

CLUSTERPRO[®] X for Linux

VMware vSphere[™] 4 システム構築ガイド

2010/12/06
第6版

CLUSTERPRO

改版履歴

版数	改版日付	内 容
1	2009/07/14	新規作成
2	2009/08/21	内部バージョン2.1.1-1に対応
3	2010/02/26	二つ目の仮想NICを追加する手順を記載 物理NICと仮想NICを結びつける手順を記載 vSphereのfirewall設定について記載 ホスト側NIC異常検出時の設定について記載 userwの設定について記載
4	2010/07/09	章立て等の文書構成を変更 リンク切れ、誤字脱字等の修正
5	2010/10/01	CLUSTERPRO X 3.0 に対応
6	2010/12/06	仮想ディスクによる共有ディスク型クラスタ構成の削除

© Copyright NEC Corporation 2010. All rights reserved.

免責事項

本書の内容は、予告なしに変更されることがあります。

日本電気株式会社は、本書の技術的もしくは編集上の間違い、欠落について、一切責任をおいませぬ。

また、お客様が期待される効果を得るために、本書に従った導入、使用および使用効果につきましては、お客様の責任とさせていただきます。

本書に記載されている内容の著作権は、日本電気株式会社に帰属します。本書の内容の一部または全部を日本電気株式会社の許諾なしに複製、改変、および翻訳することは禁止されています。

商標情報

CLUSTERPRO[®] X は日本電気株式会社の登録商標です。

AMD, AMD VirtualizationはAdvanced Micro Devices, Incの商標です。

Intel, Pentium, Xeon, Intel VTは、Intel Corporationの登録商標または商標です。

VMware, VMotionは、米国VMware, Inc.の登録商標です。

Linuxは米国及びその他の国におけるLinus Torvaldsの登録商標です。

その他のシステム名、社名、製品名等はそれぞれの会社の商標及び登録商標です。

目次

はじめに	vii
対象読者と目的	vii
関連マニュアル	viii
本書の構成	ix
本書の表記規則	x
本書で用いる用語	xi
第 1 章 構成	13
ホストOS間クラスタ	13
ゲストOS間クラスタ	15
物理サーバー仮想マシン間クラスタ	16
VMware HA連携	17
第 2 章 動作環境	19
第 3 章 注意事項	21
Service ConsoleでCLUSTERPROを利用する場合の注意事項	21
仮想マシンでCLUSTERPROを利用する場合の注意事項	21
ホストOS間クラスタの注意事項	22
ゲストOS間クラスタの注意事項	22
物理サーバー仮想マシン間クラスタの注意事項	22
VMotionを併用する場合の注意事項	23
VMware HAを併用する場合の注意事項	23
第 4 章 仮想化環境構築手順	25
VMware ESXのインストール	25
仮想スイッチの追加	25
仮想マシンの作成	27
第 5 章 クラスタ環境構築手順	29
ホストOS間クラスタの構築	29
ゲストOS間クラスタの構築	36
物理サーバー仮想マシン間クラスタの構築	38
仮想マシン制御用リソースの設定	40
仮想マシン監視用モニタの設定	46
VMware HAとの連携の設定	51
付録 A サンプルスクリプト	53

はじめに

対象読者と目的

『CLUSTERPROシステム構築ガイド』は、クラスタシステムに関して、システムを構築する管理者、およびユーザサポートを行うシステムエンジニア、保守員を対象にしています。

本書では、CLUSTERPRO環境下での動作確認が取れたソフトウェアをご紹介します。ここでご紹介するソフトウェアや設定例は、あくまで参考情報としてご提供するものであり、各ソフトウェアの動作保証をするものではありません。

関連マニュアル

本書の利用にあたっては、必要に応じて以下のマニュアルを参照してください。

1. CLUSTERPRO マニュアル

CLUSTERPRO のマニュアルは、以下の 4 つに分類されます。

『CLUSTERPRO X スタートアップガイド』(Getting Started Guide)

CLUSTERPROを使用するユーザを対象読者とし、製品概要、動作環境、アップデート情報、既知の問題などについて記載します。

『CLUSTERPRO X インストール & 設定ガイド』(Install and Configuration Guide)

CLUSTERPRO を使用したクラスタ システムの導入を行うシステム エンジニアと、クラスタシステム導入後の保守・運用を行うシステム管理者を対象読者とし、CLUSTERPRO を使用したクラスタ システム導入から運用開始前までに必須の事項について説明します。実際にクラスタ システムを導入する際の順番に則して、CLUSTERPRO を使用したクラスタ システムの設計方法、CLUSTERPRO のインストールと設定手順、設定後の確認、運用開始前の評価方法について説明します。

『CLUSTERPRO X リファレンス ガイド』(Reference Guide)

管理者、およびCLUSTERPRO を使用したクラスタ システムの導入を行うシステム エンジニアを対象とし、CLUSTERPRO の運用手順、各モジュールの機能説明、メンテナンス関連情報およびトラブルシューティング情報等を記載します。『インストール & 設定ガイド』を補完する役割を持ちます。

『CLUSTERPRO X 統合WebManager 管理者ガイド』(Integrated WebManager Administrator's Guide)

CLUSTERPRO を使用したクラスタシステムを CLUSTERPRO 統合WebManager で管理するシステム管理者、および統合WebManager の導入を行うシステム エンジニアを対象読者とし、統合WebManager を使用したクラスタ システム導入時に必須の事項について、実際の手順に則して詳細を説明します。

CLUSTERPRO マニュアルに関しては、以下を参照してください。

『CLUSTERPRO Webサイト』

<http://www.nec.co.jp/clusterpro/>

2. VMware vSphereドキュメント

VMware vSphereの詳細については、VMware vSphereドキュメントを参照してください。

『VMware vSphereドキュメント』

http://www.vmware.com/jp/support/pubs/vs_pubs.html

本書の構成

- 第 1 章 「構成」:VMware vSphereとCLUSTERPROを組み合わせることにより実現可能なクラスタシステムについて記述します。
- 第 2 章 「動作環境」:VMware vSphereとCLUSTERPROを組み合わせる場合の動作環境について記述します。
- 第 3 章 「注意事項」:VMware vSphereとCLUSTERPROを組み合わせる場合の注意事項について記述します。
- 第 4 章 「仮想化環境構築手順」:VMware vSphereとCLUSTERPROを組み合わせる場合の構築手順について記述します。

本書の表記規則

本書では、「注」および「重要」を以下のように表記します。

注: は、重要ではあるがデータ損失やシステムおよび機器の損傷には関連しない情報を表します。

重要: は、データ損失やシステムおよび機器の損傷を回避するために必要な情報を表します。

関連情報: は、参照先の情報の場所を表します。

また、本書では以下の表記法を使用します。

表記	使用方法	例
[] 角かっこ	コマンド名の前後 画面に表示される語 (ダイアログ ボックス、メニューなど) の前後	[スタート] をクリックします。 [プロパティ] ダイアログ ボックス
コマンドライン中の [] 角かっこ	かっこ内の値の指定が省略可能であることを示します。	<code>clpstat -s [-h host_name]</code>
モノスペースフォント (courier)	コマンド ライン、関数、パラメータ	<code>clpstat -s</code>
モノスペースフォント太字 (courier)	ユーザが実際にコマンドプロンプトから入力する値を示します。	以下を入力します。 <code>clpcl -s -a</code>
モノスペースフォント (courier) 斜体	ユーザが有効な値に置き換えて入力する項目	<code>clpstat -s [-h host_name]</code>

本書で用いる用語

本書で用いる用語について説明します。

用語	略語	説明
物理サーバ	SV	VMware ESXまたは他のOSが動作しているサーバです。
単体OS	OS	仮想化基盤ではなく単独で利用する通常のOSです。
ホストOS	ホスト	仮想化基盤として物理サーバにインストールされているOS、つまりVMware ESXです。
仮想マシン	VM	ホストOS上に作成される仮想的なサーバまたはクライアントです。
ゲストOS	ゲスト	仮想マシンにインストールされているOSです。
CLUSTERPRO X	CLS	CLUSTERPRO Xです。
CLUSTERPRO X SingleServerSafe	SSS	CLUSTERPRO X SingleServerSafeです。
Application	AP	業務アプリケーションです。

第 1 章 構成

VMware vSphere と CLUSTERPRO を組み合わせることで、下記構成のクラスタを構築することができます。

ホストOS間クラスタ

VMware ESX の Service Console 上に CLUSTERPRO X をインストールし、物理サーバ同士でクラスタリングを行います。通常の業務アプリケーションのフェイルオーバーのみならず、ゲスト OS をフェイルオーバーさせることができます。

また、ゲスト-ホスト間の連携により、ゲスト OS 内の業務アプリケーションの監視も可能です。

クラスタ構築手順は『ホストOS間クラスタを構築する(33ページ)』を参照してください。また、ゲスト-ホスト連携を利用する場合は『ホストOS間クラスタでゲスト-ホスト連携を利用する(35ページ)』を参照してください。

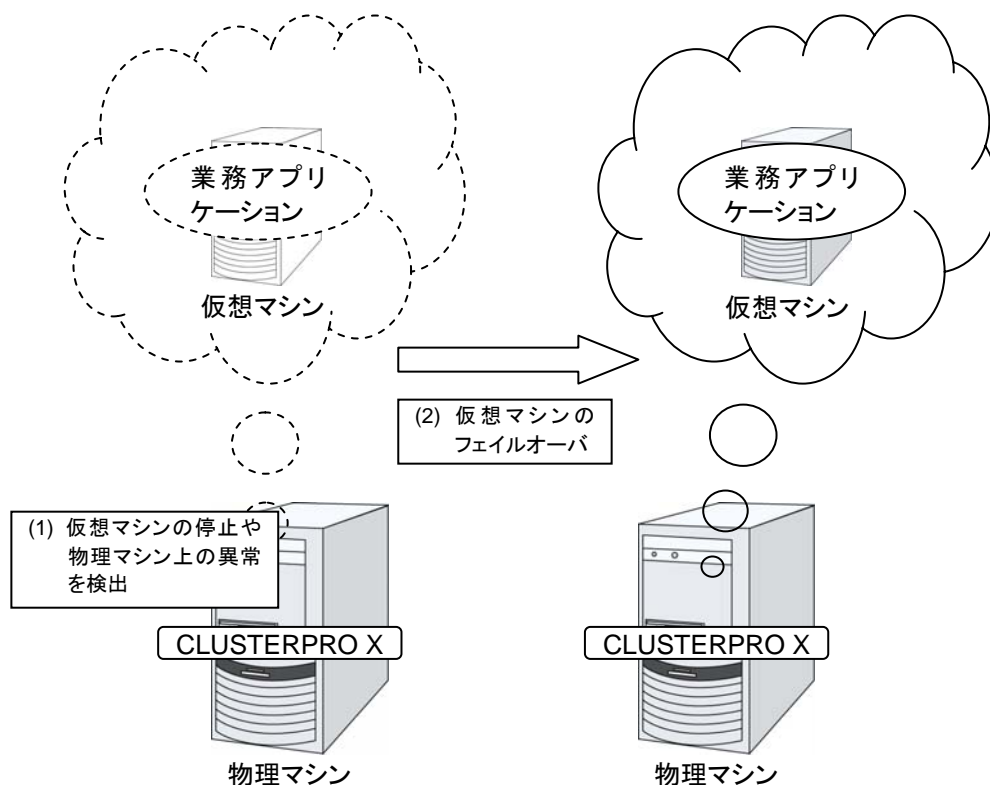


図 1 : ホスト OS 間クラスタの概要図

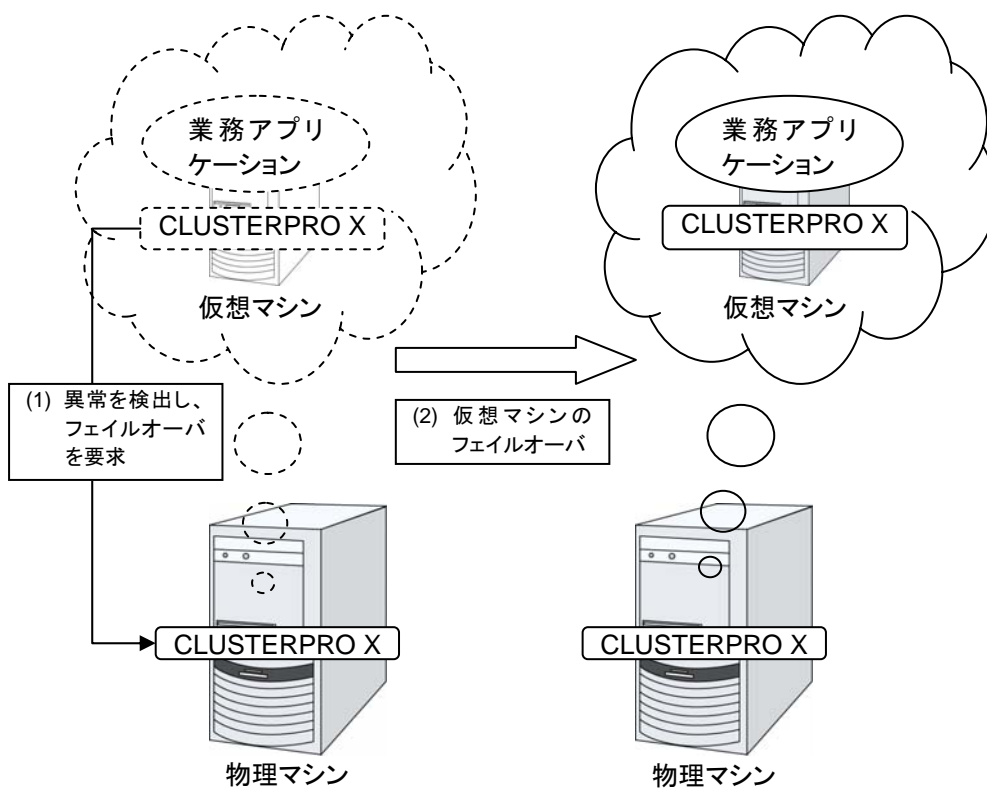


図 2 : ゲスト-ホスト連携時のホスト OS 間クラスタの概要図

ゲストOS間クラスタ

ゲスト OS 上に CLUSTERPRO X をインストールし、仮想マシン同士でクラスタリングを行います。通常のクラスタシステムと同様、業務アプリケーションのフェイルオーバーが可能で、業務の可用性を高めることができます。

