

MIRACLE CLUSTERPRO で Oracle データベース・クラスシステムを構築しよう！

ミラクル・リナックス株式会社

作成者： 鈴木 庸隆

作成日： 2005 年 6 月 30 日

目次

1.	CLUSTERPRO の概要	1
1.1.	クラスタシステム	1
1.2.	MIRACLE CLUSTERPRO	1
1.3.	データの引継ぎ	1
1.4.	CLUSTERPRO LE	2
1.5.	FastSync Option (FSO)	2
2.	CLUSTERPRO アーキテクチャー	3
2.1.	CLUSTERPRO の構成	3
2.2.	CLUSTERPRO サーバ	4
2.3.	CLUSTERPRO SE の HW 構成	5
2.4.	CLUSTERPRO LE の HW 構成	5
2.5.	クラスタオブジェクト	6
2.6.	サーバオブジェクト - ハートビートリソース	7
2.7.	グループオブジェクト - グループリソース	8
2.8.	モニタオブジェクト - モニターリソース	9
3.	Oracle DB クラスタシステムの構築	10
3.1.	MIRACLE LINUX のインストールと設定	10
3.1.1.	事前準備	10
3.1.2.	ソフトウェア構成	10
3.1.3.	Linux インストール時の設定内容	11
3.2.	Oracle 設定	13
3.2.1.	Oracle Database 10g のインストール	13
3.2.2.	データベースの作成	13
3.2.3.	共有データの管理	14
3.2.4.	OracleNet の設定	14
3.2.5.	Oracle Database 起動/停止の SQL スクリプト	15
3.3.	CLUSTERPRO 設定	17
3.4.	トレッキングツールの設定	17
3.4.1.	トレッキングツール (Linux 版)	17
3.5.	クラスタ構成ファイル作成	19
3.5.1.	事前準備	19
3.5.2.	トレッキングツールの起動	21
3.5.3.	クラスタの作成	22
3.5.4.	サーバの登録	23
3.5.5.	サーバの定義確認	29
3.5.6.	ハートビート I/F の設定	30
3.5.7.	WebManager グループの追加	32
3.5.8.	グループの追加	33
3.5.9.	フローティング IP リソースの設定	35
3.5.10.	ミラーディスクリソースの設定	38
3.5.11.	EXEC リソース (Oracle 起動/停止) の設定	43
3.5.12.	EXEC リソース (Oracle 監視プログラム起動/停止) の設定	48
3.5.13.	モニターリソースの設定	54
3.5.14.	PID モニタリソースの設定	55

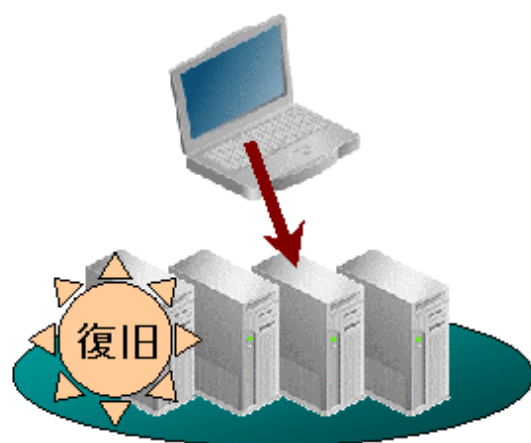
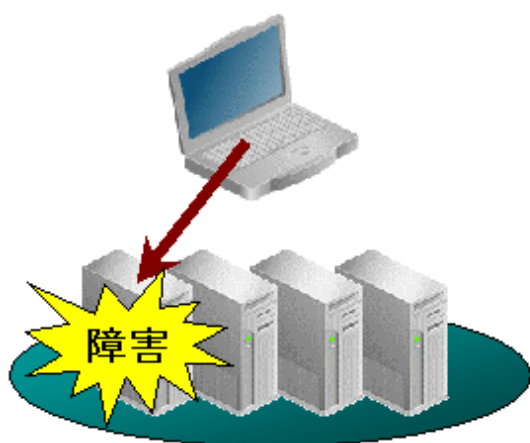
3.5.15.	ip モニタリソースの設定	57
3.5.16.	クラスタ構成情報の保存	61
3.6.	CLUSTERPRO サーバの設定	62
3.6.1.	CLUSTERPRO 本体のインストール(両ノード)	62
3.6.2.	クラスタ構成ファイルの登録準備(両ノード)	62
3.6.3.	クラスタ構成ファイルの登録(片ノードのみ)	63
3.6.4.	ライセンスキーの登録(両ノードと片ノード)	64
3.6.5.	CLUSTERPRO 全サービスの起動(両ノード)	64
4.	クラスタシステムの運用	65
4.1.	CLUSTERPRO の運用操作	65
4.2.	Web マネージャの操作	65
4.2.1.	Web マネージャ管理画面	66
4.2.2.	ミラーディスクヘルパー	67
4.2.3.	ログの収集	68
4.3.	コマンドによる運用管理	69
4.3.1.	状況確認	69
4.3.2.	グループの起動/停止	71
4.3.3.	サーバのシャットダウン/リブート	71
4.3.4.	クラスタシステム全体のシャットダウン/リブート	71
5.	最後に	71

1. CLUSTERPRO の概要

1.1. クラスタシステム

クラスタシステムとは、ネットワーク上で複数のコンピュータを協調動作させ、利用者(クライアント)からは「1 台のコンピュータ」に見せるシステムです。複数台のコンピュータを協調動作させることで、「信頼性」と「処理能力」の大幅な向上を図ることができます。

マシンの冗長化させ、障害発生時のシステム停止を最小限にするクラスタが、HA クラスタシステムということになります。



1.2. MIRACLE CLUSTERPRO

一般的には、「HA クラスタリングソフトウェア」と呼ばれるソフトウェアを利用して、このような HA クラスタシステムを構築します。MIRACLE CLUSTERPRO は、NEC 社が開発した HA クラスタウェア (CLUSTERPRO) と Oracle Database との親和性の高い MIRACLE LINUX OS をバンドルしたソリューションパッケージ製品です。

12 月 12 日に、MIRACLE LINUX V4.0 をバンドルした、MIRACLE CLUSTERPRO の新製品を出荷したばかりです。今回、カーネル 2.6 に対応し、OS 自体が高負荷時の安定性とスケーラビリティが改善されて、HA クラスタシステムを構築するのによりよい環境になっています。製品の詳細は以下のミラクル・リナックスのサイトをご覧ください。

<http://www.miraclelinux.com/products/clusterpro/>

1.3. データの引継ぎ

共有ディスクと呼ばれる外付けのストレージにプールし、複数のサーバからそのデータを参照することが一般的です。接続方式は SCSI/FibreChannel/ethernet(NAS)等が選択可能です。また、HA クラスタウェアによっては、相互のサーバの特定領域をネットワーク越しにデータミラーリングさせることでデータ共有を行うことも可能です。

CLUSTERPRO では、共有ディスクを使用するクラスタタイプが CLUSTERPRO SE (図 1) で、データミラーリングでデータ共有を図るクラスタタイプが CLUSTERPRO LE (図 2) となります。



図 1: 共有ディスクタイプ



図 2: データミラーリングタイプ

1.4.CLUSTERPRO LE

今回は、データミラーリングタイプの CLUSTERPRO LE をフォーカスして構築手順まで紹介します。CLUSTERPRO LE は、RAID1 の技術を、ネットワークをはさんだマシン間で実現するもので、ノード間のデータは常に同期が行われます。待機系マシンがそのまま、バックアップマシンと位置づけることも出来、クラスタシステムを兼ねたバックアップを安価で簡単に構築できます。

【欠点】

- ✧ 復旧してきたノードは、データの再同期が終了するまでは、クラスタに参加できない。
- ✧ データの再同期中は、ディスク I/O が発生するため、性能が劣化する。

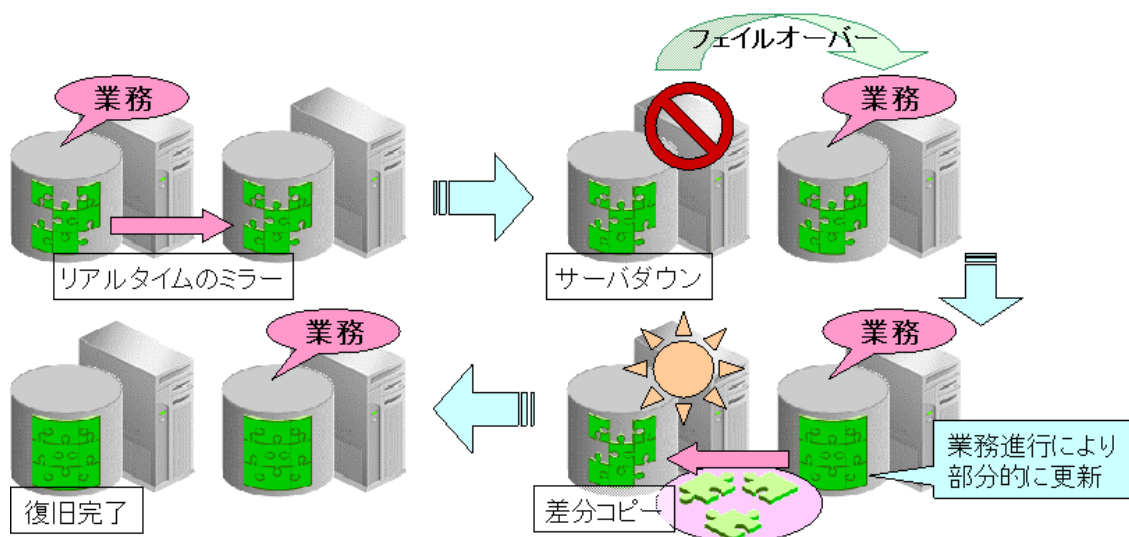
【利点】

- ✧ 共有ディスクやストレージを使用しないため、比較的安価にシステム構築可能。

1.5.FastSync Option(FSO)

ミラーディスクはその性質上、片ノードダウン時など両ノードのデータ同期できない状態になる場合があります。このような場合、ミラー復帰という作業で両ノードのミラーディスクを再同期する必要があります。ただし、このミラー再同期中はフェイルオーバー処理が行えないため、その時間はクラスタシステムの可用性が落ちてしまいます。

CLUSTERPRO では、FastSync Option(オプション製品)を使い、「ミラー再同期処理」を差分情報のみコピーすることで、データ再同期時間を大幅に短縮します。この高速ミラー再同期の機能は、国内 Linux HA ソフトでは、唯一 CLUSTERPRO が実装しています。



「FSO なし」の場合、20G のミラー復帰に 1 時間前後かかるところを、「FSO あり」の場合、数秒から数十秒でミラー復帰が完了します。

2.CLUSTERPRO アーキテクチャー

2.1.CLUSTERPRO の構成

CLUSTERPRO は、大きく分けて以下の 3 つのコンポーネントから構成されています。

☆ CLUSTERPRO サーバ

CLUSTERPRO の本体プログラムで、サーバの高可用性機能のすべてが含まれます。また、Web マネージャのサーバ側機能も含まれています。

☆ Web マネージャ

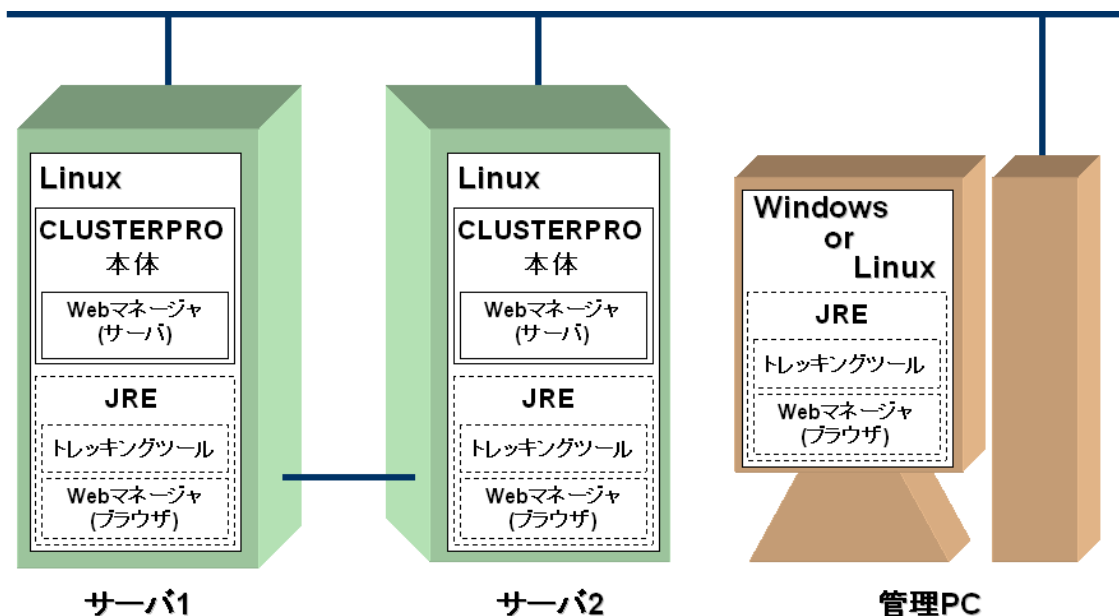
CLUSTERPRO の運用管理を行うための GUI 管理ツールです。

ユーザインターフェースとして Web ブラウザを利用します。Web マネージャのプログラムは CLUSTERPRO の本体プログラムに組み込まれていますが、操作は管理端末上の Web ブラウザで行うため、CLUSTERPRO 本体とは区別されます。

☆ トレーキングツール

CLUSTERPRO の設定情報を作成するためのツールです。

Web マネージャと同様に、ユーザインターフェースとして Web ブラウザを利用します。トレーキングツールはトレーキングツールを使用する端末上で、CLUSTERPRO 本体とは別にインストールする必要があります。

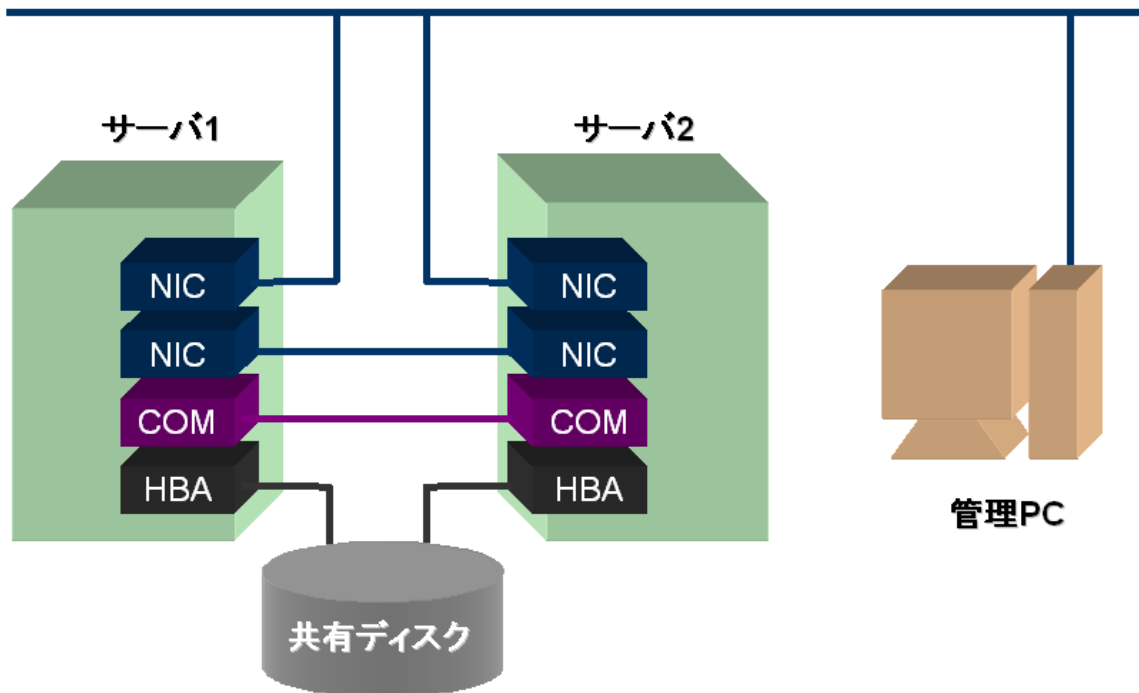


2.2.CLUSTERPRO サーバ

CLUSTERPRO サーバは、SE では5つ、LE では6つのデーモンから構成されています。

イベントデーモン clusterpro_evt	<ul style="list-style-type: none">•CLUSTERPROの各種ログ情報の記録を行う•CLUSTERPRO本体とは独立して起動可能
トランザクションデーモン clusterpro_trn	<ul style="list-style-type: none">•クラスタサーバ間でのファイル転送やライセンス同期を行う•CLUSTERPRO本体とは独立して起動可能
ミラーデーモン clusterpro_md	<ul style="list-style-type: none">•LEのみ存在する•ミラーディスクの制御を行う•CLUSTERPRO本体とは独立して起動可能
本体デーモン clusterpro	<ul style="list-style-type: none">•CLUSTERPRO本体
Webマネージャデーモン clusterpro_webmgr	<ul style="list-style-type: none">•Webマネージャサーバ側
アラート同期デーモン clusterpro_alertsvnc	<ul style="list-style-type: none">•各サーバで発行されるアラートをマージする

2.3.CLUSTERPRO SE の HW 構成



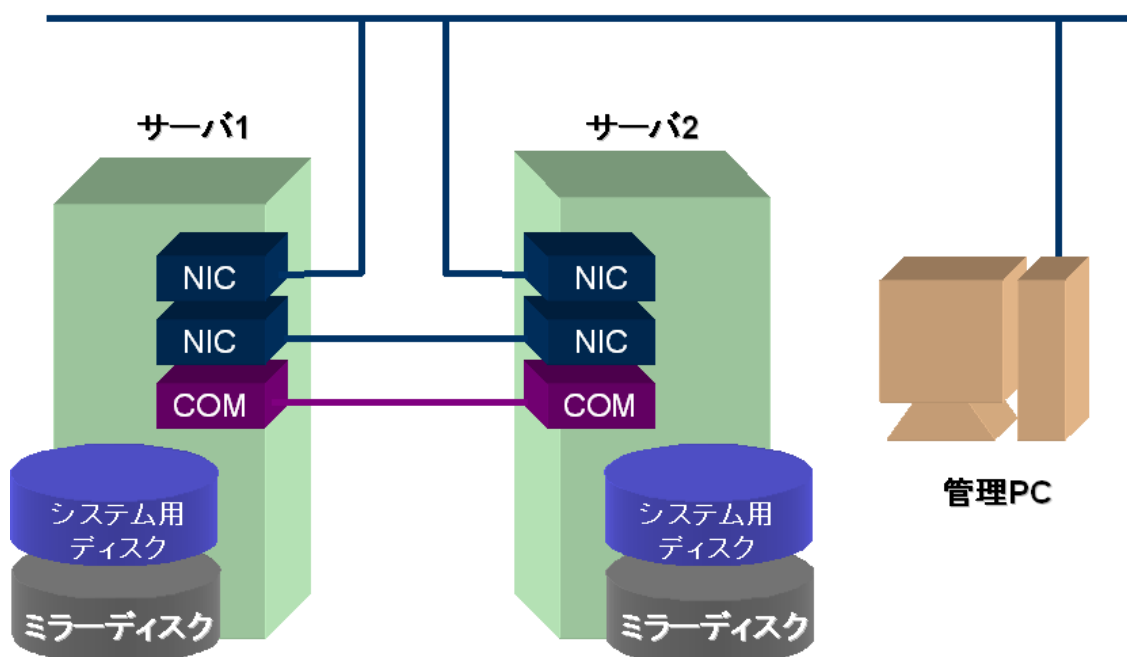
共有ディスク構成の CLUSTERPRO(CLUSTERPRO SE)の HW 構成は上図のようになります。

サーバ間の通信用に

- NIC を 2 枚 (1 枚は外部との通信と流用、1 枚は CLUSTERPRO 専用)
- RS232C クロスケーブルで接続された COM ポート
- 共有ディスクの特定領域を利用する構成が一般的です。

共有ディスクとの接続インターフェースは SCSI が FibreChannel ですが、最近では FibreChannel による接続が一般的です。

2.4.CLUSTERPRO LE の HW 構成



データミラー構成の CLUSTERPRO (CLUSTERPRO LE) は、上図のような構成になります。

共有ディスク構成と比べ、ミラーディスクデータコピー用のネットワークが必要となりますが、通常、CLUSTERPRO の内部通信用 NIC と兼用します。

また、ミラーディスクは OS インストール領域とは別ディスクである必要がありますが、接続インターフェース (IDE or SCSI) には依存しません。

ミラーディスクの容量

CLUSTERPRO LE ではディスクのミラーリングが発生するため、以下のミラーディスク容量が推奨されます。

- 30GB まで (FastSync Option なし)
- 100GB まで (Fast Sync Option あり)

2.5. クラスタオブジェクト

CLUSTERPRO では、Web マネージャを使用して管理用 PC から視覚的にクラスタシステムを運用管理します。マネージャによる運用管理の単位をクラスタオブジェクトと呼び、各オブジェクトをツリー構造で階層的に運用管理します。



◇ サーバオブジェクト

特定サーバの状態を示すオブジェクトです。

サーバ名やパブリック LAN の IP アドレス、各種ハートビートの状態を確認します。サーバオブジェクト配下にはハートビートリソースが含まれます。

◇ **グループオブジェクト**

特定グループの状態を示すオブジェクトです。

グループが所有するリソースの状態を確認できます。

グループとは、クラスタシステム内で、ある1つの業務を実行するために必要な資源の集まりのことで、フェイルオーバーを実行する単位です。

グループオブジェクト配下にはグループリソースが含まれます。

◇ **モニタオブジェクト**

監視対象の状態を示すオブジェクトです。

各種監視対象の詳細情報を確認できます。

モニタオブジェクト配下にはモニタリソースが含まれます。

2.6.サーバオブジェクト—ハートビートリソース

ハートビートリソースは、サーバオブジェクトに含まれるハートビートを監視するリソースです。



◇ **LAN ハートビートリソース**

直結の LAN、およびパブリック LAN を使用して、ハートビートを監視します。

リソース名や、ハートビートの状態、インタコネクト LAN の IP アドレスなどを確認できます。

◇ **COM ハートビートリソース**

RS232C (COM) を使用してハートビートを監視します。

リソース名や、ハートビートの状態などを確認できます。

◇ **DISK ハートビートリソース**

共有ディスクの CLUSTERPRO パーティションを使用してハートビートを監視します。リソース名や、ハートビートの状態などを確認できます。

2.7.グループオブジェクトーグループリソース

グループリソースは、サーバオブジェクトに含まれる特定の部ループに所属するリソースです。



グループリソース



フローティングIPリソース



EXECリソース



ディスクリソース



ミラーディスクリソース

◇ **フローティング IP リソース**

クライアントアプリケーションが接続する仮想的な IP アドレスで、グループの起動時 / 終了時、フェイルオーバー発生時に切り替えられます。

リソース名やリソースの状態、その他詳細情報を確認できます。

◇ **EXEC リソース**

ユーザ独自のプログラムやシェルスクリプトを登録してもので、グループの起動時 / 終了時、フェイルオーバー発生時に実行されます。

リソース名やリソースの状態、その他詳細情報を確認できます。

◇ **ディスクリソース (CLUSTERPRO SE のみ)**

クラスタシステムを構成するサーバに接続された共有ディスク上のパーティションで、グループの起動時 / 終了時、フェイルオーバー発生時に切り替えられます。

リソース名やリソースの状態、その他詳細情報を確認できます。

◇ **ミラーディスクリソース (CLUSTERPRO LE のみ)**

クラスタシステムを構成する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で、グループの起動時 / 終了時、フェイルオーバー時に切り替えられます。

