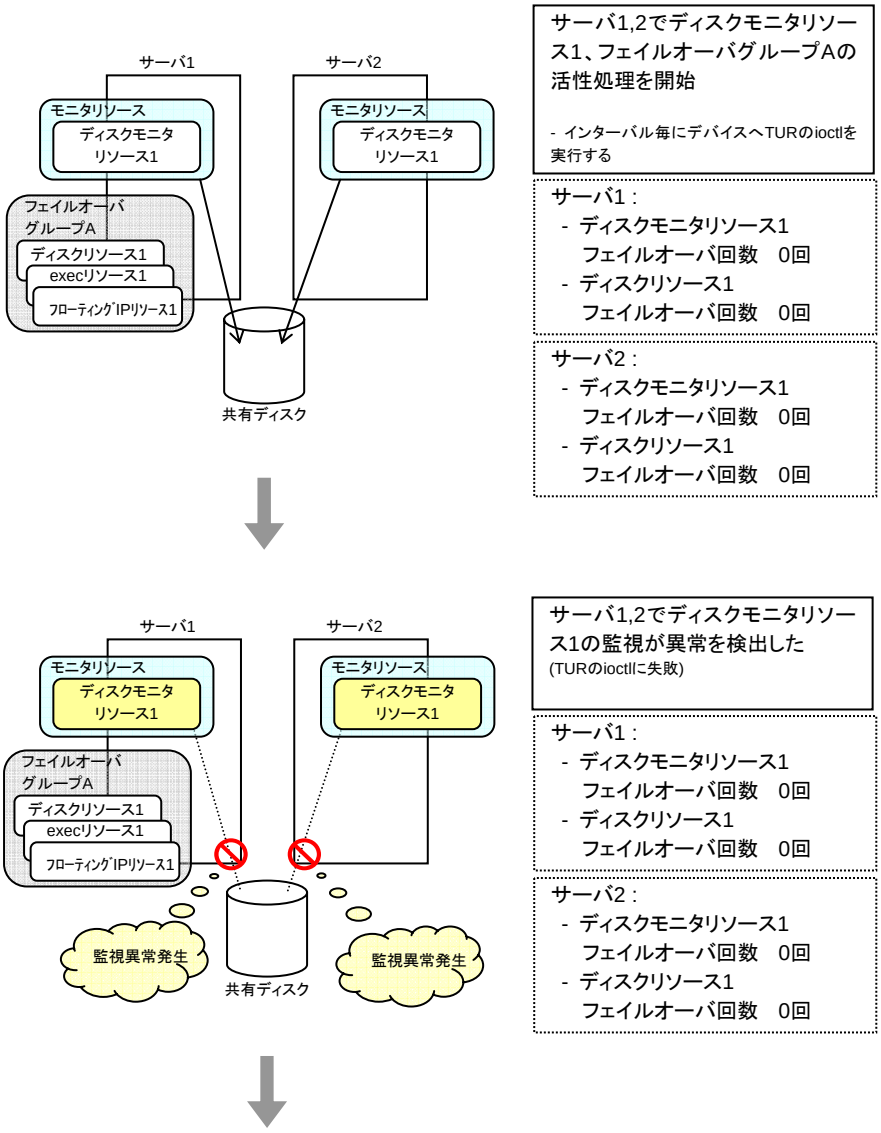
[ディスクモニタリソースの設定例]

<監視>	
インターバル	60秒
タイムアウト	120秒
リトライ回数	0回
<異常検出>	
回復対象	グループA
再活性化しきい値	0回
フェイルオーバしきい値	1回
最終動作	グループ停止
<パラメータ>	
監視方法	TUR

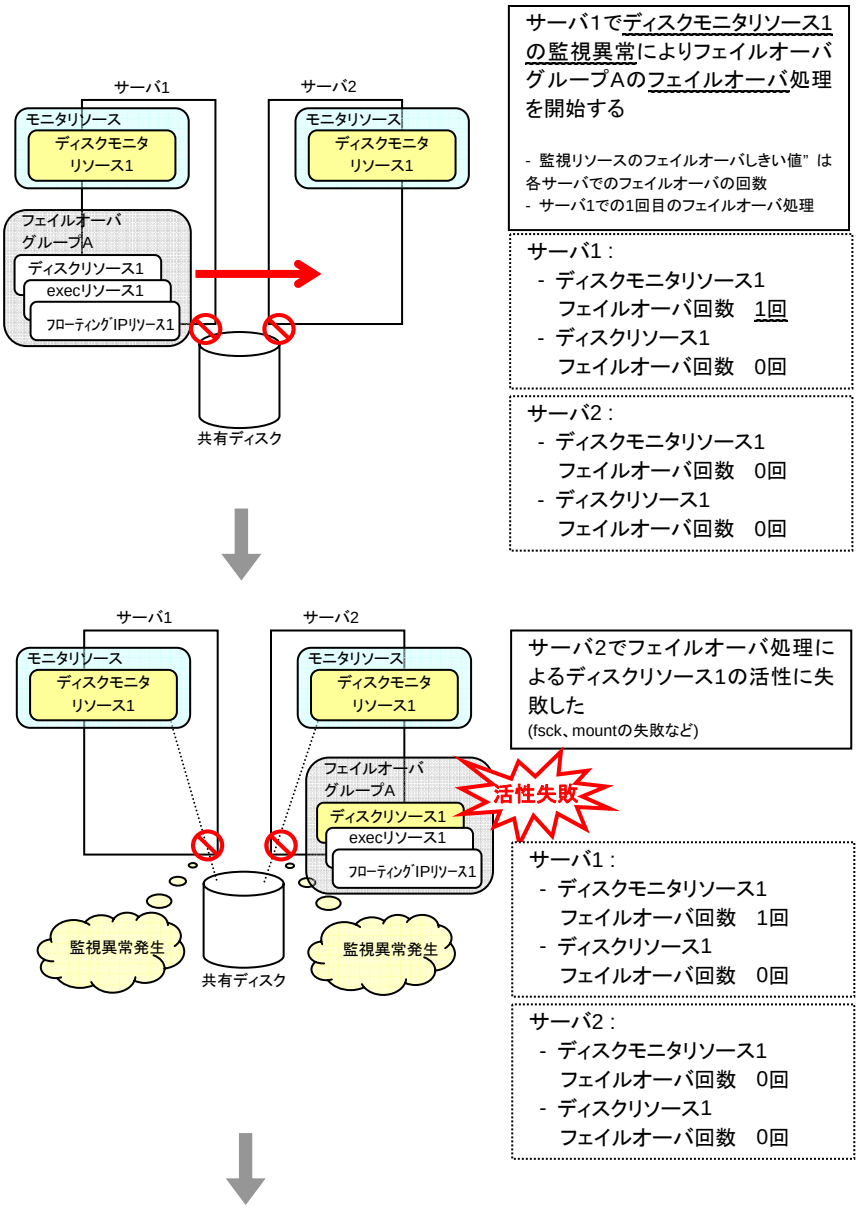
[フェイルオーバグループA： ディスクリソースの設定例]

<活性異常>	
活性リトライしきい値	0回
フェイルオーバしきい値	1回
最終動作	何もしない(次のリソースを活性しない)
<非活性異常>	
非活性リトライしきい値	0回
最終動作	クラスタデーモン停止とOSシャットダウン

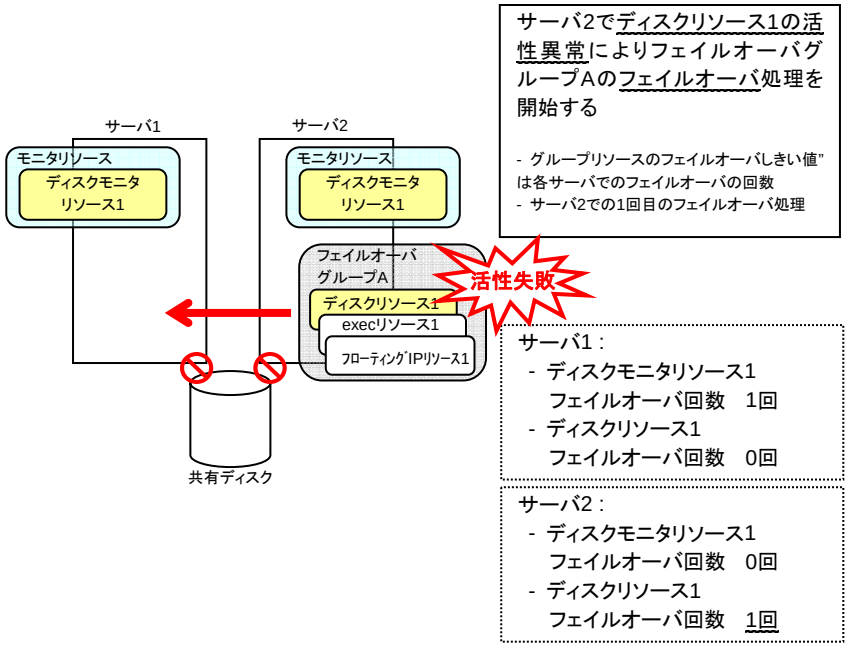
モニタリソースの再活性化しきい値とグループリソースの活性リトライしきい値は、共に設定回数が0回のため遷移図内では省略します。



ディスクデバイスの障害箇所によっては、ディスクリソースの非活性処理で異常を検出する場合があります。

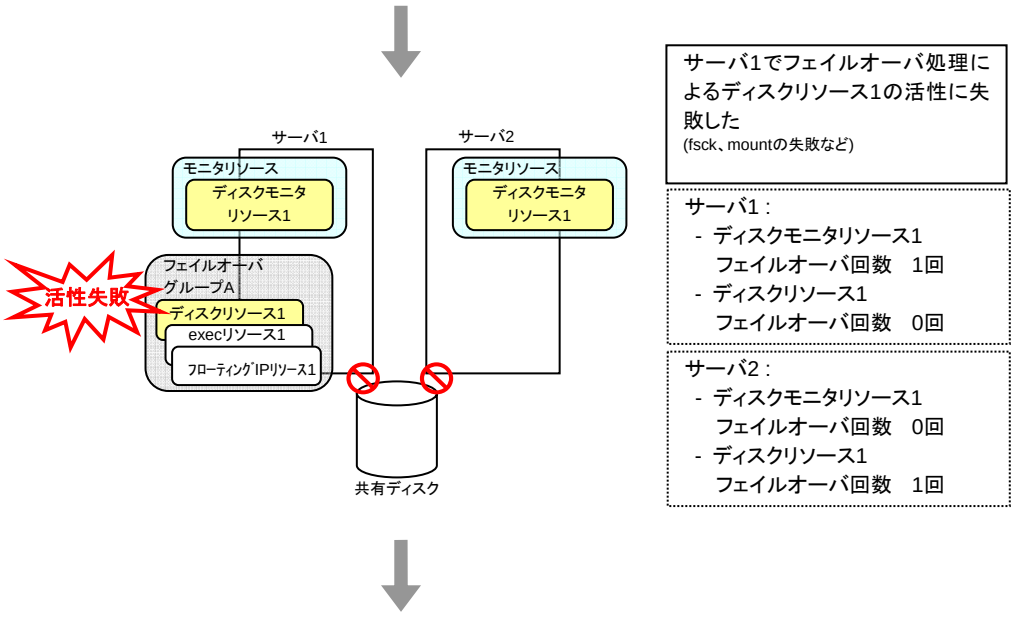


ディスクデバイスの障害箇所によっては、ディスクリソースの非活性処理で異常を検出する場合があります。

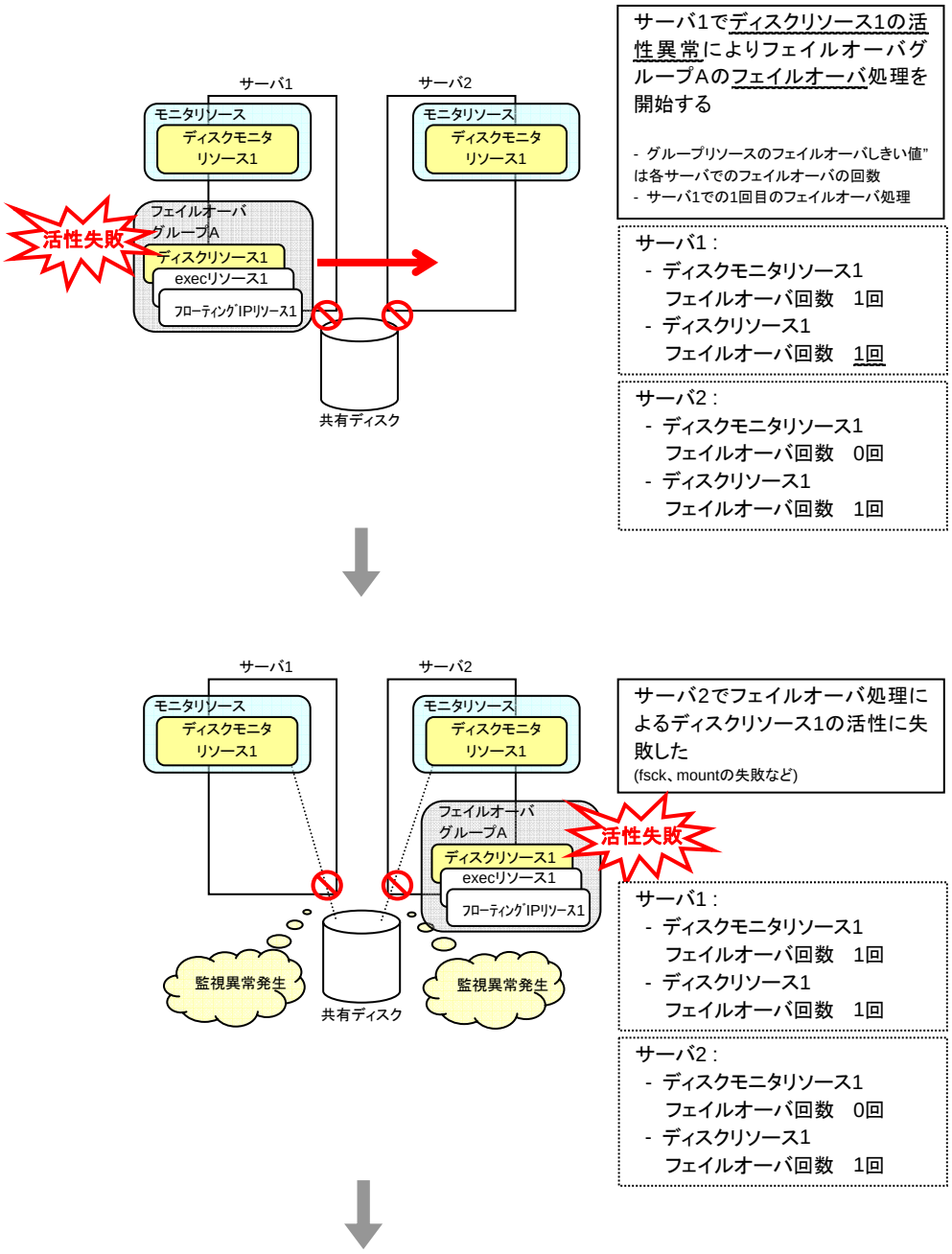


サーバ2でもサーバ1と同様にディスクモニタリソース1の異常を検出していますが、回復対象である"フェイルオーバグループA"が起動中のため回復動作は行われません。

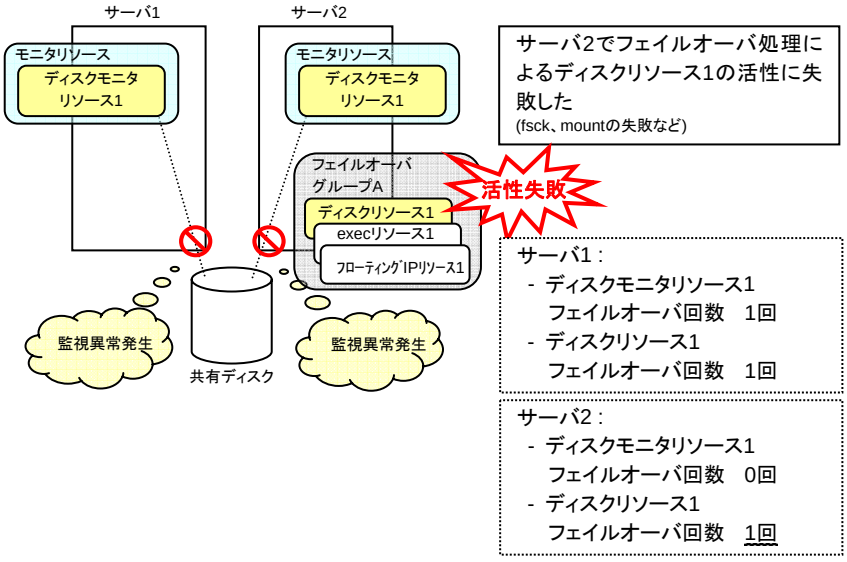
監視リソースが回復対象に対して回復動作を行う条件については、「2.1.3 異常検出」を参照してください。