クラスタ構築手順は『ゲストOS間クラスタを構築する(36ページ)』を参照してください。

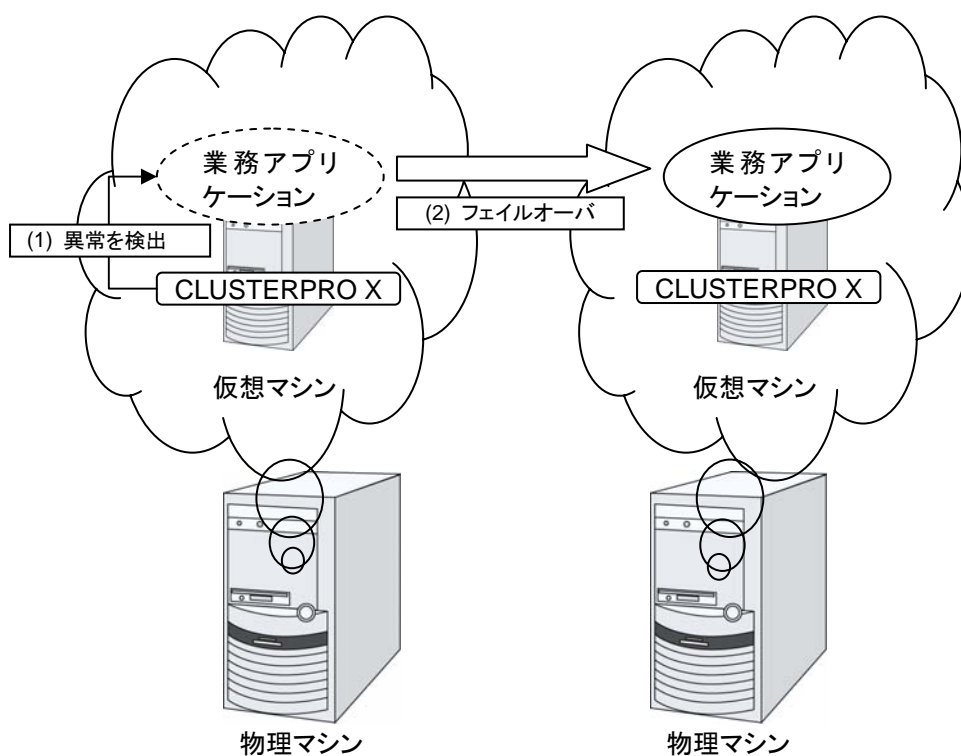


図 3 : ゲスト OS 間クラスタの概要図

物理サーバー-仮想マシン間クラスタ

物理サーバ上で動作している OS と仮想マシン上で動作している OS に CLUSTERPRO X をインストールし、物理サーバと仮想マシン間でクラスタリングを行います。通常のクラスタシステムと同様、業務アプリケーションのフェイルオーバーが可能です。

クラスタ構築手順は『物理サーバー-仮想マシン間クラスタの構築(38ページ)』を参照してください。

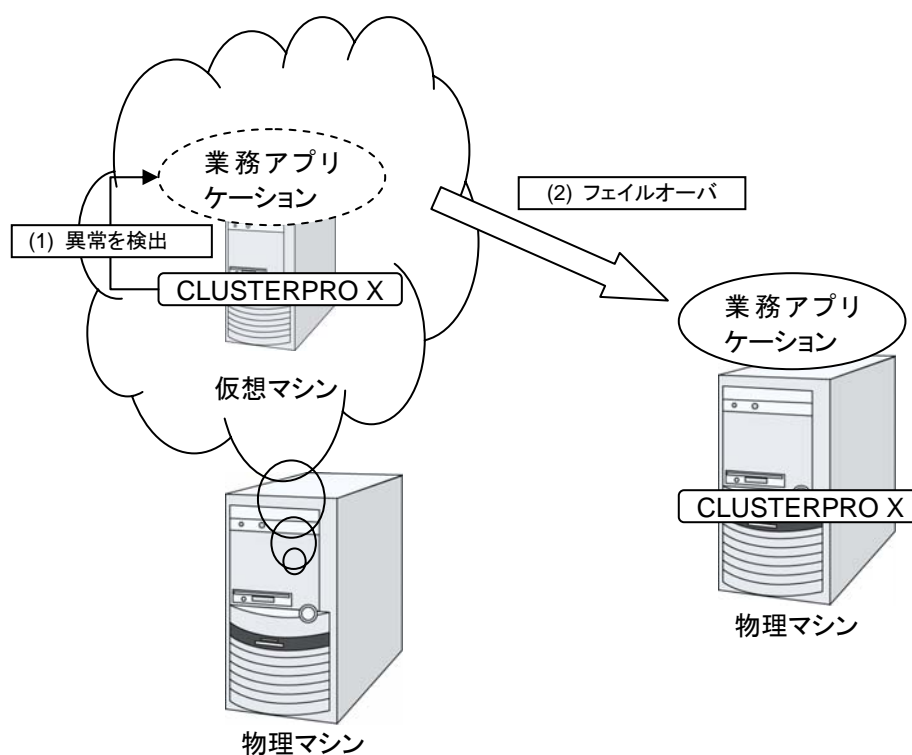


図 4：物理サーバー-仮想マシン間クラスタの概要図

VMware HA連携

仮想マシンは VMware HA を構成し、ゲスト OS ダウン時／vCenter-ESX 間ネットワーク断絶時にはゲスト OS をフェイルオーバーします。

VMware ESXのService Console上にCLUSTERPRO X SingleServerSafeをインストールすることで、VMware で検出できない物理HWの異常を検出することができます。特に、ゲストOSが格納される共有ディスクの異常検出時には速やかにESXをシャットダウンし、動作が不安定なゲストOSで業務を継続させないようにします。(図 5)

また、ゲストOS上にCLUSTERPRO Xをインストールすることで、ゲストOS上の異常(仮想HW異常、AP異常を含む)を検出し、業務をフェイルオーバーすることが可能になります。(図 6)

本構成の構築手順は『VMware HAとの連携(51ページ)』を参照してください。

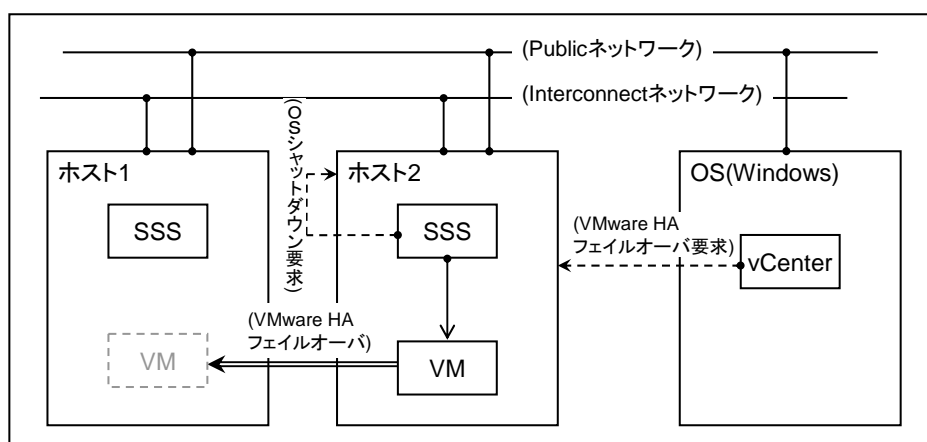


図 5 : ホスト上の CLUSTERPRO X SingleServerSafe と VMware HA の連携構成の概要図

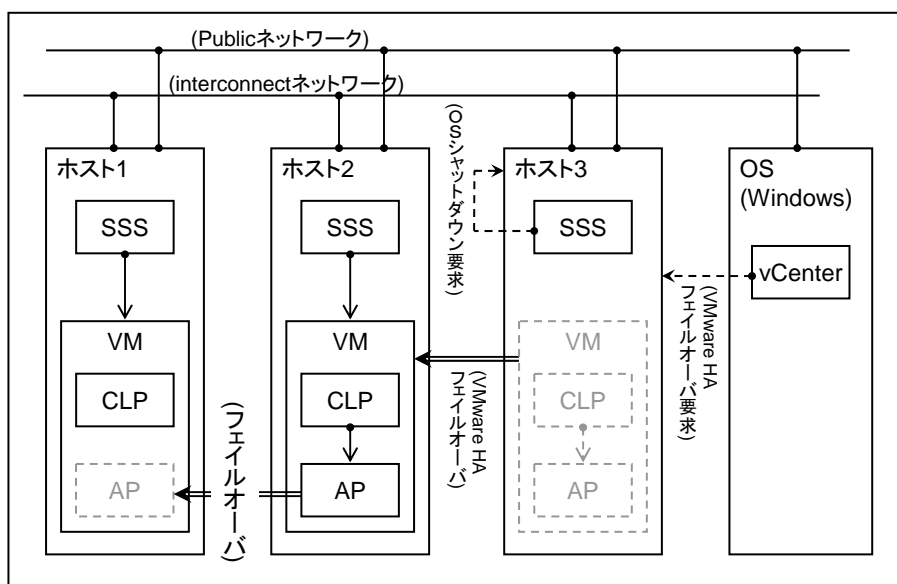


図 6 : ゲスト上の CLUSTERPRO X、ホスト上の CLUSTERPRO X SingleServerSafe と VMware HA の連携構成の概要図

第 2 章 動作環境

- 本書が対象とする VMware ESX 及び CLUSTERPRO のバージョンは下記のとおりです。

VMware ESX

- VMware ESX 4.0
- VMware ESX 4.0 update 1
- VMware ESX 4.0 update 2
- VMware ESX 4.1

CLUSTERPRO(ホストOS側)

- CLUSTERPRO X 2.1 for Linux
- CLUSTERPRO X SingleServerSafe 2.1 for Linux
- CLUSTERPRO X 3.0 for Linux
- CLUSTERPRO X SingleServerSafe 3.0 for Linux

CLUSTERPRO(ゲストOS側)

- CLUSTERPRO X 2.1 for Windows
- CLUSTERPRO X 2.1 for Linux
- CLUSTERPRO X SingleServerSafe 2.1 for Windows
- CLUSTERPRO X SingleServerSafe 2.1 for Linux
- CLUSTERPRO X 3.0 for Windows
- CLUSTERPRO X 3.0 for Linux
- CLUSTERPRO X SingleServerSafe 3.0 for Windows
- CLUSTERPRO X SingleServerSafe 3.0 for Linux

第 3 章 注意事項

Service ConsoleでCLUSTERPROを利用する場合の注意事項

- Service Consoleには、x86_64 版のrpm (clusterpro-X.X.X-x86_64.rpm) をインストールしてください。¹
- ユーザ空間監視リソース (userw) やシャットダウンストール監視を利用する場合、監視方式は keepalive を選択してください。
- 下記の機能を利用することはできません。
 - ミラーディスクリソース(md)
 - ハイブリットディスクリソース(hd)
- Service Console で CLUSTERPRO を利用する場合、CLUSTERPRO の通信用ポートにアクセスできるようにファイアウォールを設定する必要があります。
- 二つ目の NIC を使用する場合、ホストのネットワーク構成で「サービス コンソール ポート」の追加が必要です。
- ホストのネットワーク構成の各ポートに共通の IP アドレスを設定することはできません。「VMkernel」と「Service Console」を同時に使用したい場合は別々の IP アドレスを設定してください。
- VMware ESX の core の出力先は、デフォルトでは /var/core 配下となるため、CLUSTERPRO のログ収集で収集できません。

