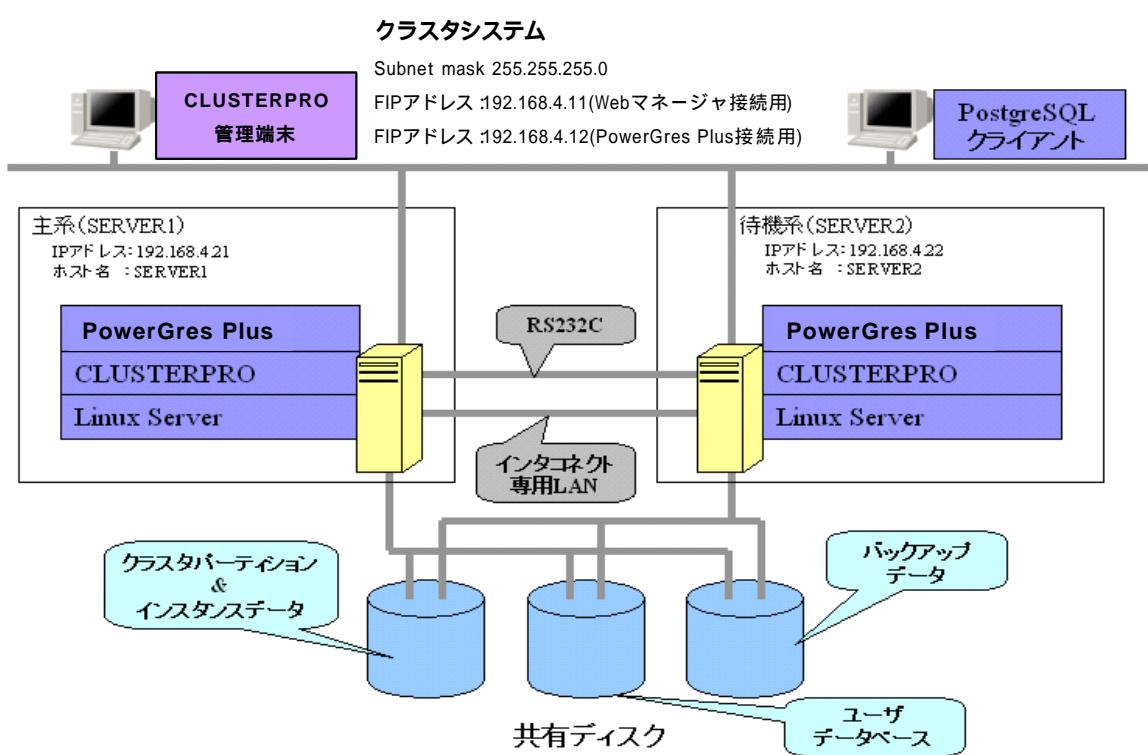


はじめに

この文書は、CLUSTERPRO SE for Linux Ver3.0（以下 CLUSTERPROと呼称）を使用し、
PowerGres Plusをクラスタリングする方法について簡単に説明しています。
CLUSTERPROを使用したフェイルオーバクラスタを構築する際に、参考情報としてご覧ください。

第1章 フェイルオーバクラスタシステムの概要

ここでは、次の図のフェイルオーバクラスタシステム（片方向スタンバイ）を構築します。このフェイルオーバクラスタシステムは、2つのサーバで構成され、各サーバのローカルディスク（IDE）に、「Linux OS」、「CLUSTERPRO」をインストールし、その上に PowerGres Plus を動作させます。PowerGres Plus のデータベースを、共有ディスク上に構築し、主系（SERVER1）にて動作させます。待機系（SERVER2）は、主系がダウンした場合に、PowerGres Plus を起動し業務を引き継ぎます。また、データベース部分は、共有ディスクを介して引き継ぎます。



また、PostgreSQL クライアントから、フェイルオーバクラスタシステムで稼働させている PowerGres Plus にアクセスする場合には、フローティング IP アドレス（192.168.4.12）を経由してアクセスします。そうすることにより、PostgreSQL クライアントからは、フェイルオーバクラスタシステムでの業務の切り替えを、意識することなくアクセスすることが可能となります。

フェイルオーバノード（SERVER1, SERVER2）に必要となるハードウェアは次のとおりです。

【NIC (ネットワークインターフェースカード) 2枚】

(インタコネクト LAN)