ディスクデバイスの障害箇所によっては、ディスクリソースの非活性処理で異常を検出する場合があります。

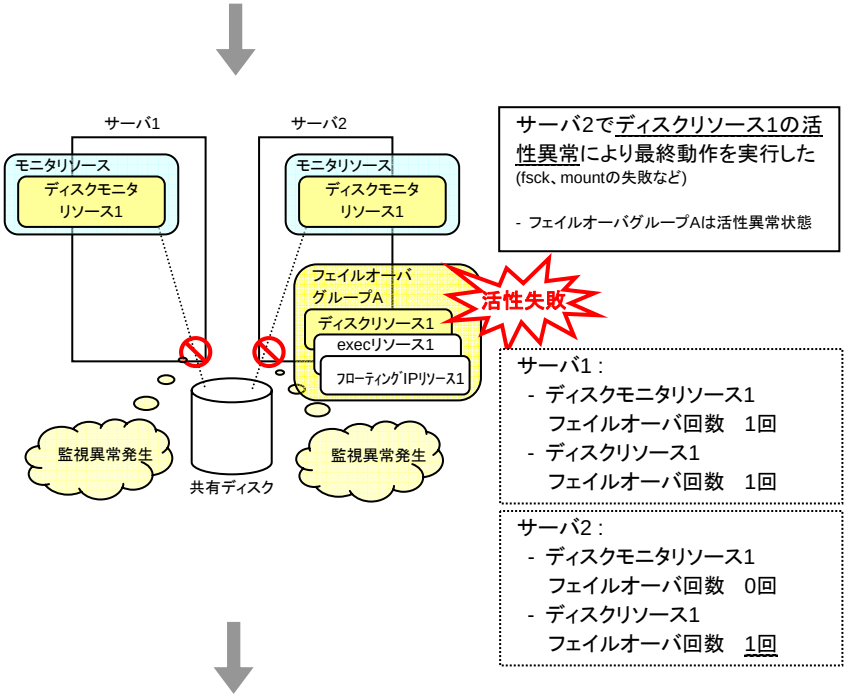


ディスクデバイスの障害箇所によっては、ディスクリソースの非活性処理で異常を検出する場合があります。

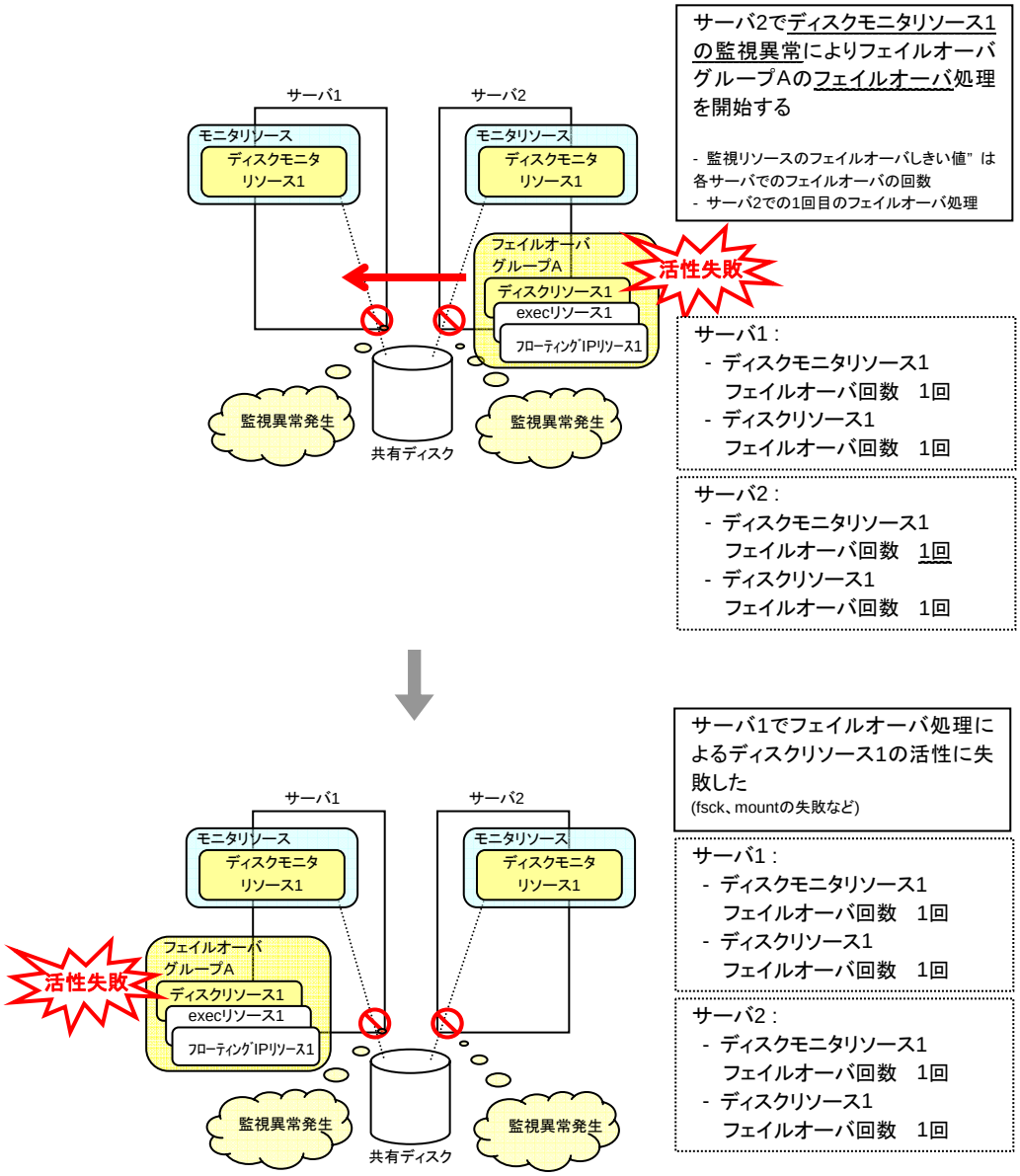


サーバ2では、ディスクリソース1の活性異常によるフェイルオーバ回数がしきい値を超えているため、最終動作を実行します。

但し、最終動作には"何もしない(次のリソースを活性しない)"が設定されているため、フェイルオーバグループAの残りのグループリソースは活性されず、起動処理は異常終了となります。



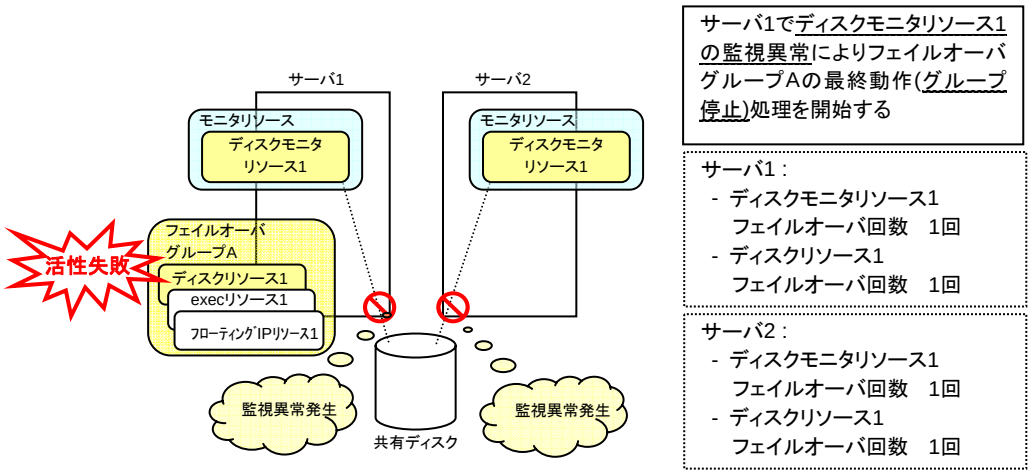
ディスクデバイスの障害箇所によっては、ディスクリソースの非活性処理で異常を検出する場合があります。



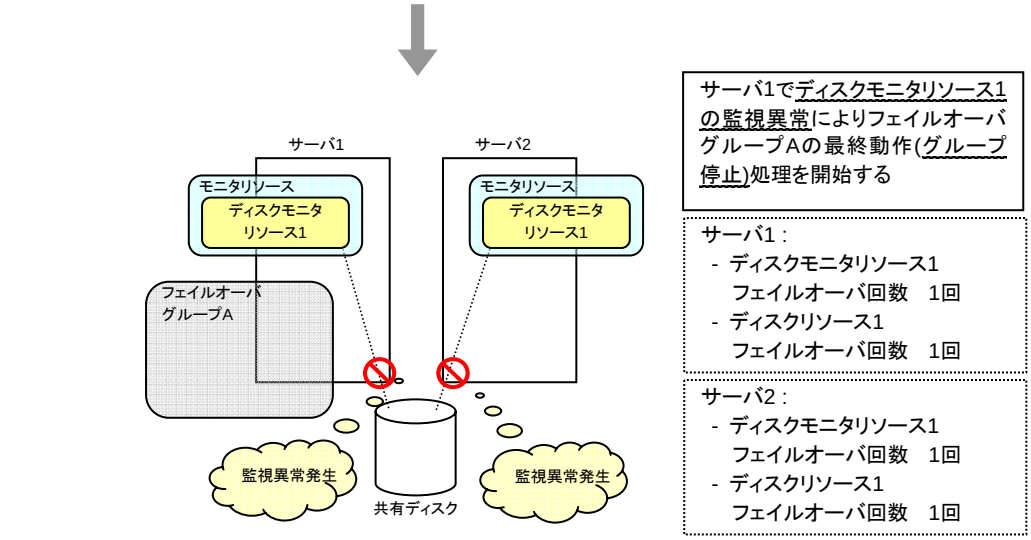
サーバ1でもサーバ2と同様に、ディスクリソース1の活性異常によるフェイルオーバ回数がしきい値を超えているため、最終動作を実行します。

但し、最終動作には"何もしない(次のリソースを活性しない)"が設定されているため、フェイルオーバグループAの残りのグループリソースは活性されず、起動処理は異常終了となります。

ディスクデバイスの障害箇所によっては、ディスクリソースの非活性処理で異常を検出する場合があります。



サーバ1では、ディスクモニタリソース1の監視異常によるフェイルオーバ回数がしきい値を超えているため、最終動作を実行します。



サーバ1で実行されたディスクモニタリソース1の最終動作によりフェイルオーバグループAが停止したため、これ以降でディスクモニタリソース1の監視異常を検出しても何も起こりません。

但し、サーバ2ではディスクモニタリソース1の最終動作が未だ実行されていないため、フェイルオーバグループAを手動で起動した場合は、ディスクモニタリソース1の最終動作が実行されます。

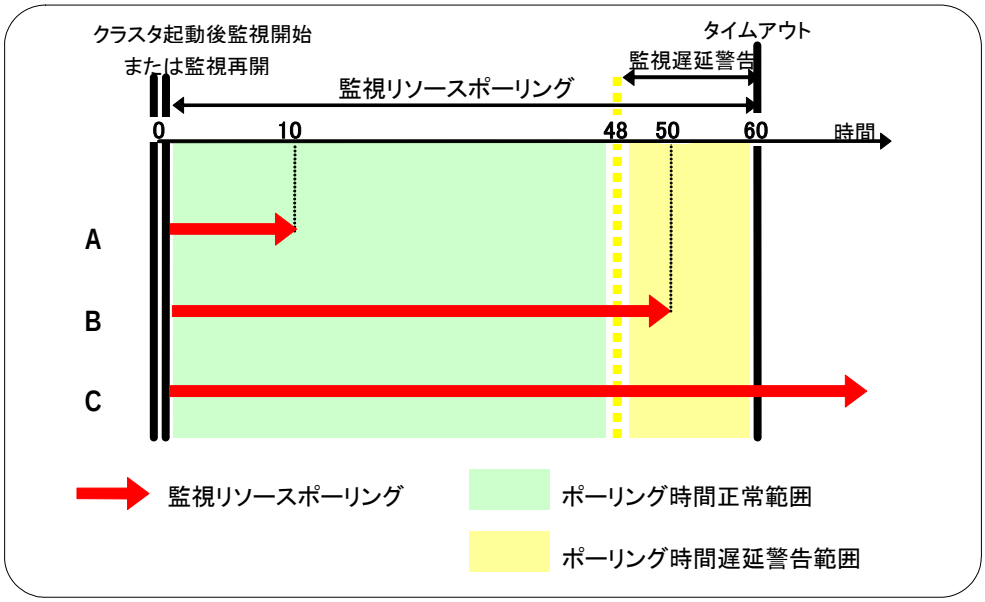
2.1.6 遅延警告

CLUSTERPRO	Version
サーバ	SE3.1-1 以降、LE3.1-1 以降、XE3.1-4 以降、SX3.1-2 以降
トレッキングツール	対応するバージョンについては動作環境編 トレッキングツールの動作環境を参照してください。

監視リソースは業務アプリケーションの集中などにより、サーバが高負荷状態になり監視タイムアウトを検出する場合があります。監視タイムアウトを検出する前に監視のポーリング時間（実測時間）が監視タイムアウト時間の何割かに達した場合、アラート通報させることが可能です。

以下は、監視リソースが遅延警告されるまでの流れを時系列で表した説明です。

監視タイムアウトに60秒、遅延警告割合には、規定値の80%を指定します。



A. 監視のポーリング時間は、10秒で監視リソースは正常状態。
この場合、アラート通報は行いません。

B. 監視のポーリング時間は、50秒で監視の遅延を検出し、監視リソースは正常状態。
この場合、遅延警告割合の80%を超えているためアラート通報を行います。

C. 監視のポーリング時間は、監視タイムアウト時間の60秒を越え監視タイムアウトを検出し、監視リソースは異常状態。
この場合、アラート通報は行いません。

また、遅延警告割合を0または、100に設定すれば以下のようなことを行うことが可能です。

+ 遅延警告割合に0を設定した場合
  監視毎に遅延警告がアラート通報されます。
  この機能を利用し、サーバが高負荷状態での監視リソースへのポーリング時間を算出し、監視リソースの監視タイムアウト時間を決定することができます。

+ 遅延警告割合に100を設定した場合
  遅延警告の通報を行いません。

ハートビートリソースについても同様にハートビートの遅延警告をアラート通報します。

* ユーザ空間モニタリソースの場合は「2.9.9.1」を参照してください。

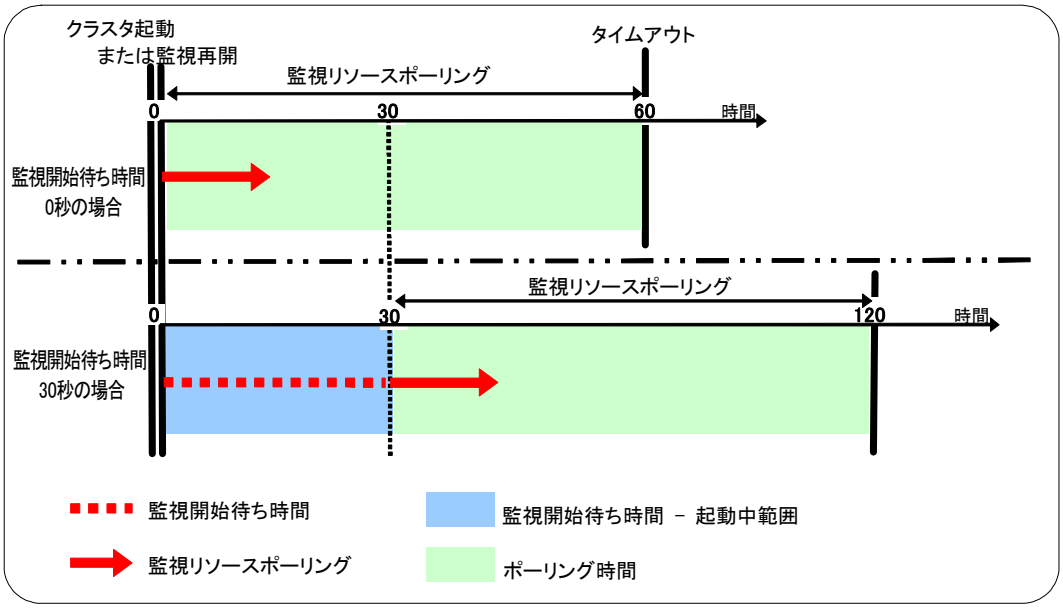
テスト運用以外で、0%等の低い値を設定しないように注意してください。

2.1.7 監視開始待ち

CLUSTERPRO	Version
サーバ	SE3.1-1 以降、LE3.1-1 以降、XE3.1-4 以降、SX3.1-2 以降
トレッキングツール	対応するバージョンについては動作環境編 トレッキングツールの動作環境を参照してください。

監視開始待ちとは、監視を指定した監視開始待ち時間後から開始することをいいます。
以下は、監視開始待ちを0秒に指定した場合と30秒に指定した場合の監視の違いを時系列で表した説明です。

[監視リソース構成]
<監視>
インターバル 30秒
タイムアウト 60秒
リトライ回数 0回
監視開始待ち時間 0秒 / 30秒

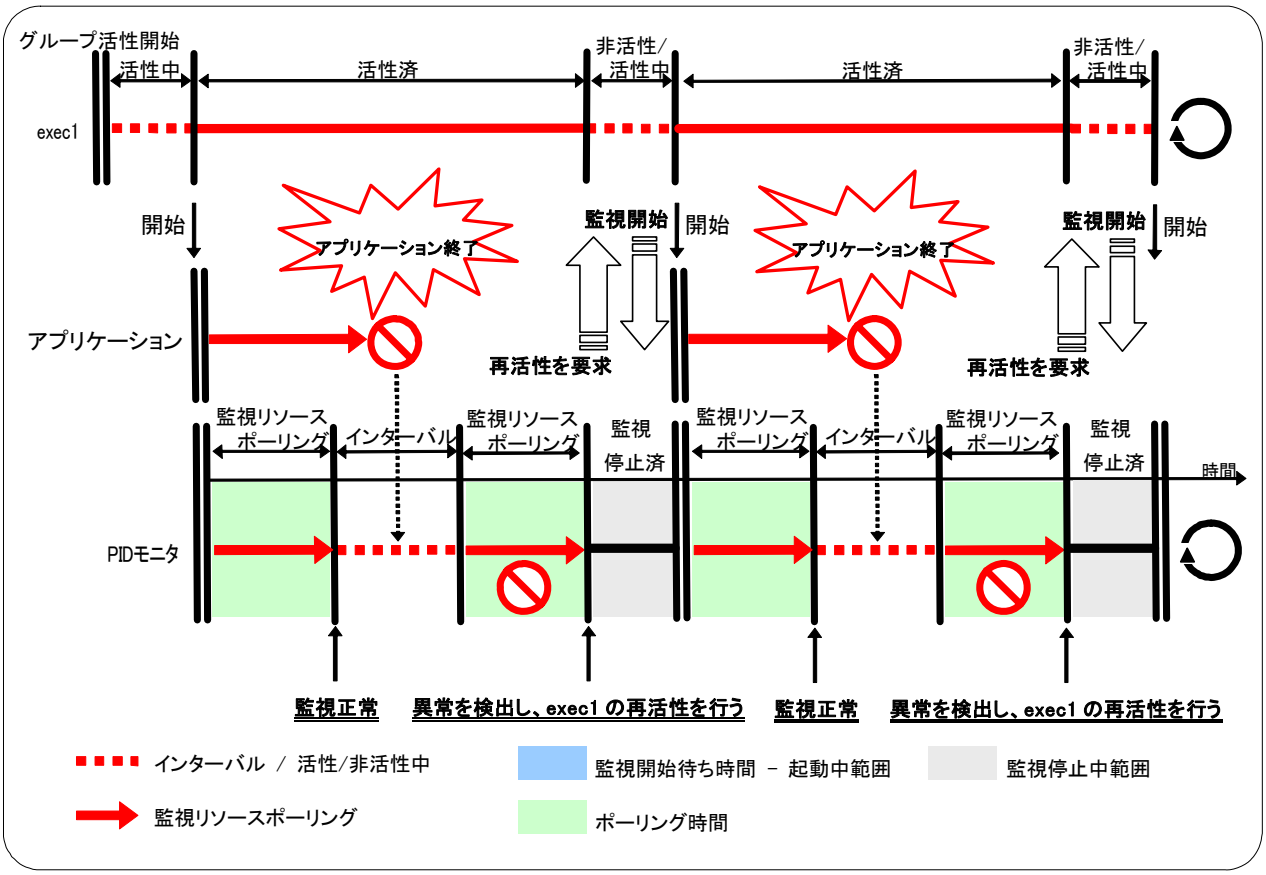


監視制御コマンドによる監視リソースの一時停止/再開を行った場合も、指定された監視開始待ち時間後に再開します。

監視開始待ち時間は、PIDモニタリソースが監視するexecリソースのようにアプリケーションの設定ミスなどにより監視開始後すぐに終了する可能性があり、再活性化では回復できない場合に使用します。