リソース名やリソースの状態、その他詳細情報を確認できます。

2.8. モニタオブジェクト—モニターリソース

モニターリソースとは、サーバオブジェクトに含まれる監視対象となるリソースです。



◇ IP モニターリソース

指定した IP アドレスを監視します。指定した IP アドレスへの ping コマンドの応答により異常を判断します。

◇ NIC モニターリソース

指定したデバイスの NIC Link down を監視します。IP モニターリソースに比べ、「ping のタイムアウト時間が必要ない」、「外部機器の生死に依存しない」といったメリットがあります。

◇ ディスクモニターリソース

指定したディスクデバイスを監視します。TUR または Dummy Read により異常を判断します。

◇ ミラーディスクモニターリソース (CLUSTERPRO LE のみ)

指定したミラーディスクデバイスを監視します。データパーティションの整合性を監視します。

◇ ミラーディスクコネクトモニターリソース (CLUSTERPRO LE のみ)

指定したミラーリング用のネットワークを監視します。ミラーデータの通信に失敗した場合に異常と判断します。

◇ PID モニターリソース

指定した exec リソースを監視します。プロセス ID の有無を監視し、プロセス ID の消滅時に異常と判断します。プロセスのストールを検知することは出来ません。

◇ ユーザ空間モニターリソース

メモリ空間のストールを異常として判断します。softdog ドライバに対してコマンドを発行し、その応答により異常を判断します。

3.Oracle DB クラスタシステムの構築

3.1.MIRACLE LINUX のインストールと設定

MIRACLE CLUSTERPRO V3.1(以下 MCP)には、現在 MIRACLE LINUX V4.0(以下 ML4.0)、と MIRACLE LINUX V3.0(以下、ML3.0)をバンドルした製品を出荷しています。
今回は、CLUSTERPRO LE と ML4.0 を使用し、Oracle Database のクラスタシステムを構築していきます。

3.1.1.事前準備

CLUSTERPRO LE(ネットワークミラー型)を使用し、Oracle Database 10g R1 の片方向スタンバイ・クラスタシステムを構築します。CLUSTERPRO LE で必須となるハードウェアは以下の通りです。

H/W	個数	解説
ローカルハードディスク	2 台	1 台目は OS 専用ディスク、2 台目は CLUSTERPRO が占有するディスクになります。
NIC	2 つ	CLUSTERPRO LE では、インターコネクト LAN 用 ハートビート兼、ネットワークミラー回線用の NIC(LAN)と、Public ネットワーク LAN 用の NIC(LAN)の 2 つが必要になります。
RS232C クロスケーブル	1 つ (任意)	サーバ間のハートビートの一つとなる COM ポート経由でのハートビートに使用します。基本的には、LAN ハートビートがあるので任意です。

CLUSTERPRO LE では、共有ディスクを使用しないので VMware(<http://www.vmware.com/jp/>)などでもテスト環境を構築することが可能です。

3.1.2.ソフトウェア構成

今回、クラスタシステムで使用するソフトウェアは以下の通りになります。

区分	詳細
オペレーティングシステム	MIRACLE LINUX V4.0 カーネル: kernel-smp-2.6.9-11.19AX GLIBC: glibc-2.3.2-95.30.1AX
CLUSTERPRO	CLUSTERPRO LE 3.1 (clusterpro-3.1-6) データベース監視オプション R3.0 (clusterpro-dbmon-3.0-4)
Oracle Database	Oracle Database 10g R1 (10.1.0.3)

OS、Oracle は、それぞれ以下サイトからの入手してください。

MIRACLE LINUX V4.0

Asianux 公式サイトからダウンロードして試使用できます。

<http://www.asianux.com/download.php>

- ・ MIRACLE LINUX V4.0 Disk1: Asianux2.0 disk1
- ・ MIRACLE LINUX V4.0 Disk2: Asianux2.0 disk2

Oracle Database

OTN(Oracle Technical Network)からダウンロードして試使用できます。

<http://otn.oracle.co.jp/software/products/oracle10g/index.html>

CLUSTERPRO

ミラクル・リナックスの以下のサイトから、MCP の試用版と試用版のお申し込みを行ってください。

<http://www.miraclelinux.com/products/clusterpro/license.html>

サーバセレクト本誌 2006 年 2 月号の付録の MIRACLE CLUSTERPRO 試用版インストール CD は、製品版と同等です。CLUSTERPRO 各種製品の試用版ライセンスキーにつきましては、上記のサイトから試用版のお申し込みを行ってください。

3.1.3.Linux インストール時の設定内容

インストールの手順は、以下のミラクル・リナックスのサイトをご参考になしてください。

<http://www.miraclelinux.com/technet/install/>

以下は、Linux インストール時のサンプルの設定内容になります。皆さんの環境に合わせて、ホスト名、IP アドレス等を変更してください。

①基本設定

今回のクラウドシステム構築に必要な、Oracle の必須パッケージや、JVM などが自動インストールされるように、インストールオプションは「すべて」を選択することをお勧めします。

項目	設定	
インストールオプション	すべて	
ディスクパーティションの設定	下記 を参照	
ブートローダーの設定	GRUB をブートローダーとして使用	
	インストールする場所	/dev/sda
ネットワークの設定	下記 を参照	
ファイアウォールの設定	ファイアウォールなし	
サポート言語の選択	Japanese	
タイムゾーン	アジア/東京	
アカウント	root パスワード	miracle
	ログイン方法	テキストログイン
デスクトップ環境	KDE	

②ネットワーク設定

項目	ノード 1	ノード 2
ホスト名	kiki	lala
ドメイン名	miraclelinux.com	miraclelinux.com
プライマリ DNS	任意	任意
セカンダリ DNS		
eth0(パブリック)		
IP アドレス	10.1.0.85	10.1.0.86
ネットマスク	255.255.255.0	255.255.255.0
ゲートウェイ	10.1.0.1	10.1.0.1
eth1(インターコネクト)		
IP アドレス	192.168.0.1	192.168.0.2
ネットマスク	255.255.255.0	255.255.255.0
ゲートウェイ		

②ローカルディスク設定

OS 用パーティション作成

インストール時に「Disk Druid」を選択し、パーティションを作成します。

ディスクデバイス	マウントポイント	サイズ	デバイス名	ファイルシステム
/dev/sda	/boot	100MB	/dev/sda1	ext3(xfs/riserfs)
	Swap	物理メモリの倍	/dev/sda2	Swap
	/	任意(6GB以上)	/dev/sda3	ext3(xfs/riserfs)

CLUSTERPRO 用パーティション作成

CLUSTERPRO LE は、CLUSTERPRO 専用ローカルディスクが1つ必要です。また、共有データが入るデータパーティション1つに対して、ミラーリング管理用にクラスタパーティション(最低10MB)を用意することになります。²

今回は、OSインストール後に、fdisk コマンドを使用して、CLUSTERPRO 用のローカルディスク(/dev/sdb)にクラスタパーティションとデータパーティションを1つずつ作成します。

ディスクデバイス	マウントポイント	サイズ	デバイス名	ファイルシステム
/dev/sdb	クラスタパーティション	20MB	/dev/sdb1	
	/oracle	任意(2GB以上)	/dev/sdb2	ext3(xfs/riserfs)
	データパーティション			

```
# fdisk /dev/sdb
```

```
コマンド (m でヘルプ): n      n 入力: 新規作成
コマンドアクション
  e   拡張
  p   基本領域 (1-4)
p
領域番号 (1-4): 1              P 入力: 基本パーティション選択
                              1 入力: パーティション番号指定
最初 シリンダ (1-4425, default 1):      リターン入力: 開始シリンダ
Using default value 1
終点 シリンダ または +サイズ または +サイズ M または +サイズ K (1-4425, default 4425):
+20M                                  20M 入力: クラスタパーティション

コマンド (m でヘルプ): n      n 入力: 新規作成
コマンドアクション
  e   拡張
  p   基本領域 (1-4)
p
領域番号 (1-4): 2              P 入力: 基本パーティション選択
                              2 入力: パーティション番号指定
最初 シリンダ (4-4425, default 4):      リターン入力: 開始シリンダ
Using default value 4
終点 シリンダ または +サイズ または +サイズ M または +サイズ K (4-4425, default 4425):
+4096M                                4096M 入力: データパーティション

コマンド (m でヘルプ): p

デバイス Boot      Start      End      Blocks  Id System
/dev/sdb1          1          3       24066   83  Linux
/dev/sdb2          4        502    4008217+  83  Linux

コマンド (m でヘルプ): w      w 入力: 保存
```


3.2.Oracle 設定

3.2.1.Oracle Database 10g のインストール

両ノードに ML4.0 のインストールと CLUSTERPRO 用パーティション作成が完了したら、次に Oracle10g Database のインストールを行います。

MIRACLE LINUX OS では、Oracle に必要なカーネルパラメータの設定や、シェル制限の設定などデフォルトで最適化されていて、いきなり Oracle のインストールを行えます。

Oracle インストール専用の Oranavi ツールを使用したインストール手順は以下のミラクル・リナックスのサイトを参照してください。

<http://www.miraclelinux.com/technet/install/oracle.html>

今回のように、Oracle Database のクラスタリングを行う際には、Oracle のソフトウェアのみをまず両ノードにインストールします。後で別途、片ノードのデータパーティションにのみ Oracle データベースを作成する手順になります。

使用アカウント	oracle
	設定パスワード:oracle
	ログインホーム:/home/oracle
ORACLE_BASE	/opt/app/oracle
ORACLE_HOME	/opt/app/oracle/product/10.1.0/db_1
ORACLE_SID	orcl
DB 管理ユーザパスワード	sys :miracle system :miracle

3.2.2.データベースの作成

3.2.2.1.データパーティションのファイルシステム作成

CLUSTERPRO LE は、初回起動時にプライマリサーバ(kiki)のデータパーティション(/dev/sdb2)のファイルシステムを自動作成し、セカンダリサーバ(lala)に対してデータミラーリングを行い、データ共有を開始します。ただし、必要に応じて、事前にプライマリサーバのデータパーティションのファイルシステムを作成しておくことも可能です。

今回は、CLUSTERPRO 初回起動時の自動ファイルシステム作成は無効にしておきます、事前にプライマリサーバ(kiki)のデータパーティション(/oracle)にファイルシステムを作成し、その領域に Oracle データベースを作成して初回起動時に Oracle データベースを含め、セカンダリサーバに対してデータミラーを行うようにします。

< プライマリサーバ(kiki)のみ >

```
# mkfs.ext3 /dev/sdb2
# mkdir /oracle
# mount /dev/sdb2 /oracle          /etc/fstab に記入しないでください。
```

3.2.2.2.データベース作成

Database Configuration Assistant(dbca)などを使用して、/oracle(データパーティション)にデータベースを作成します。今回は、/oracle/oradata 以下に作成することにします。

```
# df -h
Filesystem      サイズ  使用  残り  使用% マウント位置
/dev/sda1        99M   16M   79M   17% /boot
/dev/sda2       10G   5.4G   4.6G   43% /
none             91M    0    91M    0% /dev/shm
/dev/NMP1        4.0G  144K   4.0G    1% /oracle
# mkdir /oracle/oradata
# chown -R oracle:oinstall /oracle
```

KDE を起動した環境でコンソールを立ち上げ、oracle ユーザに Switch User し、dbca を起動します。
dbca でデータベースを作成する際に、Oracle データベースファイル(データファイル、REDO ログファイル、制御ファイル)を/oracle/oradata 以下に作成するよう指定します。

```
# xhost +
# su - oracle
$ export DISPLAY=localhost
$ xclock          X サーバが使用できるか確認
[ctrl]+c
$ dbca            dbca を起動
```

3.2.3.共有データの管理

ここまでの設定で、Oracle データベースファイル(/oracle/oradata)は、データミラーリングによって両ノード間でデータ共有できる状態になりました。これでどちらのノードにフェイルオーバーしてもデータベースが起動することが出来るわけです。

ただし、Oracle Database では、データベースファイル以外にも両ノードで共有した方が良いファイルがいくつかあります。それは、alert ログファイル(\$ORACLE_BASE/admin 以下)と初期化パラメータファイル(\$ORACLE_HOME/dbs 以下)です。今回は、\$ORACLE_HOME/dbs と\$ORACLE_BASE/admin もミラーデータ上(/oracle)に移動して、元の位置からシンボリックリンクを張るように設定します。

<kiki>

```
# mv $ORACLE_BASE/admin /oracle
# ln -s /oracle/admin $ORACLE_BASE
# mv $ORACLE_HOME/dbs /oracle
# ln -s /Oracle Databases $ORACLE_HOME
```

<lala> /oracle はアンマウント状態なので、ここでは空のシンボルリンクを作成します。

```
# ln -s /oracle/admin $ORACLE_BASE
# mv $ORACLE_HOME/dbs $ORACLE_HOME/dbs.org
# ln -s /Oracle Databases $ORACLE_HOME
```

3.2.4.OracleNet の設定

Oracle Database をクラスタ化した場合、Listener で使用する IP アドレスは、フローティング IP アドレスを指定する必要があります。そのためクラスタシステム用に listener.ora、tnsnames.ora を作成しておきます。

「orcl」接続文字列で確立される CLUSTERPRO 用の OracleNet の設定は次のようになります。

✓ \$ORACLE_HOME/network/admin/listener.ora (両ノード別々のファイルで同じ内容)

```
SID_LIST_LISTENER =
(SID_LIST =
(SID_DESC =
(GLOBAL_DBNAME = orcl.miraclelinux.com)
(ORACLE_HOME = /opt/app/oracle/product/10.1.0/db_1)
(SID_NAME = orcl)
)
)
```

```

LISTENER =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = 10.1.0.88)(PORT = 1521))
)

```

- ✓ \$ORACLE_HOME/network/admin/tnsnames.ora (両ノード別々のファイルで同じ内容)

```

orcl =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = 10.1.0.88)(PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = orcl.miraclelinux.com)
  )
)

```

3.2.5.Oracle Database 起動/停止の SQL スクリプト

CLUSTERPRO では、基本的には CLUSTERPRO サーバからシェルスクリプトを実行し、アプリケーションの起動/停止を行います。今回の Oracle Database の場合には、以下の例のように、そのシェルスクリプトの中で SQL*Plus を起動し、DB 起動/停止用の SQL スクリプトを更に実行するようになっています。

< DB 起動 >

```
su - oracle -c "export ORACLE_SID=orcl;sqlplus /nolog @/oracle/admin/orcl/sql/startup.sql"
```

< DB 停止 >

```
su - oracle -c "export ORACLE_SID=orcl;sqlplus /nolog
@/oracle/admin/orcl/sql/shutdown.sql"
```

この SQL スクリプトを共有データ(/oracle)以下に事前に作成しておきます。

- ✓ /oracle/admin/orcl/sql/startup.sql

```

connect / as sysdba
startup
exit

```

- ✓ /oracle/admin/orcl/sql/shutdown.sql

```

connect / as sysdba
shutdown immediate
exit

```

Oracle データベースの起動/停止を実際に確認してみましょう。

/oracle をマウントしているプライマサーバ(kiki)で実施

< DB 起動 >

```
# su - oracle -c "export ORACLE_SID=orcl;sqlplus /nolog @/oracle/admin/orcl/sql/startup.sql"
```

SQL*Plus: Release 10.1.0.3.0 - Production on 水 11月 30 19:31:26 2005

Copyright (c) 1982, 2005, Oracle. All rights reserved.

アイドル・インスタンスに接続しました。
ORACLE インスタンスが起動しました。

```
Total System Global Area 285212672 bytes
Fixed Size                  1218992 bytes
Variable Size               92276304 bytes
Database Buffers            188743680 bytes
Redo Buffers                 2973696 bytes
データベースがマウントされました。
データベースがオープンされました。
Oracle Database 10g Release 10.1.0.3.0 - Production との接続が切断されました。
#
```

< DB 停止 >

```
# su - oracle -c "export ORACLE_SID=orcl;sqlplus /nolog @/oracle/admin/orcl/sql/shutdown.sql"

SQL*Plus: Release 10.1.0.3.0 - Production on 水 11月 30 19:28:04 2005

Copyright (c) 1982, 2005, Oracle. All rights reserved.
接続されました。
データベースがクローズされました。
データベースがディスマウントされました。
ORACLE インスタンスがシャットダウンされました。
Oracle Database 10g Release 10.1.0.3.0 - Production との接続が切断されました。
#
```

最後に、CLUSTERPRO を起動する前に、データベースを停止して、/oracle をアンマウントしてください。
CLUSTERPRO が /oracle をマウントし、データベースを起動することになります。

3.3.CLUSTERPRO 設定

ではいよいよ CLUSTERPRO の設定を行っていきます。CLUSTERPRO でクラスタシステムを構築するまでには、「クライアント」、「サーバ」それぞれに以下のような作業が必要です。

CLUSTERPRO のマニュアルは MCP インストール CD の manual ディレクトリに格納されています。こちらも参考にしてください。

<クライアント>

- (1) トレッキングツールのインストールと JavaVM セキュリティの設定
- (2) トレッキングツールを用いてクラスタ構成ファイルを作成

<サーバ>

- (3) CLUSTERPRO サーバのインストール
- (4) (2)で作成したクラスタ構成ファイルを適応し、クラスタを生成
- (5)クラスタ生成後にライセンスキーの登録

MIRACLE CLUSTERPRO 専用インストール・スクリプト

MCP インストール CD に含まれるスクリプト(install_cluster.sh)を使用することで、CLUSTERPRO 本体とトレッキングツールの Linux サーバへのインストールが一度に行なうことが出来ます。今回は、このスクリプトを使用してインストール作業を行います。

3.4.トレッキングツールの設定

CLUSTERPRO では、クラスタシステムの設定ファイル(クラスタ構成ファイル)をまずクライアント上で作成し、その設定ファイルをサーバに登録することでクラスタシステムを構築します。

トレッキングツールはこのクラスタ構成ファイルを作成する Web ブラウザを使用する GUI ツールです。ここでは、トレッキングツールをクラスタシステムで使用する Linux サーバ上で使用するための設定を行いません。(Windows 版をご使用の場合は、別途インストールが必要となります。)

3.4.1.トレッキングツール (Linux 版)

トレッキングツール Linux 版のインストール

MCP インストール CD を使用し、トレッキングツールを Linux サーバにインストールします。MCP インストール CD をマウントし、install_cluster.sh スクリプトを root ユーザで実行します。

```
# mount /mnt/cdrom
# /mnt/cdrom/install_cluster.sh
1. Install CLUSTERPRO Server rpm
2. Install Trekking Tool rpm
3. Uninstall CLUSTERPRO Server rpm
4. Uninstall Trekking Tool rpm
5. Read the [Quick Installation Guide] *Japanese(UTF-8)

Select:(Q) > 2           2を選択し、トレッキングツールをインストール

Install the Trekking Tool RPM [Y/N]:(N) > y

    Preparing for the installation ....

    Start installing Trekking Tool ....
```

```

Preparing... ##### [100%]
  1:clusterprotrek ##### [100%]
Preparing... ##### [100%]
  1:clusterpro-monscript ##### [100%]

The installation of Trekking Tool RPM was successful

Back to the Start menu [Y/N]:(N) > n
#

```

< Windows 版 >

MCP インストール CD 内の次の exe ファイルを実行し、Windows 版トレッキングツールをインストールします。

¥CD ドライブ¥x86¥trek¥windows¥clusterprotrek-3.x-x.i386.EXE

¥CD ドライブ¥x86¥trek¥windows¥option¥SETUP.EXE

Java Run Time のインストール (ML4.0 ではインストール済み)

トレッキングツールは、Web ブラウザから JavaApplet を使用する GUI ツールです。ここでは JRE のインストールを行い、Web ブラウザの JRE プラグインの設定までを行ないます。

以下のページから Sun の Linux 版 JRE パッケージをダウンロードし、インストール作業を行ないます。ちなみに、ML4.0 の場合、デフォルトで JDK1.5.0_04 がインストールされますので、特に JRE パッケージをインストールする必要はありません。

<http://java.sun.com/j2se/>

```

# chmod +x jre-1_5_0_05-linux-i586-rpm.bin    実行権限を付与します
# ./jre-1_5_0_05-linux-i586-rpm.bin
# rpm -ivh jre-1_5_0_05-linux-i586.rpm

```

< Windows 版 >

以下のページにアクセスするとオンラインで Sun の Windows 版 JRE がインストールされます。

<http://java.com/ja/> (「今すぐ入手」をクリック)

Web ブラウザの Plugin 設定 (インストールスクリプトより自動設定)

Web ブラウザ(mozilla)からアプレットが使用できるように Plugin の設定を行ないます。JRE のバージョンによって、プラグインライブが異なります。

JRE/JDK インストール済みの環境の場合、MCP インストール CD からトレッキングツールをインストールする際に、自動的に設定されます。

JRE5.0

```

# ln -s /usr/java/jre1.5.0_05/plugin/i386/ns7/libjavaplugin_oji.so
¥ /usr/lib/mozilla/plugins/

```

JRE1.4.2

```

# ln -s /usr/java/j2re1.4.2_06/plugin/i386/ns610-gcc32/libjavaplugin_oji.so
¥ /usr/lib/mozilla/plugins/

```

Windows 版の Web ブラウザでは特に Plugin 設定は必要ありません。

JavaVM セキュリティ設定 (インストールスクリプトより自動設定)

トレッキングツールを使用するユーザのホームディレクトリに .java.policy ファイルを作成(コピー)します。

MCP インストール CD からトレッキングツールをインストールする際に、自動的に設定されます。

```
# cp /opt/nec/clptrek/etc/.java.policy $HOME ( root ユーザの場合 )
```

< Windows 版 >

トレッキングツールのインストール先ディレクトリ(C:\Program Files\nec\clptrek)の etc フォルダ内にある java_pol.zip を解凍し、.java.policy を C:\Documents and Settings\USERNAME 以下に置いてください。

.java.policy ファイルをテキストエディタ(メモ帳等)で開きます。

```
/* Trekking Tool */  
grant codeBase "file:/C:/Program Files/nec/clptrek/clptrek.jar" {  
    permission java.security.AllPermission;  
};
```