仮想マシンでCLUSTERPROを利用する場合の注意事項

- 二つ目の物理アダプタを利用したい場合、ホストのネットワーク構成で二つ目の物理アダプタに結びついた仮想スイッチを作成する必要があります。

¹ X2.1にはVMware ESX 3.5 Service Console用のrpm(clusterpro-2.1.X-X.vmware.i386.rpm)があるため、間違っても利用しないよう注意してください。

ホストOS間クラスタの注意事項

- 複数の仮想マシンから構成されるクラスタではないため、仮想マシンにインストールされた OS やアプリケーションのローリングメンテナンス（現用系で業務を継続させたまま、待機系にアップデートやパッチを適用する運用）ができません。
- ゲスト-ホスト連携を利用する場合は、ゲスト OS に CLUSTERPRO X2.1 以降の製品をインストールする必要があります。

ゲストOS間クラスタの注意事項

- 共有ディスクを利用する場合、下記の条件を満たす必要があります。
 - 共有ディスクが「Raw デバイスのマッピング」であること。
 - 仮想マシンの SCSI コントローラの「SCSI バスの共有」設定が「物理」または「仮想」であること。「仮想」を選ぶと 図 4 のような ESX ホストを跨いだゲスト OS 間クラスタは構成できません。
 - 「物理」:あらゆる ESX ホストの VM 間でハードディスクが共有できる
 - 「仮想」:同一 ESX ホストの VM 間でハードディスクが共有できる
- クラスタ運用時には、仮想マシンの[サスペンド]を行わないでください。仮想マシンの[サスペンド]を行うと、CLUSTERPRO がハートビートタイムアウトを検出し、他のサーバでフェイルオーバーグループを起動します。この状態で、[サスペンド]していた仮想マシンを[リジューム]すると、両系活性状態となり、データ保護の観点からそのフェイルオーバーグループが起動している両方の仮想マシンをシャットダウンします。

物理サーバー仮想マシン間クラスタの注意事項

- 「ゲストOS間クラスタの注意事項」を参照してください。

VMotionを併用する場合の注意事項

- VMotion を利用する場合、下記の条件を満たす必要があります。
 - VMware vCenter Server が導入されていること。
 - 各 ESX のネットワークに VMotion で使用できる「VMkernel ポート」が構成されていること。
 - 共有ストレージ装置¹が構成されており、仮想マシンが共有ストレージ上のデータストアに格納されていること。
 - 仮想マシンの SCSI コントローラの「SCSI バスの共有」設定が「なし」であること。
 - DNS が正しく構成されており、vCenter と全 ESX のサービスコンソールは互いに名前解決ができること。
 - 各 ESX に構成されているホスト名と、DNS 側にて登録されているホスト名が一致していること。
- VMotion は、共有ディスクを利用するゲスト OS 間クラスタとは併用できません。VMware vSphere の仕様により、仮想マシンの設定で SCSI コントローラの「SCSI バスの共有」を「なし」にすると、同じハードディスクを持つ仮想マシンは一つしか起動できなくなります。

表 1 : VMotion 併用可否一覧

	共有ディスク型	ミラーディスク型
ホストOS間クラスタ	可	-
ゲストOS間クラスタ	不可	可
物理サーバー仮想マシン間クラスタ	可	可

VMware HAを併用する場合の注意事項

- VMotion が併用可能な構成であること。
- 各 ESX が VMware HA を利用できるライセンスを保有していること。
- 各 ESX において、Service Console の Gateway アドレスが正しく設定されており、Gateway が ping に応答すること。
- VMware HA で仮想マシンの監視を有効にする場合、監視する仮想マシンに VMware Tools がインストールされていること。

¹共有ストレージ装置との接続方式として Fibre Channel SAN、iSCSI SANもしくはNFSが利用可能です。

第 4 章 仮想化環境構築手順

VMware ESXのインストール

VMware 社が用意しているインストールガイドにしたがって VMware ESX をインストールしてください。

- VMware サポートのリソース ドキュメント
<http://www.vmware.com/jp/support/pubs/>

Service Console で CLUSTERPRO を利用する場合は、CLUSTERPRO の通信用ポートにアクセスできるようにファイアウォールを設定してください。

ファイアウォールを無効化する場合:

```
# esxcfg-firewall --allow Incoming --allow Outgoing
```

CLUSTERPRO が利用するポート番号は下記を参照してください。

- CLUSTERPRO X for Linux スタートアップガイド
第 5 章 注意制限事項
> OS インストール後、CLUSTERPRO インストール前
>> 通信ポート番号

仮想スイッチの追加

サーバに複数の物理 NIC がある場合、デフォルトでは一方に関連付けられた仮想スイッチのみ作成されるため、もう一方の物理 NIC を利用する場合は新たに仮想スイッチを追加する必要があります。

二つ目の仮想スイッチを追加する

- (1) VMware vSphere Client から接続し、ホスト(物理サーバ)を選択し、「構成」タブの「ネットワーク」をクリックします。
- (2) 「ネットワークの追加」をクリックします。
- (3) 「仮想マシン」を選択し、「次へ」をクリックします。
- (4) 「仮想スイッチの作成」を選択し、「次へ」をクリックします。
- (5) ポットグループのプロパティの「ネットワーク ラベル」で任意の名前を入力し、「次へ」をクリックします。
- (6) 設定を確認し、「終了」をクリックします。

VMkernelポートを追加する

二つ目の仮想スイッチを VMotion で利用する場合は、「VMkernel ポート」を追加してください。

- (1) 二つ目の仮想スイッチの「プロパティ」をクリックし、「ポート」タブを選択します。
- (2) 「追加」をクリックし、「VMkernel」を選択し、「次へ」をクリックします。
- (3) ポットグループのプロパティの「ネットワーク ラベル」で任意の名前を入力し、「このポートグループを VMotion で使用」をチェック入れて、「次へ」をクリックします。

- (4) 「次の IP 設定を使用」を選択し、「IP アドレス」と「サブネットマスク」を記入し、「次へ」をクリックします。
- (5) 設定を確認し、「終了」をクリックします。
- (6) 「デフォルトゲートウェイが設定されていません。このネットワークインターフェースを使用するには、デフォルトゲートウェイの設定が必要になることがあります。今すぐ構成しますか?」という警告が出力された場合、「いいえ」をクリックします。
- (7) 仮想スイッチの「プロパティ」の「閉じる」をクリックします。

サービスコンソールポートを追加する

二つ目の仮想スイッチを Service Console で利用する場合は、「サービスコンソールポート」を追加してください。

注:「VMkernel 用」と「Service Console 用」の IP は同じ IP に設定できません。

- (1) 二つ目の仮想スイッチの「プロパティ」をクリックし、「ポート」タブを選択します。
- (2) 「追加」をクリックし、「サービス コンソール」を選択し、「次へ」をクリックします。
- (3) ポットグループのプロパティの「ネットワーク ラベル」で任意の名前を入力し、「次へ」をクリックします。
- (4) 「次の IP 設定を使用」を選択し、「IP アドレス」と「サブネットマスク」を記入し、「次へ」をクリックします。
- (5) 設定を確認し、「終了」をクリックします。
- (6) 仮想スイッチの「プロパティ」の「閉じる」をクリックします。

仮想マシンの作成

構築するクラスタシステムにあわせて仮想マシンを作成してください。下記に仮想マシンの作成例を記載しますので参考にしてください。

仮想マシンの作成手順の詳細については、VMware 社が用意している基本システム管理ガイドを参照してください。

- VMware サポートのリソース ドキュメント
<http://www.vmware.com/jp/support/pubs/>

共有ディスクを利用する仮想マシンを作成する¹

- (1) 「新規仮想マシンの作成」を実行します。
- (2) 「構成」で「標準」を選択し、「次へ」をクリックします。
- (3) 任意の名前を入力し、「次へ」をクリックします。
- (4) 適切なホストを選択し、「次へ」をクリックします。
- (5) 適切なデータストアを選択し、「次へ」をクリックします。
- (6) インストールするゲスト OS を選択し、「次へ」をクリックします。
- (7) 任意の仮想ディスクサイズを指定し、「次へ」をクリックします。ここでは OS をインストールするディスクのサイズを指定します。クラスタシステムの共有ディスク用の設定は後ほど行います。
- (8) 設定を確認し、「終了」をクリックします。
- (9) 追加した仮想マシンを右クリックし、「設定の編集」を実行する。
- (10) 「ハードウェア」タブの「追加」をクリックします。
- (11) 二つ目の VM Network を追加する必要がある場合、「イーサネット アダプタ」を選択し、「次へ」をクリックします。
- (12) 「アダプタ タイプ」を設定し、「ネットワーク ラベル」で「二つ目の仮想スイッチのネットワーク ラベル」を選択し、「次へ」をクリックします。
- (13) 設定を確認し、「終了」をクリックします。
- (14) 「ハードディスク」を選択し、「次へ」をクリックします。
- (15) 「Raw デバイスのマッピング」を選択し、「次へ」をクリックします。
なお、2 台目以降の仮想マシンの場合は、「既存の仮想ディスクを使用」を選択し、「次へ」をクリック、1 台目で作成した「*.vmdk」ファイルを選択してください。
- (16) 利用するディスク LUN を指定し、「次へ」をクリックします。
- (17) 共有ディスクの「SCSI コントローラ」を分けるため、「仮想デバイスノード」の設定で、「SCSI (X:Y)」または「IDE (X:Y)」の X の値が OS インストール用のディスクとは異なるものを選択し、「次へ」をクリックします。
- (18) 設定を確認し、「終了」をクリックします。

¹ 仮想マシンに共有ディスクを設定すると、VMotionが利用できなくなります。詳しくは『VMotionを併用する場合の注意事項(23ページ)』を参照してください。

- (19) ハードウェアのリストに「ハードディスク」以外に「SCSI コントローラ」も追加されていることを確認してください。もし「SCSI コントローラ」が追加されていない場合は、追加した「ハードディスク」の「仮想デバイスノード」の設定が間違っている可能性があります。
- (20) 「SCSI コントローラ」を選択すると表示される「SCSI バスの共有」を「物理」または「仮想」に変更し、「OK」をクリックしてください。
- (21) 以上で共有ディスクを利用する仮想マシンが作成されました。OS をインストールしていない場合はインストールしてください。

ミラーディスクを利用する仮想マシンを作成する

- (1) 『共有ディスクを利用する仮想マシンを作成する』の (14)まで同様に作業します。
- (2) 「新規仮想ディスクを作成」を選択し、「次へ」をクリックします。2 台目以降も「新規仮想ディスクを作成」を選択してください。
- (3) 「新規仮想ディスクを作成」を選択した場合は仮想ディスクの容量を指定し、「次へ」をクリックします。全ての仮想マシンで同じ容量にすることを推奨します。
- (4) そのまま、「次へ」をクリックします。
- (5) 設定を確認し、「終了」をクリックします。
- (6) 以上でミラーディスクを利用する仮想マシンが作成されました。OS をインストールしていない場合はインストールしてください。

第 5 章 クラスタ環境構築手順

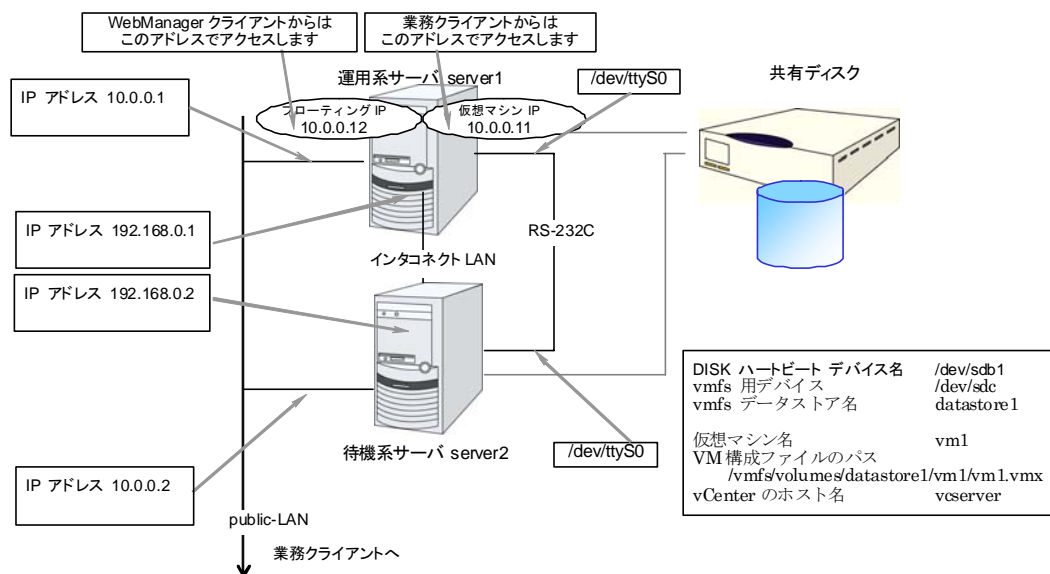
ホストOS間クラスタの構築

ホストOS間クラスタ設定例

ホスト OS 間クラスタを構築する場合、ホスト OS の CLUSTERPRO のバージョンによって使用可能な仮想マシンの制御方法が異なります。

表 2：ホスト OS 間クラスタの構築対応表

	仮想マシンの制御方法	CLUSTERPRO X2.1 for Linux	CLUSTERPRO X3.0 for Linux
パターン 1	EXEC リソースとカスタムモニタリソースを使用する	○	○
パターン 2	仮想マシンリソースと仮想マシンモニタリソースを使用する	×	○