例えば、以下のように監視開始待ち時間を0に設定すると回復動作を無限に繰り返す場合があります。

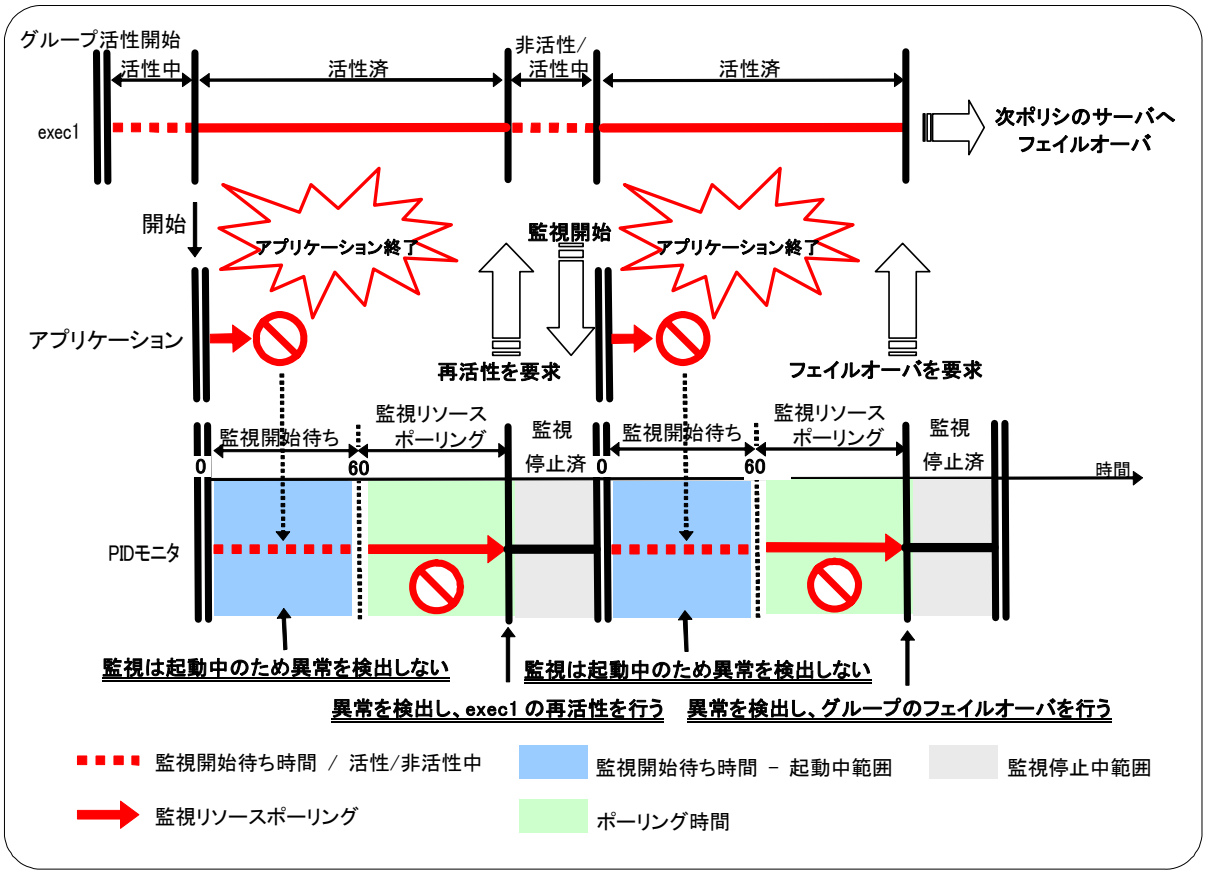
[PIDモニタリソース構成]	
<監視>	
インターバル	5秒
タイムアウト	60秒
リトライ回数	0回
監視開始待ち時間	0秒(規定値)
<異常検出>	
回復対象	exec1
再活性化しきい値	1回
フェイルオーバしきい値	1回
最終動作	グループ停止



この回復動作を無限に繰り返す原因は、初回の監視リソースポーリングが正常終了することにあります。監視リソースの回復動作の現在回数は、監視リソースが正常状態になればリセットされます。そのため、現在回数が常に0リセットされ再活性化の回復動作を無限に繰り返すことになります。

上記の現象は、監視開始待ち時間を設定することで回避できます。
監視開始待ち時間には、アプリケーションが起動後、終了しうる時間として60秒を設定しています。

```
[PIDモニタリソース構成]
    <監視>
      インターバル          5秒
      タイムアウト          60秒
      リトライ回数          0回
      監視開始待ち時間      60秒
    <異常検出>
      回復対象              exec1
      再活性化しきい値      1回
      フェイルオーバしきい値 1回
      最終動作              グループ停止
```



グループのフェイルオーバ先のサーバでもアプリケーションが異常終了した場合、最終動作としてグループ停止を行います。

2.1.8 再起動回数制限

モニタリソース異常検出時の最終動作として「クラスタデーモンの停止とOSシャットダウン」、または「クラスタデーモンの停止とOS再起動」を設定している場合に、モニタリソース異常の検出によるシャットダウン回数、または再起動回数を制限することができます。

再起動回数はサーバごとに記録されるため、最大再起動回数はサーバごとの再起動回数の上限になります。

また、グループ活性、非活性異常検出時の最終動作による再起動回数とモニタリソース異常の最終動作による再起動回数も別々に記録されます。

最大再起動回数をリセットする時間に0を設定した場合には、再起動回数はリセットされません。

以下の設定例で再起動回数制限の流れを説明します。

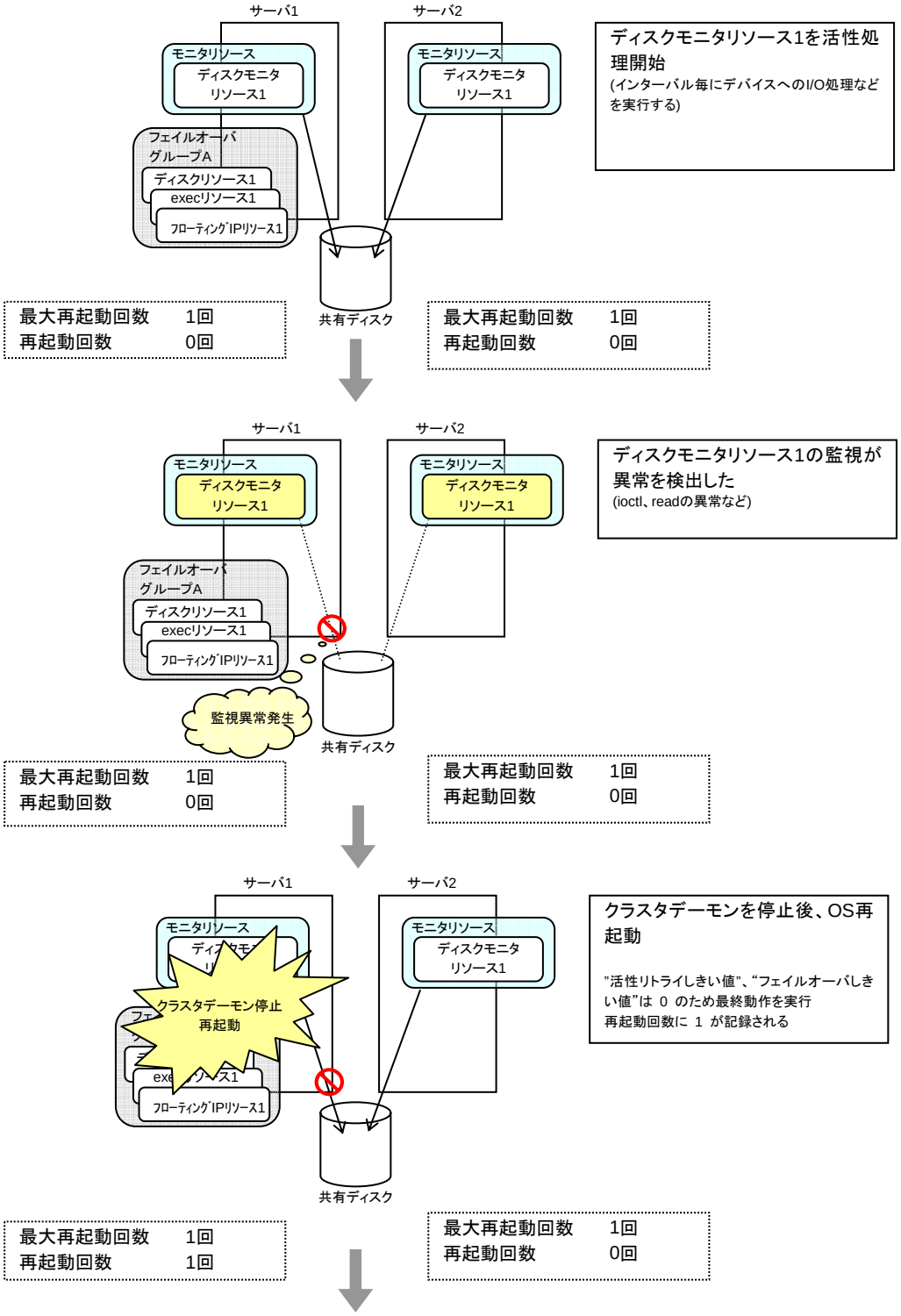
最大再起動回数が1回に設定されているため、一度だけ最終動作である「クラスタデーモンの停止とOS再起動」が実行されます。

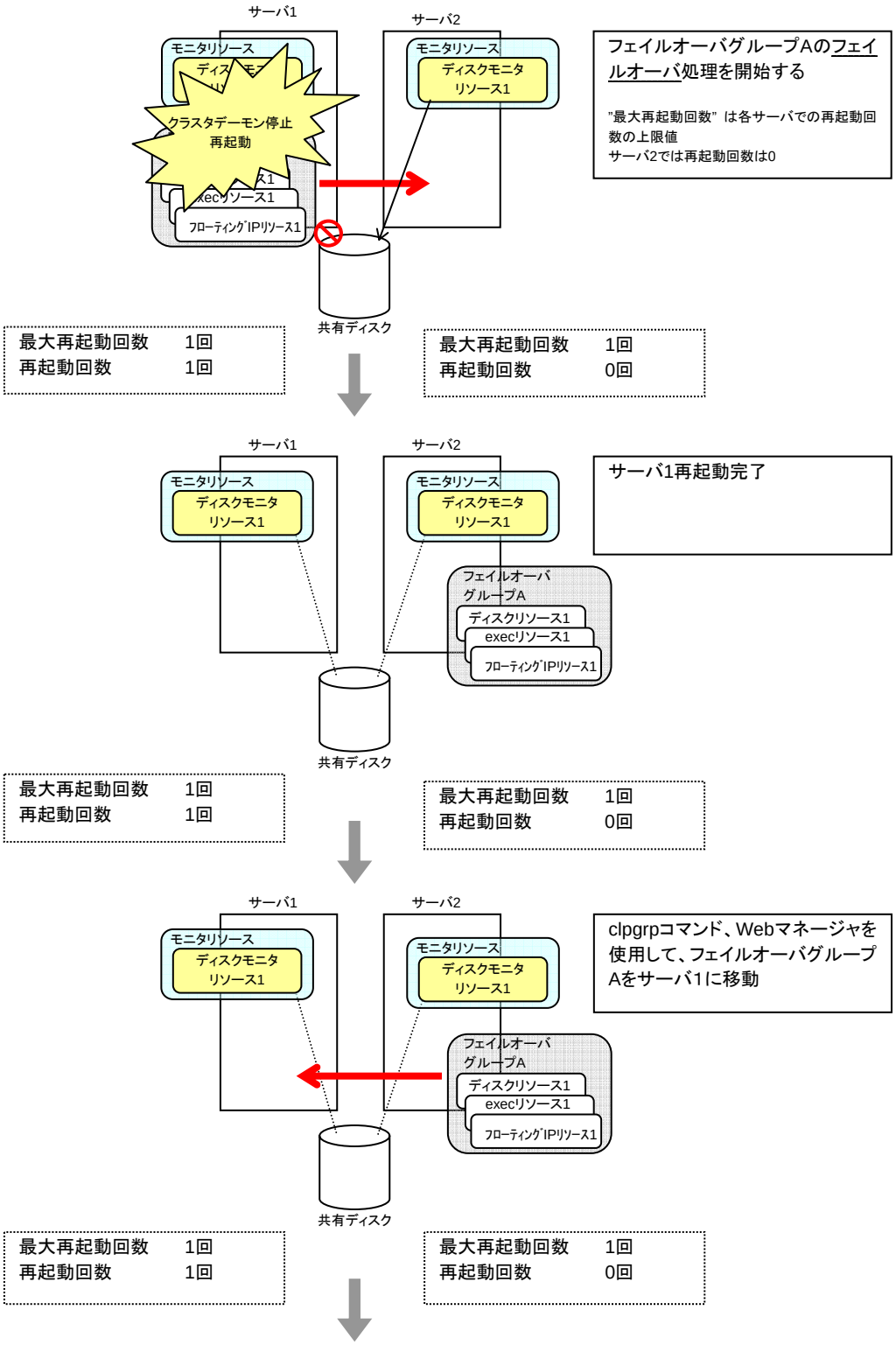
また、最大再起動回数をリセットする時間が10分に設定されているため、クラスタシャットダウン後再起動時にモニタリソースの正常状態が10分間継続した場合には、再起動回数はリセットされます。

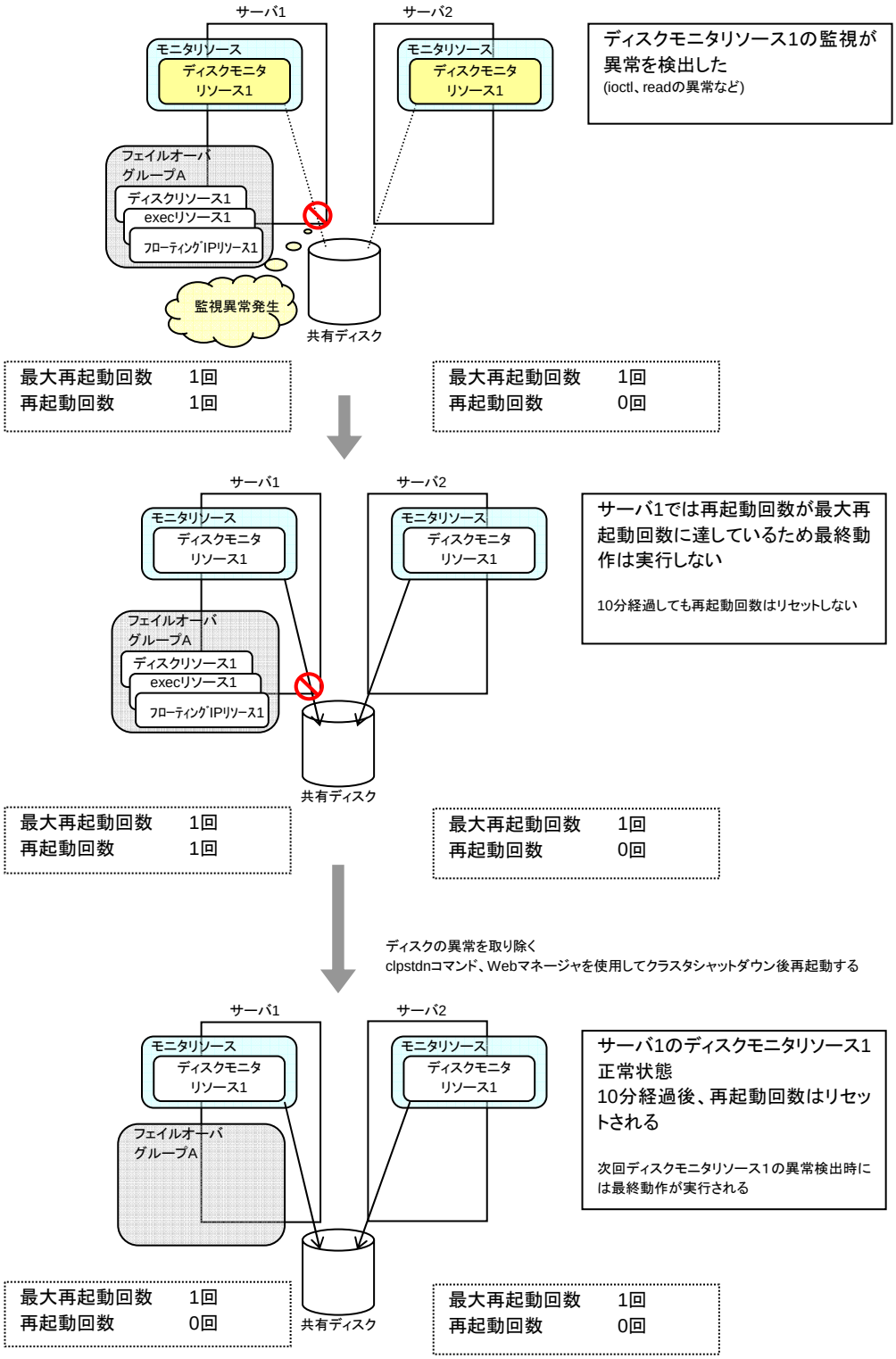
[設定例]

<監視>		
インターバル	60秒	
タイムアウト	120秒	
リトライ回数	3回	
<異常検出>		
回復対象	グループA	
再活性化しきい値	0回	
フェイルオーバしきい値	0回	
最終動作	クラスタデーモン停止とOS再起動	
<再起動回数制限>		
最大再起動回数		1回
最大再起動回数をリセットする時間		10分

を指定している場合の挙動の例


サーバ1
サーバ2
モニタリソース
ディスクモニタ
リソース1
フェイルオーバ
グループA
ディスクリソース1
execリソース1
ﾌﾛｰﾃｨﾝｸﾞIPリソース1
共有ディスク
ディスクモニタリソース1を活性処理開始
(インターバル毎にデバイスへのI/O処理などを実行する)
最大再起動回数 1回
再起動回数 0回
最大再起動回数 1回
再起動回数 0回
監視異常発生
ディスクモニタリソース1の監視が異常を検出した
(ioctl、readの異常など)
クラスタデーモン停止
再起動
クラスタデーモンを停止後、OS再起動
”活性リトライしきい値”、“フェイルオーバしきい値”は 0 のため最終動作を実行
再起動回数に 1 が記録される


サーバ1
サーバ2
モニタリソース
ディスクモニタ
リソース1
クラスタデーモン停止
再起動
execリソース1
ﾌﾛｰﾃｨﾝｸﾞIPﾘｿｰｽ1
共有ディスク
フェイルオーバグループAのフェイルオーバ処理を開始する
"最大再起動回数" は各サーバでの再起動回数の上限値
サーバ2では再起動回数は0
最大再起動回数 1回
再起動回数 1回
最大再起動回数 1回
再起動回数 0回
サーバ1再起動完了
フェイルオーバ
グループA
ディスクリソース1
execリソース1
ﾌﾛｰﾃｨﾝｸﾞIPﾘｿｰｽ1
clpgrpコマンド、Webマネージャを使用して、フェイルオーバグループAをサーバ1に移動


サーバ1
サーバ2
モニタリソース
ディスクモニタ
リソース1
フェイルオーバ
グループA
ディスクリソース1
execリソース1
ﾌﾛｰﾃｨﾝｸﾞIPﾘｿｰｽ1
監視異常発生
共有ディスク
ディスクモニタリソース1の監視が異常を検出した
(ioctl、readの異常など)
最大再起動回数 1回
再起動回数 1回
最大再起動回数 1回
再起動回数 0回
サーバ1では再起動回数が最大再起動回数に達しているため最終動作は実行しない
10分経過しても再起動回数はリセットしない
最大再起動回数 1回
再起動回数 1回
最大再起動回数 1回
再起動回数 0回
ディスクの異常を取り除く
clpstdnコマンド、Webマネージャを使用してクラスタシャットダウン後再起動する
サーバ1のディスクモニタリソース1
正常状態
10分経過後、再起動回数はリセットされる
次回ディスクモニタリソース1の異常検出時には最終動作が実行される
最大再起動回数 1回
再起動回数 0回
最大再起動回数 1回
再起動回数 0回