2行目の file:/ 以降の下線部分をトレッキングツールの解凍先ディレクトリに編集します。その場合、パスの区切りは「¥」ではなく「/」で記述します。

3.5. クラスタ構成ファイル作成

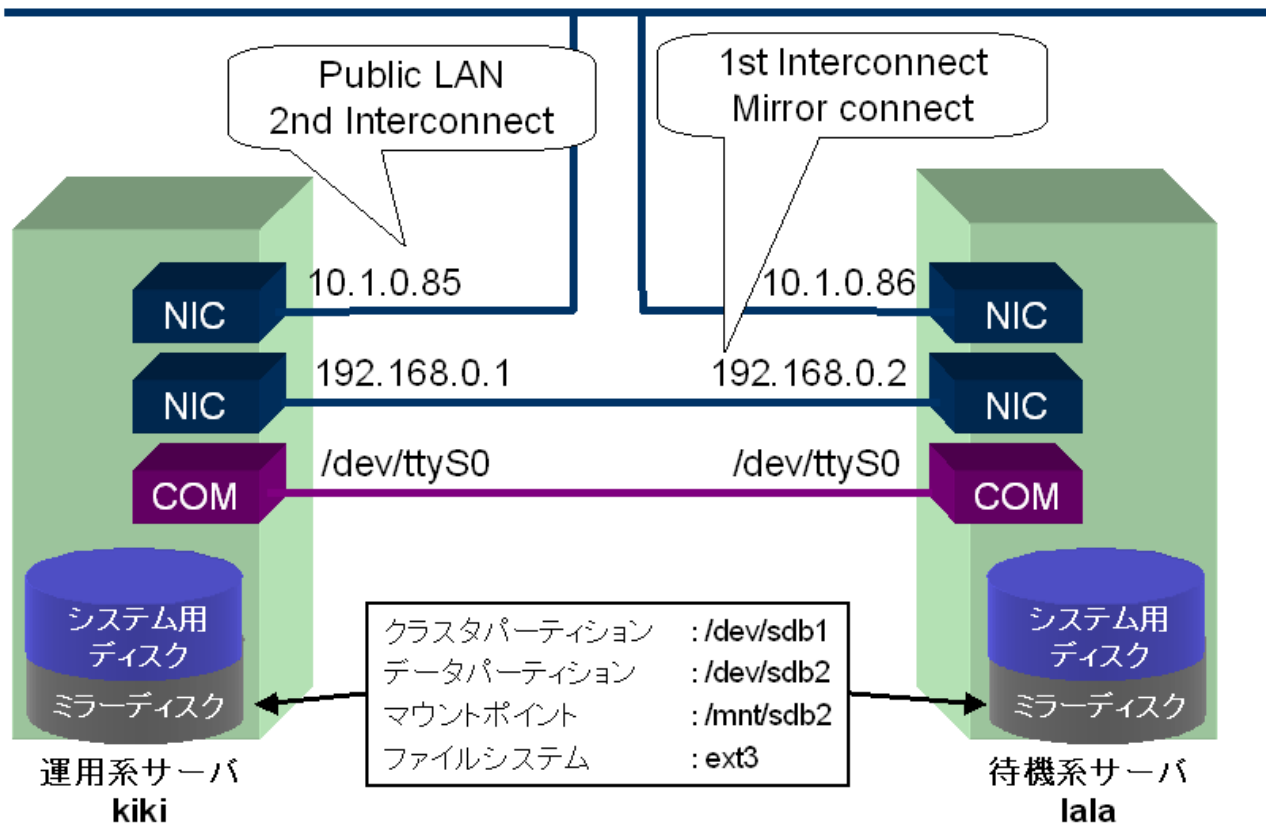
トレッキングツールを利用して CLUSTERPRO の構成情報を作成します。

なお、本テキストでは「CLUSTERPRO LE」(データミラー型)での Web サーバのクラスタを構築することにします。

3.5.1. 事前準備

トレッキングツールでのクラスタ情報生成の情報を事前に決めておきます。

Public LAN



<クラスタシステム構成>

運用系サーバの構成

- サーバ名 : kiki
- LAN ハートビート : 192.168.0.1
- LAN ハードビート : 10.1.0.85
- パブリック LAN : 10.1.0.85
- COM ハートビートデバイス : /dev/ttyS0
- ミラーディスクコネク : 192.168.0.1

待機系サーバの構成

- サーバ名 : lala
- LAN ハートビート : 192.168.0.2
- LAN ハードビート : 10.1.0.86
- パブリック LAN : 10.1.0.86
- COM ハートビートデバイス : /dev/ttyS0
- ミラーディスクコネク : 192.168.0.2

共有の構成

- WebManager 用フローティング IP : 10.1.0.87
- OracleNet 用フローティング IP : 10.1.0.88
- クラスタパーティション : /dev/sdb1
- データパーティション : /dev/sdb2
- ファイルシステム : ext3

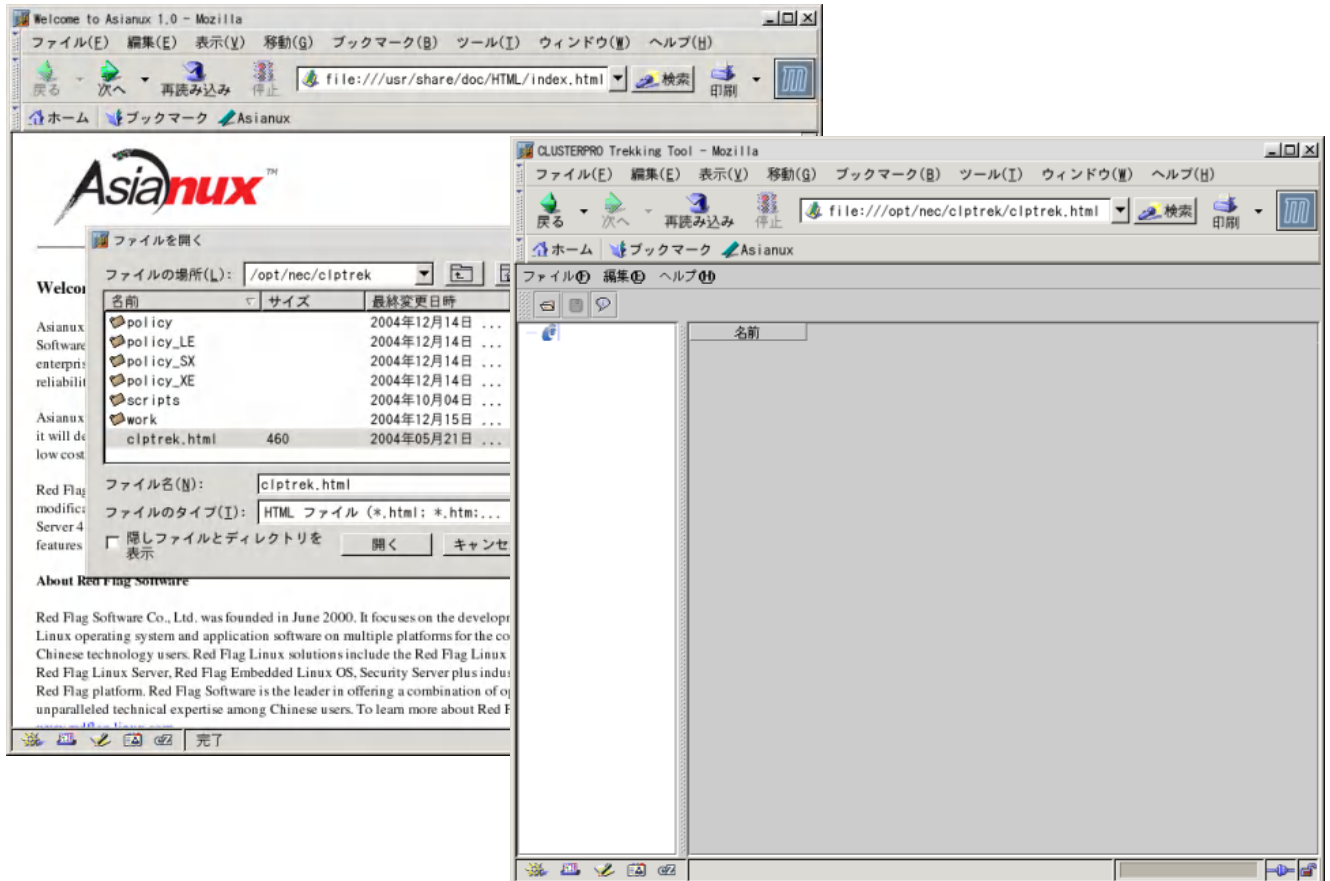
3.5.2. トレッキングツールの起動

Linux 上で mozilla を起動し、次のファイルを開きます。

```
/opt/nec/clptrek/clptrek.html
```

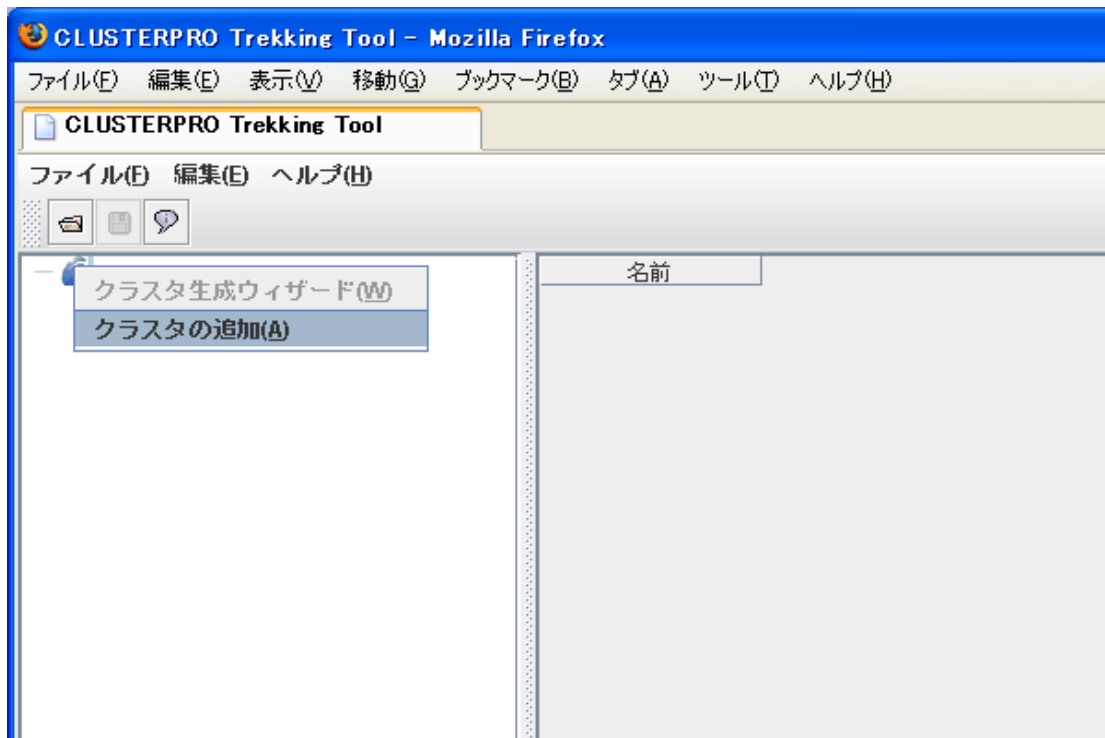
Windows 版では Web ブラウザを起動し、次のファイルを開きます。

```
C:\Program Files\Nec\clptrek\clptrek.html
```

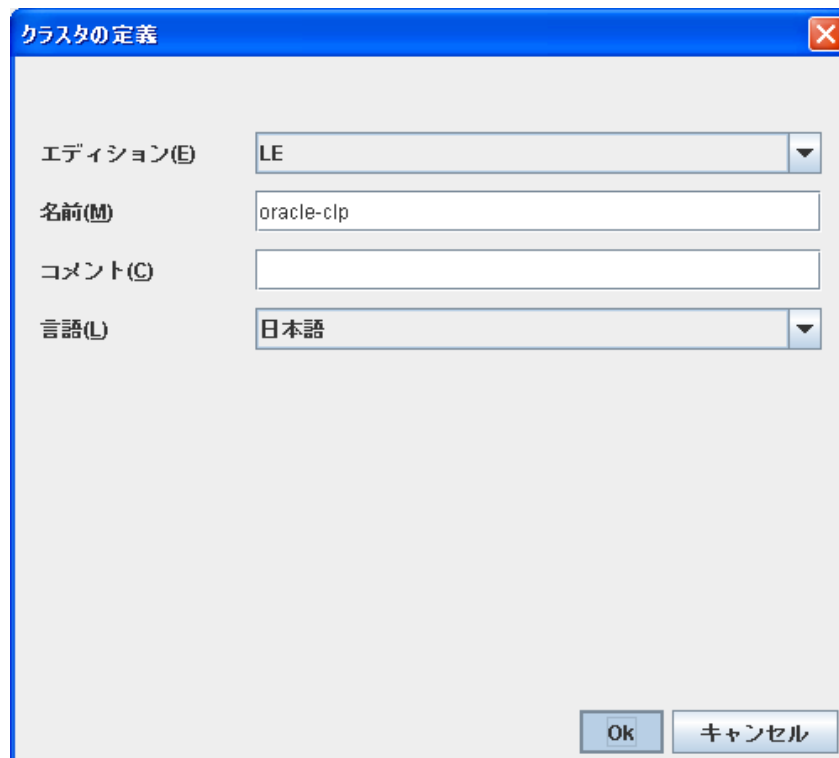


3.5.3. クラスタの作成

1. ツールビューの CLUSTERPRO アイコンを右クリックして「クラスタの追加」を選択



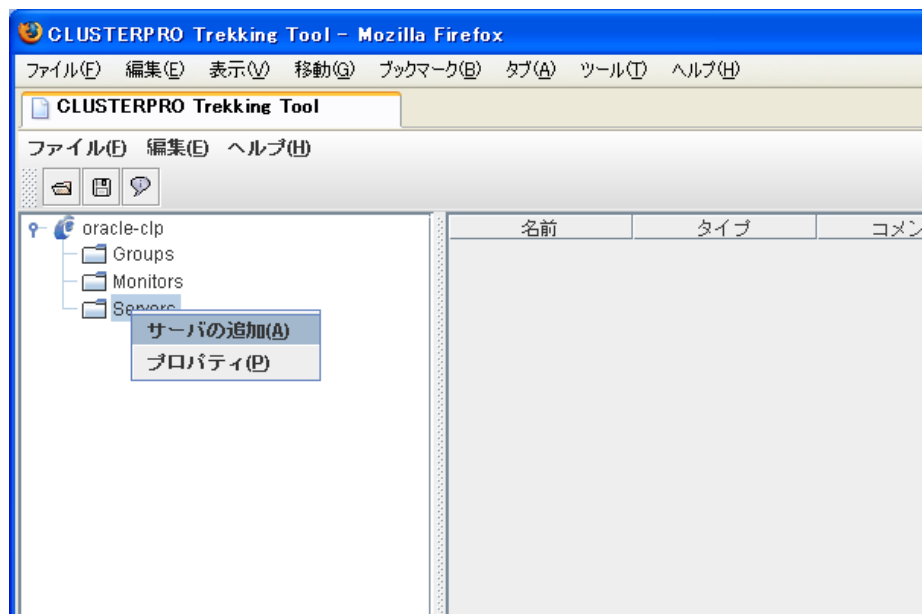
2. クラスタの定義
エディション : LE
名前 : oracle-clp
コメント : 任意
言語 : 日本語



「OK」ボタンとを選択してクラスタの定義を終了します

3.5.4.サーバの登録

1. ツリービューの Servers を右クリックして「サーバの追加」を選択



2. サーバの定義

名前 : kiki
コメント : 任意

「次へ」ボタンを選択して次の画面へ進みます。

サーバ名は、FQDN のホスト部のみを指定する。実サーバ名が "kiki.miraclelinux.com" であれば、"kiki" のみ指定します。

3.5.4.1.LAN ハートビート用インタコネクト LAN(専用)の設定

1. サーバの定義
「追加」ボタンを選択

2. IP アドレスの入力
IP アドレス : 192.168.0.1
「OK」ボタンを選択すると設定内容がインタコネクト LAN/IF 一覧に設定されます。

「次へ」ボタンを選択して次の画面へ進みます

3.5.4.2.LAN ハートビート用インタコネクト LAN(バックアップ)の設定

1. サーバの定義

サーバの定義

利用する候補を登録してください。

インタコネクトLAN IF 一覧(L)

IF 番号	IPアドレス
1	192.168.0.1

追加(D)
削除(R)
編集(E)
上へ(U)
下へ(D)
< 戻る(B) 次へ(N) > キャンセル

「追加」ボタンを選択

2. IP アドレスの入力

IP アドレス : 10.1.0.85

「OK」ボタンを選択すると設定内容がインタコネクト LAN/IF 一覧に設定されます。

IPアドレスの入力

インタフェース

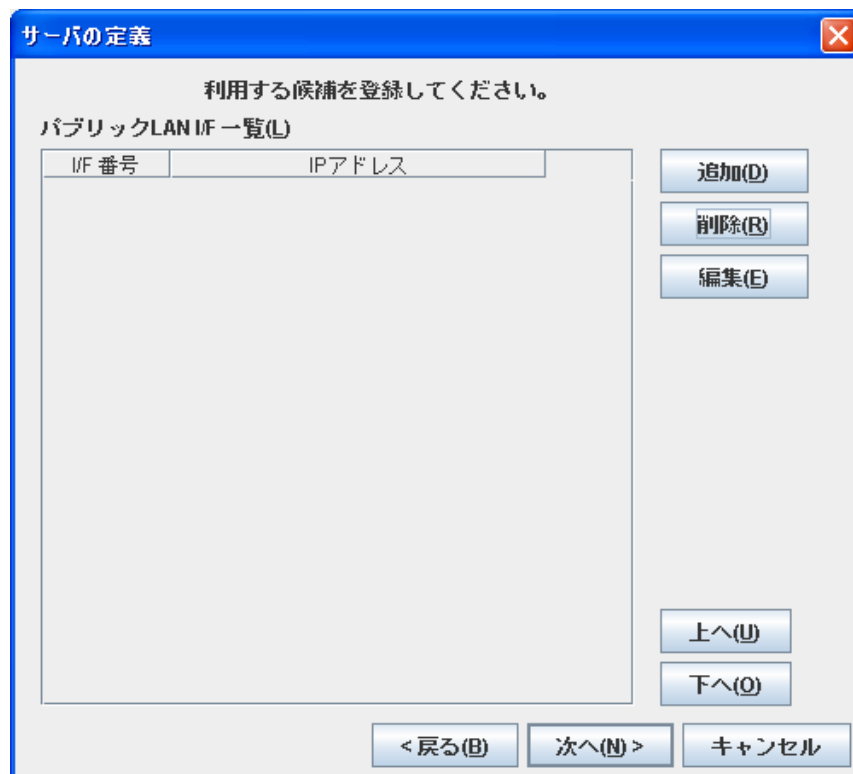
IPアドレス(D) 10.1.0.85

Ok キャンセル

「次へ」ボタンを選択して次の画面へ進みます。

3.5.4.3.LAN ハートビート用パブリック LAN の設定

1. サーバの定義

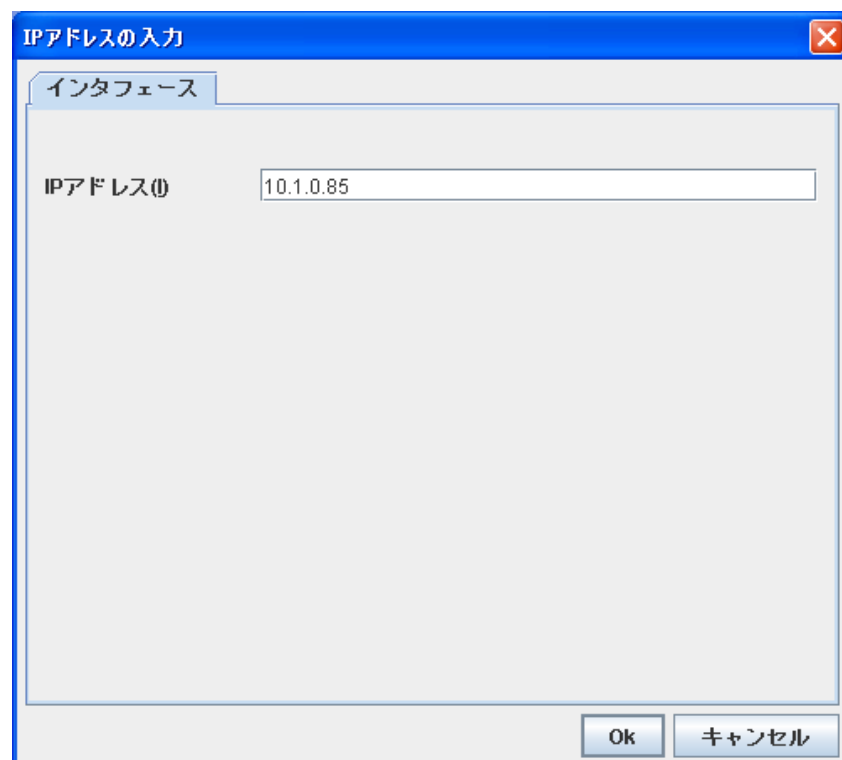


「追加」ボタンを選択

2. IP アドレスの入力

IP アドレス : 10.1.0.85

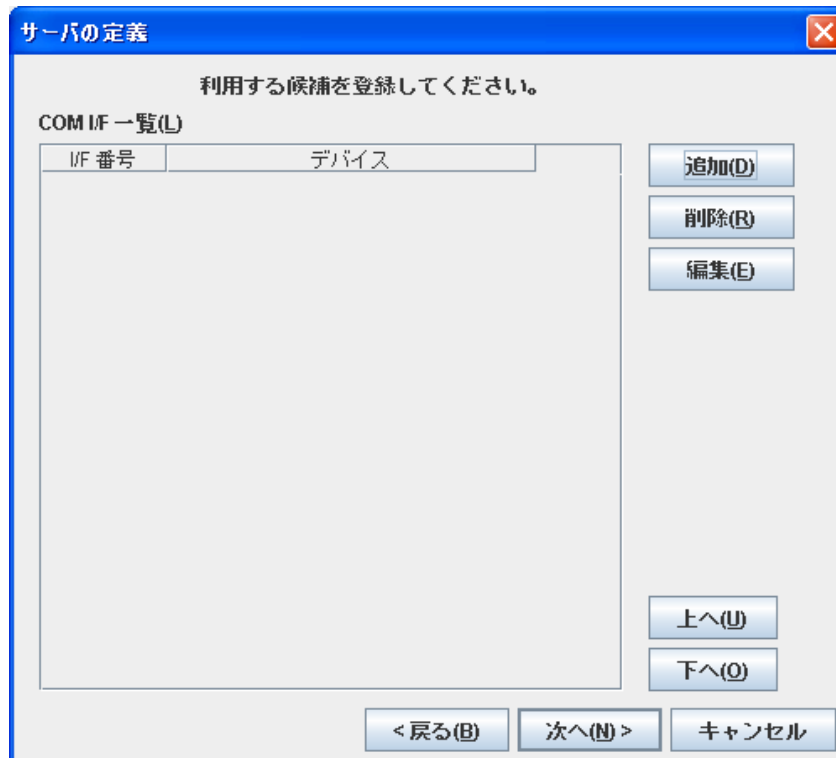
「OK」ボタンを選択すると設定内容がパブリック LAN/IF 一覧に設定されます。



「次へ」ボタンを選択して次の画面へ進みます。

3.5.4.4.COM ハートビートリソースの設定

1. サーバの定義

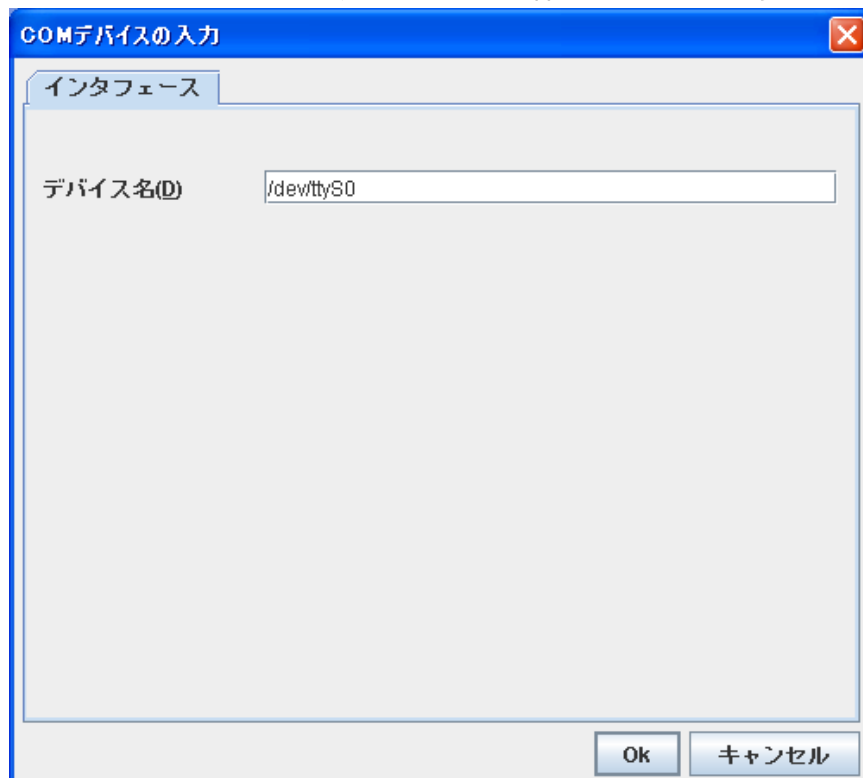


「追加」ボタンを選択

2. COM デバイスの入力

デバイス名 : /dev/ttyS0 (デフォルト値)