2 ノードの構成設定例

設定対象	設定パラメータ	設定値 (パターン1)	設定値 (パターン2)
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	2	←
	モニタ リソース数	5	←
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	←
	ディスク ハートビート数	1	←

設定対象	設定パラメータ	設定値 (パターン1)	設定値 (パターン2)
1 台目のサーバの情報 (マスタ サーバ)	サーバ名*1	server1	←
	インタコネク트의 IP アドレス (専用)	192.168.0.1	←
	インタコネク트의 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	←
	ディスク ハートビートデバイス	/dev/sdb1	←
2 台目のサーバの情報	サーバ名*1	server2	←
	インタコネク트의 IP アドレス (専用)	192.168.0.2	←
	インタコネク트의 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	←
	ディスク ハートビートデバイス	/dev/sdb1	←
管理用のグループ (WebManager 用)	タイプ	フェイルオーバー	←
	グループ名	ManagementGroup	←
	起動サーバ	全てのサーバ	←
	グループ リソース数	1	←
管理用グループのグループリソース *2	タイプ	floating ip resource	←
	グループ リソース名	ManagementIP	←
	IPアドレス	10.0.0.12	←
業務用のグループ	タイプ	フェイルオーバー	仮想マシン
	グループ名	failover-vm	virtualmachine1
	起動サーバ	全てのサーバ	←
	グループ リソース数	1	←
1 つ目のグループリソース (仮想マシン制御用リソース) *3	タイプ	execute resource	virtual machine resource
	グループ リソース名	exec-vm	vm1
	スクリプト	標準スクリプト	—
	仮想マシンの種類	—	vSphere
	仮想マシン名	—	esx_vm1
	VM構成ファイルのパス	—	/vmfs/volumes/datastore1/vm1/vm1.vmx
	vCenterを使用する	—	オン
	vCenterのホスト名	—	vcserver

設定対象	設定パラメータ	設定値 (パターン1)	設定値 (パターン2)
	リクエストタイムアウト	—	120
	仮想マシン起動待ち時間	—	30
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	タイプ	diskw monitor	←
	モニタ リソース名	diskw1	←
	監視対象	/dev/sdc	←
	監視方法	TUR	←
	異常検出時	クラスタ デーモン停止と OS シャットダウン	←
3 つ目のモニタリソース	タイプ	NIC Link Up/Down monitor	←
	モニタ リソース名	miiw1	←
	監視対象	vmnic0 (Public LANの仮想スイッチインタフェース)	←
	異常検出時	“ManagementGroup” グループのフェイルオーバー	←
4 つ目のモニタリソース	タイプ	NIC Link Up/Down monitor	←
	モニタ リソース名	miiw2	←
	監視対象	vmnic0 (Public LANの仮想スイッチインタフェース)	←
	異常検出時	“failover-vm” グループのフェイルオーバー	←
5 つ目のモニタリソース (仮想マシン監視用モニタ) *3*4	タイプ	Custom monitor	virtual machine monitor
	モニタ リソース名	genw-vm	vmw1
	スクリプト	標準スクリプト	—
	仮想マシンリソース	—	vm1
	外部マイグレーション発生時の待ち時間	—	15(秒)
	監視タイミング	活性時	常時
	対象リソース	exec-vm	—
	異常検出時	“failover-vm” グループのフェイルオーバー	“vm1”リソースのリトライ(3回)後フェイルオーバー(1回)

- *1: 「ホスト名」は原則として FQDN 形式からドメイン名を除いたショートネームのことを指します。
- *2: WebManager に接続するフローティング IP を 用意します。この IP により、障害発生時も Web ブラウザから動作している方のサーバが実行する WebManager にアクセスできます。
- *3: 網掛けした項目が仮想マシンに関する設定項目です。
- *4: X 3.0 利用時は”virtual machine resource”を設定すると自動的に作成されます。

ホストOS間クラスタを構築する

- (1) 『CLUSTERPRO Xインストール&設定ガイド』¹に従い、各ホストOSにCLUSTERPRO Xをインストールしてください。インストールするRPMはx86_64版になります。
また、ユーザ空間モニタリソース及びシャットダウンストール監視を利用する場合は、監視方法を「keepalive」に設定してください。
- (2) CLUSTERPRO Builder を起動してください。
- (3) 『CLUSTERPRO Xインストール&設定ガイド』に従い、「ホストOS間クラスタ設定例 (29ページ)」の例を参考にクラスタを構築します。
- (4) (3)の手順の中で、仮想マシンリソースについては、本書の「仮想マシン制御用リソースの設定 (40ページ)」を参考に設定します。
- (5) (3)の手順の中で、仮想マシンモニタについては、本書の「仮想マシン監視用モニタの設定 (46ページ)」を参考に設定します。
- (6) その他に必要な設定があれば追加します。
- (8) 設定を反映します。Builder のメニューの [ファイル] から [設定の反映] または [情報ファイルのアップロード] を選択し、構成情報をアップロードしてください。
- (9) クラスタを起動させる前に、フェイルオーバー対象とする仮想マシンが存在する VMware ESX の Service Console 上で下記のコマンドを実行し、仮想マシンを unregister してください。unregister 時には仮想マシンの電源状態は OFF にする必要があります。
`vmware-cmd -s unregister vm_path`
- (10) CLUSTERPRO を起動し、仮想マシンが正常に起動することを確認してください。
- (11) 以上でクラスタの構築は終了です。

¹ <http://www.nec.co.jp/clusterpro/> から入手可能です。

ホストOS間クラスタの動作を確認する

- (1) WebManager または clpcl コマンドでクラスタを起動してください。
- (2) WebManager または clpgrp コマンドで仮想マシングループを起動してください。グループが起動しているサーバで、ゲスト OS が動作していることを確認してください。
- (3) WebManager または clpgrp コマンドで仮想マシングループを移動してください。グループの移動先のサーバで、ゲスト OS が動作していることを確認してください。
- (4) WebManager または clpgrp コマンドで仮想マシングループのライブマイグレーションを実行してください。グループの移動先のサーバで、ゲスト OS が動作していることを確認してください。
- (5) WebManager または clpdown コマンドで、仮想マシングループが起動している物理サーバのシャットダウンまたはリブートを行ってください。この時、グループが他のサーバへ移動し、ゲスト OS が動作していることを確認してください。
- (6) ゲスト OS をシャットダウンすると、仮想マシン監視用モニタが異常を検出し、回復対象の再活性またはフェイルオーバを行うことを確認してください。また、フェイルオーバ後にゲスト OS が再起動されていることを確認してください。
- (7) CLUSTERPRO 以外から物理サーバをシャットダウンした場合に、他方のサーバで相手サーバの停止を検出し、仮想マシングループを起動され、ゲスト OS が再起動されることを確認してください。
- (8) 上記に加え、『CLUSTERPRO X インストール & 設定ガイド 第 8 章 動作チェックを行う 動作確認テストを行う』に記載されている項目を適宜実施してください。

ホストOS間クラスタでゲスト-ホスト連携を利用する

- (1) 『ホストOS間クラスタを構築する(33ページ)』と同様にホストOS間クラスタを構築してください。
- (2) 『CLUSTERPRO X SingleServerSafeインストール & 設定ガイド』¹に従い、ゲスト上にCLUSTERPRO X SingleServerSafeをインストールしてください。
- (3) CLUSTERPRO Builder でゲスト側のクラスタ構成情報を編集します。スクリプトの内容は付録を参照してください。
 - A) 監視対象のモニタリソース (pid モニタリソース、appli モニタリソース、oracle モニタリソースなど) を追加します。
 - B) モニタリソースの “最終動作前スクリプトを実行する” を有効にして、[設定]ボタンを選択します。
 - C) ゲストOSがLinuxの場合、[置換] ボタンでpreaction.shの内容を vmpreaction.shで置き換えます。
ゲスト OS が Windows の 場 合 、 [置 換] ボ タ ン で preaction.bat の 内 容 を vmpreaction.batで置き換えます。
 - D) その他の設定は適宜行ってください。
- (4) CLUSTERPRO Builder で作成した構成情報をアップロードします。Builder のメニューの [ファイル] から [情報ファイルのアップロード] を選択し、構成情報をアップロードしてください。

¹ http://www.nec.co.jp/clusterpro/sss/sss_index.html から入手可能です。

ゲストOS間クラスタの構築

ゲストOS間クラスタを構築する

- (1) 仮想マシンを作成していない場合は、「仮想マシンの作成 (27ページ)」を参考に仮想マシンを作成してください。
- (2) CLUSTERPRO がサポートするゲスト OS を仮想マシンにインストールしてください。
- (3) 『CLUSTERPRO X インストール & 設定ガイド』に従い、ゲスト OS に CLUSTERPRO をインストールしてください。
- (4) 『CLUSTERPRO X インストール & 設定ガイド』に従い、CLUSTERPRO Builder でクラスタを構築してください。
- (5) 設定を反映します。Builder のメニューの [ファイル] から [設定の反映] または [情報ファイルのアップロード] を選択し、構成情報をアップロードしてください。

ゲストOS間クラスタの動作を確認する

- (1) WebManager または clpcl コマンドでクラスタを起動してください。
- (2) WebManager または clpgrp コマンドでフェイルオーバーグループを移動してください。フェイルオーバーグループの移動先のサーバで、フェイルオーバーグループが起動していることを WebManager または clpstat コマンドで確認してください。
- (3) WebManager または clpdown コマンドで、フェイルオーバーグループが起動している仮想マシンのシャットダウンまたはリブートを行ってください。この時、フェイルオーバーグループが他のサーバで起動していることを WebManager または clpstat コマンドで確認してください。
- (4) CLUSTERPRO 以外から物理サーバの電源を落とした場合に、他方のサーバで相手サーバの停止を検出し、フェイルオーバーグループを起動していることを WebManager または clpstat コマンドで確認してください。
- (5) 上記に加え、『CLUSTERPRO X インストール & 設定ガイド 第 8 章 動作チェックを行う 動作確認テストを行う』に記載されている項目を適宜実施してください。

物理サーバー仮想マシン間クラスタの構築

物理サーバー仮想マシン間クラスタを構築する

- (1) 仮想マシンを作成していない場合は、「仮想マシンの作成 (27ページ)」を参考に仮想マシンを作成してください。
- (2) CLUSTERPRO がサポートするゲスト OS を仮想マシンにインストールしてください。
- (3) 『CLUSTERPRO X インストール & 設定ガイド』に従い、物理サーバ OS 及びゲスト OS に CLUSTERPRO をインストールしてください。
- (4) 『CLUSTERPRO X インストール & 設定ガイド』に従い、CLUSTERPRO Builder でクラスタを構築してください。
- (5) 設定を反映します。Builder のメニューの [ファイル] から [設定の反映] または [情報ファイルのアップロード] を選択し、構成情報をアップロードしてください。

物理サーバー仮想マシン間クラスタの動作を確認する

- (1) WebManager または clpcl コマンドでクラスタを起動してください。
- (2) WebManager または clpgrp コマンドでフェイルオーバーグループを移動してください。フェイルオーバーグループの移動先のサーバで、フェイルオーバーグループが起動していることを WebManager または clpstat コマンドで確認してください。
- (3) WebManager または clpdown コマンドで、フェイルオーバーグループが起動しているサーバのシャットダウンまたはリブートを行ってください。この時、フェイルオーバーグループが他のサーバで起動していることを WebManager または clpstat コマンドで確認してください。
- (4) CLUSTERPRO 以外から物理サーバの電源を落とした場合に、他方のサーバで相手サーバの停止を検出し、フェイルオーバーグループを起動していることを WebManager または clpstat コマンドで確認してください。
- (5) 上記に加え、『CLUSTERPRO X インストール & 設定ガイド 第 8 章 動作チェックを行う 動作確認テストを行う』に記載されている項目を適宜実施してください。