2.1.9 監視プライオリティ

OS高負荷時にモニタリソースへの監視を優先的に行うため、ユーザ空間モニタリソースを除く全てのモニタリソースでnice値を設定することができます。
nice値は 19(優先度低) ~ -20(優先度高) の範囲で指定することが可能です。

* nice値の優先度を上げることで監視タイムアウトの検出を抑制することが可能です。

2.2 ディスクモニタリソース

ディスクデバイスの監視を行います。

ディスクモニタリソース(TUR方式)が使用できない共有ディスクでは、RAWモニタでの監視を推奨します。

2.2.1 CLUSTERPROのバージョン

以下のCLUSTERPROのバージョンでサポートします。

CLUSTERPRO	Version
サーバ	SE3.0-1 以降、LE3.0-1 以降、XE3.0-1 以降、SX3.1-2 以降
トレッキングツール	対応するバージョンについては動作環境編 トレッキングツールの動作環境を参照してください。

2.2.2 監視方法

ディスクモニタリソースの監視方法は大きく分けてReadとTURがあります。

* TUR共通の注意事項
  - SCSIのTest Unit ReadyコマンドやSG_IOコマンドをサポートしていないディスク、ディスクインタフェース(HBA)では使用できません。
    ハードウェアがサポートしている場合でもドライバがサポートしていない場合があるのでドライバの仕様も合わせて確認してください。
  - S-ATAインタフェースのディスクの場合には、ディスクコントローラのタイプや使用するディストリビューションにより、OSにIDEインタフェースのディスク(hd)として認識される場合とSCSIインタフェースのディスク(sd)として認識される場合があります。
    IDEインタフェースとして認識される場合には、すべてのTUR方式は使用できません。
    SCSIインタフェースとして認識される場合には、TUR(legacy)が使用できます。
    TUR(generic)は使用できません。
  - Read方式に比べてOSやディスクへの負荷は小さくなります。
  - Test Unit Readyでは、実際のメディアへのI/Oエラーは検出できない場合があります。

TURの監視方法は、下記の3つが選択可能です。

* TUR
  + CLUSTERPRO Version 3.1-5 までの場合
    ioctrl(Test Unit Ready)を使って監視を行います。指定されたデバイスへSCSIコマンドとして定義されている Test Unit Ready(TUR)コマンドを発行してその結果で判断します。

  + CLUSTERPRO Version 3.1-5以降の場合
    指定されたデバイスへ以下の手順でioctlを発行して、その結果で判断します。
    ioctl(SG_GET_VERSION_NUM)コマンドを実行します。このioctlの戻り値とSGドライバのversionを見て判断します。
    ioctlコマンド成功かつSGドライバのversionが3.0以上ならSGドライバを使用したioctl TUR(SG_IO)を実行します。
    ioctlコマンド失敗またはSGドライバのversionが3.0未満ならSCSIコマンドとして定義されているioctl TURを実行します。

* TUR(legacy)
  + CLUSTERPRO Version 3.1-6 以降の場合
    ioctl(Test Unit Ready)を使って監視を行います。指定されたデバイスへSCSIコマンドとして定義されている Test Unit Ready(TUR)コマンドを発行してその結果で判断します。

* TUR(generic)
  + CLUSTERPRO Version 3.1-6 以降の場合
    ioctl TUR(SG_IO)を使って監視を行います。指定されたデバイスへSCSIコマンドとして定義されているioctl(SG_IO)コマンドを発行してその結果で判断します。
    SG_IO はSCSIディスクであってもOSやディストリビューションによって動作しないことがあります。SG_IOの動作可否に関しては「2.2.3」を参照してください。

READの監視方法は、下記になります。

* Dummy Read
  + 指定されたデバイス(ディスクデバイスまたはパーティションデバイス)上を指定されたサイズをreadしてその結果(readできたサイズ)で判断します。
  + 指定されたサイズがreadできたことを判断します。readしたデータの正当性は判断しません。
  + readするサイズを大きくするとOSやディスクへの負荷が大きくなります。
  + readするサイズについては次ページを留意して設定をしてください。
  + rawデバイスが使用できる場合には、RAWモニタリソースの使用を推奨します。

2.2.3 SG_IO の動作確認済みOSとkernelバージョン

TUR(generic)、Version 3.1-5以降のTURで使用するSG_IOの動作検証済みのディストリビューションとkernelバージョンは以下になります。

以下は、SCSIインタフェースのディスクの場合です。S-ATAインタフェースのディスクでOSにSCSIインタフェースのディスク(sd)として認識されてもSG_IOは使用できません。

－ia32版－

ディストリビューション	kernelバージョン [11]	SG_IO動作可否
Turbolinux 8 Server	2.4.18-23smp	×
Turbolinux Enterprise Server 8 powered by UnitedLinux	2.4.21-292-smp	×
Turbolinux 10 Server	2.6.8-5smp	○
Red Hat Enterprise Linux AS/ES 2.1 Update6	2.4.9-e.62smp	×
Red Hat Enterprise Linux AS/ES 3 Update7	2.4.21-40.ELsmp	×
Red Hat Enterprise Linux AS/ES 4 Update3	2.6.9-34.ELsmp	○
MIRACLE LINUX 2.1	2.4.9-e.25.90mlsmp	×
MIRACLE LINUX V3.0 SP2 R2	2.4.21-37.18AXsmp	×
MIRACLE LINUX V4.0	2.6.9-11.25AXsmp	○
Novell SUSE LINUX Enterprise Server 9 SP3	2.6.5-7.244-smp	○

[11] kernelバージョンについては 当社で確認をした環境のバージョンを記載しています。SG_IOが動作するkernelバージョンを限定するものではありません。

－ EM64T, x86_64版－

ディストリビューション	kernelバージョン [12]	SG_IO動作可否
Turbolinux 10 Server	2.6.12-1smp	○
Red Hat Enterprise Linux AS/ES 3 Update7	2.4.21-40.EL	×
Red Hat Enterprise Linux AS/ES 4 Update3	2.6.9-34.ELsmp	○
MIRACLE LINUX V3.0 SP2 R2	2.4.21-37.18AX	×
MIRACLE LINUX V4.0	2.6.9-11.25AXsmp	○
Novell SUSE LINUX Enterprise Server 9 SP3	2.6.5-7.244-smp	○

－ IA64版－

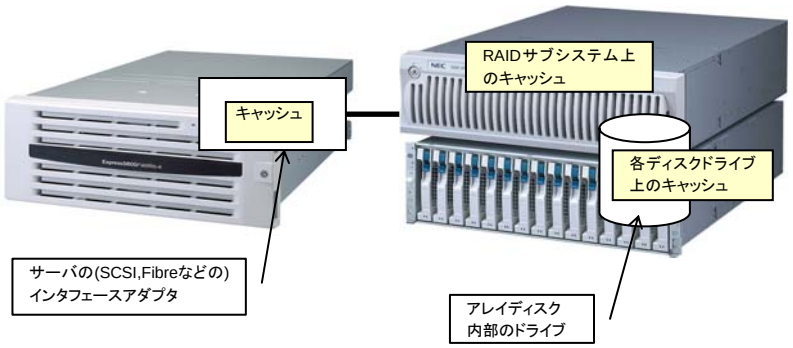
ディストリビューション	kernelバージョン [12]	SG_IO動作可否
Red Hat Enterprise Linux AS/ES 3 Update7	2.4.21-40.EL	×
Red Hat Enterprise Linux AS/ES 4 Update3	2.6.9-34.EL	○
Asianux 2.0 準拠 ディストリビューション	2.6.9-11.25AX	○
Novell SUSE LINUX Enterprise Server 9 SP3	2.6.5-7.244-default	○

[12] kernelバージョンについては 当社で確認をした環境のバージョンを記載しています。SG_IOが動作するkernelバージョンを限定するものではありません。

2.2.4 I/Oサイズ

監視方法でDummy Readを選択した場合のDummy Readを行うサイズを指定します。

- 使用する共有ディスクやインタフェースにより、様々なread用のキャッシュが実装されている場合があります。
- そのためDummy Readのサイズが小さい場合にはキャッシュにヒットしてしまいreadのエラーを検出できない場合があります。
- Dummy Readのサイズは共有ディスクの障害を発生させて障害の検出ができることを確認してください。



(注意) 上の図は共有ディスクの一般的な概念図を表したもので、必ずしもすべてのアレイ装置に当てはまるものではありません

2.3 RAWモニタリソース

RAWモニタとはディスクモニタリソース(Dummy Read方式)と似ていますが、Read対象にrawデバイスを使用します。OSがバッファリングをしないので比較的短時間に確実に異常を検出できます。

ディスクモニタリソース(TUR方式)が使用できない共有ディスクでは、RAWモニタでの監視を推奨します。

2.3.1 CLUSTERPROのバージョン

以下のCLUSTERPROのバージョンでサポートします。

CLUSTERPRO	Version
サーバ	SE3.0-4 以降、LE3.0-4 以降、XE3.1-4 以降、SX3.1-2 以降
トレッキングツール	対応するバージョンについては動作環境編 トレッキングツールの動作環境を参照してください。

2.3.2 RAWモニタリソースに関する注意事項

* 2.4系kernelの場合
  RAWモニタリソースを設定する場合、既にmountしているパーティションまたはmountする可能性のあるパーティションの監視はできません。whole device（ディスク全体を示すデバイス）をデバイス名に設定してください。

* 2.6系kernelの場合
  RAWモニタリソースを設定する場合、既にmountしているパーティションまたはmountする可能性のあるパーティションの監視はできません。また、既にmountしているパーティションまたはmountする可能性のあるパーティションのwhole device(ディスク全体を示すデバイス)をデバイス名に設定して監視することもできません。監視専用のパーティションを用意してRAWモニタリソースに設定してください。（監視用のパーティションサイズは、10M以上を割り当ててください）

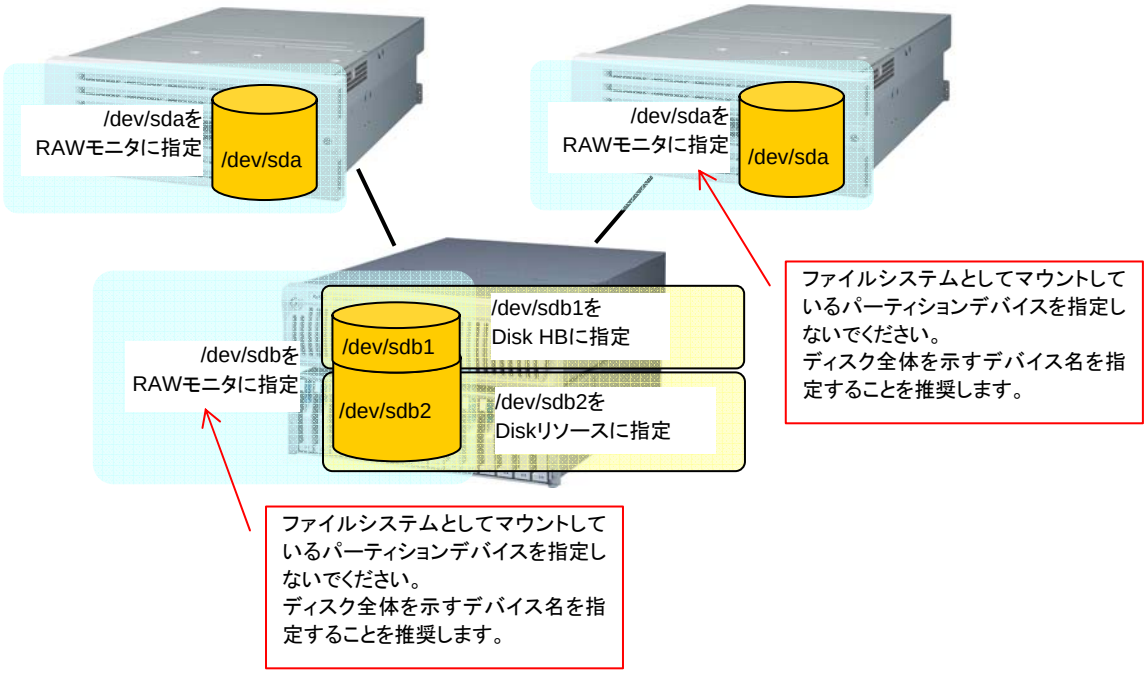
* 既にサーバプロパティの「ディスク I/F一覧」、「RAWリソース」または「VxVMボリュームリソース」に登録されているRAWデバイスは登録しないでください。VxVMボリュームリソースのRAWデバイスについては「1.7.6 CLUSTERPROで制御する際の注意」を参照してください。

* DISKハートビートが使用しているrawデバイスをRAWモニタリソースで監視する場合、トレッキングツールで「監視対象RAWデバイス名」にDISKハートビートで使用しているrawデバイスを指定し、「デバイス名」は入力しないでください。

2.3.3 2.4系kernelでのRAWモニタリソースの設定例

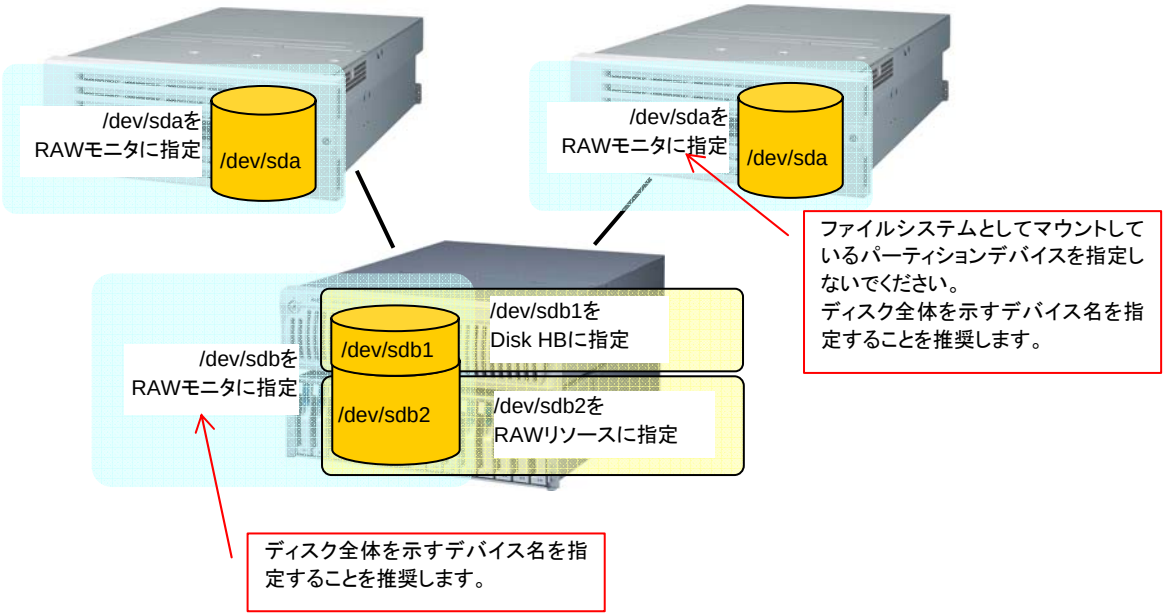
(1) Diskリソース、RAWモニタの設定例

+ ディスクリソース
+ RAWモニタ (両サーバの内蔵HDDを監視)
+ RAWモニタ (共有ディスクを監視)



(2)　RAWリソース、RAWモニタの設定例

+ RAWリソース
+ RAWモニタ (両サーバの内蔵HDDを監視)
+ RAWモニタ (共有ディスクを監視)