「OK」ボタンを選択すると設定内容が COM/IF 一覧に設定されます。



「次へ」ボタンを選択して次の画面へ進みます。

3.5.4.5.ミラーディスクコネクトリソースの設定

1. サーバの定義

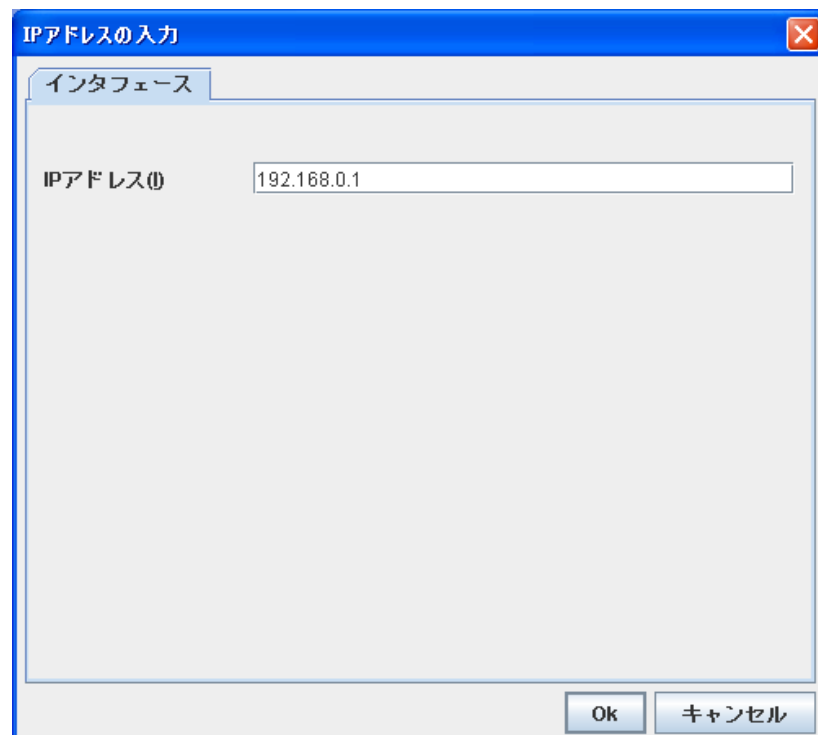


「追加」ボタンを選択

2. IP アドレスの入力

IP アドレス : 192.168.0.1

「OK」ボタンを選択すると設定内容がミラーディスクコネクタ/IF 一覧に設定されます



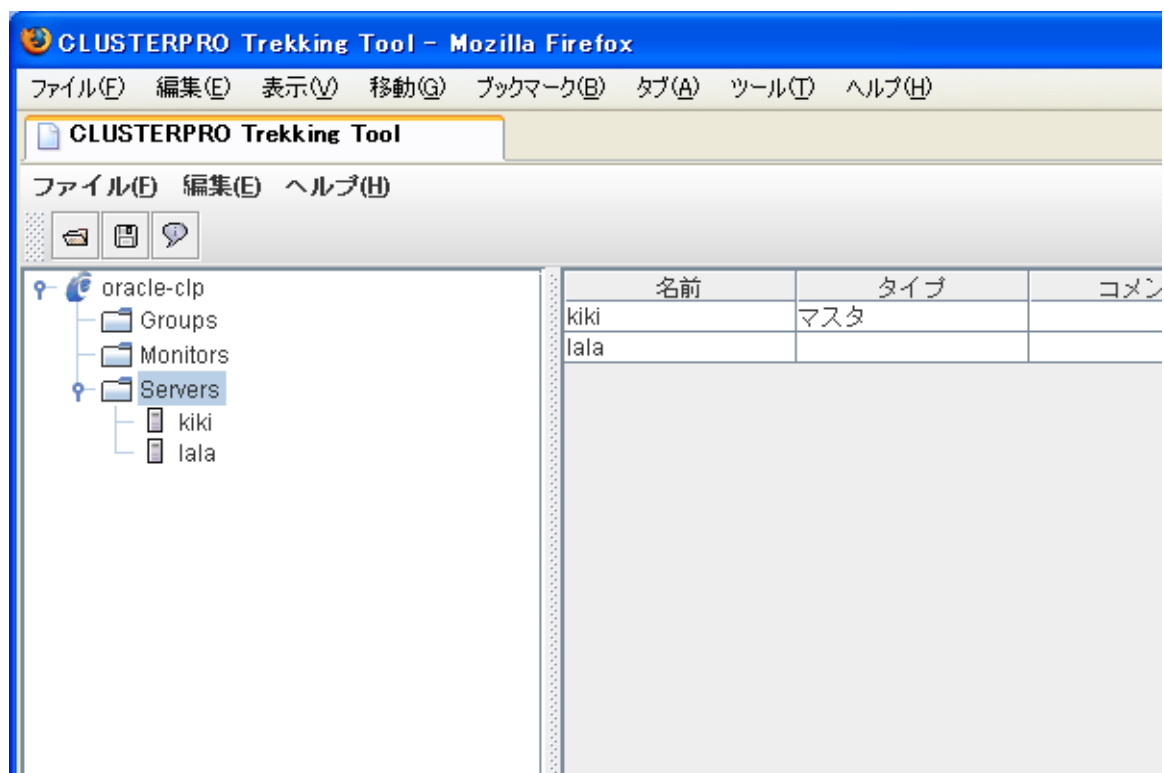
「次へ」ボタンを選択して次の画面へ進みます。

今回設定した以下の構成は、最も標準的であるが、インタコネクトが 1 本とミラーコネクトが 1 本あれば(共通でも可)、クラスタは生成可能です。

- ・NIC(LAN)は 2 本
- ・NIC :プライマリインタコネクト LAN 兼、ミラーコネクト
- ・NIC :セカンダリインタコネクト LAN 兼パブリック LAN

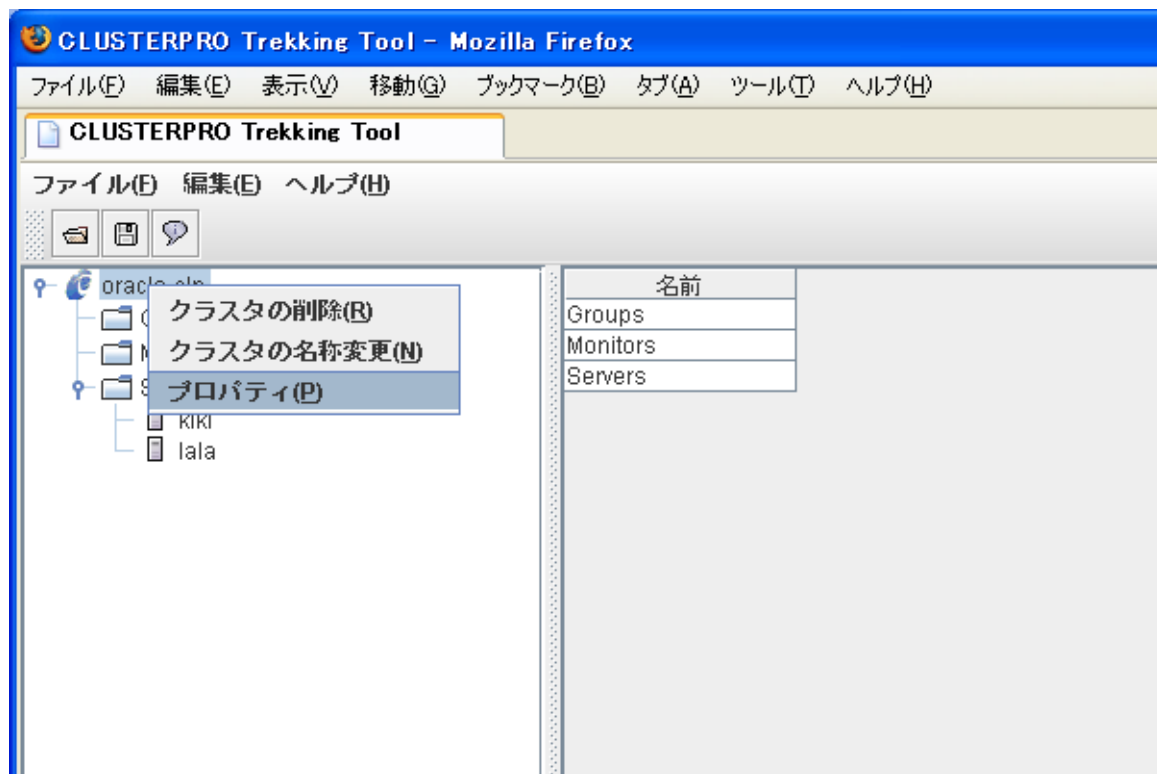
3.5.5.サーバの定義確認

2 台目のサーバについても同様の操作で、サーバ情報を定義します。なお、1 台目に設定したサーバがデフォルトでマスターサーバとなります。

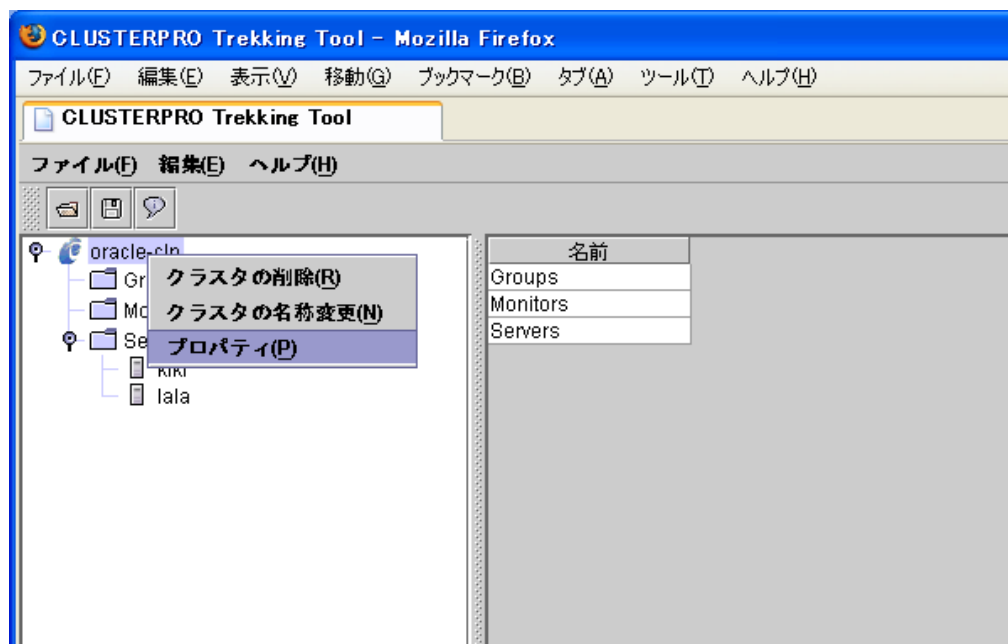


上図は、2 台のサーバを定義した状態です。

3.5.6.ハートビート I/F の設定



1. ツリービューの oracle-clp を右クリックして「プロパティ」を選択



- ツリービューの oracle-clp を右クリックして「プロパティ」を選択
「oracle-clp」クラスタのプロパティの「ハートビート I/F」タブを選択

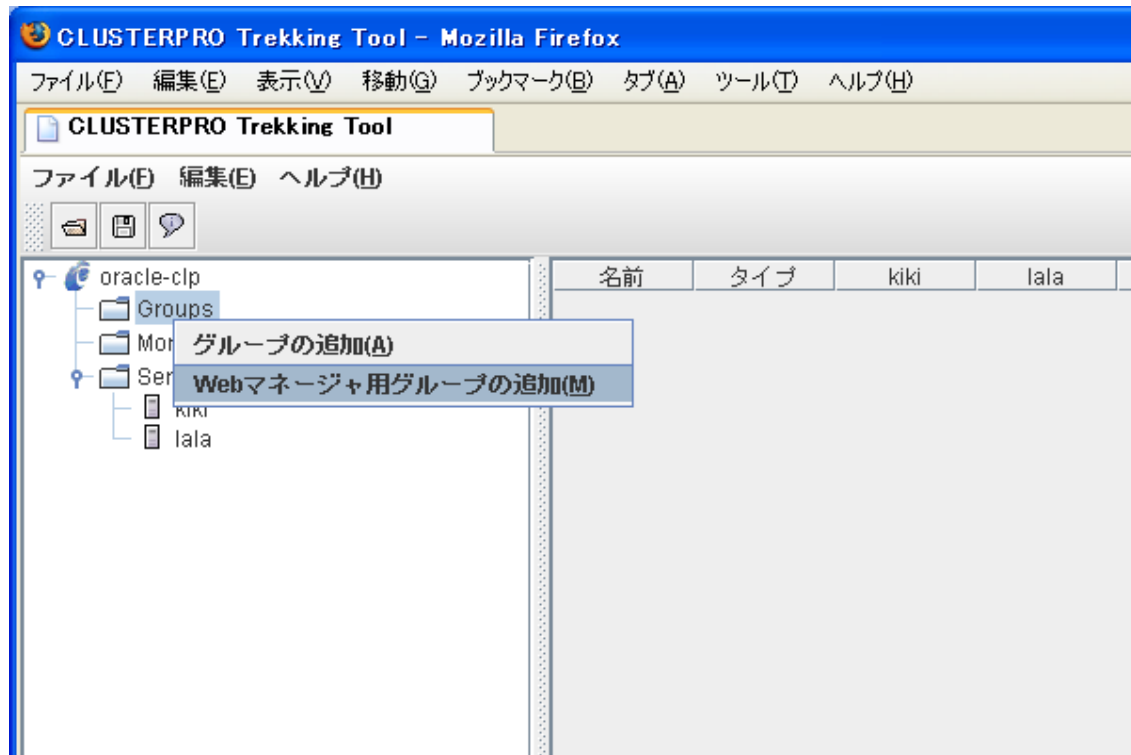


「<追加」、「削除>」、「上へ」、「下へ」ボタンを選択して「ハートビート I/F の優先順位一覧」を以下のように設定します。

順位 1	タイプ LAN	I/F 番号 1
順位 2	タイプ LAN	I/F 番号 2
順位 3	タイプ COM	I/F 番号 1

「OK」ボタンを選択してハートビートの優先順位設定を終了します

3.5.7.WebManager グループの追加



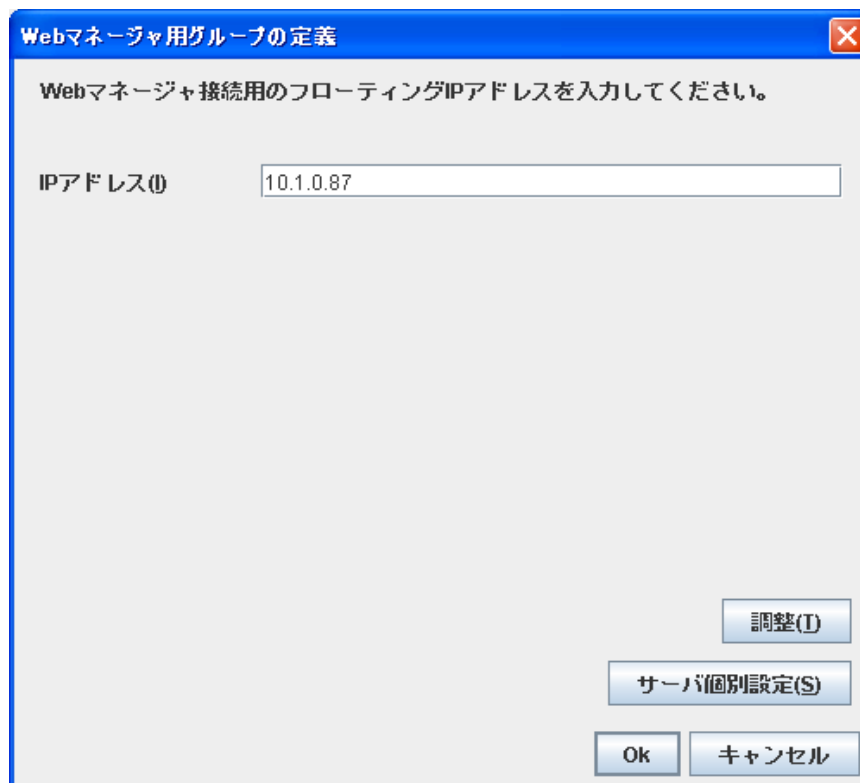
1. ツリービューの Groups を右クリックし「Web マネージャ用グループの追加」を選択

WebManager は、トレッキングツール同様 Web ブラウザ上で JavaApplet を使用し、クラスタ停止、サーバ停止、業務(グループ)の起動・停止などを行う GUI ツールです。この WebManager 用グループは、WebManager サーバ専用のグループで、フローティング IP アドレスのみのリソースを持つフェイルオーバーグループになります。