仮想マシン制御用リソースの設定

EXECリソースを利用する場合

- (1) CLUSTERPRO Builder の画面で、ツリービューのフェイルオーバーグループ名 (failover-vm) を右クリックし、[リソースの追加] をクリックしてください。
- (2) 仮想マシン制御用 EXEC リソースを登録します。[タイプ] から [execute resource] を選択してください。exec リソースに名前 (exec-vm) を設定し、[次へ]をクリックしてください。

リソースの定義

タイプ(T)

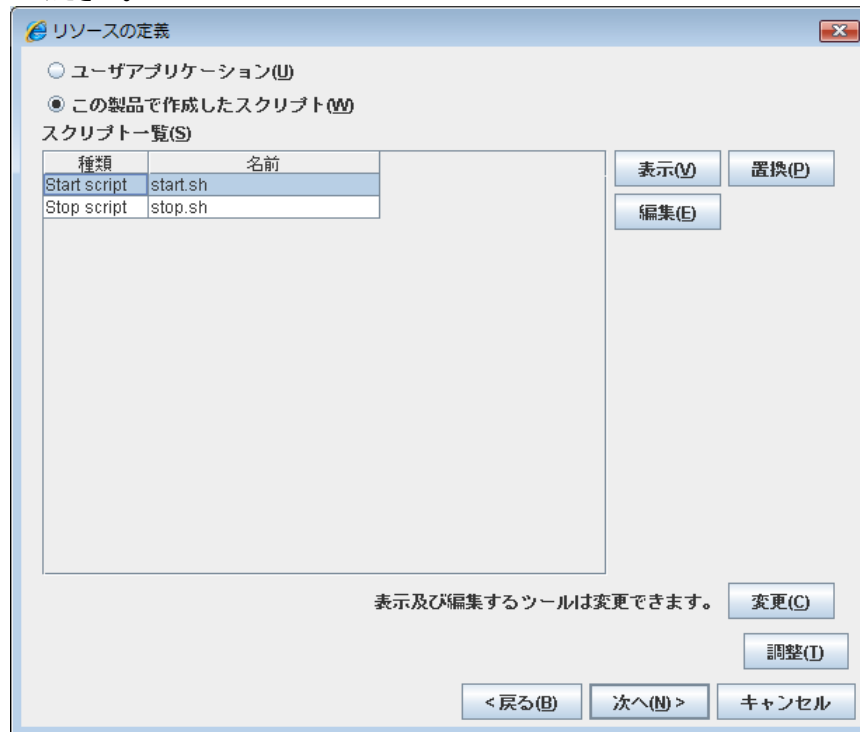
名前(N)

コメント(C)

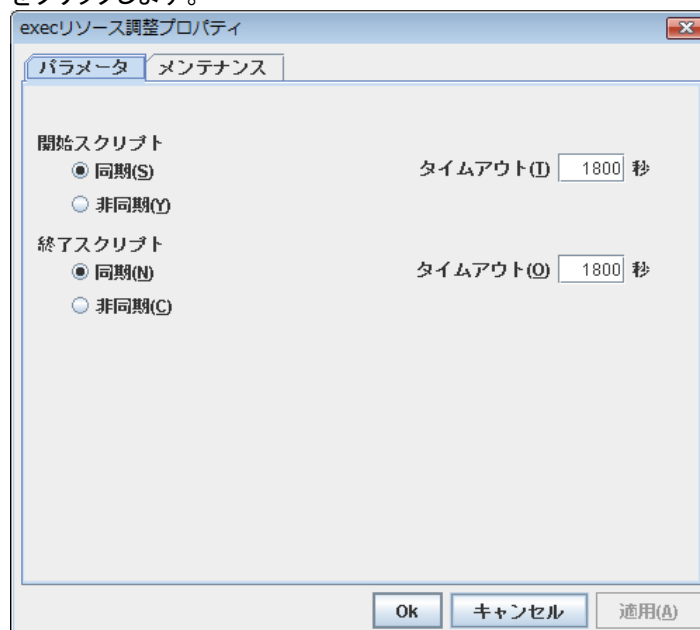
継続するには[次へ]をクリックしてください。

<戻る(B) 次へ(N)> キャンセル

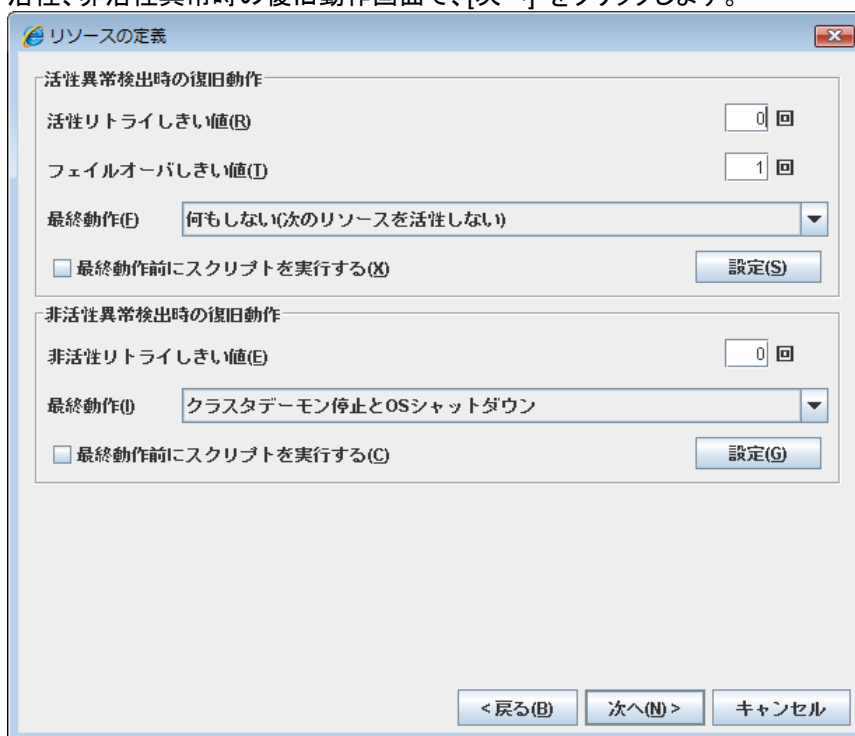
- (3) [Start script] を選択し、[置換] をクリックしてください。ファイル選択画面が表示されるので、本書の サンプルスクリプト vmpower.start.pl (53ページ) を選択し、start.shを置換してください。



- (4) [Stop script] を選択し、[置換]をクリックしてください。ファイル選択画面が表示されるので、本書の サンプルスクリプト vmpower.stop.pl (58ページ) を選択し、stop.shを置換してください。
- (5) [調整] をクリックし、非大域ゾーンの活性、非活性タイムアウトを適宜変更します。[次へ] をクリックします。



(6) 活性、非活性異常時の復旧動作画面で、[次へ] をクリックします。

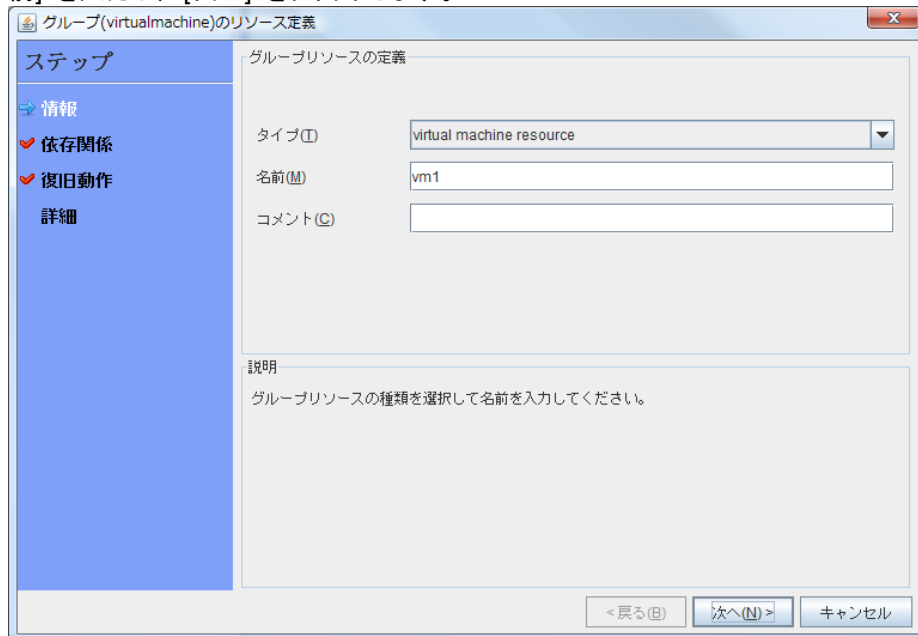


(7) リソースの依存関係画面で、[完了] をクリックします。

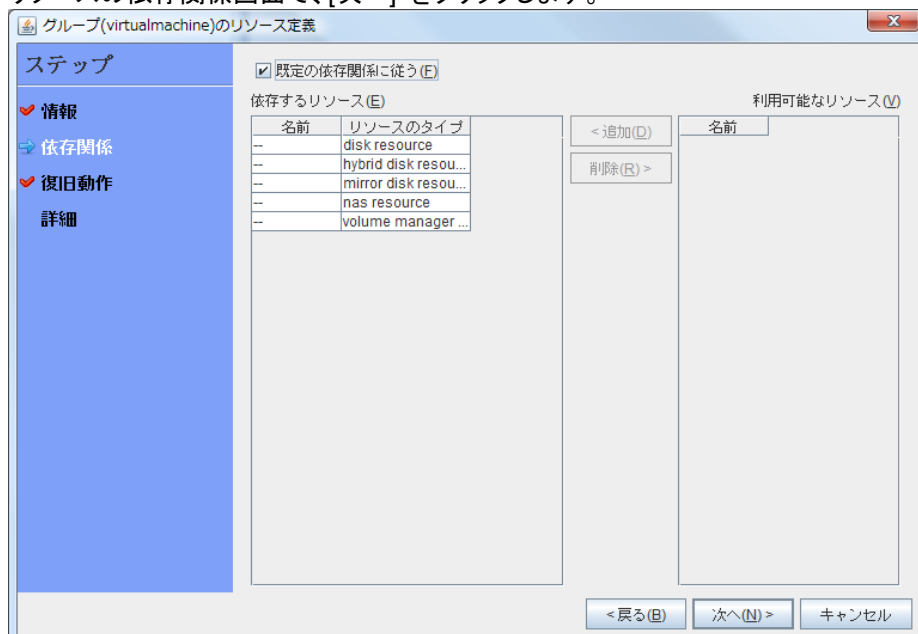


仮想マシンリソースを利用する場合(X3.0以降)

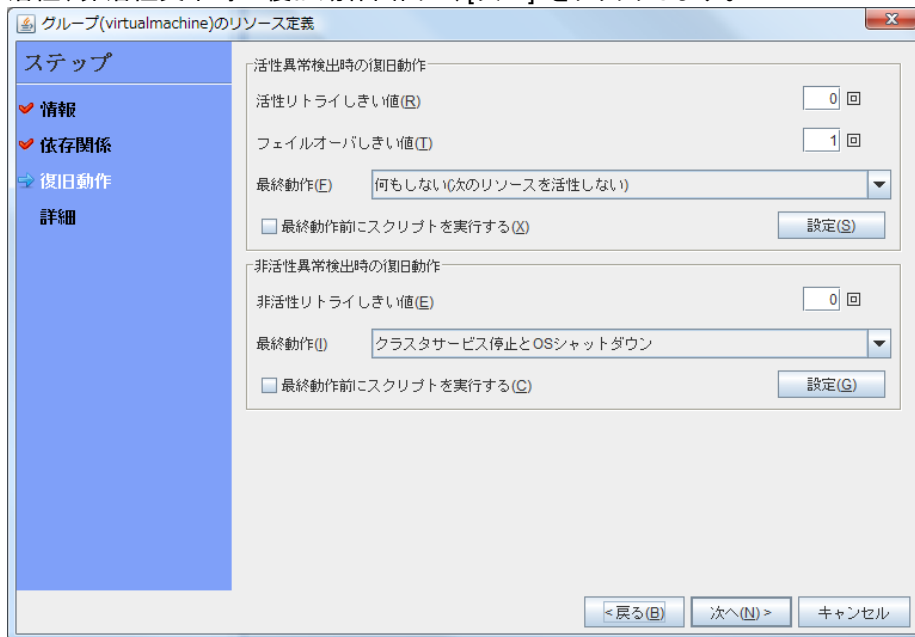
- (1) CLUSTERPRO Builder の画面で、ツリービューの仮想マシングループ名 (virtualmachine1) を右クリックし、[リソースの追加] をクリックしてください。
- (2) 仮想マシンリソースを登録します。[タイプ] に [virtual machine resource] を選択、[名前] を入力し、[次へ] をクリックします。