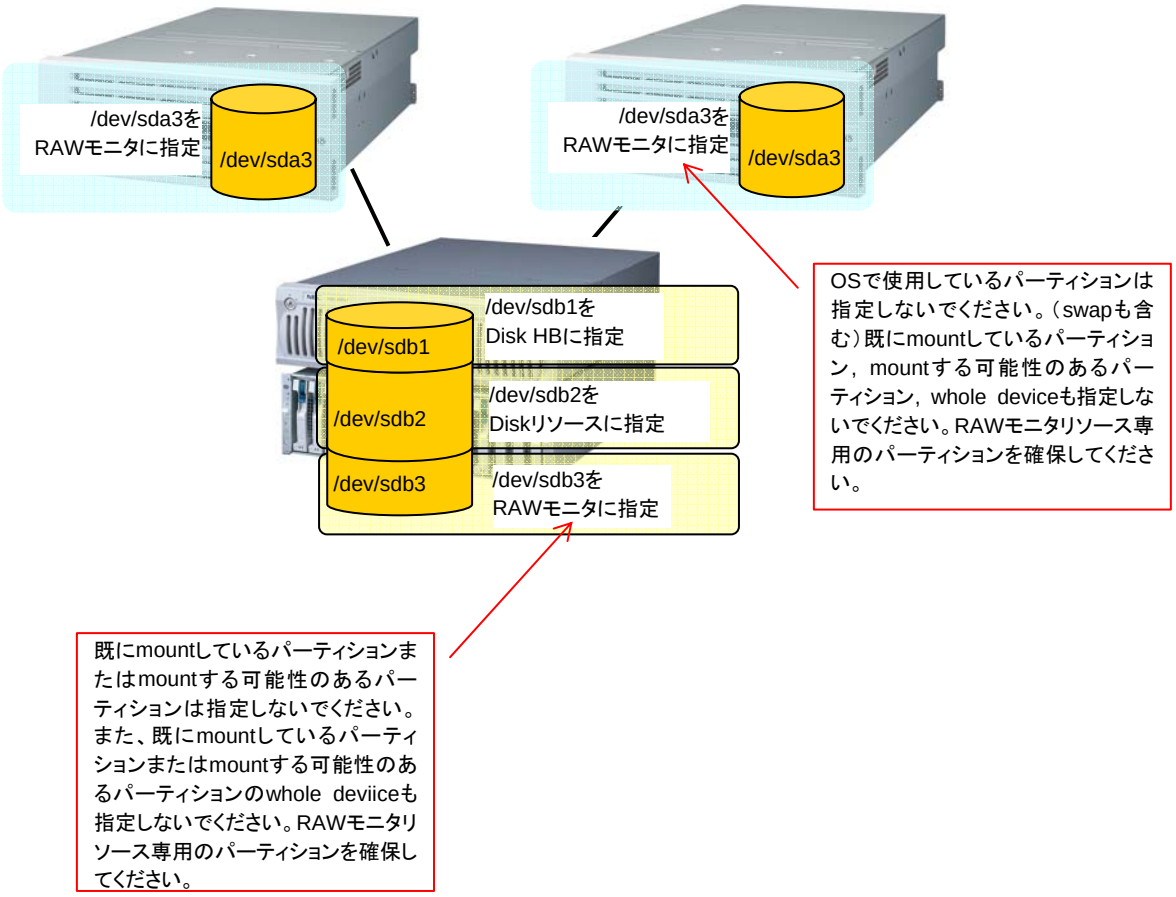
(3) VxVM rootdg を監視するRAWモニタの設定例

VxVM rootdg を監視するRAWモニタの設定例は、「1.7.7.2 CLUSTERPRO環境のサンプル」を参照してください。

2.3.4 2.6系kernelでのRAWモニタリソースの設定例

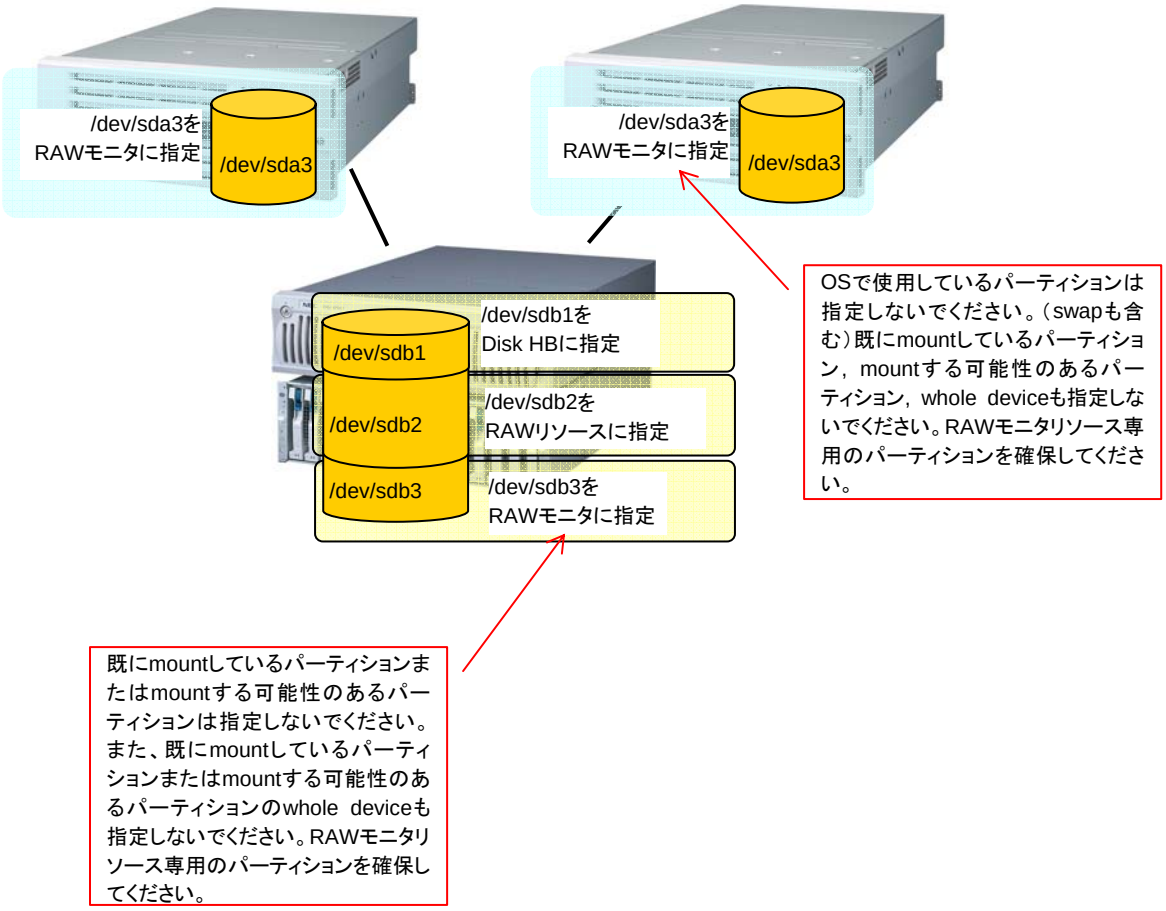
(1) Diskリソース、RAWモニタの設定例

+ ディスクリソース
+ RAWモニタ (両サーバの内蔵HDDを監視)
+ RAWモニタ (共有ディスクを監視)



(2)　RAWリソース、RAWモニタの設定例

+ RAWリソース
+ RAWモニタ (両サーバの内蔵HDDを監視)
+ RAWモニタ (共有ディスクを監視)



2.4 IPモニタリソース

pingコマンドを使用して、IPアドレスの監視を行います。

2.4.1 CLUSTERPROのバージョン

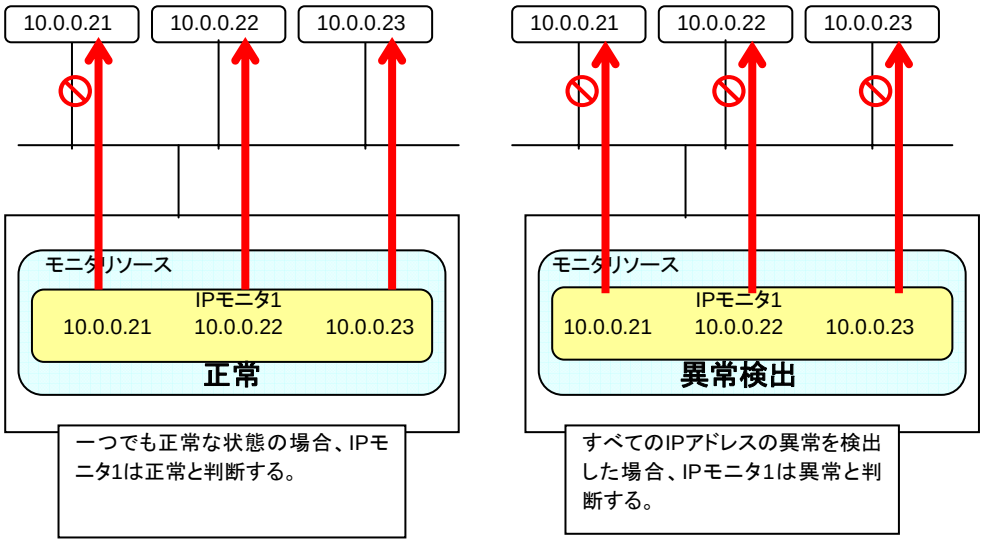
以下のCLUSTERPROのバージョンでサポートします。

CLUSTERPRO	Version
サーバ	SE3.0-1 以降、LE3.0-1 以降、XE3.0-1 以降、SX3.1-2 以降
トレッキングツール	対応するバージョンについては動作環境編 トレッキングツールの動作環境を参照してください。

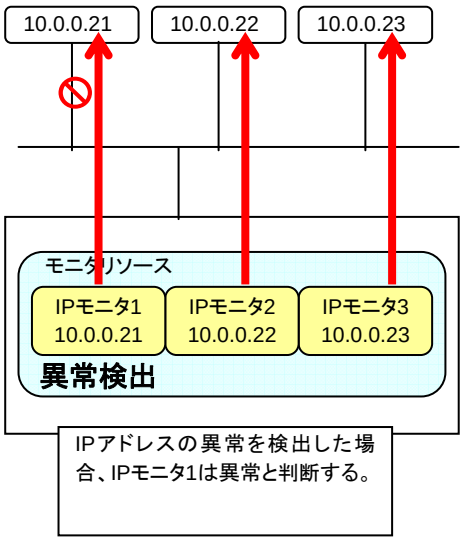
2.4.2 監視方法

指定したIPアドレスをpingコマンドで監視します。指定したIPアドレスすべての応答がない場合に異常と判断します。

* 複数のIPアドレスについてすべてのIPアドレスが異常時に異常と判断したい場合、1つのIPモニタリソースにすべてのIPアドレスを登録してください。



* 複数のIPアドレスについてどれか1つが異常時に異常と判断したい場合、個々のIPアドレスについて1つずつのIPモニタリソースを作成してください。



2.5 NIC Link Up/Downモニタリソース

2.5.1 動作確認情報

2.5.1.1 CLUSTERPROのバージョン

以下のCLUSTERPROのバージョンでサポートします。

CLUSTERPRO	Version
サーバ	SE3.1-1 以降、LE3.1-1 以降、XE3.1-4 以降、SX3.1-2 以降
トレッキングツール	対応するバージョンについては動作環境編 トレッキングツールの動作環境を参照してください。

2.5.1.2 ディストリビューション

以下のバージョンで動作確認しています。

Distribution	Arch	kernel
RedHat AS2.1	ia32	2.4.9-e.49/smp/enterprise
	ia64	2.4.18-e.31smp
RedHat AS3	ia32	2.4.21-9.0.1.ELsmp 2.4.21-15.EL/smp/hugemem 2.4.21-27.EL/smp/hugemem
	x86_64	2.4.21-20.EL 2.4.21-27.EL
	ia64	2.4.21-4.EL
TurboLinux ES8	ia32	2.4.21-231-smp 2.4.21-273/smp
MIRACLE LINUX V2.1	ia32	2.4.9-e.25.78ml/smp
MIRACLE LINUX V3.0	ia32	2.4.21-9.30AXsmp 2.4.21-9.34AXsmp 2.4.21-9.38AX/smp/hugemem
	x86_64	2.4.21-20.19AX
	ia64	2.4.21-15.8AX/smp/hugemem
Suse8	ia32	2.4.21-138-default/smp

2.5.1.3 ネットワークインタフェース

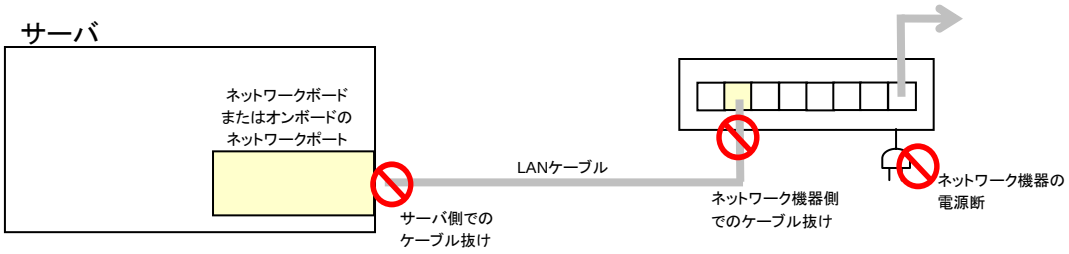
以下のネットワークインタフェースで動作確認しています。

Ethernet Controller(Chip)	Bus	Driver	Driver version
Intel 82557/8/9	PCI	e100	2.1.15 2.1.29-k2 2.3.13-k1-1 2.3.40 3.0.27-k2-NAPI
Intel 82544EI	PCI	e1000	5.1.11-k1
Intel 82546EB	PCI	e1000	5.3.19-k2
Broadcom BCM5700 - 5703	PCI	tg3	v2.2 v3.10 7.1.9e

2.5.2 注意事項

NICのドライバによっては、必要なioctl( )がサポートされていない場合があります。
その場合には このモニタリソースは使用できません。

2.5.3 監視の構成及び範囲



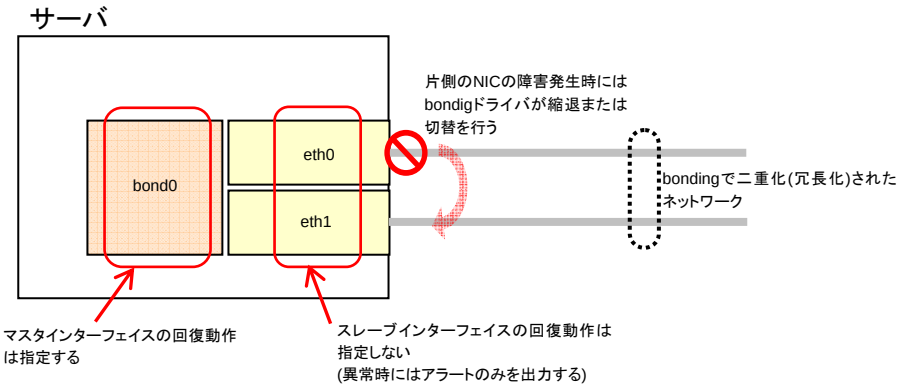
* NICのドライバへのioctl( )によりネットワーク(ケーブル)のリンク確立状態を検出します。
(IPモニタの場合は、指定されたIPアドレスへのpingの反応で判断をします。)

* インタコネクト(ミラーコネクト)専用のNICを監視することもできますが、2ノードでクロスケーブルで直結している場合には 片サーバダウン時に(リンクが確立しないため)残りのサーバ側でも 異常を検出します。
監視異常時の回復動作の設定に注意してください。
例えば、回復動作に"CLUSTERPROデーモン停止及びOS再起動"すると、残りのサーバ側は無限にOS再起動を繰り返すことになります。

また、ネットワークをbondig化している場合には、bondingによる可用性を活かしたまま下位のスレーブインターフェイス(eth0, eth1...)だけでなくマスタインターフェイス(bond0...)も監視することが可能です。その場合には、下記の設定を推奨します。

* スレーブインターフェイス
  + 異常検出時の回復動作:何もしない
    片方のネットワークケーブルのみ(eth0)の異常時にはCLUSTERPROは回復動作を実行せず、アラートのみ出力します。
    ネットワークの回復動作は、bondingが行います。

* マスタインターフェイス
  + 異常検出時の回復動作:フェイルオーバやシャットダウンなどを設定する
    全てのスレーブインターフェイスの異常時(マスタインターフェイスがダウン状態)にCLUSTERPROは、回復動作を実行します。



2.6 ミラーディスクコネクトモニタリソース

2.6.1 CLUSTERPROのバージョン

以下のCLUSTERPROのバージョンでサポートします。

CLUSTERPRO	Version
サーバ	LE3.0-1 以降
トレッキングツール	対応するバージョンについては動作環境編 トレッキングツールの動作環境を参照してください。

2.6.2 注意事項

ミラーリング用のネットワークを監視します。本リソースは1つ目のミラーディスクリソースを追加した時に自動的に登録されます。ミラーディスクリソースは同じミラーディスクコネクトI/Fを使用するので1つのみ自動登録されます。

2.7 ミラーディスクモニタリソース

2.7.1 CLUSTERPROのバージョン

以下のCLUSTERPROのバージョンでサポートします。

CLUSTERPRO	Version
サーバ	LE3.0-1 以降
トレッキングツール	対応するバージョンについては動作環境編 トレッキングツールの動作環境を参照してください。

2.7.2 注意事項

本リソースはミラーディスクリソースを追加した時に自動的に登録されます。各ミラーディスクリソースに対応するミラーディスクモニタリソースが自動登録されます。

2.8 PIDモニタリソース

2.8.1 CLUSTERPROのバージョン

以下のCLUSTERPROのバージョンでサポートします。

CLUSTERPRO	Version
サーバ	SE3.0-1 以降、LE3.0-1 以降、XE3.0-1, SX3.1-2 以降
トレッキングツール	対応するバージョンについては動作環境編 トレッキングツールの動作環境を参照してください。

2.8.2 注意事項

活性に成功したexecリソースを監視します。execリソースの起動時の設定が[非同期]の場合のみ監視できます。

2.9 ユーザ空間モニタリソース

2.9.1 CLUSTERPROのバージョン

以下のCLUSTERPROのバージョンでサポートします。

CLUSTERPRO	監視方法	Version
サーバ	softdog	SE3.0-1 以降、LE3.0-1 以降、XE3.0-1 以降、SX3.1-2 以降
サーバ	ipmi	SE3.1-4 以降、LE3.1-4 以降。XE3.1-4 以降、SX3.1-4 以降
サーバ	keepalive	SE3.1-6 以降、LE3.1-6 以降, XE3.1-6 以降、SX3.1-6 以降
サーバ	none	SE3.1-4 以降、LE3.1-4 以降、XE3.1-4 以降、SX3.1-4 以降
トレッキングツール	-	対応するバージョンについては動作環境編 トレッキングツールの動作環境を参照してください。

* CLUSTERPRO Version 3.1-5 までの場合
  ユーザ空間監視を複数設定することができません。

* CLUSTERPRO Version 3.1-6 以降の場合
  監視方法の異なるユーザ空間監視を複数設定することが可能です。同じ監視方法のユーザ空間監視は複数設定することができません。

2.9.2 依存するドライバ

(1) 監視方式 softdog

- softdog

* 監視方法がsoftdogの場合、このドライバが必要です。
* SE、LEの場合、ローダブルモジュール構成にしてください。スタティックドライバでは動作しません。
* softdogドライバが使用できない場合、監視を開始することは出来ません。

(2) 監視方式 keep alive

- clpka
- clpkhb

* 監視方法がkeepaliveの場合、CLUSETRPOのclpkhbドライバ，clpkaドライバが必要です。
* 監視方法をkeepaliveに設定する場合、カーネルモードLANハートビートを設定することを推奨します。カーネルモードLANハートビートを使用するにはclpkhbドライバが必要です。
* clpkaドライバとclpkhbドライバはCLUSTERPROが提供するドライバです。サポート範囲については「動作環境編」を参照してください。
* clpkhbドライバ, clpkaドライバが使用できない場合、監視を開始することは出来ません。

2.9.3 依存するrpm

- ipmiutil

* 監視方法がipmiの場合、このrpmをインストールしておく必要があります。
* このrpmがインストールされていない場合、監視を開始することは出来ません。

2.9.4 監視方法

ユーザ空間モニタリソースの監視方法は以下のとおりです。

(1) 監視方法 softdog

監視方法がsoftdogの場合、OSのsoftdogドライバを使用します。

(2) 監視方法 ipmi

監視方法がipmiの場合、ipmiutilを使用します。
ipmiutilがインストールされていない場合、インストールする必要があります。

(3) 監視方法 keepalive

監視方法がkeepaliveの場合、clpkhbドライバとclpkaドライバを使用します。

clpkhbドライバ，clpkaドライバが動作するディストリビューション，kernelバージョンについては必ず「動作環境編」で確認してください。
ディストリビューターがリリースするセキュリティパッチを既に運用中のクラスタへ適用する場合(kernelバージョンが変わる場合)にも確認してください。

(4) 監視方法 none

監視方法 noneは、評価用の設定です。ユーザ空間監視のoptionだけを実行します。本番環境では設定しないでください。

2.9.5 監視の拡張設定

ユーザ空間監視を拡張させる設定として、ダミーファイルのオープン/クローズ、 ダミーファイルへの書き込み、 ダミースレットの作成があります。各設定に失敗するとタイマの更新を行ないません。設定したタイムアウト値またはハートビートタイムアウト時間内に各設定が失敗し続けるとOSをリセットします。

(1) ダミーファイルのオープン/クローズ

監視間隔毎にダミーファイルの作成、ダミーファイルのopen、ダミーファイルのclose、ダミーファイルの削除を繰り返します。

* この拡張機能を設定している場合、ディスクの空き容量がなくなるとダミーファイルのopenに失敗してタイマの更新が行なわれず、OSをリセットします。

(2) ダミーファイルへの書き込み

監視間隔毎にダミーファイルに設定したサイズを書き込みます。

* この拡張機能は、ダミーファイルのオープン/クローズが設定されていないと設定できません。

(3) ダミースレットの作成

監視間隔毎にダミースレットを作成します。

2.9.6 監視ロジック

監視方法の違いによる処理内容、特徴は以下の通りです。
シャットダウンストール監視では各処理概要のうち1のみの挙動になります。

2.9.6.1 IPMI

(1) 処理概要
以下の2～7 の処理を繰り返します。
1. IPMIタイマセット
2. ダミーファイルのopen
3. ダミーファイルwrite()
4. ダミーファイルfdatasync()
5. ダミーファイルのclose
6. ダミースレッド作成
7. IPMIタイマ更新