2. フローティング IP リソースの定義

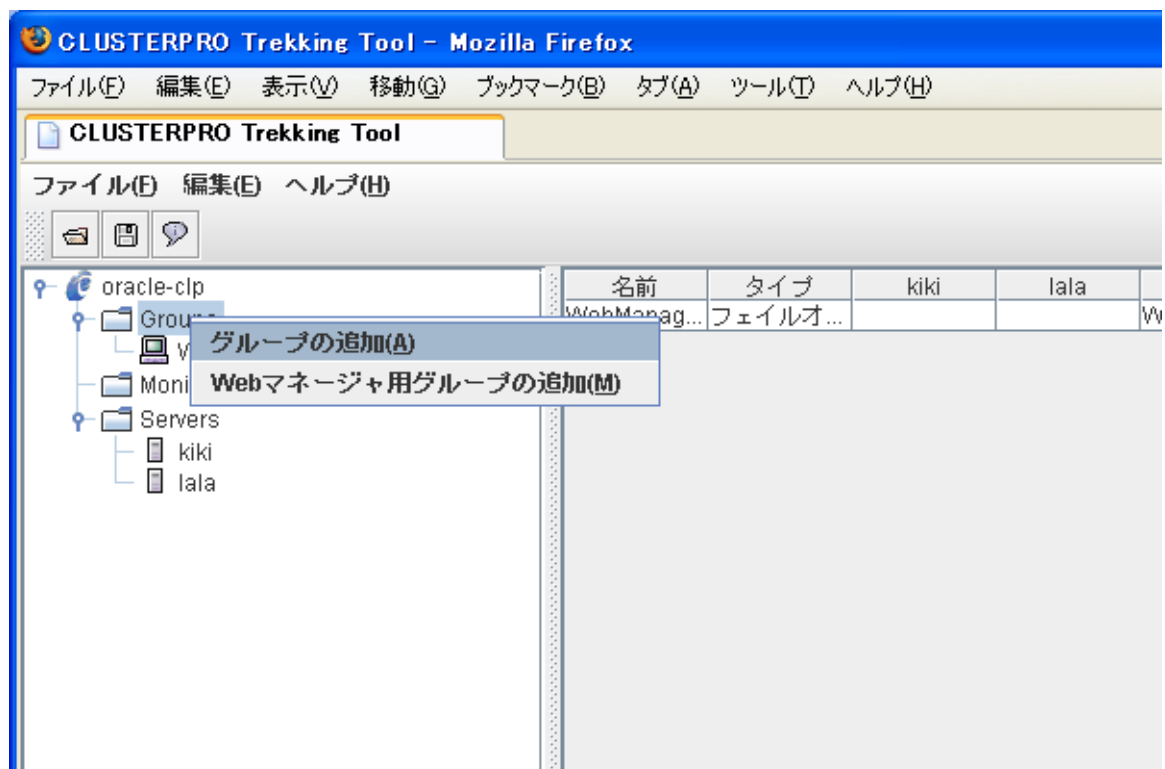
IP アドレス : 10.1.0.87

「OK」ボタンを選択して Web マネージャグループの追加を終了します。



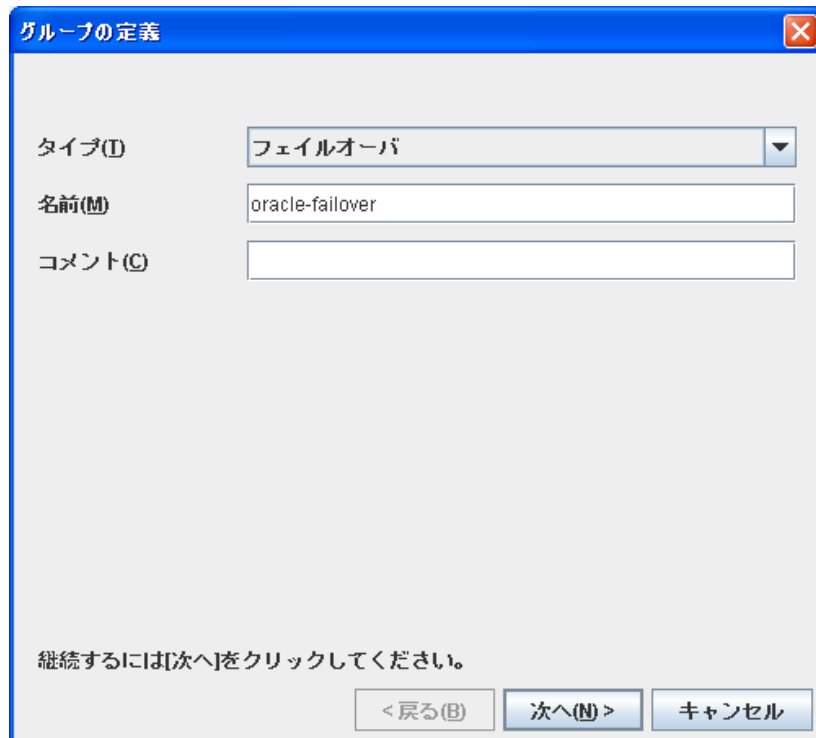
3.5.8. グループの追加

1. ツリービューの Groups を右クリックして「グループの追加」を選択



2. グループの定義

タイプ : フェイルオーバー
名前 : oracle-failover
コメント : 任意



グループの定義

タイプ(T) フェイルオーバー ▼

名前(M) oracle-failover

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

「次へ」ボタンを選択して次の画面へ進みます。

3. グループの定義

「<追加>」、「削除>」、「上へ」、「下へ」ボタンを選択して「起動可能なサーバ」を以下のように設定します。

順位 1 kiki
順位 2 lala

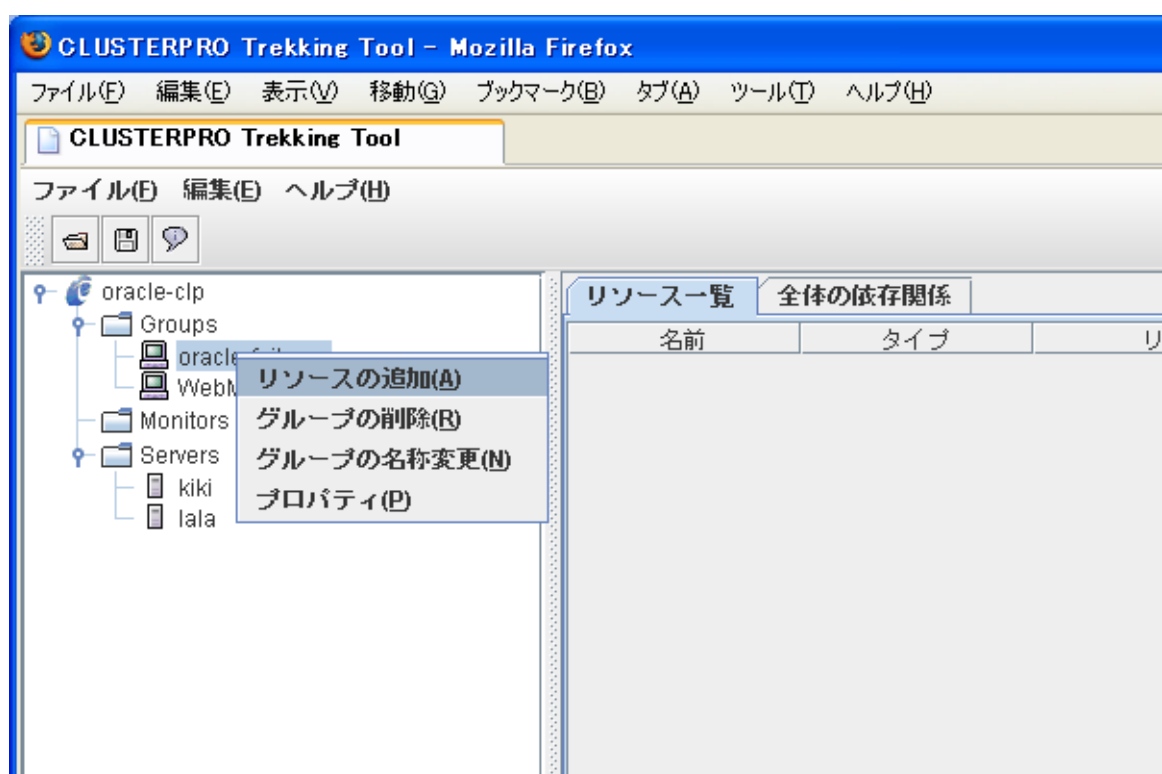
「完了」ボタンを選択してグループの定義を終了します。



このグループ起動優先度の最も高いサーバがその業務についての「現用系サーバ」である。

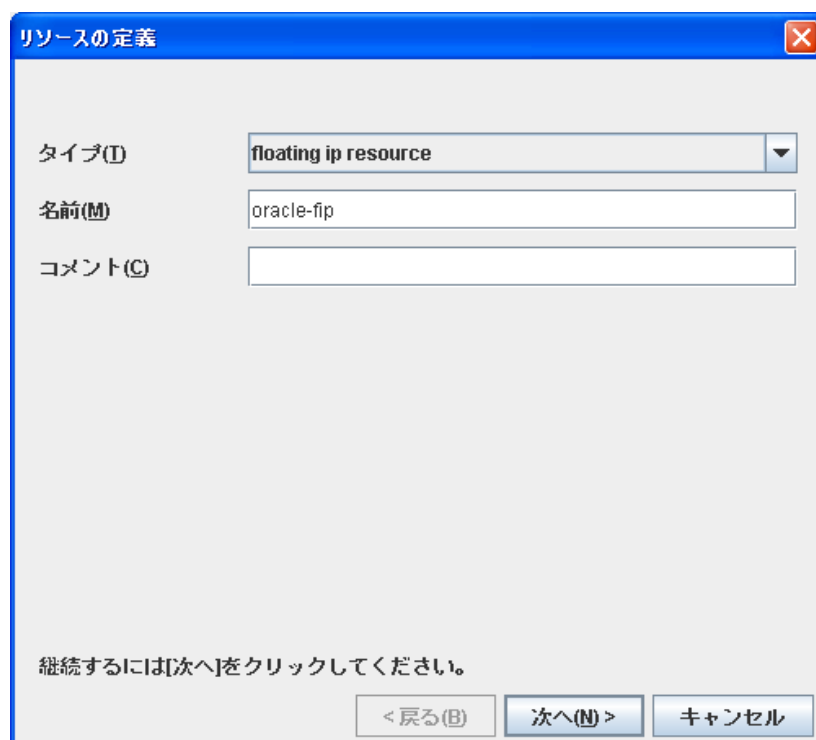
3.5.9.フローティング IP リソースの設定

1. ツリービューの Groups 配下の oracle-failover を右クリックして「リソースの追加」を選択



2. リソースの定義

タイプ : floating ip resource
名前 : oracle-fip
コメント : 任意



リソースの定義

タイプ(I) floating ip resource

名前(M) oracle-fip

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

「次へ」ボタンを選択して次の画面へ進みます。

3. リソースの定義 IP アドレス : 10.1.0.88



リソースの定義

IPアドレス(I) 10.1.0.88

調整(I)

サーバ個別設定(S)

< 戻る(B) 次へ(N) > キャンセル

「次へ」ボタンを選択して次の画面へ進みます。

MIRACLE LINUX V3.0/V4.0 は、デフォルトで NIC 二重化機能を実現する Bonding ドライバがあります。この NIC 二重化の Bonding 機能を利用する場合、NIC ドライバのデバイスは bond0 となります。CLUSTERPRO で Bonding 機能を使用する場合には、フローティング IP アドレスの値は、"10.1.0.88%bond0"と入力する必要があります。

4. リソースの定義

活性リトライしきい値 : 5

フェイルオーバーしきい値 : 1

最終動作 : 何もしない (次のリソースを活性しない)

非活性リトライしきい値 : 0

最終動作 : クラスターデーモン停止と OS シャットダウン

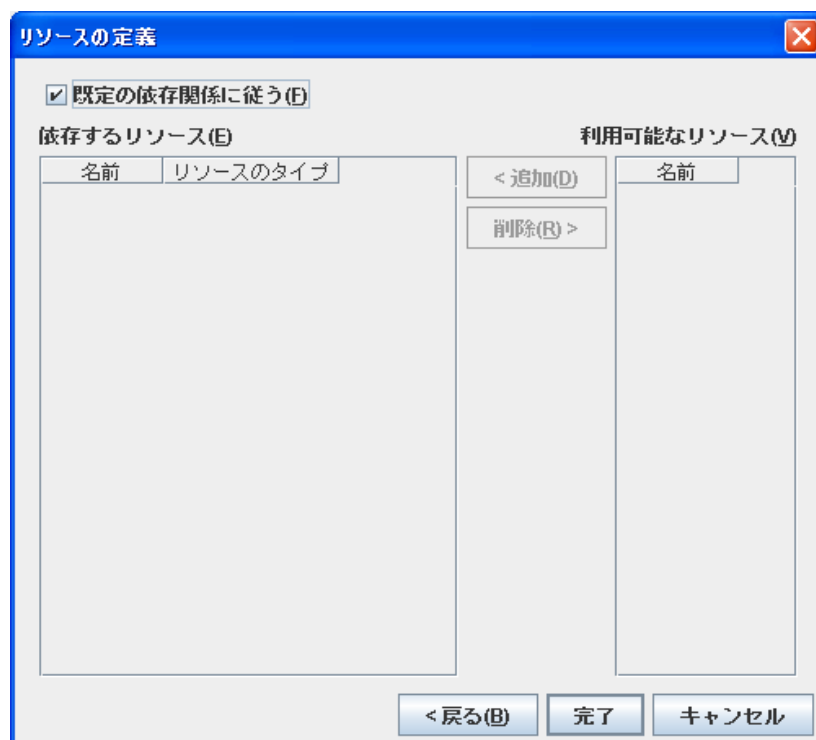
「次へ」ボタンを選択して次の画面へ進みます。

上記の設定の場合、フローティング IP アドレスの活性化に失敗すると、以下の動作となる。

1. 5 回までは再活性を試みる
2. 5 回の再活性に失敗すると相手サーバにフェイルオーバーする
3. 相手サーバ上でも活性に失敗した場合、5 回までは再活性を行う
4. 相手サーバ上での再活性が 5 回失敗した場合、フェイルバックしてくる
5. それでも起動に失敗すると、再活性もフェイルオーバーも実施済みなため、最終動作に移行する
6. 最終動作として、次の別リソースは起動せず、活性失敗状態のままとなる

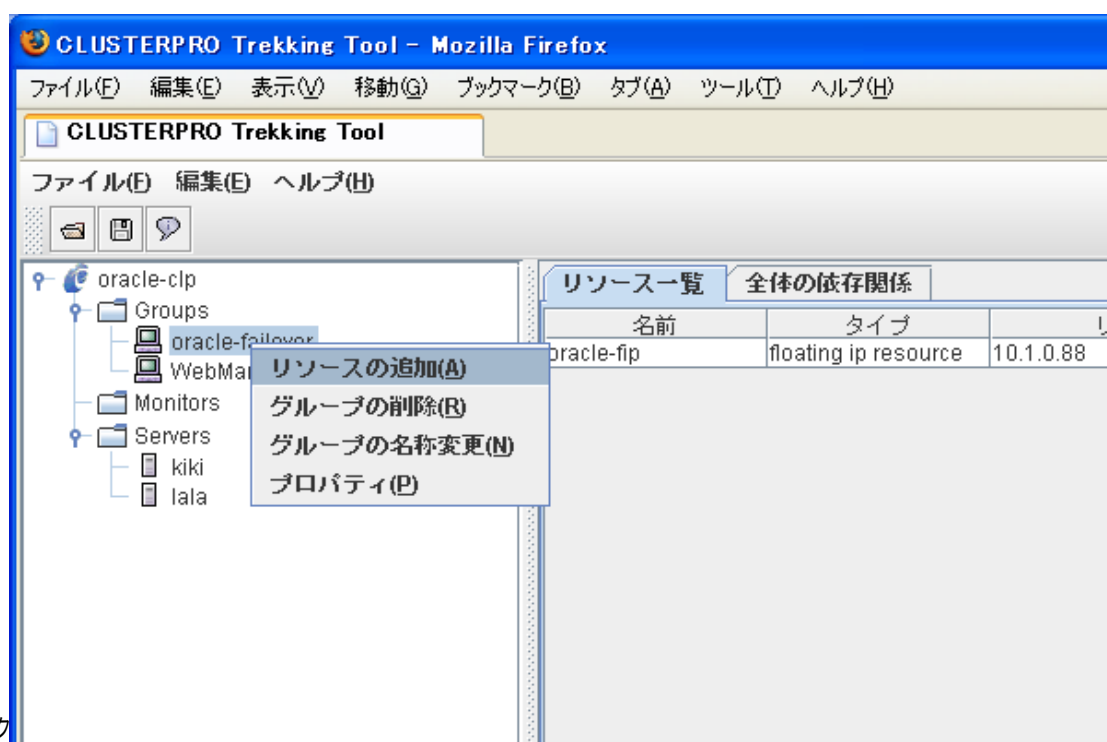
5. リソースの定義

「既定の依存関係に従う」チェックボックスにチェックが入っていることを確認し、「完了」ボタンを選択してフローティング IP アドレスリソースの設定を終了します。



3.5.10.ミラーディスクリソースの設定

1. ツリービューの Groups 配下の oracle-failover を右クリックして「リソースの追加」を選択



2. リソースの定義

タイプ : mirror disk resource
名前 : oracle-md
コメント : 任意



リソースの定義

タイプ(T) mirror disk resource ▼

名前(M) oracle-md

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

「次へ」ボタンを選択して次の画面へ進みます。

3. リソースの定義

ミラーパーティションデバイス名 : /dev/NMP1
ミラーマウントポイント : /oracle
データパーティションデバイス名 : /dev/sdb2
クラスタパーティションデバイス名 : /dev/sdb1
ディスクデバイス名 : /dev/sdb
ファイルシステム : ext3
ミラーデータポート番号 : 29051

リソースの定義

ミラーパーティションデバイス名(M)

ミラーマウントポイント(M)

データパーティションデバイス名(D)

クラスタパーティションデバイス名(C)

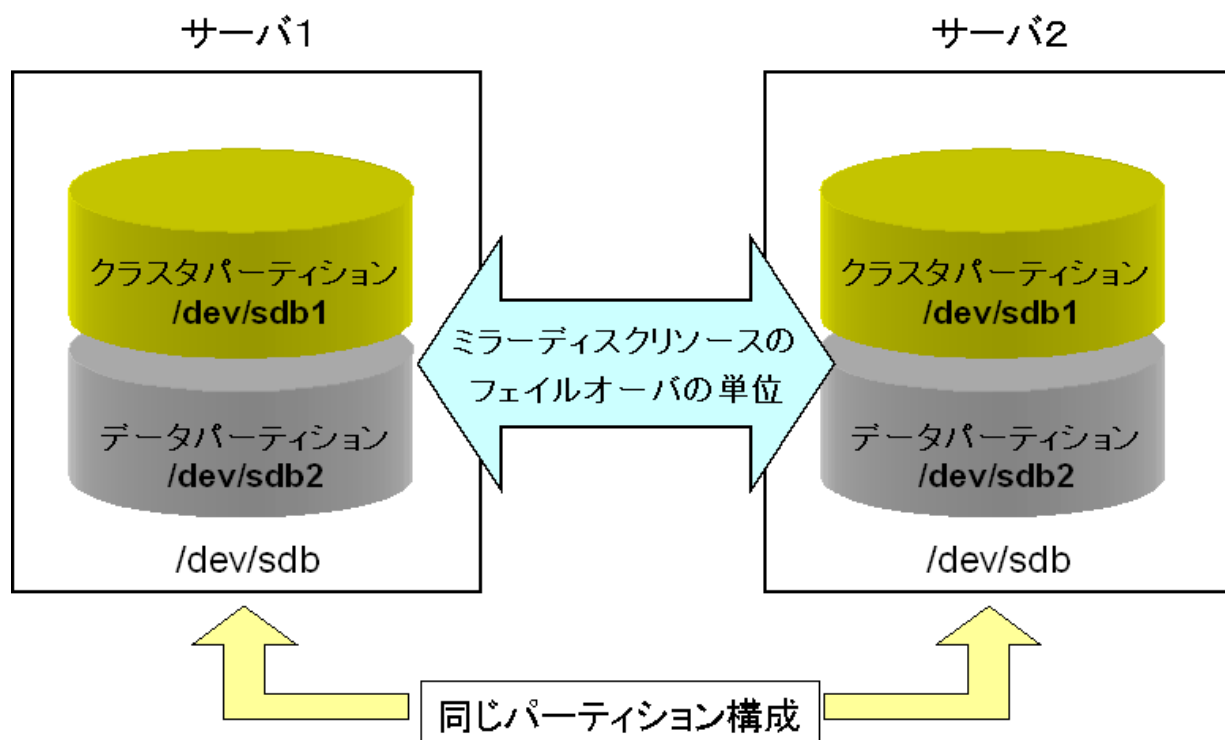
ディスクデバイス名(K)

ファイルシステム(F)

ミラーデータポート番号(R)

「調整」ボタンを選択して次の画面へ進みます。

< ポイント解説 > ミラーディスクリソース

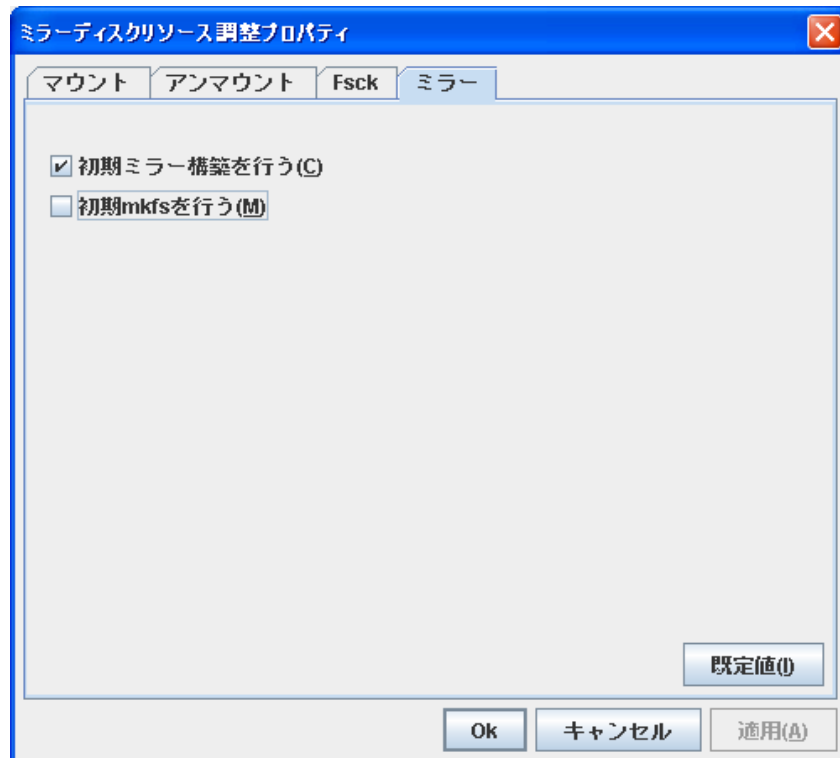


ミラーディスクは、クラスタシステムを構成する2台のサーバ間でデータのミラーリングを行う一組のデバイスのことです。ミラーディスクとして使用するためには、ディスクのタイプやパーティション構成を同じにする必要があります。

また、OSで使用しているディスクはミラーディスクには出来ません。別途ミラーディスク用のディスクを増設する必要があります。

4. ミラーディスクリソース調整プロパティ

「初期 mkfs を行う」チェックボックスにチェックをはずし、「OK」ボタンを選択して次の画面に進みます。



今回はプライマリサーバ(kiki)のデータパーティションに事前にファイルシステムを作成し、データベースを作成しています。初期起動時には、ミラー構築だけを行うように設定します。

5. リソースの定義

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

リソースの定義

活性異常検出時の復旧動作

活性リトライしきい値(R) 回

フェイルオーバーしきい値(I) 回

最終動作(F) 何もしない(次のリソースを活性しない)

非活性異常検出時の復旧動作

非活性リトライしきい値(E) 回

最終動作(I) クラスターデーモン停止とOSシャットダウン

< 戻る(B) 次へ(N) > キャンセル

「次へ」ボタンを選択して次の画面へ進みます。

6. リソースの定義

「既定の依存関係に従う」チェックボックスにチェックが入っていることを確認し、「完了」ボタンを選択してフローティング IP アドレスリソースの設定を終了します。

リソースの定義

☒ 既定の依存関係に従う(E)

依存するリソース(E)

名前	リソースのタイプ
--	floating ip resource

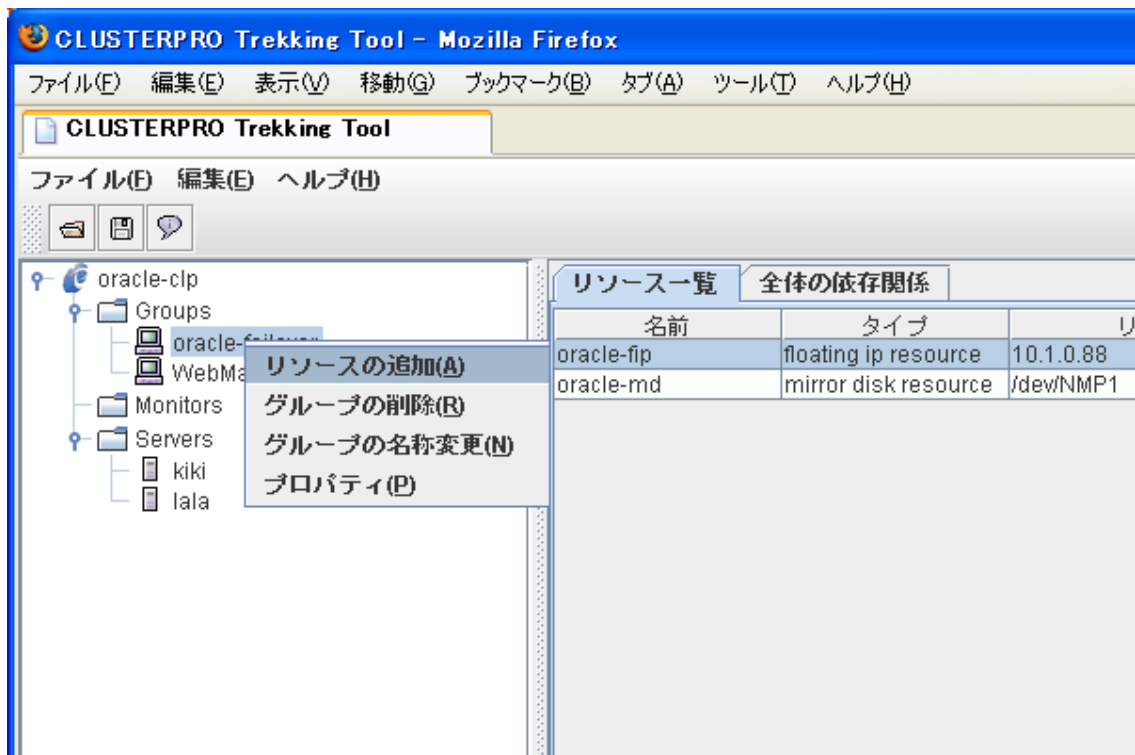
利用可能なリソース(M)

< 追加(D) 削除(R) >

< 戻る(B) 完了 キャンセル

3.5.11.EXEC リソース（Oracle 起動/停止）の設定

1. ツリービューの Groups 配下の oracle-failover を右クリックして「リソースの追加」を選択



2. リソースの定義

タイプ : execute resource
名前 : oracle-start
コメント : 任意

The 'リソースの定義' (Define Resource) dialog box is shown. It has three input fields:

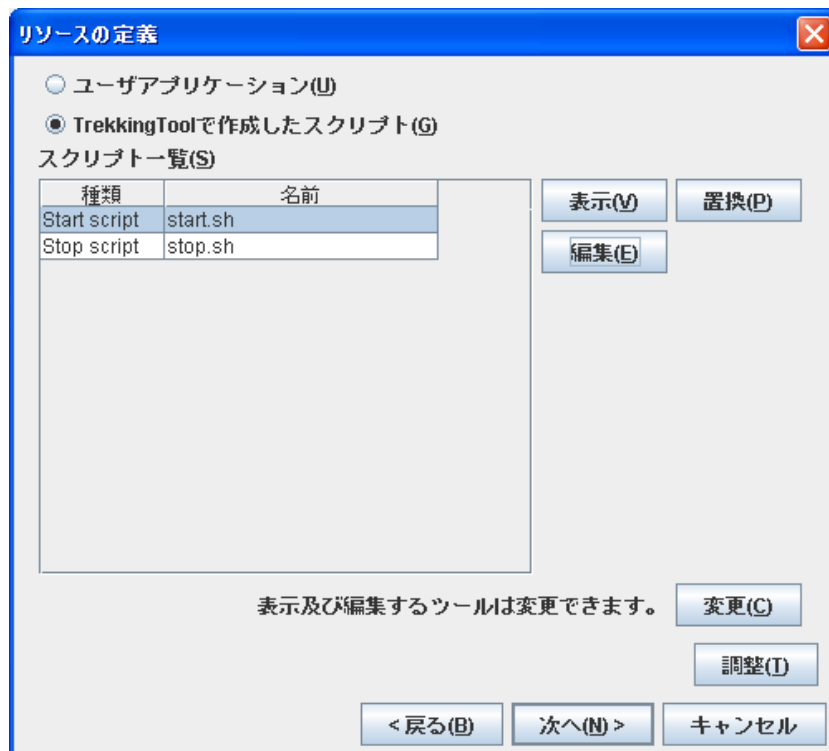
- タイプ(T): execute resource (selected in a dropdown)
- 名前(M): oracle-start
- コメント(C): (empty)

At the bottom, there is a message: '継続するには次へをクリックしてください。' (Click Next to continue). Below this are three buttons: '< 戻る(B)' (Back), '次へ(F) >' (Next), and 'キャンセル' (Cancel).