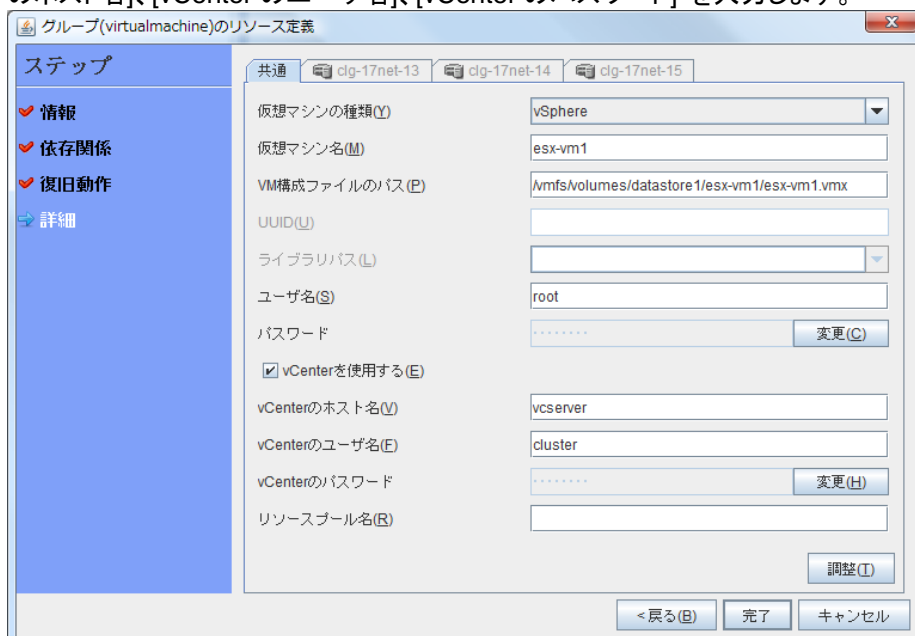
- (3) リソースの依存関係画面で、[次へ] をクリックします。



- (4) 活性、非活性異常時の復旧動作画面で、[次へ] をクリックします。



- (5) [仮想マシンの種類] に [vSphere] を選択し、[仮想マシン名]、[VM 構成ファイルのパス]、[ユーザ名]、[パスワード] を設定します。
vCenter を使用しない場合は、[vCenter を使用する] チェックボックスがオフであることを確認します。
vCenter を使用する場合は、[vCenter を使用する] チェックボックスをオンにし、[vCenter のホスト名]、[vCenter のユーザ名]、[vCenter のパスワード] を入力します。



- (6) [調整] ボタンをクリックして調整プロパティを表示し、[リクエストタイムアウト]、[仮想マシン起動待ち時間]、[仮想マシン停止待ち時間] を入力し、[OK] をクリックします。

仮想マシンリソース調整プロパティ

パラメータ

リクエストタイムアウト(B)	120	秒
仮想マシン起動待ち時間(V)	30	秒
仮想マシン停止待ち時間(I)	240	秒

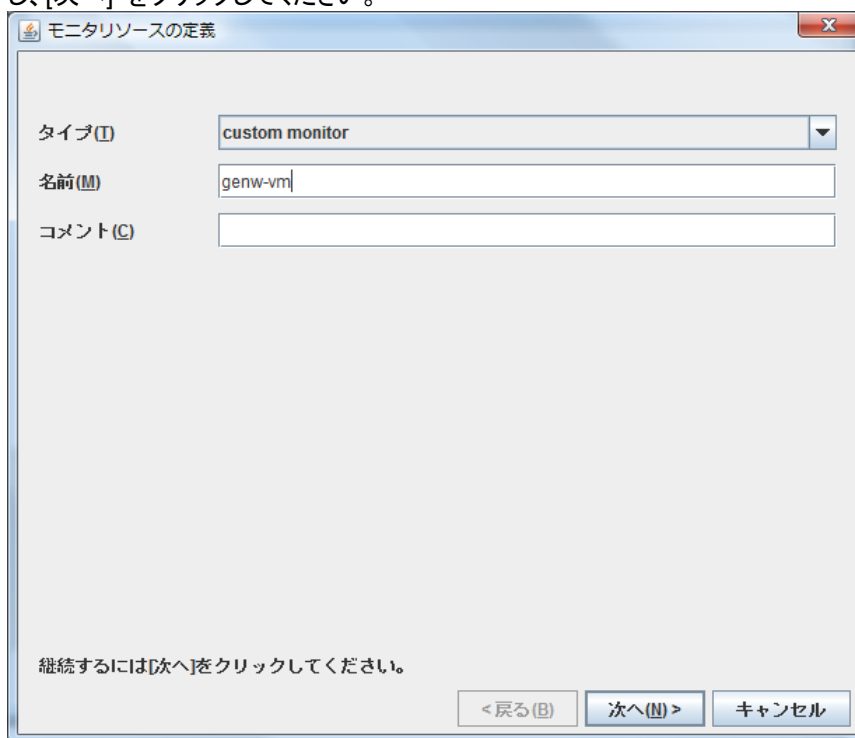
既定値(I)

OK キャンセル 適用(A)

仮想マシン監視用モニタの設定

カスタムモニタリソースを利用する場合

- (1) CLUSTERPRO Builder の画面で、ツリービューの [Monitors] を右クリックし、[モニタリソースの追加] をクリックしてください。
- (2) 仮想マシン監視用カスタムモニタリソースを登録します。[タイプ] から [custom monitor resource] を選択してください。カスタムモニタリソースに任意の名前 (genw-vm) を設定し、[次へ] をクリックしてください。



- (3) [置換] をクリックし、genw.shを本書の サンプルスクリプト clpvmmmon.pl (62ページ) で置換し、[監視タイプ] が[同期]になっていることを確認したら [次へ] をクリックしてください。

モニタリソースの定義

ユーザアプリケーション(U)
 この製品で作成したスクリプト(W)

ファイル(F)

監視タイプ
 同期(S) 非同期(Y)

ログ出力先(L)

正常な戻り値(M)

表示及び編集するツールは変更できます。

- (4) [監視タイミング] に [活性時] を選択し、対象リソースに仮想マシン制御用リソース [exec-vm] を選択し、[タイムアウト]、[インターバル]、[リトライ回数] を環境に合わせて変更したら [次へ] をクリックします。

モニタリソースの定義

インターバル(I) 秒
タイムアウト(T) 秒
リトライ回数(R) 回
監視開始待ち時間(S) 秒

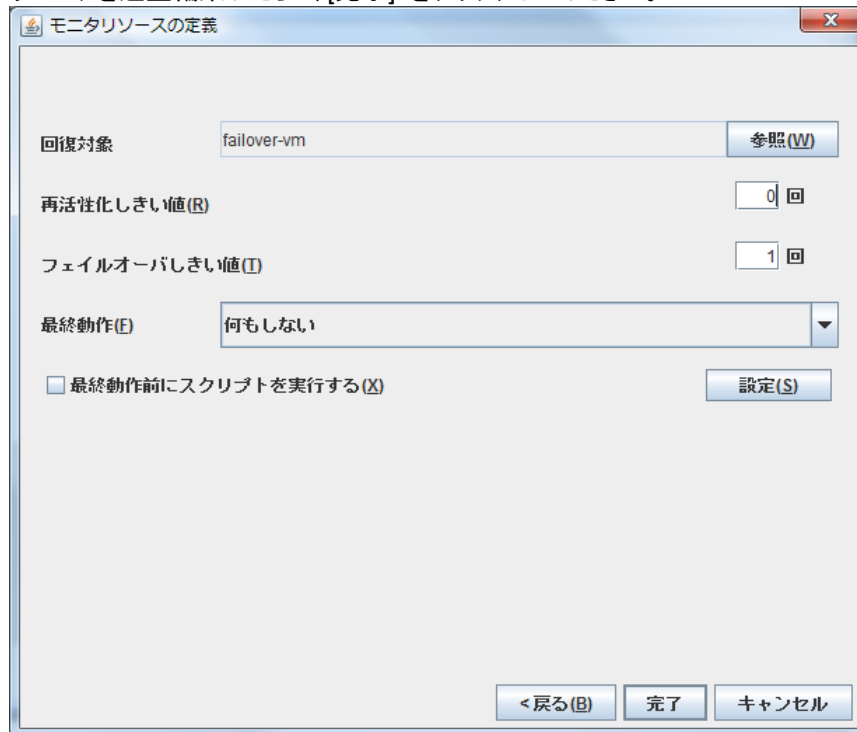
監視タイミング
 常時(L)
 活性時(C)

対象リソース

nice値(E)

監視を行うサーバを選択する

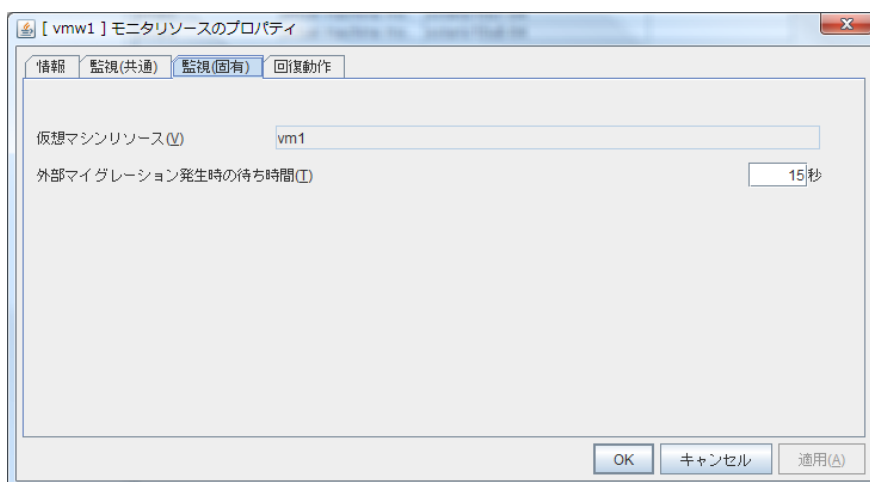
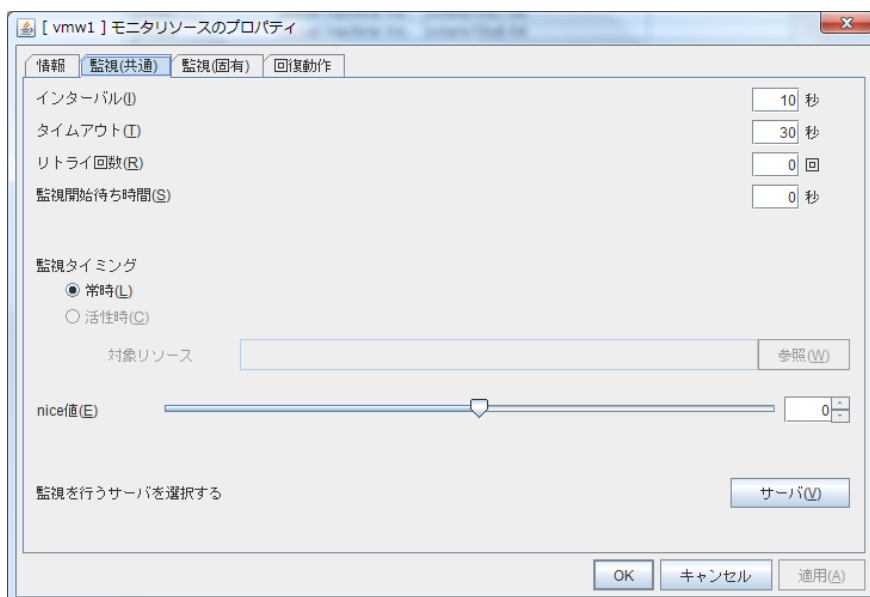
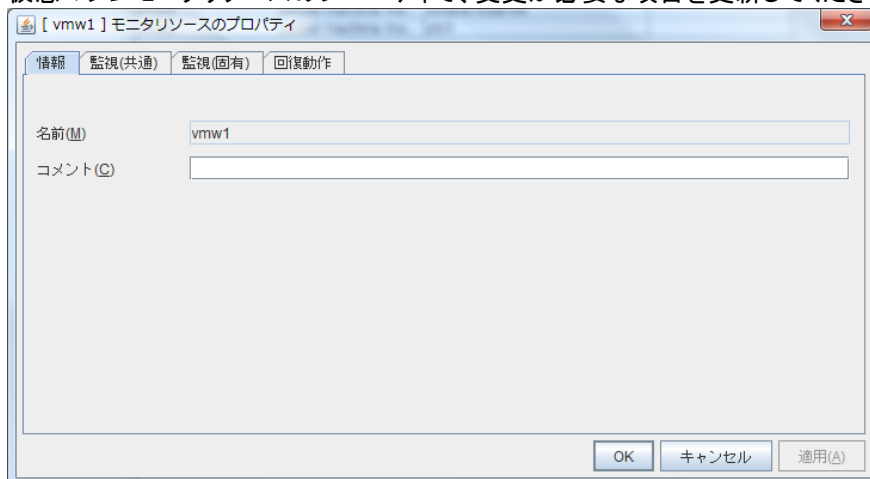
- (5) [回復対象]として仮想マシン制御用リソース (exec-vm) が属するフェイルオーバーグループ名 (failover-vm) を選択し、[OK] をクリックしてください。[再活性化しきい値] などのパラメータを適宜編集したら¹、[完了] をクリックしてください。