* 処理概要 2～6は監視の拡張設定の処理です。各設定を行なっていないと処理を行ないません。

(2) タイムアウトしない(上記2～7が問題なく処理される)場合の挙動
リセットなどのリカバリ処理は実行されません

(3) タイムアウトした(上記2～7の何れかが停止または遅延した)場合の挙動
BMC(サーバ本体のマネージメント機能)によりリセットを発生させます

(4) メリット
= BMC(サーバ本体のマネージメント機能)を使用するためkernel空間の障害を受けにくく、リセットができる確率が高くなります。

(5) デメリット
= H/Wに依存しているためIPMIをサポートしていないサーバ、ipmiutilが動作しないサーバでは使用できません。
= ESMPRO/ServerAgentを使用しているサーバでは使用できません。
= その他サーバベンダが提供するサーバ監視ソフトウェアと共存できない可能性があります。
= 一部のアーキテクチャではipmiutilが提供されていません。

2.9.6.2 softdog

(1) 処理概要
以下の2～7の処理を繰り返します。
1. softdogセット
2. ダミーファイルのopen
3. ダミーファイルwrite()
4. ダミーファイルfdatasync()
5. ダミーファイルのclose
6. ダミースレッド作成
7. softdogタイマ更新

* 処理概要 2～6は監視の拡張設定の処理です。各設定を行なっていないと処理を行ないません。

(2) タイムアウトしない(上記2～7が問題なく処理される)場合の挙動
リセットなどのリカバリ処理は実行されません

(3) タイムアウトした(上記2～7の何れかが停止または遅延した)場合の挙動
softdog.o(softdog.ko)によりリセット(machine_restart)を発生させます

(4) メリット
= H/Wに依存しないためsoftdog kernelモジュールがあれば使用できます。
(一部のディストリビューションでは デフォルトでsoftdogが用意されていないものもがありますので 設定する前にsoftdogの有無を確認してください)

(5) デメリット
= softdogがkernel空間のタイマロジックに依存しているためkernel空間に障害が発生した場合にリセットされない場合があります。

2.9.6.3 監視方法 keepalive

(1) 処理概要
以下の2～7の処理を繰り返します。
1. keepaliveタイマセット
2. ダミーファイルのopen
3. ダミーファイルwrite()
4. ダミーファイルfdatasync()
5. ダミーファイルのclose
6. ダミースレッド作成
7. keepaliveタイマ更新

* 処理概要 2～6は監視の拡張設定の処理です。各設定を行なっていないと処理を行ないません。

(2) タイムアウトしない(上記2～7が問題なく処理される)場合の挙動
リセットなどのリカバリ処理は実行されません

(3) タイムアウトした(上記2～7の何れかが停止または遅延した)場合の挙動
= clpkhb.o(clpkhb.ko)を経由して他のサーバへ「自サーバのリセット」をアナウンスします
= clpka.o(clpka.ko)によりリセット(machine_restart)を発生させます

(4) メリット
= clpkhbにより他サーバへ「自サーバのリセット」をアナウンスすることで、他のサーバ上で記録(ログ)を残すことができます。

(5) デメリット
= 動作できる(ドライバを提供している)ディストリビューション,アーキテクチャ,カーネルバージョンが制限されます。
= clpkaがkernel空間のタイマロジックに依存しているためkernel空間に障害が発生した場合にリセットされない場合があります。

2.9.7 ipmi動作可否の確認方法

サーバ本体のipmiutilの対応状況を確認する方法は、以下の手順で簡易確認することができます。

(1) ダウンロードしたipmiutilのrpmパッケージをインストールする。[13]
(2) /usr/sbin/wdtを実行する。
(3) 実行結果を確認する。

A. 以下のように表示される場合
(以下は表示例です。H/Wにより表示される値が異なる場合があります。)

```
wdt ver 1.8
-- BMC version 0.8, IPMI version 1.5
wdt data: 01 01 01 00 31 17 31 17
Watchdog timer is stopped for use with BIOS FRB2. Logging
                pretimeout is 1 seconds, pre-action is None
                timeout is 593 seconds, counter is 593 seconds
                action is Hard Reset
```

→ipmiutilは使用できます。監視方法にipmiを選択することが可能です。

B. 以下のように表示される場合

```
wdt version 1.8
ipmignu_cmd timeout, after session activated
```

→ipmiutilは使用できません。監視方法にIPMIを選択しないでください。

[13]一部のディストリビューションではディストリビューションと共にインストールされています。その場合にはipmi-util rpmパッケージのインストールは不要です。

2.9.8 ipmiコマンド

使用しているipmiutilのコマンド

ユーザモードストール監視、シャットダウンストール監視ではipmiutilのうち以下のコマンド、オプションを使用します。

コマンド	オプション	使用するタイミング	
		ユーザモードストール監視	シャットダウンストール監視
wdt	-e (タイマ開始)	開始時	監視開始時
	-d (タイマ停止)	終了時	終了時（SIGTERM有効）
	-r (タイマ更新)	開始時/監視間隔毎	監視開始時
	-t (タイムアウト値セット)	開始時/監視間隔変更時	監視開始時

2.9.9 注意事項

2.9.9.1 全監視方法での共通の注意事項

* クラスタを追加すると自動で作成されます。

* OSのsoftdogドライバが存在しないまたはCLUSTERPROのclpkhbドライバ、clpkaドライバが存在しない、ipmiutilのrpmがインストールされていないなどの理由によりユーザ空間モニタリソースの活性に失敗した場合、Webマネージャのアラートビューに”Monitor userw failed.” というメッセージが表示されます。Webマネージャのツリービュー、clpstatコマンドでの表示ではリソースステータスは「正常」が表示され、各サーバのステータスは「停止済」が表示されます。

* CLUSTERPROのバージョンが3.1-8以降の場合、設定されているタイムアウト値の80%を超える場合、syslogとアラートに警告を出力します。
遅延警告割合を変更することはできません。

2.9.9.2 softdogによる監視の注意事項

* 監視方法にsoftdogを設定する場合、OS標準添付のheatbeatを動作しない設定にしてください。

2.9.9.3 ipmiによる監視の注意事項

* userwのIPMIによる監視はCLUSTERPROとipmiutilが連携をして実現しています。
* CLUSTERPROにipmiutilは添付しておりません。ユーザ様ご自身で別途ipmiutil rpmパッケージをダウンロードしてインストールを行ってください。
* IPMIを使用する場合、syslogに下記のkernelモジュール警告ログが多数出力されます。

```
modprobe: modprobe: Can`t locate module char-major-10-173
```

このログ出力を回避したい場合は、/dev/ipmikcsをrenameしてください。

* ipmiutilは 2005年3月15日現在以下のURLよりダウンロードが可能です。
  http://ipmiutil.sourceforge.net/
  (2005年8月現在、このURLでは x86_64のipmiutilは提供されていません)

* 監視方法にipmiを設定する場合、OS標準添付のbmc-watchdogを動作しない設定にしてください。

* ipmiutilに関し以下の事項は弊社は対応いたしません。ユーザ様の判断、責任にてご使用ください。
  = ipmiutil自体に関するお問い合わせ
  = ipmiutilの動作保証
  = ipmiutilの不具合対応、不具合が原因の障害
  = 各サーバのipmiutilの対応状況のお問い合わせ

* ご使用予定のサーバ(ハードウェア)のipmiutil対応可否についてはユーザ様にて事前に確認ください。

* ハードウェアとしてIPMI規格に準拠している場合でも実際にはipmiutilが動作しない場合がありますので、ご注意ください。

以下の組み合わせにて動作検証を行いました。

ディストリビューション	kernel verison	ipmiutilのバージョン	サーバ
Red Hat Enterprise Linux AS 3	2.4.21-27.EL.smp	ipmiutil-1.5-6-1.i386.rpm	Express5800/120Lf
Red Hat Advanced Server 2.1	2.4.9-e.49smp	ipmiutil-1.5-5-1.i386.rpm	Express5800/420Ma
Red Hat Enterprise Linux AS 3	2.4.21-20.EL.smp	ipmiutil-1.5-5-1386.rpm	Express5800/120Rg-2

ESMPRO/ServerAgentなどサーバベンダが提供するサーバ監視ソフトウェアを使用する場合には 監視方法にIPMIを選択しないでください。
これらのサーバ監視ソフトウェアとipmiutilは共にサーバ上のBMC(Baseboard Management Controller)をするため競合が発生して正しく監視が行うことができなくなります。

2.9.9.4 keepalive による監視の注意事項

* 他サーバへのresetの通知はカーネルモードLANハートビートリソースが設定されている場合に限ります。この場合、下記のログがsyslogに出力されます。

  kernel: clpkhb: Keepalive: <server priority: %d> <reason: %s> <process name: %s>system reboot.

2.10 VxVMデーモンモニタリソース

2.10.1 CLUSTERPROのバージョン

以下のCLUSTERPROのバージョンでサポートします。

CLUSTERPRO	Version
サーバ	SE3.0-4 以降
トレッキングツール	対応するバージョンについては動作環境編 トレッキングツールの動作環境を参照してください。

2.10.2 注意事項

VxVMデーモンモニタリソースについて、詳細設定はありません。VxVMの環境で設定してください。

2.11 VxVMボリュームモニタリソース

2.11.1 CLUSTERPROのバージョン

以下のCLUSTERPROのバージョンでサポートします。

CLUSTERPRO	Version
サーバ	SE3.0-4 以降
トレッキングツール	対応するバージョンについては動作環境編 トレッキングツールの動作環境を参照してください。

2.11.2 注意事項

ボリュームRAWデバイスのファイルシステムがvxfsではない場合、VxVMボリュームモニタリソースで監視できません。VxVMの環境で設定してください。

2.12 マルチターゲットモニタリソース

複数の監視リソースの監視を行います。

2.12.1 CLUSTERPROのバージョン

以下のCLUSTERPROのバージョンでサポートします。

CLUSTERPRO	Version
サーバ	SE3.1-6 以降、LE3.1-6 以降、SX3.1-6 以降、XE3.1-6 以降
トレッキングツール	対応するバージョンについては動作環境編 トレッキングツールの動作環境を参照してください。

2.12.2 マルチターゲットモニタリソースのステータス

マルチターゲットモニタリソースのステータスは登録されている監視リソースのステータスによって判断します。

2つの監視リソースが登録されている場合、マルチターゲットモニタリソースのステータスは以下のようになります。

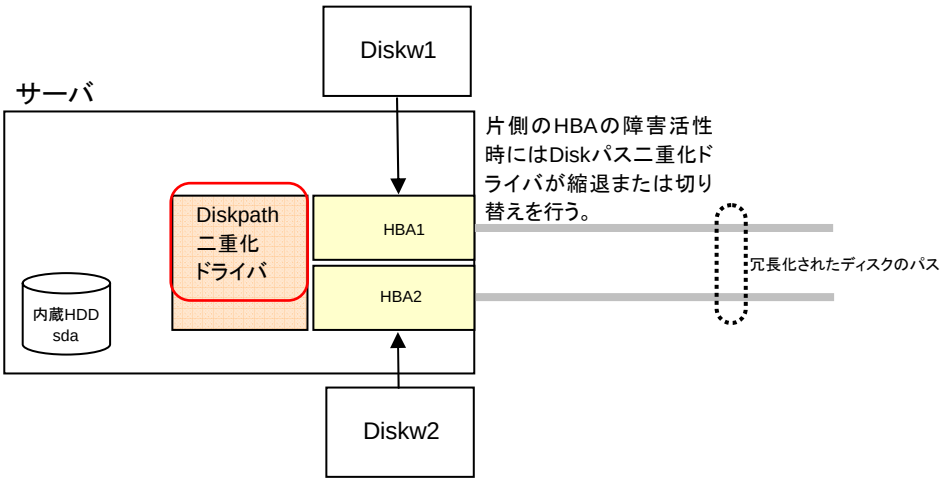
マルチターゲットモニタリソース ステータス		監視リソース1 ステータス		
		正常 (normal)	異常 (error)	停止済み (offline)
監視リソース2 ステータス	正常 (normal)	正常 (normal)	正常 (normal)	正常 (normal)
	異常 (error)	正常 (normal)	異常 (error)	異常 (error)
	停止済み (offline)	正常 (normal)	異常 (error)	正常 (normal)

* マルチターゲットモニタリソースは登録されている監視リソースのステータスを監視しています。
ステータスが正常(normal)な監視リソースがない場合、マルチターゲットモニタリソースは異常(error)を検出します。
登録されている全ての監視リソースのステータスが停止済み(offline)の場合、マルチターゲットモニタリソースのステータスは正常(normal)となります。

* 登録されている監視リソースのステータスが異常(error)となっても、その監視リソースの異常時アクションは実行されません。
マルチターゲットモニタリソースが異常(error)になった場合のみ、マルチターゲットモニタリソースの異常時アクションが実行されます。

2.12.3　設定例

* Diskパス二重化ドライバの使用例



+ マルチターゲットモニタリソース（mtw1）
  登録されている監視リソース
  - Diskw1
  - Diskw2

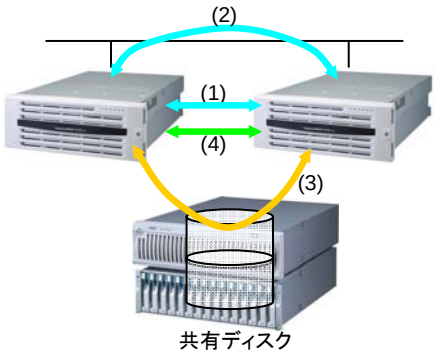
マルチターゲットモニタリソースに登録されている監視リソース
+ ディスクモニタリソース（Diskw1）
  sdbを監視
+ ディスクモニタリソース（Diskw2）
  sdcを監視

* 上記の設定の場合、マルチターゲットモニタリソースの監視リソースに登録されているDiskw1とDiskw2のどちらかが異常を検出しても異常となった監視リソースの異常時アクションを行いません。

* Diskw1とDiskw2が共に異常となった場合、2つの監視リソースのステータスが異常(error)と停止済み(offline)になった場合、マルチターゲットモニタリソースに設定された異常時アクションを実行します。

3 ハートビートリソース

クラスタ内のサーバは他のサーバの死活監視をおこないます。サーバ間の死活監視はハートビートリソースを使用します。ハートビートデバイスには以下の種類があります。



(1)インタコネクト専用LANハートビート
(1)インタコネクト専用LANハートビート
（カーネルモード）
(2)パブリックLANハートビート
(2)パブリックLANハートビート
（カーネルモード）
(3)ディスクハートビート
(4)COMハートビート

モニタリソース名	略称	機能概要
LANハートビートリソース (1)(2)	lanhb	LANを使用してサーバの死活監視をおこないます クラスタ内の通信でも使用します
カーネルモードLANハートビートリソース (1)(2)	lankhb	カーネルモードのモジュールがLANを使用してサーバの死活監視をおこないます クラスタ内の通信でも使用します
ディスクハートビートリソース (3)	diskhb	共有ディスク上の専用パーティションを使用してサーバの死活監視をおこないます
COMハートビートリソース (4)	comhb	2台のサーバ間をCOMケーブルで接続してサーバの死活監視をおこないます

* LANハートビートは最低一つ設定する必要があります。二つ以上の設定を推奨します。LANハートビートリソースとカーネルモードLANハートビートを同時に設定することを推奨します。

* ディスクハートビート及びCOMハートビートのI/Fは、以下の基準で設定してください。

+ SE、SXの場合

共有ディスクを使用するとき	[サーバ数 2台まで] 基本的にCOM I/F方式とディスクI/F方式 [サーバ数 3台以上] ディスクI/F方式
共有ディスクを使用しないとき	[サーバ数 2台まで] COM I/F方式

+ XEの場合
  ディスクI/F方式
+ LEの場合
  COM I/F方式

3.1 LANハートビートリソース

3.1.1 動作確認情報

3.1.1.1 CLUSTERPROのバージョン

以下のCLUSTERPROのバージョンでサポートします。

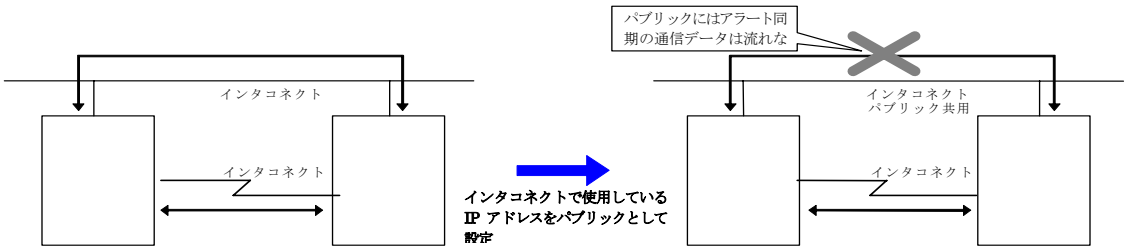
CLUSTERPRO	Version
サーバ	SE、SX、XE、LEの全てのバージョン
トレッキングツール	全てのバージョン

3.1.2 注意事項

* LANハートビートリソースは最低一つ設定する必要があります。インタコネクト専用のLANハートビートリソースと、インタコネクトとパブリック共用のLANハートビートリソースの二つ以上の設定を推奨します。

* パブリックLAN I/Fに登録していないインタコネクトI/Fにはアラート同期の通信データが流れます。ネットワークトラフィックを考慮して設定してください。

* インタコネクトLAN I/Fとパブリック LAN I/Fは同じIPアドレスを設定することができますが、その場合はアラート同期の通信データが流れなくなります。



3.2 カーネルモードLANハートビートリソース

3.2.1 動作確認情報

3.2.1.1 CLUSTERPROのバージョン

ディストリビューション、カーネルバージョンに依存するため、設定前に必ず「動作環境編」を参照してください。

以下のCLUSTERPROのバージョンでサポートします。

CLUSTERPRO	Version
サーバ	SE3.1-6 以降、LE3.1-6 以降
トレッキングツール	3.1-6 以降

3.2.2 カーネルモードLANハートビートリソース

LANハートビートと同様の機能をカーネルモードのドライバモジュールを使用して実現します。以下のような特徴があります。

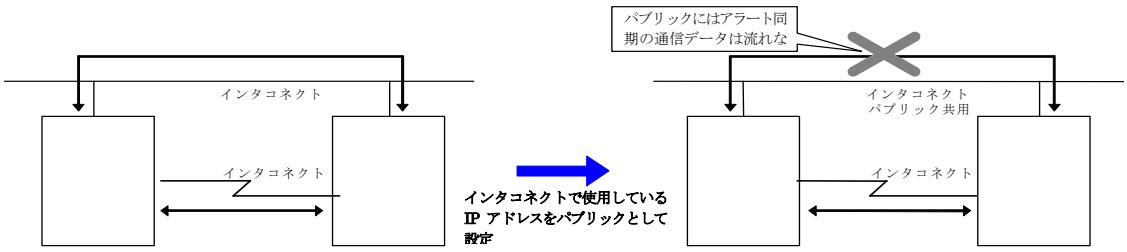
+ カーネルモードのドライバを使用するため、負荷に影響されにくくインタコネクト断線の誤認が少なくなります。
+ ユーザモードストール監視のkeepalive方式と同時に設定することで、ユーザモードストール検出時のリセットを他のサーバで記録することが可能になります。