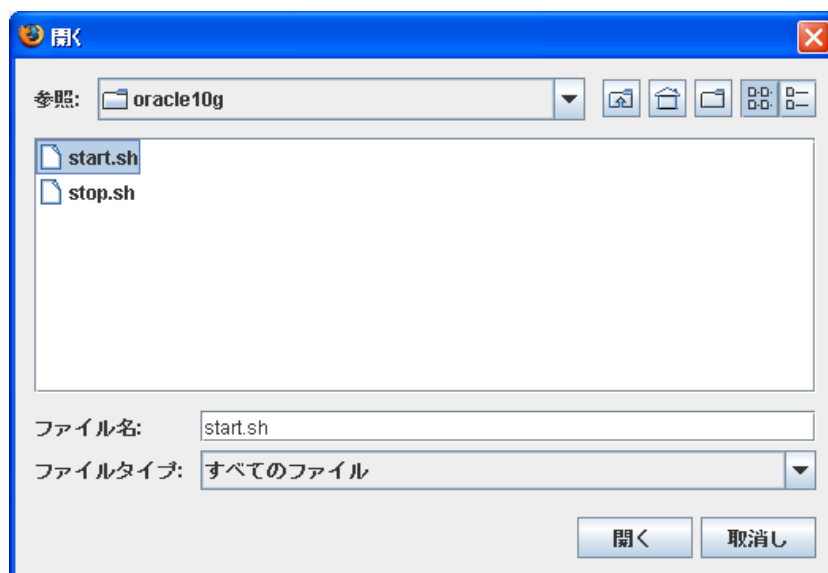
「次へ」ボタンを選択して次の画面へ進みます。

3. リソースの定義

「Trekking Tool で作成したスクリプト」チェックボックスにチェックが入っていることを確認します。



4. 「置換」ボタンを押し、デフォルトで提供されている起動/停止スクリプト(start.sh/stop.sh)を Oracle10g DB 起動/停止用スクリプトに変更します。



トレッキングツール用のデータベース監視 オプションをインストールすると、Oracle10g DB 起動/停止用のスクリプトが次のディレクトリに作成されます。

Windows : 「C:\Program Files\nec\clptrek\scripts\linux\oracle10g」

Linux : 「/opt/nec/clptrek/scripts/linux/oracle10g」

5. 置換した Oracle10g DB 起動/停止用スクリプトの start.sh, stop.sh を[編集]ボタンで vi(Linux 版)、notepad(Windows 版)で開き内容を編集します。

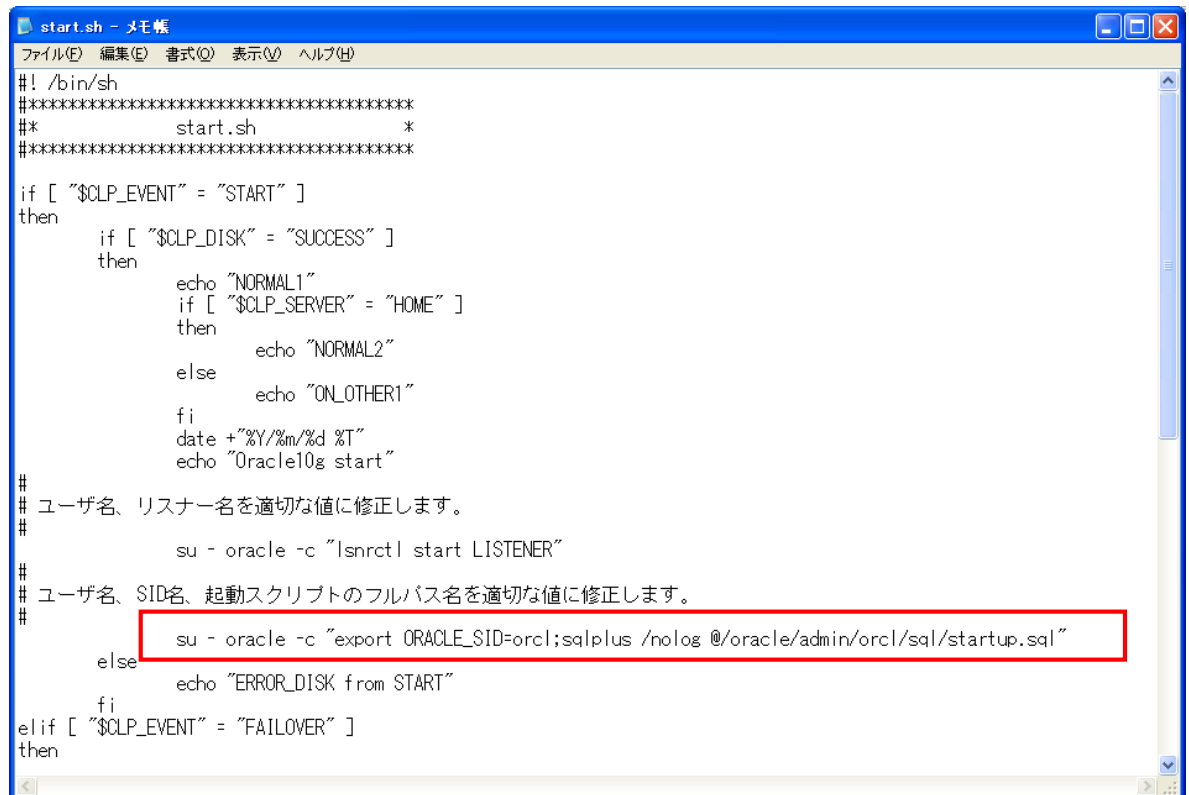
Oracle Database を起動/停止を行っている箇所を次のように変更する必要があります。

各スクリプトには、変更の必要がある箇所が 2 箇所ありますので注意してください。

< start.sh >

```
su - oracle -c "export ORACLE_SID=orcl;sqlplus /nolog @/xxxx/startup.sql"
```

```
su - oracle -c "export ORACLE_SID=orcl;sqlplus /nolog @/oracle/admin/orcl/sql/startup.sql"
```



```
start.sh - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
#!/bin/sh
#####
##                                *
#####

if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
        date +%Y/%m/%d %T
        echo "Oracle10g start"
    #
    # ユーザ名、リスナー名を適切な値に修正します。
    #
        su - oracle -c "lsnrctl start LISTENER"
    #
    # ユーザ名、SID名、起動スクリプトのフルパス名を適切な値に修正します。
    #
        su - oracle -c "export ORACLE_SID=orcl;sqlplus /nolog @/oracle/admin/orcl/sql/startup.sql"
    else
        echo "ERROR_DISK from START"
    fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
```

< stop.sh >

```
su - oracle -c "export ORACLE_SID=orcl;sqlplus /nolog @/xxxx/startup.sql"
```

```
su - oracle -c "export ORACLE_SID=orcl;sqlplus /nolog @/oracle/admin/orcl/sql/startup.sql"
```

```
stop.sh - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
#!/bin/sh
#*****
#*          stop.sh          *
#*****

if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
        date +"%Y/%m/%d %T"
        echo "Oracle10g stop"
    fi
# ユーザ名、SID名、終了スクリプトのフルパス名を適切な値に修正します。
#
# su - oracle -c "export ORACLE_SID=orcl;sqlplus /nolog @/oracle/admin/orcl/sql/shutdown.sql"
# ユーザ名、リスナー名を適切な値に修正します。
#
# su - oracle -c "lsnrctl stop LISTENER"
#
# else
#     echo "ERROR_DISK from START"
# fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
```

6.

編集が完了したら、「次へ」ボタンを選択して次の画面へ進みます。

start.sh, stop.sh スクリプト内で実行されている startup.sql, shutdown.sql の SQL スクリプトは、「Oracle の設定」で作成した、Oracle Database の起動/停止用の SQL スクリプトになります。

7. リソースの定義

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

リソースの定義

活性異常検出時の復旧動作

活性リトライしきい値(R) 回

フェイルオーバーしきい値(I) 回

最終動作(E) 何もしない(次のリソースを活性しない)

非活性異常検出時の復旧動作

非活性リトライしきい値(E) 回

最終動作(I) クラスターデーモン停止とOSシャットダウン

< 戻る(B) 次へ(N) > キャンセル

「次へ」ボタンを選択して次の画面へ進みます。

8. リソースの定義

「既定の依存関係に従う」チェックボックスにチェックが入っていることを確認
「完了」ボタンを選択してフローティング IP アドレスリソースの設定を終了します。

リソースの定義

☒ 既定の依存関係に従う(E)

依存するリソース(E)

名前	リソースのタイプ
--	floating ip resource
--	mirror disk resou...
--	nas resource

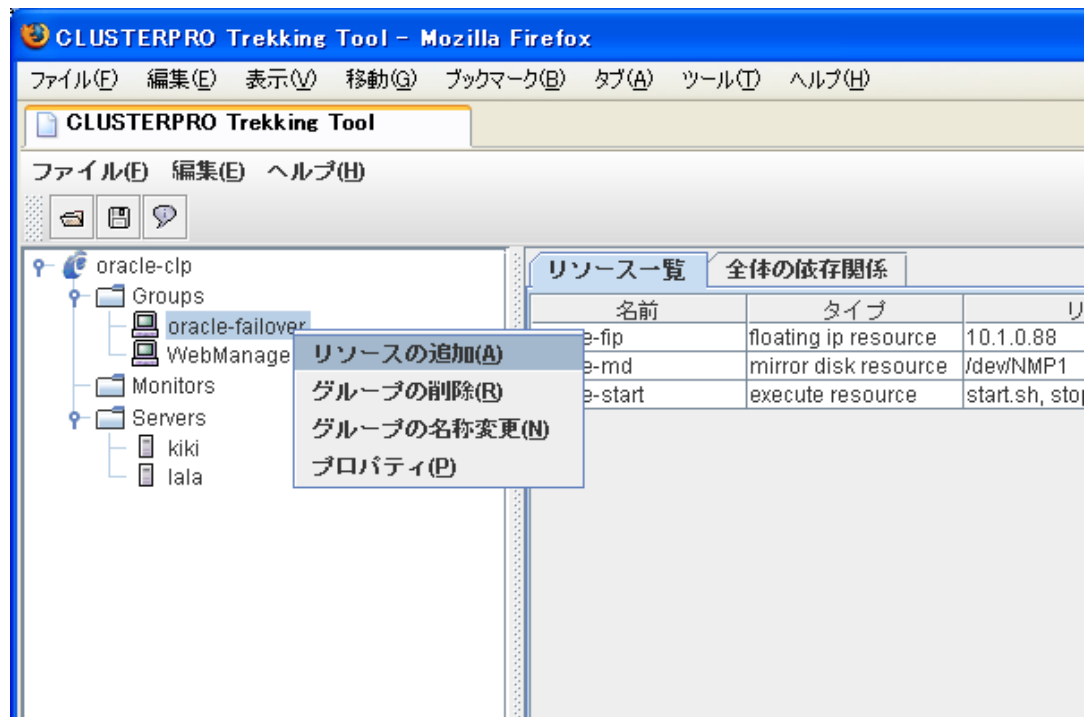
利用可能なリソース(M)

< 追加(D) 削除(R) >

< 戻る(B) 完了 キャンセル

3.5.12.EXEC リソース（Oracle 監視プログラム起動/停止）の設定

1. ツリービューの Groups 配下の oracle-failover を右クリックして「リソースの追加」を選択



2. リソースの定義

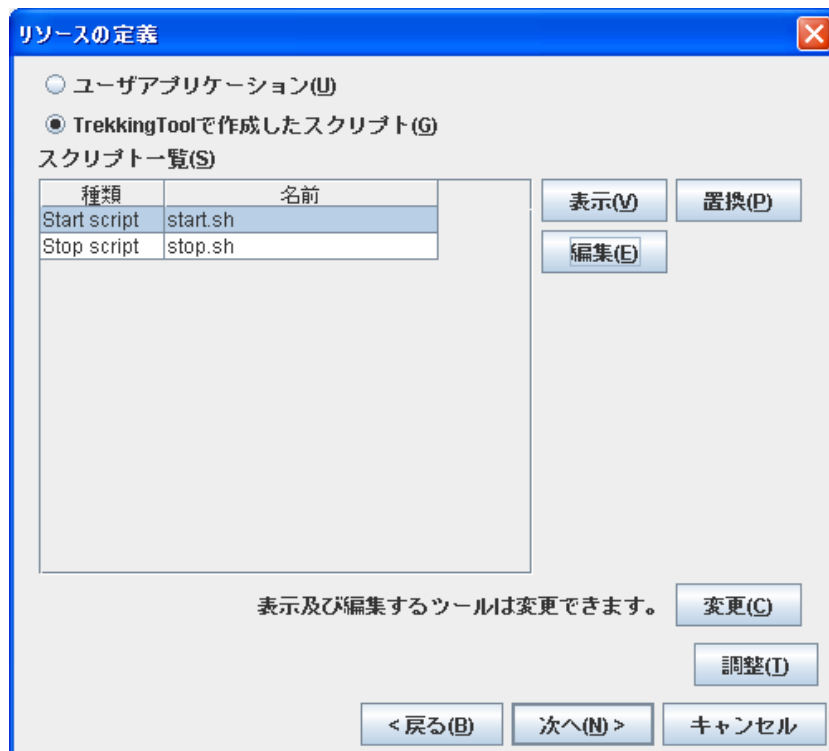
タイプ : execute resource
名前 : oracle-monitor
コメント : 任意



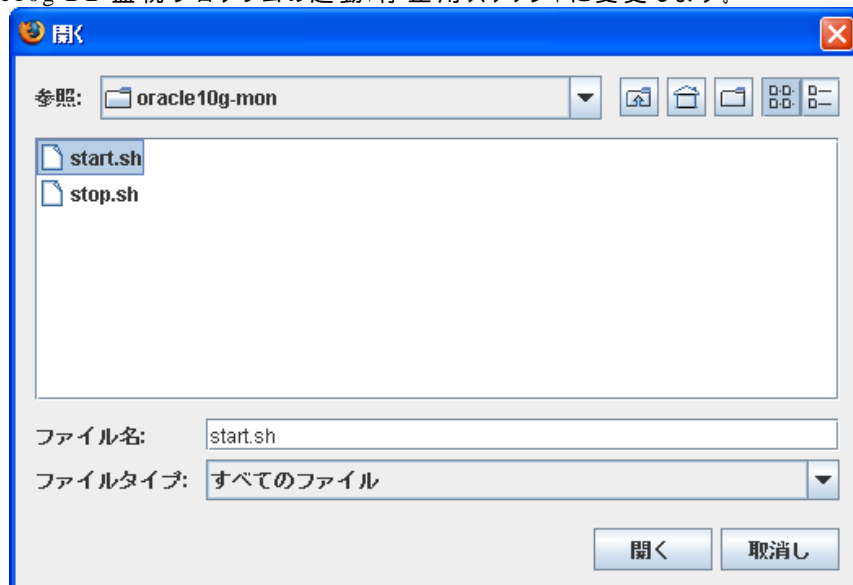
「次へ」ボタンを選択して次の画面へ進みます。

3. リソースの定義

「Trekking Tool で作成したスクリプト」チェックボックスにチェックが入っていることを確認します。



4. 「置換」ボタンを押し、デフォルトで提供されている起動/停止スクリプト(start.sh/stop.sh)を Oracle10g DB 監視プログラムの起動/停止用スクリプトに変更します。



トレッキングツール用のデータベース監視オプションをインストールすると、Oracle10g DB 監視プログラムの起動/停止用のスクリプトが次のディレクトリに作成されます。

Windows : 「C:\Program Files\nec\clptrek\scripts\linux\oracle10g-mon」

Linux : 「/opt/nec/clptrek/scripts/linux/oracle10g-mon」

5. 置換した Oracle10g DB 監視用プログラムの起動/停止用スクリプトの start.sh, stop.sh を[編集] ボタンで vi(Linux 版)、notepad(Windows 版)で開き内容を編集します。

Oracle Database 監視の起動/停止を行っている箇所を次のように変更する必要があります。

各スクリプトには、変更の必要がある箇所が 2 箇所ありますので注意してください。

< start.sh >

```
export ORACLE_HOME=/u01/app/oracle/product/10.1.0/db_1
clp_ora10mon orawatch -d xxxx
```

```
ex port ORACLE_HOME=/opt/app/oracle/product/10.1.0/db_1
clp_ora10mon orawatch -d oracle -u system -p miracle
```

The screenshot shows two Notepad++ windows. The top window is titled 'start.sh - メモ帳' and contains the following script content:

```
*****
**          start.sh          *
*****

if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
        date +%Y/%m/%d %T"
        echo "clp_ora10mon start"
    #
    # ORACLE_HOMEを適切な値に修正します。
    #
    export ORACLE_HOME=/opt/app/oracle/product/10.1.0/db_1
    export LD_LIBRARY_PATH=$ORACLE_HOME/lib
    #
    # NLS_LANG/パラメータを適切な値に修正します。
    #
    export NLS_LANG=AMERICAN_AMERICA.JA16EUC
    # データベース名を適切な値に修正します。
    #
    clp_ora10mon orawatch -d orcl -u system -p miracle
    else
        echo "ERROR_DISK from START"
    fi
fi

if [ "$CLP_DISK" = "SUCCESS" ]
then
    echo "NORMAL1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
        echo "NORMAL2"
    else
        echo "ON_OTHER1"
    fi
    #
    # ORACLE_HOMEを適切な値に修正します。
    #
    export ORACLE_HOME=/opt/app/oracle/product/10.1.0/db_1
    export LD_LIBRARY_PATH=$ORACLE_HOME/lib

    date +%Y/%m/%d %T"
    echo "clp_ora10mon stop"
    clp_ora10mon orawatch --stop
else
    echo "ERROR_DISK from START"
fi
```

The bottom window is titled 'stop.sh - メモ帳' and contains the following script content:

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

if [ "$CLP_EVENT" = "STOP" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
        date +%Y/%m/%d %T"
        echo "clp_ora10mon stop"
        clp_ora10mon orawatch --stop
    else
        echo "ERROR_DISK from START"
    fi
fi
```

Red boxes in the original image highlight the following lines in both scripts:

- In start.sh: `export ORACLE_HOME=/opt/app/oracle/product/10.1.0/db_1` and `clp_ora10mon orawatch -d orcl -u system -p miracle`
- In stop.sh: `clp_ora10mon orawatch --stop`

Oracle10g DB 監視用の clp_ora10mon コマンドのオプション解説は以下の通りです。

【監視の開始】