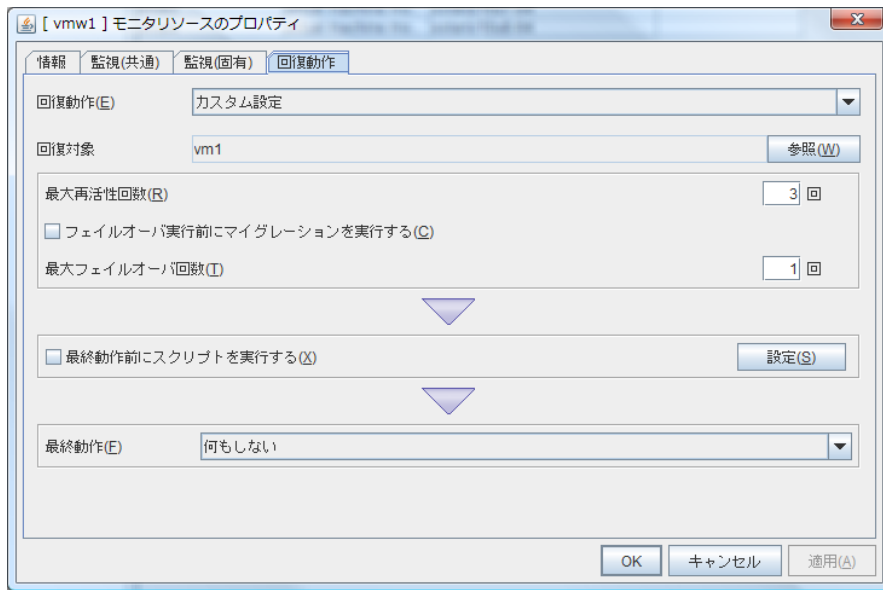


¹ 『CLUSTERPRO X リファレンスガイド 第 6 章 モニタリソースの詳細』を参考に、各パラメータを調整してください。

仮想マシンモニタリソースを利用する場合(X3.0以降)

- (1) 仮想マシンリソースが作成されると自動的に仮想マシンモニタリソースが作成されます。仮想マシンモニタリソースのプロパティで、変更が必要な項目を更新してください。





VMware HAとの連携の設定

ホスト上のCLUSTERPRO X SingleServerSafeとVMware HAを連携させる

- (1) 必要な数の仮想マシンを作成します。
- (2) vCenter から VMware HA を構築します。
 - A) インベントリ上のデータセンターを右クリックして、[新規クラスター]を選択し、クラスターを作成します。
 - B) インベントリ上のホストを作成したクラスターへドラッグ & ドロップし、ホストをクラスターへ参加させます。ゲスト OS のフェイルオーバー先にしたいホストを全てクラスターに参加させてください。
 - C) 作成したクラスターを右クリックして、[設定の編集]を選択し、クラスターの設定画面を表示します。
 - D) クラスターの設定画面上の左ペイン[クラスター機能]を選択し、[VMware HA の有効化]をチェックします。
 - E) VMware HAによる仮想マシンの監視が必要であれば、クラスターの設定画面上の左ペイン[VMware HA]->[仮想マシンの監視]で[仮想マシンの監視ステータス]欄の[仮想マシンの監視を有効にする]をチェックします。¹
- (3) 『CLUSTERPRO X インストール & 設定ガイド』に従い、VMware ESX の Service Console 上に CLUSTERPRO X SingleServerSafe (SSS) をインストールします。
- (4) インストールした SSS に、ゲスト OS が格納される共有ディスクを監視するよう diskw を設定します。diskw の監視タイプには「READ(O_DIRECT)」、最終動作には「OS シャットダウン」を選択してください。また、ユーザ空間モニタリソース及びシャットダウンストール監視を利用する場合は、監視方法を「keepalive」に設定してください。
 - A) 監視対象のモニタリソース (NIC Link up/down モニタリソースなど) を追加します。
 - B) モニタリソースの “最終動作前スクリプトを実行する” を有効にして、[設定]ボタンを選択します。
 - C) ゲストOSがLinuxの場合、[置換] ボタンでpreaction.shの内容を vmpreaction.shで置き換えます。
ゲスト OS が Windows の場合、[置換] ボタンで preaction.bat の内容を vmpreaction.batで置き換えます。
- (5) 任意でその他の監視を設定します。

¹ 本設定を有効にするには、ゲスト上へVMware Toolsをインストールし動作させる必要があります。

ゲスト上のCLUSTERPRO X及びホスト上のCLUSTERPRO X SingleServerSafeとVMware HAを連携させる

- (1) 『ホスト上のCLUSTERPRO X SingleServerSafeとVMware HAを連携させる (51ページ)』の (2)まで実行します。
- (2) 『CLUSTERPRO X インストール & 設定ガイド』に従い、ゲスト OS に CLUSTERPRO をインストールしてください。
- (3) 『CLUSTERPRO X インストール & 設定ガイド』に従い、CLUSTERPRO Builder でクラスタを構築してください。ユーザ空間モニタリソース及びシャットダウンストール監視を利用する場合は、監視方法を「keepalive」に設定してください。
- (4) 『ホスト上のCLUSTERPRO X SingleServerSafeとVMware HAを連携させる (51ページ)』の (3)以降を実行します。

付録 A サンプルスクリプト

ホスト OS 間クラスタを構築するために必要なスクリプトのサンプルです。のスクリプト内でアンダーラインになっている箇所は、お使いになられる環境に合わせて適宜編集してください。

vmpower.start.pl

仮想マシンを起動するためのスクリプトです。アンダーラインの箇所を適宜編集してお使いください。

```
#!/usr/bin/perl -w

#
# Script for power on the Virtual Machine
#

use strict;

#-----
# Configuration
#-----
# The path to VM configuration file. This must be absolute UUID-based path.
my $cfg_path = "/vmfs/volumes/<datastore-uuid>/vm1/vm1.vmx";

# The interval to check the vm status. (second)
my $interval = 1;
# The maximum count to check the vm status.
my $max_cnt = 100;
# The timeout to start the vm. (second)
my $start_to = 10;
#-----
my $vmname = $cfg_path; # VMname to be outputted on log.
$vmname =~ s/^(.*\%/)(.*)\%\.vmx/$2/;

# VM operation command path
my $vmcmd = "/usr/bin/vmware-cmd";

# VM execution state map
my %state = (
    "VM_EXECUTION_STATE_ON" => "on",
    "VM_EXECUTION_STATE_OFF" => "off",
    "VM_EXECUTION_STATE_SUSPENDED" => "suspended",
    "VM_EXECUTION_STATE_STUCK" => "stuck",
    "VM_EXECUTION_STATE_UNKNOWN" => "unknown"
);

#-----
# Main
#-----
exit 1 if (!&RegisterVm());

if (&IsPoweredOn()){
    exit 0;
}
```

```

else{
  if (&PowerOn()){
    if (&WaitPoweredOnDone()){
      exit 0;
    }
    else{
      exit 1;
    }
  }
  else{
    exit 1;
  }
}

#-----
# Functions
#-----

sub RegisterVm{
  my $svop = "-s register";
  my $vmcmd_list = $vmcmd . " -l";
  my @vmlist = ` $vmcmd_list `;
  my $ret = 0;
  my $opn_ret;
  my $line;

  foreach (@vmlist){
    if (/ $cfg_path/){
      &Log("[I] [$vmname] at localhost already registered.¥n");
      return 1;
    }
  }

  $opn_ret = open(my $fh, $vmcmd . " " . $svop . " " . $cfg_path . " 2>&1 |");
  if (!$opn_ret){
    &Log("[E] [$vmname] at localhost: $vmcmd $svop could not be executed.¥n");
    return 0;
  }

  $line = <$fh>;
  if (defined($line)){
    &Log("[E] [$vmname] at localhost: Could not register VM: $line¥n");
  }else{
    $ret = 1;
    &Log("[I] [$vmname] at localhost: Registered.¥n");
  }

  close($fh);

  return $ret;
}

#-----
sub IsPoweredOn{

  if (&IsEqualState($state{"VM_EXECUTION_STATE_ON"})){
    return 1;
  }else{
    return 0;
  }
}

```

```

}
}
#-----
sub IsEqualState{
    my $vmop = "getstate";
    my $state = shift;
    my $ret = 0;
    my $opn_ret;
    my $line;

    $opn_ret = open(my $fh, $vmcmd . " " . $cfg_path . " " . $vmop . " 2>&1 |");
    if (!$opn_ret){
        &Log("[E] [$vmname] at localhost: $vmcmd $vmop could not be executed.¥n");
        return 0;
    }

    $line = <$fh>;
    if (defined($line)){
        chomp($line);
        if ($line =~ /^$vmop¥(¥)¥s=¥s(.+)$/) {
            $ret = 1 if ($1 eq $state);
            &Log("[D] [$vmname] at localhost: VM execution state is $1.¥n");
        }else{
            &Log("[E] [$vmname] at localhost: Could not get VM execution state: $line¥n");
        }
    }

    close($fh);

    return $ret;
}
#-----
sub PowerOn{
    my $vmop = "start";
    my $ret = 0;
    my $opn_ret;
    my $line;

    $opn_ret = open(my $fh, $vmcmd . " " . $cfg_path . " " . $vmop . " 2>&1 |");
    if (!$opn_ret){
        &Log("[E] [$vmname] at localhost: $vmcmd $vmop could not be executed.¥n");
        return 0;
    }

    eval {
        local $$SIG{ALRM} = sub { die "timeout" };
        alarm($start_to);
        $line = <$fh>;
        alarm(0);
    };
    alarm(0);

    if ($?) {
        if ($? =~ /timeout/) {
            &Log("[E] [$vmname] at localhost: Cound not start VM: timeout($start_to
second)¥n");

```

```

        if (&IsEqualState($state{"VM_EXECUTION_STATE_STUCK"})) {
            $ret = 1 if (&ResolveVmStuck());
        }
    }
}
else{
    if (defined($line)){
        chomp($line);
        if ($line =~ /^$vmop¥(¥)¥s=¥s(.+)$/){
            if ($1 == 1){
                $ret = 1;
                &Log("[I] [$vmname] at localhost: Started.¥n");
            }else{
                &Log("[E] [$vmname] at localhost: Cound not start VM: $1¥n");
            }
        }else{
            &Log("[E] [$vmname] at localhost: Cound not start VM: $line¥n");

            if (&IsEqualState($state{"VM_EXECUTION_STATE_STUCK"})) {
                $ret = 1 if (&ResolveVmStuck());
            }
        }
    }

    close($fh);
}

return $ret;
}
#-----
sub ResolveVmStuck{
    my $vmop = "answer";
    my $ret = 0;
    my $opn_ret;
    my $line;

    $opn_ret = open(my $fh, "| ". $vmcmd . " " . $cfg_path . " " . $vmop);
    if (!$opn_ret){
        &Log("[E] [$vmname] at localhost: $vmcmd $vmop could not be executed.¥n");
        return 0;
    }

    # Answering "1) I _moved it" to keep vm config.
    print($fh "1¥n");

    close($fh);

    if (&IsEqualState($state{"VM_EXECUTION_STATE_STUCK"})) {
        &Log("[E] [$vmname] at localhost: VM stuck could not be resolved.¥n");
    }else{
        $ret = 1;
        &Log("[I] [$vmname] at localhost: VM stuck is resolved.¥n");
    }

    return $ret;
}
#-----

```

```

sub WaitPoweredOnDone {

    for (my $i = 0; $i < $max_cnt; $i++){
        if (&IsEqualState($state{"VM_EXECUTION_STATE_ON"})){
            &Log("[I] [$vmname] at localhost: Powered on done. ($i)¥n");
            return 1;
        }
        sleep $interval;
    }

    &Log("[E] [$vmname] at localhost: Not powered on done. ($max_cnt)¥n");

    return 0;
}
#-----
sub Log {
    my ($sec, $min, $hour, $mday, $mon, $year, $yday, $yday, $isdst) = localtime(time);
    $year += 1900;
    $mon += 1;
    my $date = sprintf "%d/%02d/%02d %02d:%02d:%02d", $year, $mon, $mday, $hour, $min, $sec;
    print "$date $_[0]";
    return 0;
}