3.2.3 注意事項

* インタコネクト専用のカーネルモードLANハートビートリソースと、インタコネクトとパブリック共用のカーネルモードLANハートビートリソースの二つ以上の設定を推奨します。

* パブリックLAN I/Fに登録していないインタコネクトI/Fにはアラート同期の通信データが流れます。ネットワークトラフィックを考慮して設定してください。

* インタコネクトLAN I/Fとパブリック LAN I/Fは同じIPアドレスを設定することができますが、その場合はアラート同期の通信データが流れなくなります。



3.3 ディスクハートビートリソース

3.3.1 動作確認情報

3.3.1.1 CLUSTERPROのバージョン

以下のCLUSTERPROのバージョンでサポートします。

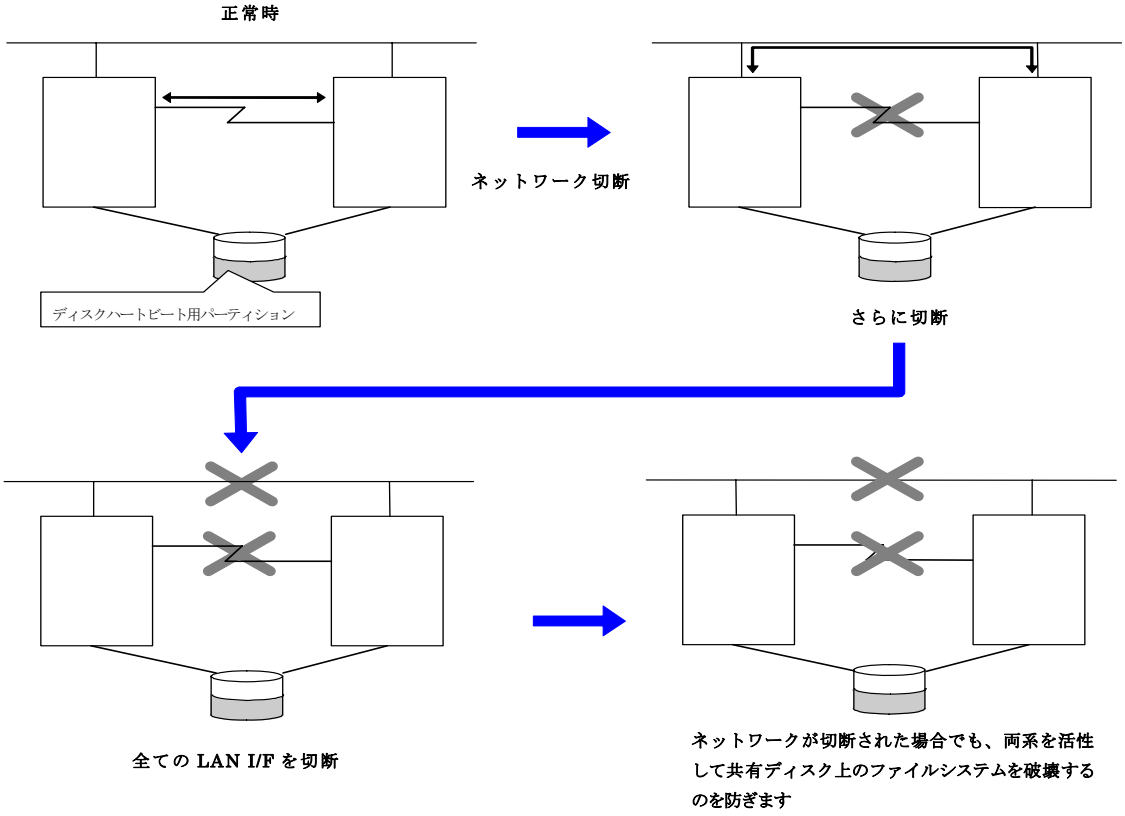
CLUSTERPRO	Version
サーバ	SE、SX、XEの全てのバージョン
トレッキングツール	全てのバージョン

3.3.2 ディスクハートビートリソース

ディスクハートビートリソースを使用するためには、以下の設定が必要です。

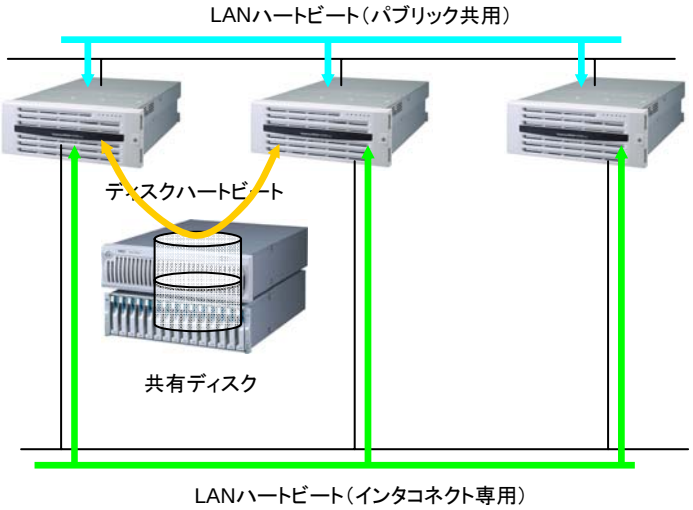
+ 共有ディスク上に専用のパーティションを確保してください。(ファイルシステムを作成する必要はありません。)
+ 全てのサーバから、共有ディスク上の専用パーティションが同じデバイス名でアクセスできるように設定してください。

ディスクハートビートリソースを使用すると、ネットワークが切断された場合でも他サーバの生存を確認することが可能になります。



クラスタが3台以上のサーバで構成されている場合に、以下のようにディスクハートビートリソースを使用する構成が可能です。クラスタ内の共有ディスクを使用するサーバ間でのみディスクハートビートリソースを使用するように設定することができます。

詳細については「トレッキングツール編」を参照してください。

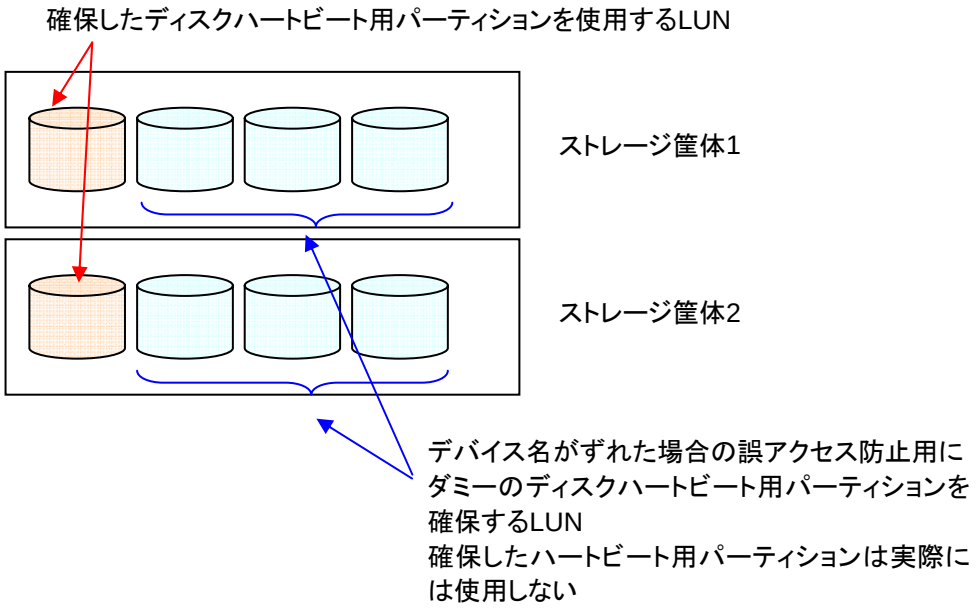


3.3.3 注意事項

* 共有ディスクを使用する場合には、LANハートビートリソースとディスクハートビートリソースの併用を推奨します。

* 複数のLUNを使用している場合でも、ディスクハートビートリソースはクラスタ内で1つまたは2つの使用を推奨します。ディスクハートビートリソースはハートビートインターバルごとにディスクへのread/writeを行うためディスクへの負荷を考えて設定してください。

* 各LUNにディスクハートビート専用パーティションを確保してください。ディスクの故障などでデバイス名がずれた場合にファイルシステムを破壊することがありますので、ディスクハートビートを使用しないLUNにもダミーのパーティションを確保してください。
ディスクハートビート専用パーティションのパーティション番号が各LUNで同じになるように確保してください。



3.4 COMハートビートリソース

3.4.1 動作確認情報

3.4.1.1 CLUSTERPROのバージョン

以下のCLUSTERPROのバージョンでサポートします。

CLUSTERPRO	Version
サーバ	SE、SX、LEの全てのバージョン
トレッキングツール	全てのバージョン

3.4.2 注意事項

* ネットワークが断線した場合に両系で活性することを防ぐため、COMが使用できる環境であればCOMハートビートリソースを使用することを推奨します。

4 シャットダウンストール監視

4.1 CLUSTERPROのバージョン

以下のCLUSTERPROのバージョンでサポートします。

CLUSTERPRO	監視方法	Version
サーバ	softdog	SE3.0-1 以降、LE3.0-1 以降、XE3.0-1 以降、SX3.1-2 以降
サーバ	ipmi	SE3.1-4 以降、LE3.1-4 以降
サーバ	keepalive	SE3.1-6 以降、LE3.1-6 以降、XE3.1-6 以降、SX3.1-6 以降
トレッキングツール	-	対応するバージョンについては動作環境編 トレッキングツールの動作環境を参照してください。

4.2 シャットダウン監視

CLUSTERPROのコマンドでクラスタシャットダウンまたはサーバシャットダウンを実行したときに、OSがストールしているか否か監視します。
クラスタデーモンはOSがストールしていると判断すると強制的にリセットします。

[する]
シャットダウン監視をします。
ハートビート タイムアウト(トレッキングツール編を参照してください。)アプリケーションを含めてOSがシャットダウンする時間より長い時間にする必要があります。共有ディスクまたはミラーディスクを使用する場合は[する]を選択することを推奨します。

[しない]
シャットダウン監視をしません。

4.2.1 監視方法

シャットダウンストール監視の監視方法は以下のとおりです。

(1) 監視方法 softdog

監視方法がsoftdogの場合、softdogドライバを使ってタイマーを設定します。

(2) 監視方法 ipmi

監視方法がipmiの場合、ipmiutilを使ってタイマーを設定します。
ipmiutilがインストールされていない場合、インストールする必要があります。
ipmiについては 2.9 ユーザ空間モニタリソース を参照してください。

(3) 監視方法 keepalive

監視方法がkeepaliveの場合、CLUSTERPROのclpkhbドライバとclpkaドライバを使ってタイマーを設定します。

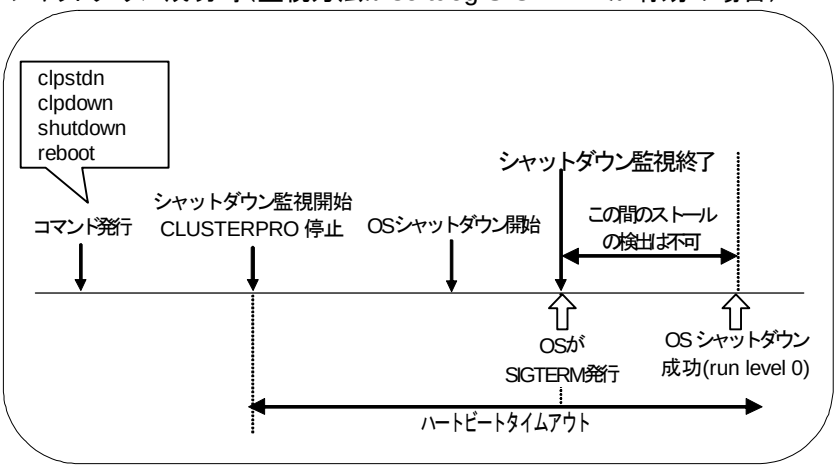
clpkhbドライバ，clpkaドライバが動作するディストリビューション，kernelバージョンについては必ず「動作環境編」で確認してください。
ディストリビュータがリリースするセキュリティパッチを既に運用中のクラスタへ適用する場合(kernelバージョンが変わる場合)にも確認してください。

4.2.2 SIGTERMの設定

OSシャットダウン時に発行されるsignal “SIGTERMを有効にする”の設定によりシャットダウンストール監視の有効範囲とOSが正常にシャットダウンした際の挙動が変わります。

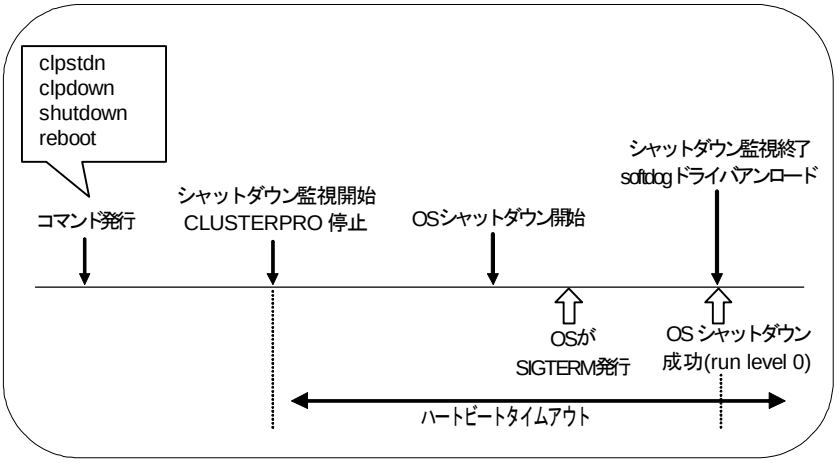
(1) 監視方法 softdog

* シャットダウン成功時（監視方法がsoftdog SIGTERMが有効の場合）



+ SIGTERMを有効にした場合、シャットダウン処理の途中でOSがSIGTERMを発行するとシャットダウン監視が終了するので、ストールを検出できません。

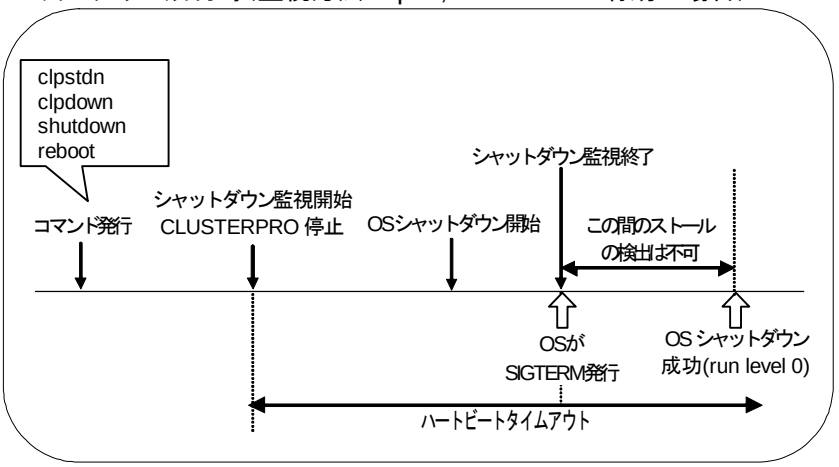
* シャットダウン成功時（監視方法がsoftdogの場合, SIGTERMが有効でない場合）



* 監視方法がsoftdogの場合、SIGTERMを無効にすることを推奨します。

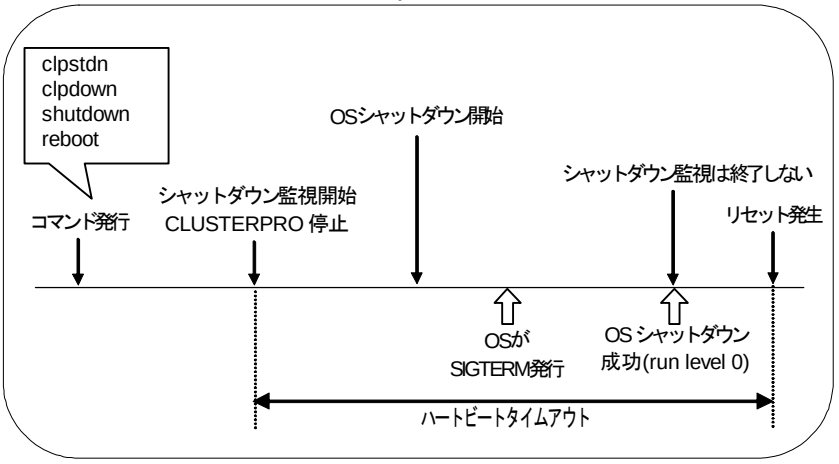
(2) 監視方法 ipmi

* シャットダウン成功時（監視方法がipmi, SIGTERMが有効の場合）



+ SIGTERMを有効にした場合、シャットダウン処理の途中でOSがSIGTERMを発行するとシャットダウン監視が終了するので、ストールを検出できません。

* シャットダウン成功時（監視方法がipmiの場合, SIGTERMが有効でない場合）

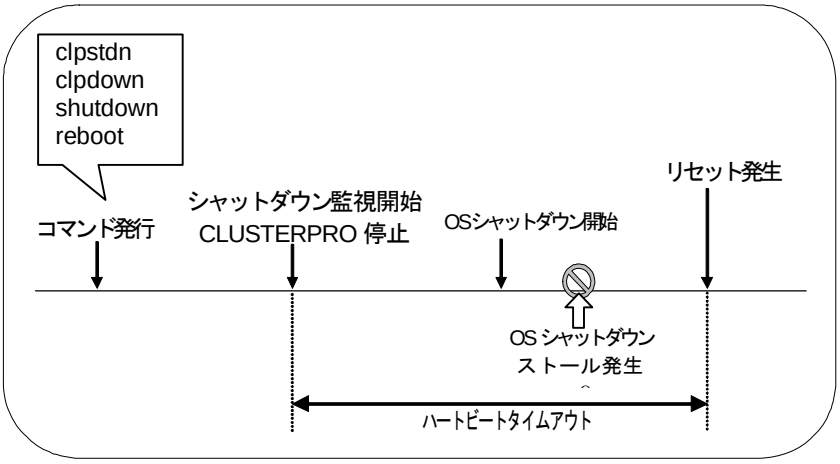


+ ストールが発生しないで正常にシャットダウンが完了した場合もipmiによってリセットが発生します。
+ ソフトウェア電源offが可能なサーバではリセットは発生しません。

* 監視方法がipmiの場合、SIGTERMを有効にする設定を推奨します。

(3) OSシャットダウンでストールが発生した場合

* シャットダウンストール検出時



5 付録1

5.1 bonding

5.1.1 FIPリソース

5.1.1.1 動作確認情報

5.1.1.1.1 CLUSTERPROのバージョン

以下のCLUSTERPROのバージョンでサポートします。

CLUSTERPRO	Version
サーバ	SE3.0-4 以降、LE3.0-4 以降、XE3.1-4 以降、SX3.1-2 以降
トレッキングツール	対応するバージョンについては動作環境編 トレッキングツールの動作環境を参照してください。