```
clp_ora10mon 識別子 -d 接続文字列 [-u ユーザ名] [-p パスワード] [-t テーブル名]
[-i 監視間隔] [-c リトライ回数] [-r 応答待ち時間]
```

【監視の終了】

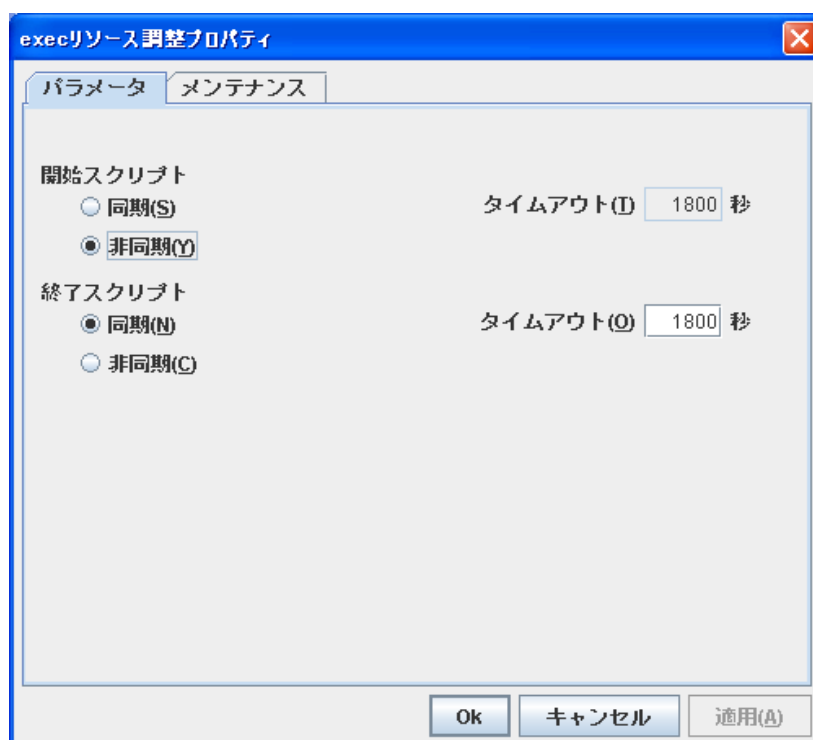
```
clp_ora10mon 識別子 --stop [終了待ち時間]
```

- + -d: 接続文字列 監視するデータベースに対応する接続文字列を指定します。(必須)
- + -u: ユーザ名 データベースにログインする際のユーザ名を指定します。(必須、規定値: sys)
- + -p: パスワード データベースにログインする際のパスワードを指定します。(必須、規定値: change_on_install)
- + -t: テーブル名 データベース上に作成する監視用テーブル名を指定します。(既定値: orawatch)
- + -i: 監視間隔 データベース監視の監視間隔(1 ~ 10000)を秒数で指定します。(既定値: 60)
- + -c: リトライ回数 データベース監視で障害を検出したときのリトライ回数(1 ~ 10000)を指定します。(既定値: 2)
- + -r: 応答待ち時間 データベース監視処理の応答待ち時間(1 ~ 10000)を秒数で指定します。(既定値: 120)
- + --stop: 監視コマンドを終了します。

2. リソース定義

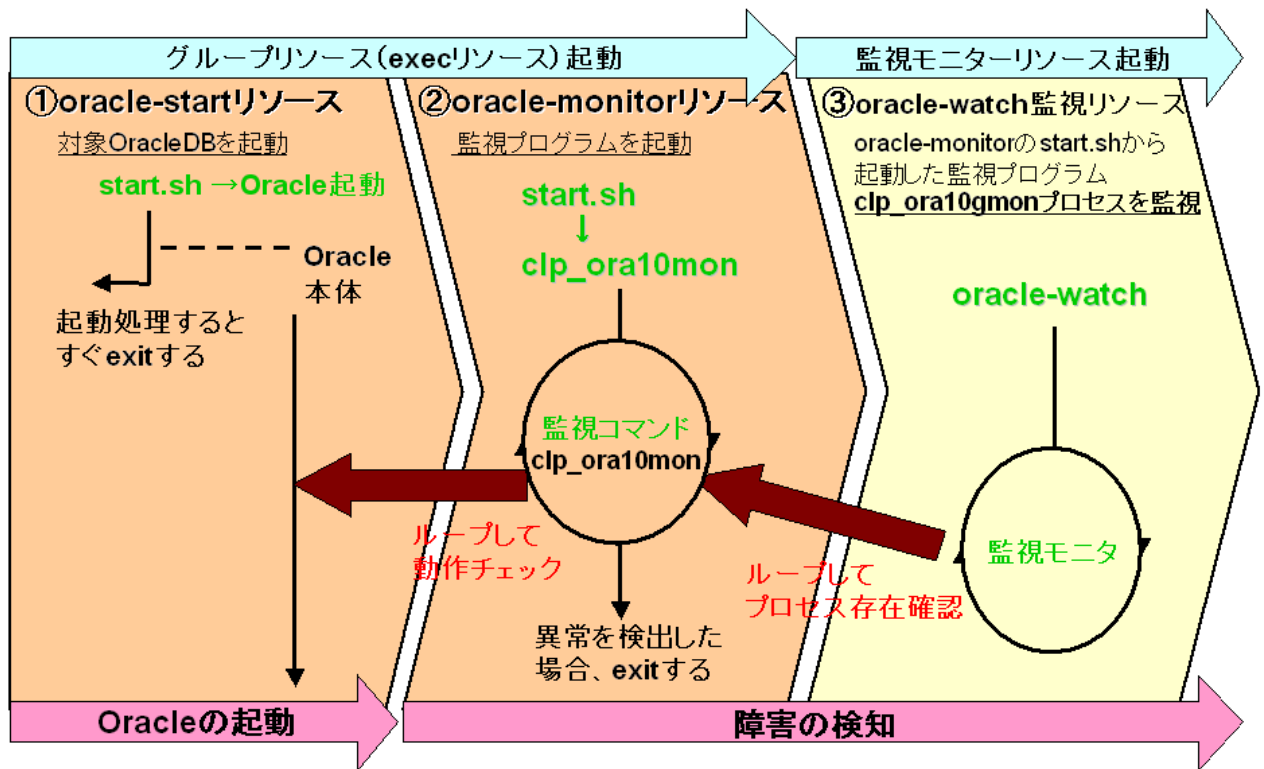
「調整」ボタンを選択し、開始スクリプトを「非同期」に設定します。

「OK」ボタンを選択して「exec リソース調整プロパティ」を終了します。



監視用の start.sh スクリプトは、スクリプト実行時にスクリプト終了を待たない「非同期」で設定する必要があります。EXEC リソースの start.sh スクリプトを非同期に設定したものだけ、PID モニタリソースからプロセス監視することが可能になります。

< ポイント解説 > アプリケーション監視の仕組み



アプリケーション (Oracle Database) の状況を監視する監視コマンド (clp_ora10mon) は、ループしながら定期的にアプリケーションの動作をチェックします。異常を検知したときに初めてループから exit し、監視スクリプトが終了するようになっています。

そのため clp_ora10mon コマンドを実行する start.sh スクリプトは、その実行終了を待たない「非同期」に指定する必要があります。

9. リソースの定義

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

リソースの定義

活性異常検出時の復旧動作

活性リトライしきい値(R) 回

フェイルオーバーしきい値(I) 回

最終動作(F)

非活性異常検出時の復旧動作

非活性リトライしきい値(E) 回

最終動作(I)

< 戻る(B) 次へ(N) > キャンセル

「次へ」ボタンを選択して次の画面へ進みます。

10. リソースの定義

「既定の依存関係に従う」チェックボックスにチェックを外し、「<追加>」、「削除>」ボタンを選択して、「oracle-start」を「依存するリソース」とします。

リソースの定義

☐ 既定の依存関係に従う(E)

依存するリソース(E)

名前	リソースのタイプ
oracle-start	execute resource

利用可能なリソース(M)

名前
oracle-fip
oracle-md

< 追加(D) 削除(B) >

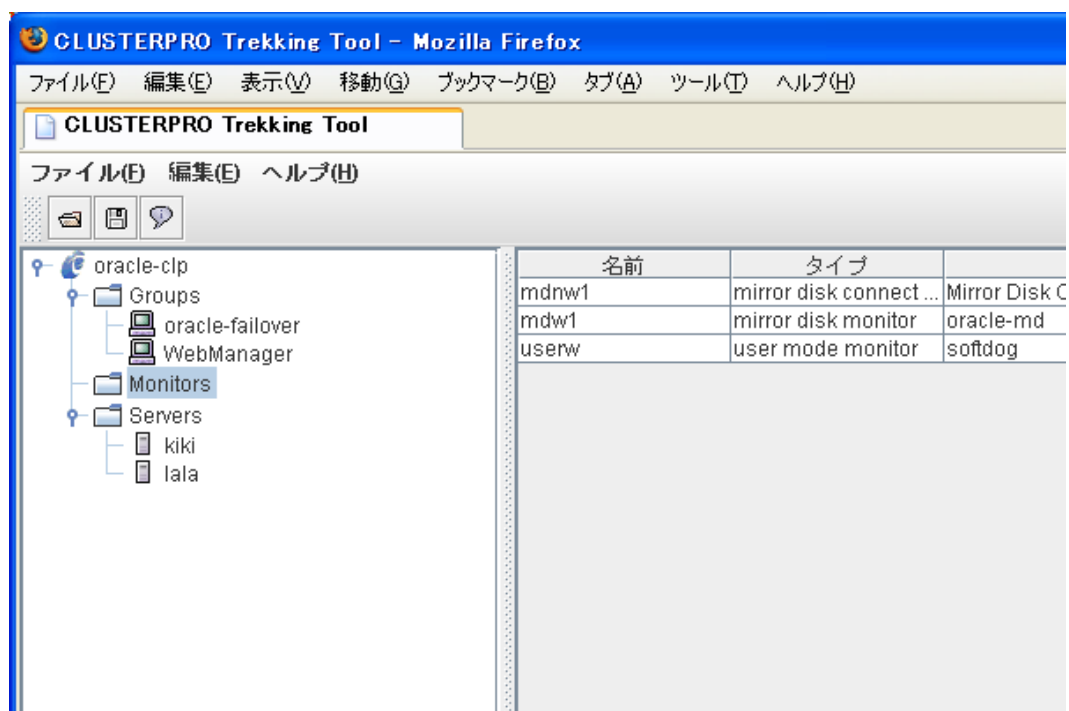
< 戻る(B) 完了 キャンセル

Oracle Database 監視の oracle-monitor は、“監視対象の Oracle Database が起動してから”監視を始めないと、起動前に監視を開始し、異常であると誤認してしまいます。そのため、Oracle Database 起動のに依存する設定が必要です。

3.5.13.モニターリソースの設定

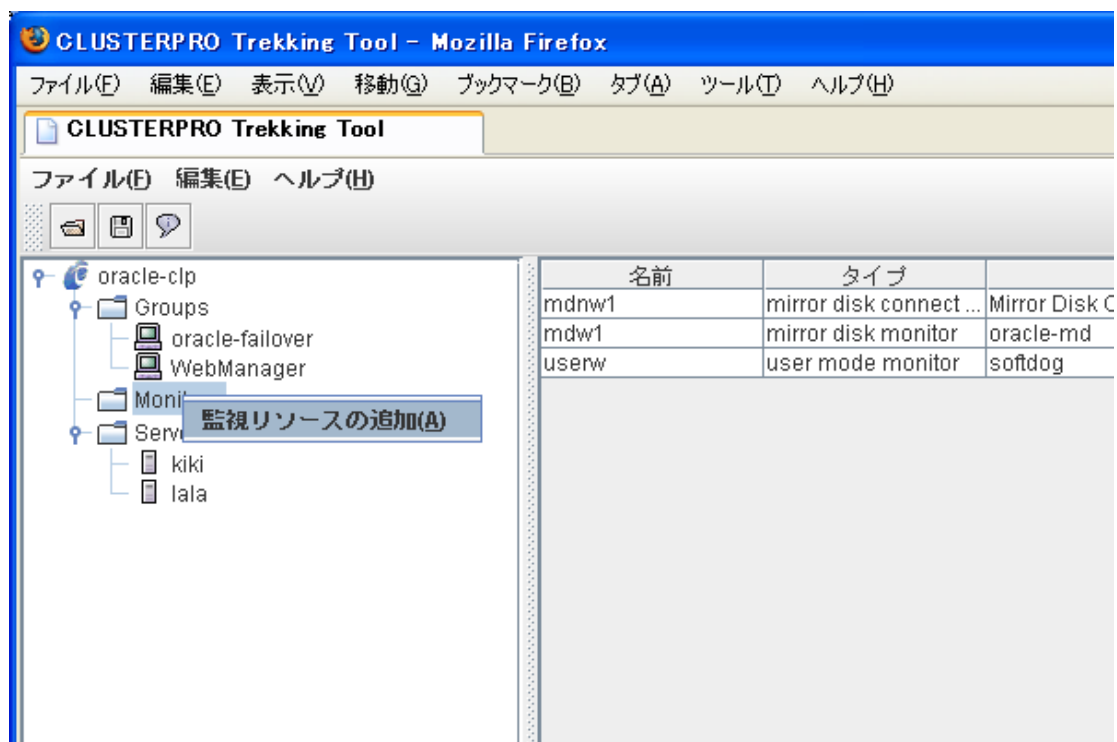
クラスタ定義時、およびミラーディスクリソースを設定した時点で以下のモニターリソースがデフォルトで定義されています。

- user mode monitor
- mirror disk connect monitor
- mirror disk monitor



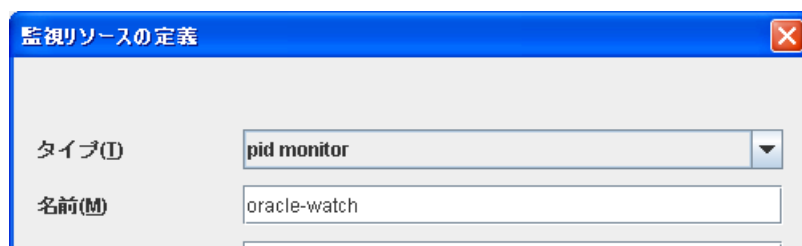
3.5.14.PID モニタリソースの設定

1. ツリービューの Monitor を右クリックして「監視リソースの追加」を選択



2. 監視リソースの定義

タイプ : pid monitor
 名前 : oracle-watch
 コメント : 任意



「次へ」ボタンを選択して次の画面へ進みます。

3. 監視リソースの定義

「インターバル」、「タイムアウト」、「リトライ回数」を確認し、「監視開始待ち時間」を 30 秒に設定し、対象リソースとして oracle-monitor を選択します



The dialog box titled "監視リソースの定義" (Define Monitoring Resource) contains the following fields and controls:

- インターバル(I): 5 秒
- タイムアウト(T): 60 秒
- リトライ回数(R): 0 回
- 監視開始待ち時間(S): 30 秒
- 監視タイミグ:
 - ☐ 常時(L)
 - ☒ 活性時(C)
- 対象リソース(O): oracle-monitor (with a 参照(W) button)
- nice値(E): A slider set to 0
- 監視を行うサーバを選択する (with a サーバ(R) button)
- Navigation buttons at the bottom: < 戻る(B), 次へ(N) >, キャンセル

「次へ」ボタンを選択して次の画面へ進みます。

このプロセス監視モニター(pidw)は、oracle-monitor リソースから Oracle Database 監視プログラムとして起動される clp_oral0mon プロセスの死活監視を行います。そのため、この clp_oral0mon が Oracle Database の監視を正常に開始するまでの猶予として 30 秒間「監視開始待ち時間」を設定しています

4. 監視リソースの定義

回復対象として oracle-start を選択し、以下の値を設定します。

回復対象	: exec1
再活性化しきい値	: 1
フェイルオーバーしきい値	: 1
最終動作	: 何もしない

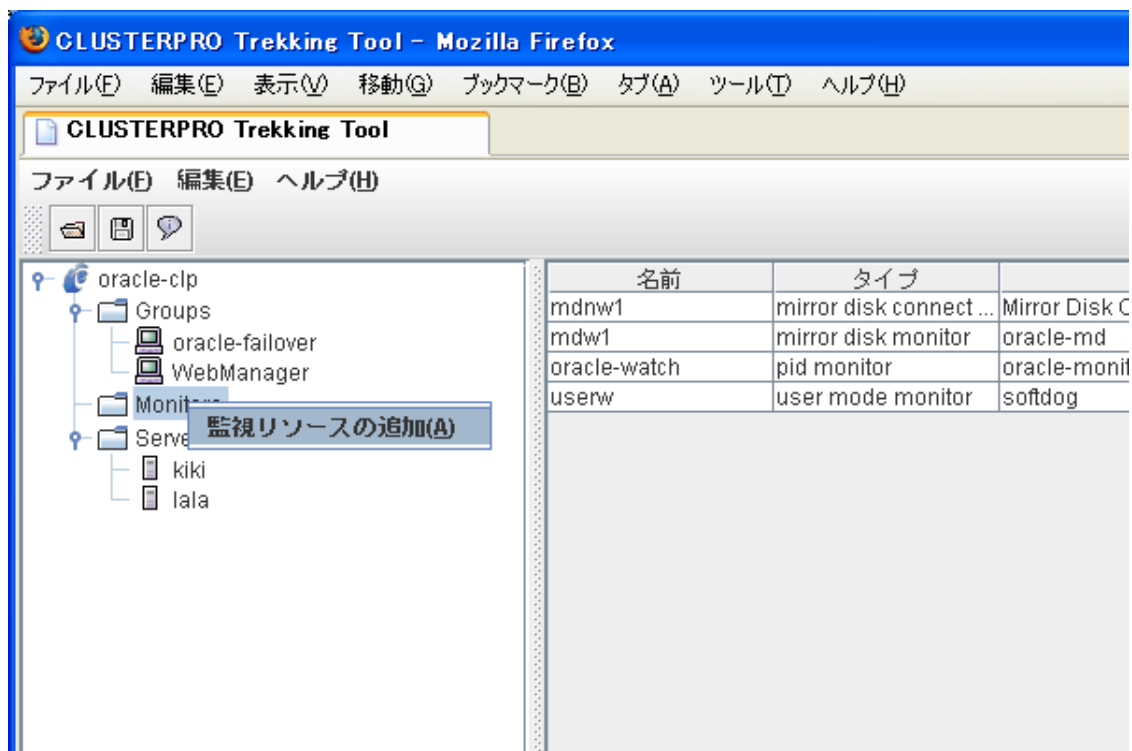
「完了」ボタンを選択して PID モニタリソースの設定を終了します。

各監視モニタは、回復対象として「グループリソース」、「グループ」、「クラスタ」のレベルで指定し、この回復対象に対してどのような処理を行うかを細かく規定することが可能です。

今回は、oracle-monitor(clp_oral0mon プロセス)の監視を行う PID 監視モニタでは、回復対象を oracle-start リソース(Oracle Database 起動/停止)とし、再活性化(再実行)を 1 度行うことで障害(ここでは Oracle Database の異常終了)を回復し、無用にフェイルオーバーを発生させないような設定に

3.5.15.ip モニタリソースの設定

1. ツリービューの Monitor を右クリックして「監視リソースの追加」を選択



2. 監視リソースの定義

タイプ : ip monitor
 名前 : ping_to_gw
 コメント : 任意

監視リソースの定義

タイプ(T)

ip monitor

名前(M)

ping_to_gw

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

「次へ」ボタンを選択して次の画面へ進みます。

3. 監視リソースの定義

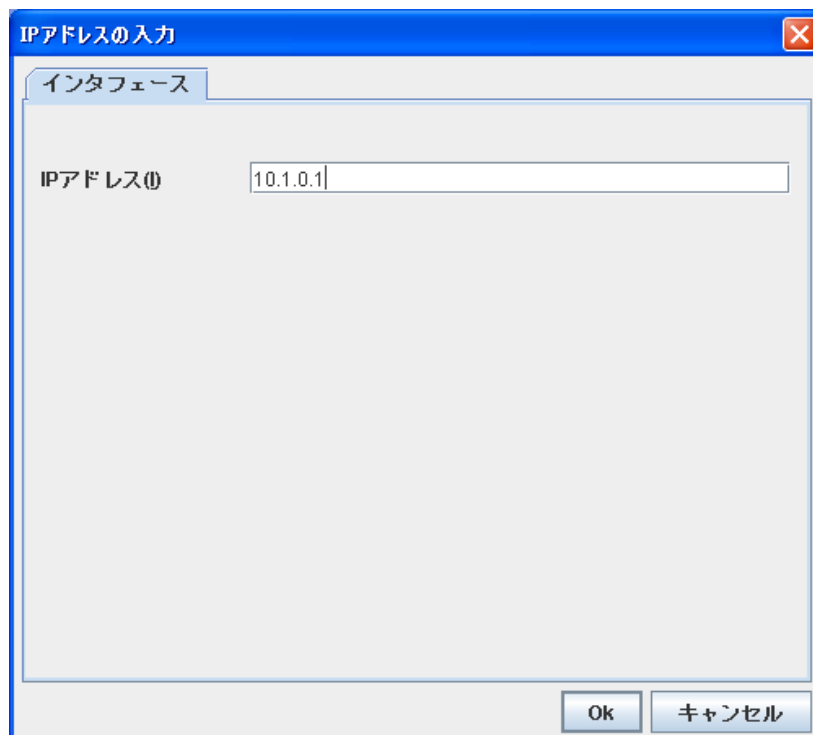
「追加」ボタンを選択して次の画面へ進みます。



4. IP アドレスの追加

IP アドレス : 10.1.0.1 (Gateway)

「OK」ボタンを選択すると定義内容が監視リソースの定義に追加されます。



「次へ」ボタンを選択して次の画面へ進みます。

5. 監視リソースの定義

「インターバル」、「タイムアウト」、「リトライ回数」、「監視開始待ち時間」を確認し、次の画面へ進みます。



監視リソースの定義 dialog box. The dialog has a blue title bar with the text "監視リソースの定義" and a close button. It contains the following fields and controls:

- インターバル(I): 30 秒
- タイムアウト(T): 30 秒
- リトライ回数(R): 0 回
- 監視開始待ち時間(S): 0 秒
- 監視タイミング:
 - ☒ 常時(L)
 - ☐ 活性時(C)
- 対象リソース(G): [empty text box] 参照(W)
- nice値(E): [slider bar] 0
- 監視を行うサーバを選択する: サーバ(R) button
- Navigation buttons at the bottom: < 戻る(B), 次へ(N) >, キャンセル

6. 監視リソースの定義

回復対象として oracle-failover を選択し、以下の値を設定します。

回復対象 : oracle-failover
再活性化しきい値 : 0
フェイルオーバーしきい値 : 1
最終動作 : 何もしない



監視リソースの定義 dialog box. The dialog has a blue title bar with the text "監視リソースの定義" and a close button. It contains the following fields and controls:

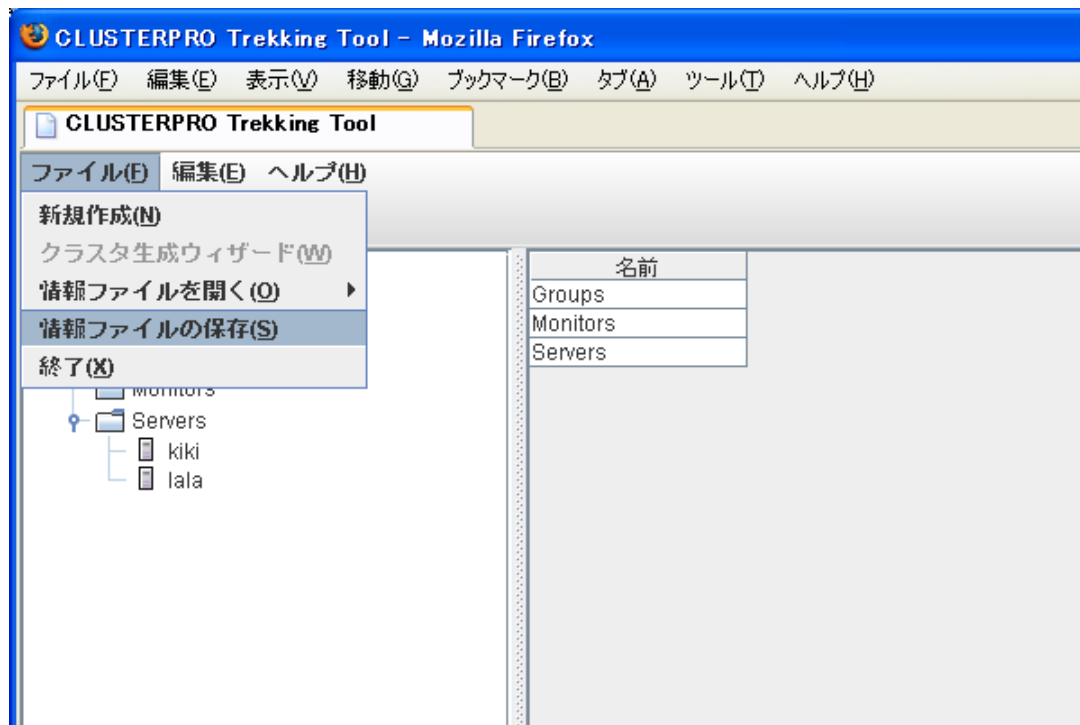
- 回復対象(R): oracle-failover 参照(W)
- 再活性化しきい値(E): 0 回
- フェイルオーバーしきい値(T): 1 回
- 最終動作(F): 何もしない (dropdown menu)
- Navigation buttons at the bottom: < 戻る(B), 完了, キャンセル

「完了」ボタンを選択して IP モニタリソースの設定を終了します。

この設定では、Gateway(10.1.0.1)への Ping に失敗した場合、ネットワーク障害と判断し、即時フェイルオーバーを実施します。

3.5.16. クラスタ構成情報の保存

クラスタ構成情報は、ローカルディレクトリに保存するか、フロッピーディスクへ保存します。



この作成したクラスタ構成情報をサーバへコピーし、CLUSTERPRO サーバのクラスタ設定に使用します。

Linux サーバ上のトレッキングツールを使用している場合には、直接 /root/cluster/conf()ディレクトリに格納してください。事前にディレクトリは作成してください。

Windows クライアント上のトレッキングツールを使用している場合には、直接フロッピーディスクに保存するか、保存したフォルダ (conf など) ごと Linux サーバ上 (/root/cluster/) へコピーしてください。

3.6.CLUSTERPRO サーバの設定

3.6.1.CLUSTERPRO 本体のインストール (両ノード)

再び、MCP インストール CD の `install_cluster.sh` スクリプトを使用して、CLUSTERPRO サーバを両ノードの Linux サーバにインストールしていきます。今回は、ミラーデータタイプの CLUSTERPRO LE をインストールします。

```
# mount /mnt/cdrom
# /mnt/cdrom/install_cluster.sh

1. Install CLUSTERPRO Server rpm
2. Install Trekking Tool rpm
3. Uninstall CLUSTERPRO Server rpm
4. Uninstall Trekking Tool rpm
5. Read the [Quick Installation Guide]

Select:(Q) > 1

Install the CLUSTERPRO Server RPM [Y/N]:(N) > y

Select the type of CLUSTERPRO [SE/LE]:(Q) > le

    Preparing for the installation ....

    Start installing CLUSTERPRO LE ....