```

vmpower.stop.pl

仮想マシンを停止するためのスクリプトです。アンダーラインの箇所を適宜編集してお使いください。

```
#!/usr/bin/perl -w

#
# Script for power off the Virtual Machine
#

use strict;

#-----
# Configuration
#-----
# The path to VM configuration file. This must be absolute UUID-based path.
my $cfg_path = "/vmfs/volumes/<datastore-uuid>/vm1/vm1.vmx";

# The interval to check the vm status. (second)
my $interval = 5;
# The maximum count to check the vm status.
my $max_cnt = 100;
#-----
my $vmname = $cfg_path; # VMname to be outputted on log.
$vmname =~ s/^(.*\$/)(.*)\$.vmx)/$2/;

# VM operation command path
my $vmcmd = "/usr/bin/vmware-cmd";

# VM execution state map
my %state = (
    "VM_EXECUTION_STATE_ON" => "on",
    "VM_EXECUTION_STATE_OFF" => "off",
    "VM_EXECUTION_STATE_SUSPENDED" => "suspended",
    "VM_EXECUTION_STATE_STUCK" => "stuck",
    "VM_EXECUTION_STATE_UNKNOWN" => "unknown"
);

#-----
# Main
#-----
if (&IsPoweredOn()){
    if (&PowerOff()){
        if (!&WaitPoweredOffDone()){
            exit 1;
        }
    }else{
        exit 1;
    }
}

if (&UnRegisterVm()){
    exit 0;
}else{
```

```

    exit 1;
}

#-----
# Functions
#-----
sub IsPoweredOn{

    if (&IsEqualState($state{"VM_EXECUTION_STATE_ON"})) {
        return 1;
    }else{
        return 0;
    }
}

#-----
sub IsEqualState{
    my $vmop = "getstate";
    my $state = shift;
    my $ret = 0;
    my $opn_ret;
    my $line;

    $opn_ret = open(my $fh, $vmcmd . " " . $cfg_path . " " . $vmop . " 2>&1 |");
    if (!$opn_ret){
        &Log("[E] [$vmname] at localhost: $vmcmd $vmop could not be executed.¥n");
        return 0;
    }

    $line = <$fh>;
    if (defined($line)){
        chomp($line);
        if ($line =~ /^$vmop¥(¥)¥s=¥s(.+)¥/){
            $ret = 1 if ($1 eq $state);
            &Log("[D] [$vmname] at localhost: VM execution state is $1.¥n");
        }else{
            &Log("[E] [$vmname] at localhost: Could not get VM execution state: $line¥n");
        }
    }

    close($fh);

    return $ret;
}

#-----
sub PowerOff{
    my $ret;

    # Soft stop.
    $ret = &PowerOffOpMode("soft");

    # Hard stop if Soft stop failed.
    if (!$ret){
        $ret = &PowerOffOpMode("hard");
    }

    return $ret;
}

```

```

#-----
sub PowerOffOpMode{
    my $vmop = "stop";
    my $powerop_mode = shift;
    my $ret = 0;
    my $opn_ret;
    my $line;

    return 0 if ($powerop_mode !~ /^hard|soft$/);

    $opn_ret = open(my $fh, $vmcmd . " " . $cfg_path . " " . $vmop . " " . $powerop_mode .
" 2>&1 |");
    if (!$opn_ret){
        &Log("[E] [$vmname] at localhost: $vmcmd $vmop $powerop_mode could not be
executed.¥n");
        return 0;
    }

    $line = <$fh>;
    if (defined($line)){
        chomp($line);
        if ($line =~ /^$vmop¥($powerop_mode¥)¥s=¥s(.+)¥/){
            if ($1 == 1){
                $ret = 1;
                &Log("[I] [$vmname] at localhost: Stopped. ($powerop_mode)¥n");
            }else{
                &Log("[E] [$vmname] at localhost: Cound not stop ($powerop_mode) VM: $1¥n");
            }
        }else{
            &Log("[E] [$vmname] at localhost: Cound not stop ($powerop_mode) VM: $line¥n");
        }
    }else{
        if ($powerop_mode eq "soft"){
            $ret = 1;
            &Log("[I] [$vmname] at localhost: Stopped. ($powerop_mode)¥n");
        }
    }

    close($fh);

    return $ret;
}
#-----
sub WaitPoweredOffDone{

    for (my $i = 0; $i < $max_cnt; $i++){
        if (&IsEqualState($state{"VM_EXECUTION_STATE_OFF"})){
            &Log("[I] [$vmname] at localhost: Powered off done. ($i)¥n");
            return 1;
        }
        sleep $interval;
    }

    &Log("[E] [$vmname] at localhost: Not powered off done. ($max_cnt)¥n");

    return 0;
}

```

```

#-----
sub UnRegisterVm{
    my $svop = "-s unregister";
    my $vmcmd_list = $vmcmd . " -l";
    my @vmlist = `vmcmd_list`;
    my $ret = 0;
    my $opn_ret;
    my $flag = 0;
    my $line;

    foreach (@vmlist){
        if (/ $cfg_path/){
            $flag = 1;
        }
    }

    if ($flag == 0){
        &Log("[I] [$vmname] at localhost already unregistered.¥n");
        return 1;
    }else{
        $opn_ret = open(my $fh, $vmcmd . " " . $svop . " " . $cfg_path . " 2>&1 |");
        if (!$opn_ret){
            &Log("[E] [$vmname] at localhost: $vmcmd $svop could not be executed.¥n");
            return 0;
        }

        $line = <$fh>;
        if (defined($line)){
            &Log("[E] [$vmname] at localhost: Could not unregister VM: $line¥n");
        }else{
            $ret = 1;
            &Log("[I] [$vmname] at localhost: Unregistered.¥n");
        }

        close($fh);
    }

    return $ret;
}
#-----
sub Log{
    my ($sec, $min, $hour, $mday, $mon, $year, $wday, $yday, $isdst) = localtime(time);
    $year += 1900;
    $mon += 1;
    my $date = sprintf "%d/%02d/%02d %02d:%02d:%02d", $year, $mon, $mday, $hour, $min,
    $sec;
    print "$date $_[0]";
    return 0;
}

```

clpvmmmon.pl

仮想マシンの起動状態を確認するためのスクリプトです。アンダーラインの箇所を適宜編集してお使いください。

```
#!/usr/bin/perl -w

#
# Script for monitoring the Virtual Machine
#

use strict;

#-----
# Configuration
#-----
# The path to VM configuration file. This must be absolute UUID-based path.
my $cfg_path = ""/vmfs/volumes/<datastore-uuid>/vm1/vm1.vmx"";

# The interval to check the vm status. (second)
my $interval = 1;
#-----
my $vmname = $cfg_path; # VMname to be outputted on log.
$vmname =~ s/^(.*\./)(.*)\.(.*)\.vmx/$2/;

# VM operation command path
my $vmcmd = ""/usr/bin/vmware-cmd"";

# VM execution state map
my %state = (
    "VM_EXECUTION_STATE_ON" => "on",
    "VM_EXECUTION_STATE_OFF" => "off",
    "VM_EXECUTION_STATE_SUSPENDED" => "suspended",
    "VM_EXECUTION_STATE_STUCK" => "stuck",
    "VM_EXECUTION_STATE_UNKNOWN" => "unknown"
);

#-----
# Main
#-----
system("`ulimit` -s unlimited");
while (&IsHblIncrease()){
}
exit 0;

#-----
# Functions
#-----
sub IsHblIncrease{
    my $last_hb = -1;

    for (my $i = 0; $i < 2; $i++){
        if (&IsPoweredOn()){
            my $hb = &GetHb();
            if ($hb == -1){
```

```

        return 0;
    }

    if ($hb == $last_hb){
        &Log("[I] [$vmname] is stalled.¥n");
        return 0;
    }else{
        $last_hb = $hb;
    }
}
}else{
    &Log("[I] [$vmname] is powered off.¥n");
    return 0;
}
}
sleep $interval;
}

return 1
}
#-----
sub GetHb{
    my $vmop = "getheartbeat";
    my $ret = -1;
    my $opn_ret;
    my $line;

    $opn_ret = open(my $fh, $vmcmd . " " . $cfg_path . " " . $vmop . " 2>&1 |");
    if (!$opn_ret){
        &Log("[E] [$vmname] at localhost: $vmcmd $vmop could not be executed.¥n");
        return -1;
    }

    $line = <$fh>;
    if (defined($line)){
        chomp($line);
        if ($line =~ /^$vmop¥(¥)¥s=¥s(.+)¥/){
            $ret = $1;
#            &Log("[D] [$vmname] at localhost: Got VM heartbeat count $1.¥n");
        }else{
            &Log("[E] [$vmname] at localhost: Could not get VM heartbeat count: $line¥n");
        }
    }

    close($fh);

    return $ret;
}
#-----
sub IsPoweredOn{

    if (&IsEqualState($state{"VM_EXECUTION_STATE_ON"})){
        return 1;
    }else{
        return 0;
    }
}
#-----
sub IsEqualState{

```

```

my $vmop = "getstate";
my $state = shift;
my $ret = 0;
my $opn_ret;
my $line;

$opn_ret = open(my $fh, $vmcmd . " " . $cfg_path . " " . $vmop . " 2>&1 |");
if (!$opn_ret) {
    &Log("[E] [$vmname] at localhost: $vmcmd $vmop could not be executed.¥n");
    return 0;
}

$line = <$fh>;
if (defined($line)) {
    chomp($line);
    if ($line =~ /^$vmop¥(¥)¥s=¥s(.+)$/) {
        $ret = 1 if ($1 eq $state);
#        &Log("[D] [$vmname] at localhost: VM execution state is $1.¥n");
    } else {
        &Log("[E] [$vmname] at localhost: Could not get VM execution state: $line¥n");
    }
}

close($fh);

return $ret;
}
#-----
sub Log{
    my ($sec, $min, $hour, $mday, $mon, $year, $yday, $isdst) = localtime(time);
    $year += 1900;
    $mon += 1;
    my $date = sprintf "%d/%02d/%02d %02d:%02d:%02d", $year, $mon, $mday, $hour, $min,
$sec;
    print "$date $_[0]";
    return 0;
}

```

vmpreaction.sh

ゲスト OS 側の CLUSTERPRO からホスト OS 側の CLUSTERPRO にフェイルオーバー要求を発行する Linux 用スクリプトです。アンダーラインの箇所を適宜編集してお使いください。

1. ホスト OS の CLUSTERPRO のバージョンが X2.1 の場合

```
#!/bin/sh
#*****
#*           preaction.sh           *
#*****
ulimit -s unlimited

echo START
echo $CLP_MONITORNAME

# 仮想マシン制御用グループリソース名
CLPRSC="exec_vm"
# 各ホストのIPアドレス(カンマ区切り)
CLPIP="10.0.0.1,10.0.0.2"

/opt/nec/clusterpro/bin/clptrnreq -t GRP_FAILOVER -r $CLPRSC -h $CLPIP
exit 0
```

2. ホスト OS の CLUSTERPRO のバージョンが X3.0 以降の場合

```
#!/bin/sh
#*****
#*           preaction.sh           *
#*****
ulimit -s unlimited

echo START
echo $CLP_MONITORNAME

# 仮想マシン制御用グループリソース名
CLPRSC="vm1"
# 各ホストのIPアドレス(カンマ区切り)
CLPIP="10.0.0.1,10.0.0.2"

/opt/nec/clusterpro/bin/clprexec --failover -r $CLPRSC -h $CLPIP
exit 0
```

注:ゲスト OS の CLUSTERPRO のバージョンが X2.1 の場合、clprexec コマンドが存在しません。clptrnreq コマンドを使用するか、CLUSTERPRO CD から clprexec コマンドを取得して使用してください。

vmpreaction.bat

ゲスト OS 側の CLUSTERPRO からホスト OS 側の CLUSTERPRO にフェイルオーバー要求を発行する Windows 用バッチファイルです。アンダーラインの箇所を適宜編集してお使いください。

1. ホスト OS の CLUSTERPRO のバージョンが X2.1 の場合

```
rem *****  
rem *           preaction.bat           *  
rem *****  
  
echo START  
echo %CLP_MONITORNAME%  
  
rem 仮想マシン制御用グループリソース名  
SET CLPRSC="exec-vm"  
rem 各ホストのIPアドレス(カンマ区切り)  
SET CLPIP="10.0.0.1,10.0.0.2"  
  
clptrnreq.exe -t GRP_FAILOVER -r %CLPRSC% -h %CLPIP%  
echo EXIT
```

2. ホスト OS の CLUSTERPRO のバージョンが X3.0 以降の場合

```
rem *****  
rem *           preaction.bat           *  
rem *****  
  
echo START  
echo %CLP_MONITORNAME%  
  
rem 仮想マシン制御用グループリソース名  
SET CLPRSC="vm1"  
rem 各ホストのIPアドレス(カンマ区切り)  
SET CLPIP="10.0.0.1,10.0.0.2"  
  
clprexec.exe --failover -r %CLPRSC% -h %CLPIP%  
echo EXIT
```

注:ゲスト OS の CLUSTERPRO のバージョンが X2.1 の場合、clprexec コマンドが存在しません。clptrnreq コマンドを使用するか、CLUSTERPRO CD から clprexec コマンドを取得して使用してください。