5.1.1.1.2 ディストリビューション

以下のバージョンで動作確認しています。

Distribution	kernel	note
RedHat ES/AS3	2.4.21-9.0.1.ELsmp	bonding v2.2.14 e100 2.3.30-k1 e1000 5.2.20-k1
TurboLinux ES8	2.4.21-231-smp	bonding v2.2.14 e100 2.3.27 e1000 5.2.16
MIRACLE LINUX V3.0	2.4.21-9.30AXsmp	bonding v2.2.14 e100 2.3.40 e1000 5.2.39

5.1.1.1.3 ネットワークインタフェース

以下のネットワークインタフェースで動作確認しています。

Ethernet Controller(Chip)	Bus	Driver
Intel 82557/8/9	PCI	e100
Intel 82544EI	PCI	e1000

5.1.1.2 注意事項

bondingモードに"active-backup"を指定すると、スレーブインターフェイスの切り替えの際、一時的に通信が途絶えることがあるため推奨しません。

5.1.1.3 bonding設定例

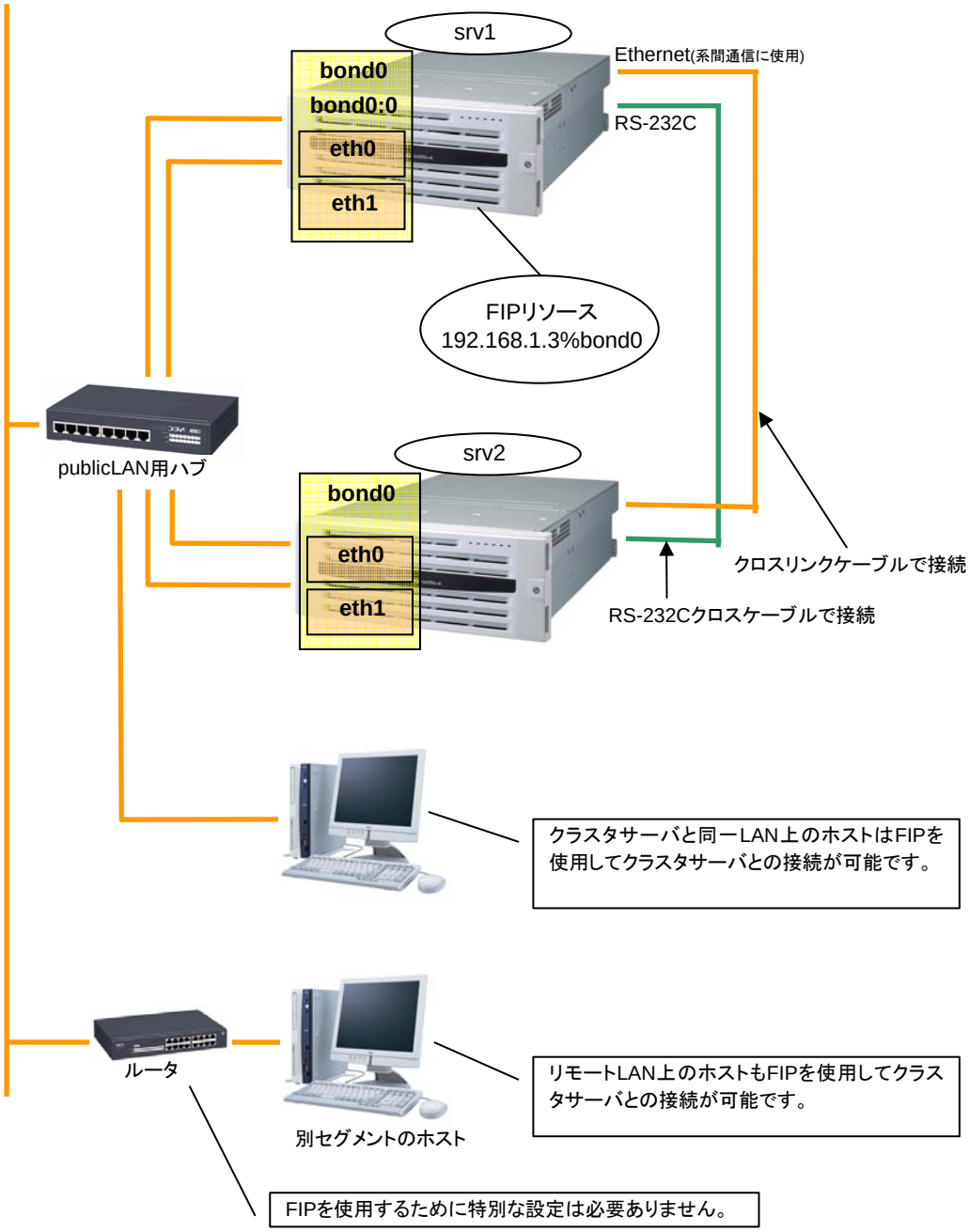
トレッキングツールでFIPリソースを設定する際、以下のようにIPアドレスとbondingデバイスを"%"で区切って指定してください。

例）デバイス名 bond0、IPアドレス 192.168.1.3 を設定する場合

インタコネクトのIPアドレス設定には、IPアドレスのみ設定してください。

bondig上にFIPリソースを使用する設定例を示します。

bonding			
Cluster Server	Device	Slave	Mode
srv1	bond0	eth0 eth1	- active-backup(1) - balance-tlb(5)
srv2	bond0	eth0 eth1	- active-backup(1) - balance-tlb(5)



srv1でのifconfigによるFIPリソースの活性状態は以下のようになります。
（bonding modeは、"balance-tlb(5)"を指定。）

```
$ ifconfig
bond0     Link encap:Ethernet  HWaddr 00:00:01:02:03:04
          inet addr:192.168.1.1 Bcast:192.168.1.255 Mask:255.255.255.0
          UP BROADCAST RUNNING MASTER MULTICAST  MTU:1500  Metric:1
          RX packets:6807 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2970 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:670032 (654.3 Kb)  TX bytes:189616 (185.1 Kb)

bond0:0   Link encap:Ethernet  HWaddr 00:00:01:02:03:04
          inet addr:192.168.1.3 Bcast:192.168.1.255 Mask:255.255.255.0
          UP BROADCAST RUNNING MASTER MULTICAST  MTU:1500  Metric:1
          RX packets:236 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2239 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:78522 (76.6 Kb)  TX bytes:205590 (200.7 Kb)

eth0      Link encap:Ethernet  HWaddr 00:00:01:02:03:04
          UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500  Metric:1
          RX packets:3434 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1494 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:332303 (324.5 Kb)  TX bytes:94113 (91.9 Kb)
          Interrupt:18 Base address:0x2800 Memory:fc041000-fc041038

eth1      Link encap:Ethernet  HWaddr 00:00:05:06:07:08
          UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500  Metric:1
          RX packets:215  errors:0 dropped:0 overruns:0 frame:0
          TX packets:1627 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:77162 (75.3 Kb)  TX bytes:141394 (138.0 Kb)
          Interrupt:19 Base address:0x2840 Memory:fc042000-fc042038

eth2      Link encap:Ethernet  HWaddr 00:00:09:10:11:12
          inet addr:192.168.2.1 Bcast:192.168.2.255 Mask: 255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:47 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1525 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2820 (2.7 Kb)  TX bytes:110113 (107.5 Kb)
          Interrupt:24 Base address:0x3000 Memory:fc500000-fc500038
```

① ②

③

① eth0, eth1 を bonding 化したデバイス
パブリックLAN、2番目のインタコネクトに使用

② bond0 上で活性した FIP

③ 1番目のインタコネクトに使用

5.1.2 ミラーコネクト

5.1.2.1 動作確認情報

5.1.2.1.1 CLUSTERPROのバージョン

以下のCLUSTERPROのバージョンでサポートします。

CLUSTERPRO	Version
サーバ	LE3.1-1 以降
トレッキングツール	対応するバージョンについては動作環境編 トレッキングツールの動作環境を参照してください。

5.1.2.1.2 ディストリビューション

以下のバージョンで動作確認しています。

Distribution	kernel	note
TurboLinux ES8	2.4.21-231-default 2.4.21-231-smp	bonding v2.2.14 e100 2.3.27 e1000 5.2.16

5.1.2.1.3 ネットワークインタフェース

以下のネットワークインタフェースで動作確認しています。

Ethernet Controller(Chip)	Bus	Driver
Intel 82557/8/9	PCI	e100
Intel 82540EM	PCI	e1000
Intel 8086:1026	PCI	e1000

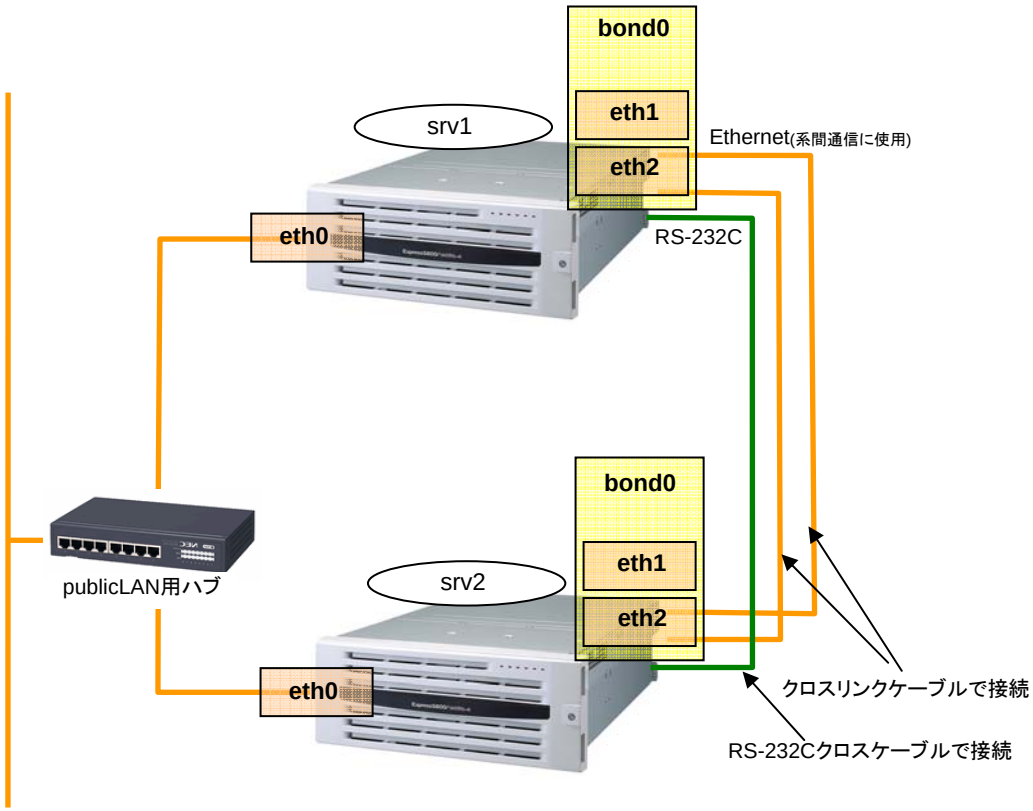
5.1.2.2 注意事項

bonding上でミラーコネクトを使用すると、スレーブインターフェイスの切り替えの際、一時的に通信が途絶えることがあるため推奨しません。

5.1.2.3 bonding設定例

bondig上にミラーコネクトを使用する設定例を示します。

bonding			
Cluster Server	Device	Slave	Mode
srv1	bond0	eth1 eth2	- balance-rr(0) - active-backup(1) - balance-tlb(5)
srv2	bond0	eth1 eth2	- balance-rr(0) - active-backup(1) - balance-tlb(5)



6 付録2

6.1 業務の洗い出し

CLUSTERPROを導入する場合、まず可用性を向上しなければならないアプリケーションを、洗い出す必要があります。また、洗い出したアプリケーションが、CLUSTERPROの環境下で動作するのに適しているかどうかを、見極めなければなりません。

洗い出したアプリケーションが、CLUSTERPROでのクラスタ対象として適しているかどうかは、次節からの内容を十分検討して判断してください。

6.2 CLUSTERPRO環境下でのアプリケーション

ここでは、CLUSTERPRO環境下で動作できるアプリケーションについて、留意すべき事項を述べます。

6.2.1 サーバアプリケーション

対象アプリケーションがどのようなスタンバイ形態で実行するかで注意事項が異なります。
注意事項については「6.2.2 サーバアプリケーションについての注意事項」に対応します。

* 片方向スタンバイ[運用-待機] 注意事項: 1 2 3 ~~4~~ 5
  クラスタ内で、あるアプリケーションの稼動サーバが常に一台である運用形態です。
* 双方向スタンバイ[運用-運用] 注意事項: 1 2 3 4 5
  クラスタ内で、あるアプリケーションの稼動サーバが複数台である運用形態です。
* 共存動作 注意事項: ~~1 2 3 4~~ 5
  クラスタシステムによるフェイルオーバの対象とはせず、共存動作する運用形態です。

6.2.2 サーバアプリケーションについての注意事項

(1) 障害発生後のデータ修復

障害発生時にアプリケーションが更新していたファイルは、待機系にてアプリケーションがそのファイルにアクセスするときデータとして完結していない状態にある場合があります。

非クラスタ(単体サーバ)での障害後のリブートでも同様のことが発生するため、本来アプリケーションはこの状態に備えておく必要があります。クラスタシステム上ではこれに加え人間の関与なしに(スクリプトから)復旧が行える必要があります。

共有ディスクまたはミラーディスクのファイルシステムにfsckが必要な場合には、CLUSTERPROがfsckを行います。

(2) アプリケーションの終了

CLUSTERPROが業務グループを停止・移動(オンラインフェイルバック)する場合、その業務グループが使用していたファイルシステムをアンマウントします。このため、アプリケーションへの終了指示にて、共有ディスクまたはミラーディスク上の全てのファイルに対するアクセスを停止する必要があります。

通常は終了スクリプトでアプリケーション終了指示コマンドを実行しますが、終了指示コマンドが(アプリケーションの終了と)非同期で完了してしまう場合注意が必要です。

(3) データ格納位置

CLUSTERPROがサーバ間で引き継ぐことのできるデータは次の通りです。

+ 共有ディスクまたはミラーディスク上のデータ

アプリケーションはサーバ間で引き継ぎたいデータと引き継ぎたくないデータを分離できる必要があります。

データの種類	例	配置場所
引き継ぎたいデータ	ユーザデータなど	共有ディスクまたはミラーディスク
引き継ぎたくないデータ	プログラム、 設定情報など	サーバのローカルディスク

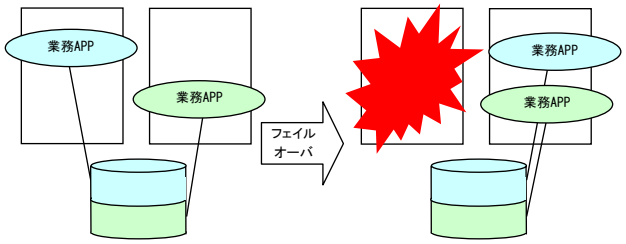
(4) 複数業務グループ

双方向スタンバイの運用形態では、(障害による縮退時)、1つのサーバ上で同一アプリケーションによる複数業務グループが稼動することを想定しなくてはなりません。

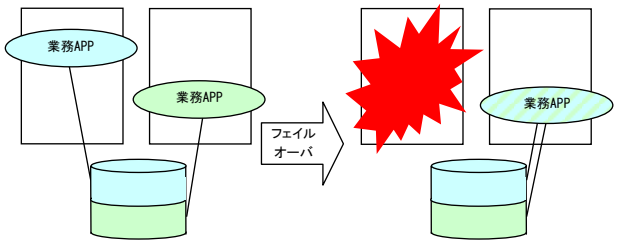
アプリケーションは次のいずれかの方法で引き継がれた資源を引き取り、単一サーバ上で複数業務グループを実行できなければなりません。

ミラーディスクも同じ考え方です。

* 複数インスタンス起動
  新たに別インスタンス(プロセス)を起動する方法です。アプリケーションが複数動作できる必要があります。

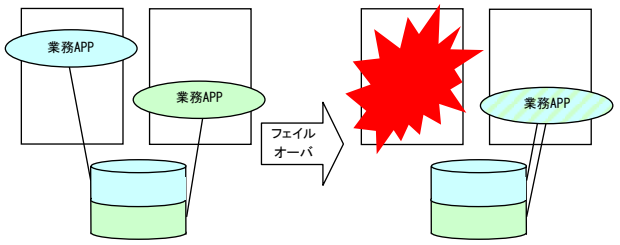


* アプリケーション再起動
  もともと動いていたアプリケーションを一旦停止し、再起動することで、追加された資源を扱えるようにする方法です。



業務APPを再起動することでデータを引き継ぐ

* 動的追加
  動作中のアプリケーションに対して、自動またはスクリプトからの指示により資源を追加する方法です。



実行中の業務APPに動的にデータを追加することでデータを引き継ぐ

(5) アプリケーションとの相互干渉(相性問題)

CLUSTERPROの機能や動作に必要なOS機能との相互干渉によってアプリケーションまたはCLUSTERPROが動作できない場合があります。

* 切替パーティションとミラーパーティションのアクセス制御
  非活性状態の共有ディスクは書込み禁止の設定になります。
  非活性状態のミラーディスクは読込み,書込み禁止の設定になります。
  アプリケーションは非活性状態の(つまりアクセス権利のない)共有ディスクまたはミラーディスクにアクセスしてはいけません。
  通常、クラスタスクリプトから起動されるアプリケーションは、それが起動された時点でアクセスすべき切替パーティションまたはミラーパーティションが既にアクセス可となっていることを想定してかまいません。

* マルチホーム環境及びIPアドレスの移動
  クラスタシステムでは、通常、一つのサーバが複数のIPアドレスを持ち、あるIPアドレス(フローティングIPアドレスなど)はサーバ間で移動します。

* アプリケーションの共有ディスクまたはミラーディスクへのアクセス
  共存動作アプリケーションには、業務グループの停止が通知されません。もし、業務グループの停止のタイミングでそのグループが使用している切替パーティションまたはミラーパーティションにアクセスしている場合、アンマウントに失敗してしまいます。

  システム監視サービスを行うようなアプリケーションの中には、定期的に全てのディスクパーティションをアクセスするようなものがあります。この場合、監視対象パーティションを指定できる機能などが必要になります。

6.2.3 注意事項に対する対策

6.2.2の注意事項に対応する番号		
問題点	**対策**	
データファイル更新中に障害が発生した場合、待機系にてアプリケーションが正常に動作しない	プログラム修正	(1)
アプリケーションを停止しても一定時間の間、共有ディスクまたはミラーディスクへアクセスしつづける	停止スクリプト中にsleepコマンドを使用し待ち合わせる	(2)
一台のサーバ上で同一アプリケーションを複数起動できない	双方向スタンバイ運用では、フェイルオーバ時にアプリケーションを再起動し共有データを引き継ぐ	(4)

6.3 業務形態の決定

6.2章全体を踏まえた上で、業務形態を決定してください。

* どのアプリケーションをいつ起動するか
* 起動時やフェイルオーバ時に必要な処理は何か
* 共有ディスクまたはミラーディスクに置くべき情報は何なのか

また、以下を運用の中に必ず組み込んでください。

* 共有ディスクまたはミラーディスクの定期的なバックアップ