Preparing... ##### [100%]
  1:clusterpro ##### [100%]
Preparing... ##### [100%]
  1:clusterpro-fsmon ##### [ 33%]
  2:clusterpro-ison ##### [ 67%]
  3:clusterpro-dbmon ##### [100%]

    The installation of CLUSTERPRO Server RPM was successful

[INFO]: Start-up clusterpro event, trnsv processes
----- start processes -----
Starting clusterpro event: [ OK ]
Starting clusterpro trnsv: [ OK ]

Back to the Start menu [Y/N]:(N) > n
```

3.6.2.クラスタ構成ファイルの登録準備 (両ノード)

先に作成したクラスタ構成ファイルとライセンスキーを CLUSTERPRO サーバに登録する前に、全てのノードで次の2つのサービスを起動する必要があります。MCP インストール CD のインストールスクリプトでインストールした場合には、自動的に起動されています。

```
# service clusterpro_evt start
# service clusterpro_trn start
```

- **CLUSTERPRO 本体の起動について**

CLUSTERPRO で起動するサービス(デーモン)は以下の通りになります。CLUSTERPRO インストール時に、この順番で自動起動する設定になっています。

1. clusterpro_evt:	ログ出力、メール通達	
2. clusterpro_trn:	ノード間通信	
3. clusterpro_md:	ミラーディスクの制御	CLUSTERPRO LE のみ
clusterpro:	本体	
4. clusterpro_webmgr:	Web マネージャ	
5. clusterpro_alertsync:	syslog 収集	

手動でサービスの起動する際には、service コマンドを使い上から順番に起動し、下から順番に停止します。

< 起動 >

```
# service clusterpro_evt start
# service clusterpro_trn start
# service clusterpro_md start
# service clusterpro start
# service clusterpro_webmgr start
# service clusterpro_alertsync start
```

< 停止 >

```
# service clusterpro_alertsync stop
# service clusterpro_webmgr stop
# service clusterpro stop
# service clusterpro_md stop
# service clusterpro_trn stop
# service clusterpro_evt stop
```

3.6.3. クラスタ構成ファイルの登録 (片ノードのみ)

次に、トレッキングツールで作成したクラスタ構成情報ファイルを CLUSTERPRO サーバに登録します。

以下のコマンドを用いて、ローカルディレクトリ(/root/cluster/conf)に保存されているクラスタ構成ファイルを、ディレクトリを指定し、CLUSTERPRO サーバに登録します。

```
# clpcfctrl --push -l -x /root/cluster/conf

file delivery to server 192.168.102.1 success.
file delivery to server 192.168.102.2 success.

success.(code:0)
```

Windows 上で作成したファイルの場合には、以下の構文のように「-w」で実行します。

FD に保存したファイルの場合には、「-x ディレクトリ」を省略して実行します。

【構文】

```
# clpcfctrl --push -l -x [clf.conf が含まれるディレクトリ]
+ clf.conf : トレッキングツールで作成するクラスタ構成ファイル
+ -w : Windows 版のトレッキングツールで作成した場合
+ -l : Linux 版のトレッキングツールで作成した場合
+ -x [ディレクトリ] を省略した場合、/dev/fd0(FD)が参照されます。
```

3.6.4.ライセンスキーの登録（両ノードと片ノード）

弊社から送られたライセンスキーを Linux サーバ上にコピーしてください。今回は、`/root/cluster/licence` 以下にコピーしたものとします。このライセンスキーを使用して、CLUSTERPRO サーバを試用用途で利用できます。

```
# ls -l /root/cluster/licence
20060201_DBMON30_01.key : データベース監視オプション R3.0
20060201_DBMON30_02.key : データベース監視オプション R3.0
20060201_FSMON30_01.key : ファイルサーバ監視オプション R3.0
20060201_FSMON30_02.key : ファイルサーバ監視オプション R3.0
20060201_FS030.key : Fast Sync Option R3.0
20060201_ISMON30_01.key : インターネットサーバ監視オプション R3.0
20060201_ISMON30_02.key : インターネットサーバ監視オプション R3.0
20060201_LE30.key : CLUSTERPRO LE v3.x
20060201_SE30.key : CLUSTERPRO SE v3.x
```

以下のコマンドを使い、CLUSTERPRO LE、FastSync Option、データベース監視オプションのライセンスキーを登録します。

- CLUSTERPRO LE ライセンス

```
# clplcncs -i /mnt/cdrom/license/20060201_LE30.key -p LE30 片ノードのみ
```

- Fast Sync Option ライセンス

```
# clplcncs -i /mnt/cdrom/license/20060201_FS030.key -p FS030 片ノードのみ
```

- データベース監視オプション ライセンス × 2

```
# clplcncs -i /mnt/cdrom/license/20060201_DBMON30_01.key -p DBMON30 片ノードずつ(kiki)
# clplcncs -i /mnt/cdrom/license/20060201_DBMON30_02.key -p DBMON30 片ノードずつ(lala)
```

【構文】

```
# clplcncs -i [Key ファイル] -p 製品略名
+ Key ファイル：弊社から送られたライセンスキー
+ -i：ライセンスキーのファイルを指定
+ -p：製品の略名を指定する
  ・ CLUSTERPRO SE： SE30
  ・ CLUSTERPRO LE： LE30
  ・ Fast Sync Option： FS030
  ・ データベース監視オプション： DBMON30
  ・ ファイルサーバ監視オプション： FSMON30
  ・ インターネット監視オプション： ISMON30
```

3.6.5.CLUSTERPRO 全サービスの起動（両ノード）

クラスタ構成ファイルの登録と、試用版ライセンスキーの登録が完了したら、両ノードで残りの全サービスを起動し、データベース・クラスタシステムを開始します。または、両ノードを再起動してもかまいません。

```
# service clusterpro_md start
# service clusterpro start
# service clusterpro_webmgr start
# service clusterpro_alertsync start
```

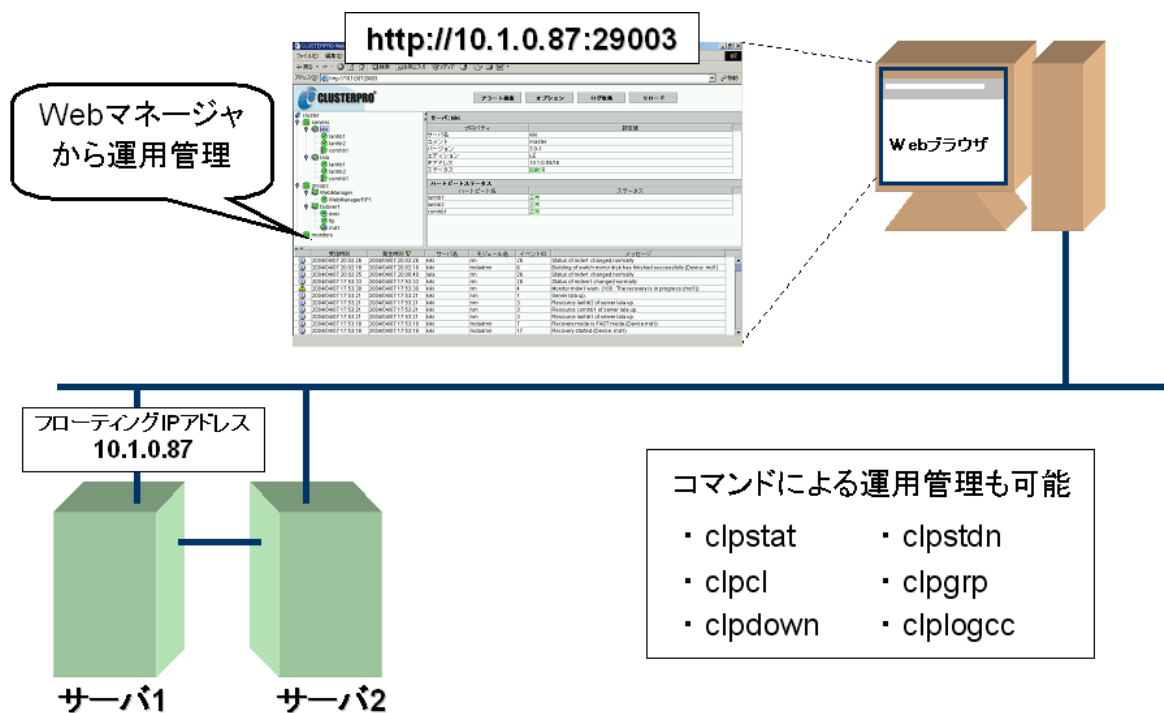
これらの CLUSTERPRO のサービスは、インストール時に自動起動の設定がされています。次回からは、再起動時に自動的にクラスタシステムが起動していきます。

```
# chkconfig --list | grep clusterpro
clusterpro_evt 0:off 1:off 2:off 3:on 4:off 5:on 6:off
clusterpro_md 0:off 1:off 2:off 3:on 4:off 5:on 6:off
clusterpro_alertsync 0:off 1:off 2:off 3:on 4:off 5:on 6:off
clusterpro_trn 0:off 1:off 2:off 3:on 4:off 5:on 6:off
clusterpro 0:off 1:off 2:off 3:on 4:off 5:on 6:off
clusterpro_webmgr 0:off 1:off 2:off 3:on 4:off 5:on 6:off
```

4. クラスタシステムの運用

4.1. CLUSTERPRO の運用操作

CLUSTERPRO におけるクラスタシステムの運用管理は、GUI ベースの Web マネージャを使用するか、専用のコマンドで行います。Web マネージャとコマンドラインから、クラスタの状況監視、サーバやグループの起動/停止、クラスタ動作ログの収集が可能です。

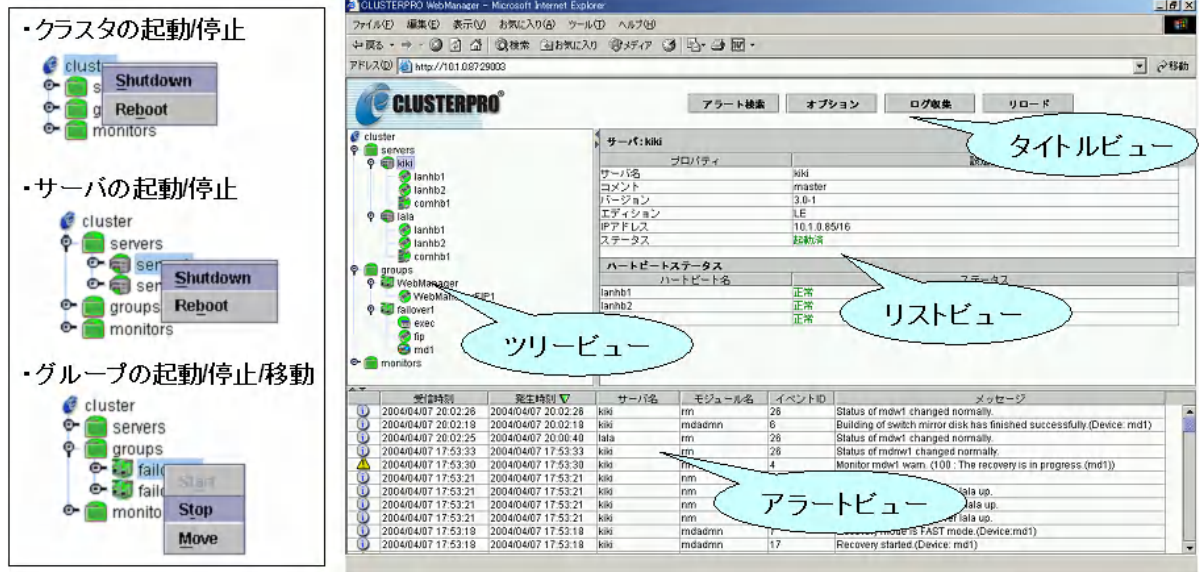


Web マネージャの接続先には Web マネージャのフローティング IP アドレスを指定します。フローティング IP アドレスを指定することで、一方のサーバがダウンした場合でももう一方のサーバに自動的に接続が切り替わります。

また、Web マネージャで指定できるポート番号 (初期値 29003) は、トレッキングツールで変更可能です。

4.2. Web マネージャの操作

Web マネージャは、クラスタの状況監視、サーバ/グループの起動・停止、クラスタ動作ログの収集を行う GUI ツールです。

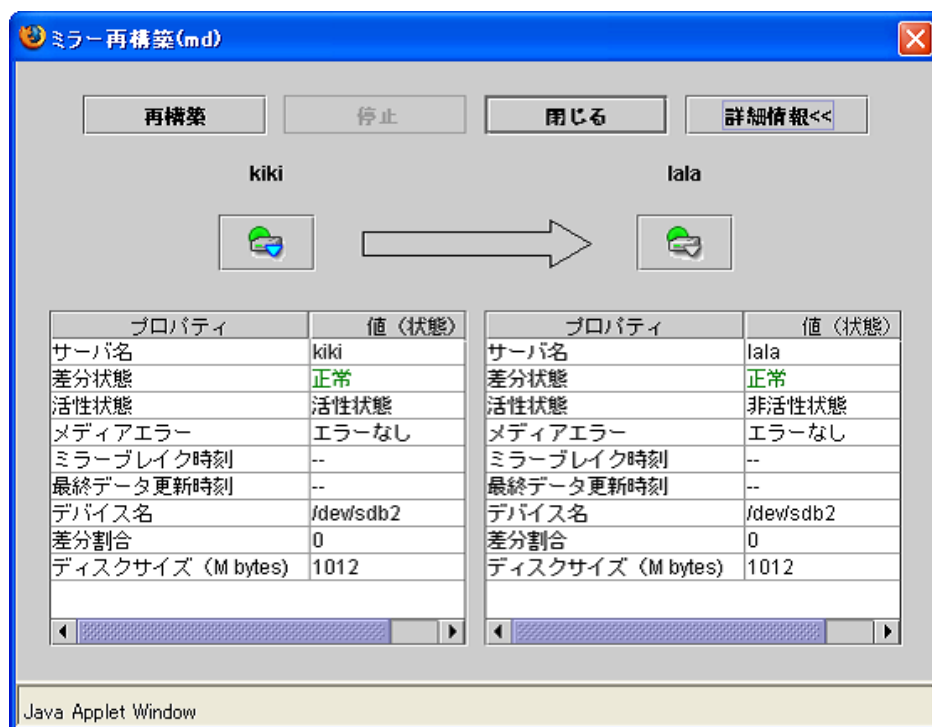


4.2.1.Web マネージャ管理画面

- **タイトルビュー**
 - **アラート検索**
アラートログの検索。検索対象とする期間や検索対象などを指定し、必要なアラートログを抽出できます。
 - **オプション**
スプリットバーの表示/非表示。Web マネージャの各ビューを仕切っているバーの表示/非表示の切り替え。
 - **ログ収集**
CLUSTERPRO の動作にかかわるログを全ノードから収集し、ローカルにダウンロードします。
 - **リロード**
画面のリロード。Web マネージャを最新の状態に更新します。
- **ツリービュー**
サーバ、グループ、モニタリソースなどの各リソースの状態を確認します。隔離ソースを表すアイコンの形や色により状態の確認を行います。
- **リストビュー**
ツリービューで選択した各リソースについての詳細情報が表示されます。
- **アラートビュー**
CLUSTERPRO の動作状況がメッセージとして表示されます。

4.2.2.ミラーディスクヘルパー

ミラーディスクヘルパーは、Web マネージャから起動するツールで、ミラーディスクの状態確認や復旧作業を行う GUI ツールです。ミラーディスクヘルパーは、ミラーディスクリソースを右クリックし、起動します。

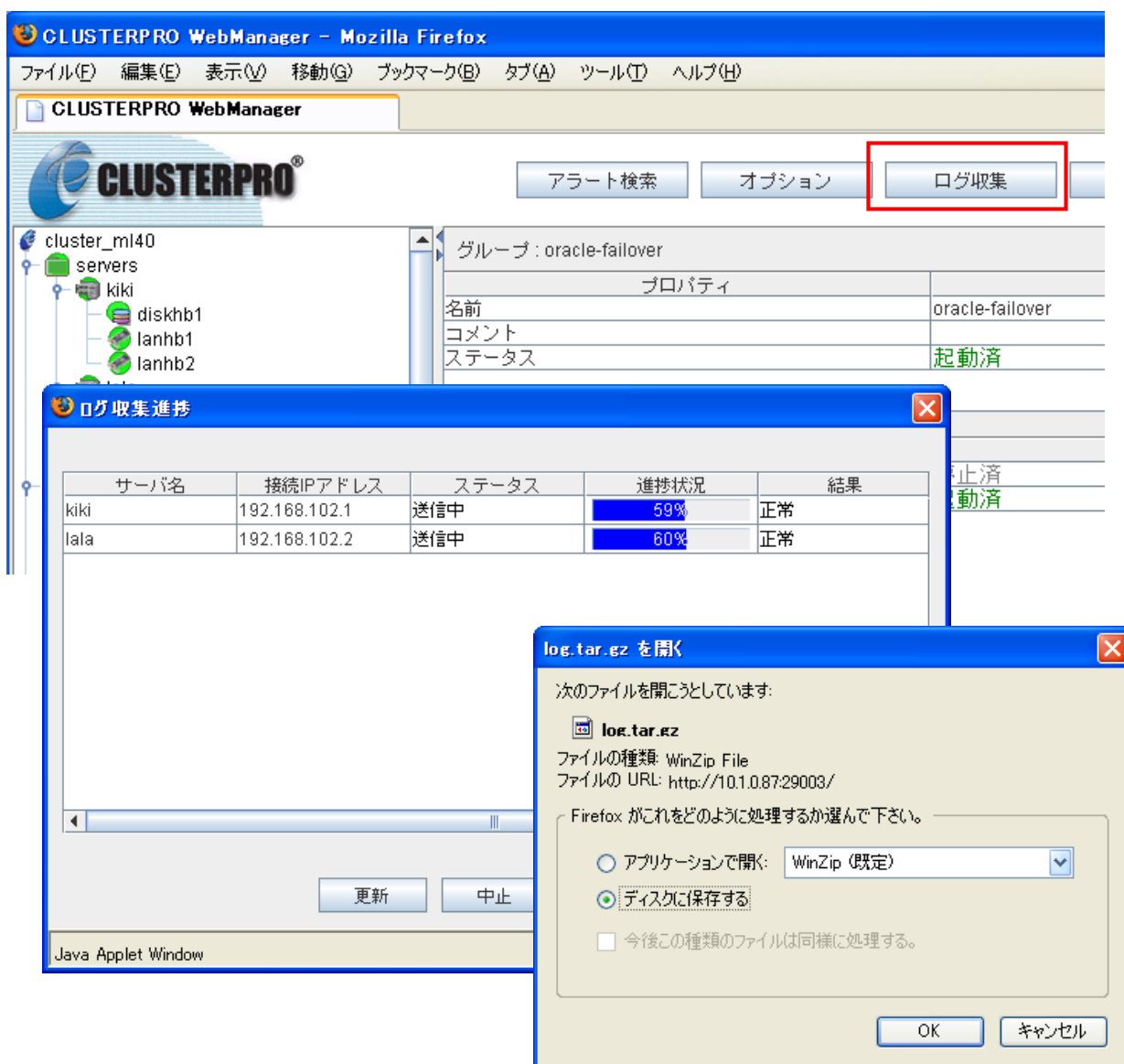


ミラーディスクヘルパーでは、4 つのボタンとミラーディスクの状態を表すアイコンを使用してミラーディスクの運用管理を行います。

- **再構築**
両サーバのミラーディスクに差分がある場合、ミラー復帰を行います。
両サーバのミラーディスクに差分がない場合、強制ミラー復帰を行うかを確認するダイアログがポップアップします。
- **停止**
選択した処理の停止を行います。
- **閉じる**
ミラーディスクヘルパーの管理画面を終了します。
- **詳細**
各ミラーディスクの詳細情報が表示されます。

4.2.3.ログの収集

Web マネージャのタイトルビュー「ログ収集」ボタンにより、CLUSTERPRO の全ノードの動作ログを収集されます。このログは、Web マネージャで管理するログだけでなく、OS で管理している syslog などを含む CLUSTERPRO に関連する全てのログを収集します。



今回の新バージョンの MCP インストール CD からインストールした場合、CLUSTERPRO のログ収集のスクリプトに、以下のログも収集するように調整されています。

- Oracle Database の Alert ログや、各種設定ファイル
- mcinfo の結果 (MIRACLE LINUX 独自のログ収集コマンド)

これまでの CLUSTERPRO によるログ収集に加え、OS/Oracle の情報も一度に収集しています。ミラクル・リナックスの MIRACLE CLUSTERPRO サポートでは、これら一つにまとまった OS/クラスタ/Oracle ログ情報をもとにより円滑、迅速にお客様のクラスタシステムをサポートします。

4.3.コマンドによる運用管理

CLUSTERPRO では、Web マネージャ以外にもクラスタシステムを運用管理するためのあらゆるコマンドが用意されています。ここでは特に利用頻度の高い CLUSTERPRO コマンドをいくつか紹介します。

4.3.1.状況確認

clpstat コマンドを使用することで、Web マネージャで確認できる情報をコマンドラインから確認することが可能です。Web マネージャは JavaApplet で動作するため、clpstat コマンドで確認する方がリアルタイムに状況を確認できます。

clpstat [-s] clpstat -g clpstat -m clpstat -n clpstat -i	全体ステータス表示 グループの起動状態を表示 モニタリソースの状態を表示 ネットワークの状態を表示 クラスタの情報を表示
--	--

```
# clpstat
===== CLUSTER STATUS =====
Cluster : ml-cluster
<server>
*kiki .....: Normal
  lanhb1      : Normal      LAN Heartbeat
  lanhb2      : Normal      LAN Heartbeat
  comhb1      : Normal      COM Heartbeat
  lala .....: Normal
  lanhb1      : Normal      LAN Heartbeat
  lanhb2      : Normal      LAN Heartbeat
  comhb1      : Normal      COM Heartbeat
<group>
WebManager .....: Online      WebManager group
  current      : kiki
WebManagerFIP1 : Online      floating ip resource
failover .....: Online
  current      : kiki
  exec1        : Online
  exec2        : Online
  fip          : Online
  md           : Online
<monitor>
diskw          : Normal
ipw            : Normal
mdnw1          : Normal      mirror disk connect monitor
mdw1           : Normal      mirror disk monitor
pidw           : Normal
userw          : Normal      user mode monitor
=====
```

サーバ

機能	状態	説明
ハートビートリソース状態 表示	Online	起動中
	Offline	停止中
	Caution	ハートビートリソースが異常
	Unknown	状態不明

ハートビート

機能	状態	説明
状態表示	Normal	正常
	Caution	一部異常
	Error	全部異常
	Unused	未使用
	Unknown	状態不明

グループ

機能	状態	説明
状態表示	Online	起動済
	Offline	停止済
	Online Pending	起動処理中
	Offline Pending	停止処理中
	Error	異常
	Unknown	状態不明

グループリソース

機能	状態	説明
状態表示	Online	起動済
	Offline	停止済
	Online Pending	起動処理中
	Offline Pending	停止処理中
	Online Failure	起動失敗
	Offline Failure	停止失敗
	Unknown	状態不明

モニタリソース

機能	状態	説明
状態表示	Normal	正常
	Caution	一部異常
	Error	全部異常
	Unused	未使用
	Unknown	状態不明

4.3.1.1.CLUSTERPRO 本体デーモンの起動/停止

CLUSTERPRO 本体デーモンのみを停止することで、ソフトウェアレベルでクラスタシステムを停止することが可能です。本体デーモンが起動/停止する際には、CLUSTERPRO 管理下のアプリケーションも同時に起動/停止されます。

clpcl -s clpcl -t clpcl --suspend clpcl --resume	clusterpro起動 clusterpro停止 clusterproサスペンド clusterproレジューム
-a option	全サーバに対して実行

4.3.2.グループの起動/停止

clpgrp -s [group_name] clpgrp -t [group_name] clpgrp -m group_name	グループ起動 グループ停止 グループ移動
-f option -a [dest_name] option	強制的に操作を行う 移動先を指定



4.3.3.サーバのシャットダウン/リブート

CLUSTERPRO がインストールされた環境では、OS のシャットダウン/リブート作業は、原則として CLUSTERPRO から操作することになります。

また、CLUSTERPRO 環境化で 1 台だけのサーバを落とすという作業は、クラスタ内でサーバダウンが発生したことに同義であり、状況に応じてフェイルオーバーが発声する。

また、CLUSTERPRO LE の場合、ダウンしたサーバ側のデータが更新されなくなるため、サーバ起動時に「ミラー復帰」という作業が必要になります。

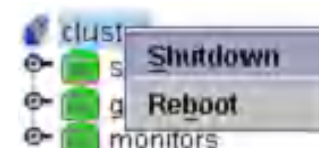
clpdown	サーバシャットダウン
-r option	再起動を行う



4.3.4.クラスタシステム全体のシャットダウン/リブート

計画停電などがあり、クラスタシステム全体をシャットダウン/リブートしたい場合、クラスタシャットダウン/リブート作業を行います。これはサーバ単体のシャットダウンと異なり、クラスタシステム全体で行われるシャットダウン/リブートとなります。

clpstdn	クラスタシャットダウン
-r option	再起動を行う



5.最後に

今回は、新製品の MIRACLE CLUSTERPRO V3.1 (MIRACLE LINUX4.0 バンドル) を使い、Oracle Database のクラスタシステムを構築しました。CLUSTERPRO LE は、IA サーバ 2 台あればクラスタシステムを手軽に構築できます。皆さんも同等スペックのサーバ 2 台用意して、または VMware などを利用して HA クラスタシステム構築にチャレンジしてください。思った以上に簡単に構築できると思います。

お問合せ先: info@miraclelinux.com