フェイルオーバ型クラスタの通信バスで、ハートビートを行うとともにサーバ間の情報交換に使用します。

今回は、インタコネクト専用 LAN を、各サーバごとに以下 IP アドレスで設定します。

SERVER1 : 192.168.0.1/255.255.255.0

SERVER2 : 192.168.0.2/255.255.255.0

インタコネクト専用 LAN は、通常 2 ノードクラスタシステムであれば、クロスケーブルで接続します。

(パブリック LAN)

クライアントと接続する通信バスです。

インタコネクト LAN と兼用設定を行うことも可能です。

【RS-232C】

クロスケーブルで接続し、ハートビートを行います。インタコネクト LAN が全断線した場合も、RS-232C 経由でのサーバ生存確認が可能です。(但し、グループ移動などの操作はできません)

【共有ディスク】

アプリケーションサーバで使用するデータを格納し、リソースの引き継ぎに使用します。また、クラスタパーティションというクラスタサーバ間通信用のパーティションを確保することで、クラスタパーティション経由でのサーバ生存確認が可能になります。(但し、RS-232C 経由と同様、インタコネクト LAN 全断線時にはグループ移動などの操作はできません)

【HBA (Host Bus Adapter)】

SCSI インタフェースカードもしくはファイバチャネルカードを使用し、共有ディスクと PC サーバを接続します。

また、フェイルオーバクラスタを構成するノード以外に、以下のノードを用意します。

【CLUSTERPRO 管理端末】

CLUSTERPRO Web マネージャを利用することで、管理端末から、クラスタの状態監視、サーバの停止、グループの起動/停止及び、クラスタ動作ログの収集を行なうことが可能です。管理端末から、CLUSTERPRO Web マネージャへの接続は、一般的な Web ブラウザを利用します。

【PostgreSQL クライアント】

PowerGres Plus サーバに接続する確認クライアントに使用します。

今回は、Linux OS 上に PostgreSQL クライアントが導入されているものとします。

第2章 インストール

2.1 前準備

SCSIインターフェースカードを使用して共有ディスクとサーバを接続する場合、まず、フェイルオーバクラスタを構築する両サーバに SCSI ホストバスアダプタの BIOS 設定を施します。今回のクラスタシステムの共有ディスクへの接続には、SCSI ホストバスアダプタ (Adaptec 29160) を経由し、サーバと接続します。両サーバの SCSI ホストバスアダプタの BIOS 設定を以下のように設定します。なお、今回使用する共有ディスクは、マルチホスト対応を前提とします。

SCSI Bus Interface Definitions	
Host Adapter SCSI ID	7
SCSI Parity Checking	Enabled
Host Adapter SCSI Termination	
LVD/SE Connector	Automatic
SE Connector	Automatic
Additional Options	
Boot Device Options	
Boot SCSI ID	0
Boot LUN Number	0
SCSI Device Configuration	
Sync Transfer Rate (MB/Sec)	160
Initiate Wide Negotiation	Yes
Enable Disconnection	Yes
Send Start Unit Command	No
Enable Write Back Cache	N/C
BIOS Multiple LUN Support	No
Include in BIOS Scan	Yes
Advanced Configuration Options	
Reset SCSI Bus at IC Initialization	Disabled
Display <Ctrl><A> Message During BIOS Initialization ..	Enabled
Extended BIOS Translation for DOS Drives > 1 GByte ..	Enabled
Verbose/Silent Mode	Verbose
Host Adapter BIOS	Disabled:scan bus
Domain Validation	Enable
Support Removable Disks Under BIOS as Fixed Disks	Boot Only
BIOS Support for Bootable CD-ROM	Disabled
BIOS Support for Int13 Extensions	Enable

2.2 CLUSTERPRO トレッキングツールのセットアップ

管理端末にトレッキングツールをインストールします。

管理端末にトレッキングツールをインストールする前にトレッキングツールの動作環境を確認してください。

トレッキングツールの動作環境については、以下のマニュアルを参照してください。
「CLUSTERPRO for Linux Ver3.0 動作環境編」

トレッキングツールのインストール方法については、以下のマニュアルを参照してください。
「CLUSTERPRO for Linux Ver3.0 クラスタ生成編」

2.3 CLUSTERPRO サーバのセットアップ

CLUSTERPRO サーバRPMをインストールすることでセットアップされます。

CLUSTERPRO サーバRPMはrootユーザでインストールしてください。

以下の手順でCLUSTERPRO サーバRPMをすべてのフェイルオーバノード (SERVER1, SERVER2) にインストールします。

1. インストールCD-ROMの媒体をmountします。

2. rpmコマンドを実行してパッケージファイルをインストールします。

CD-ROM内の/Linux/3.0/jpn/server/SEに移動して、

`rpm -i clusterpro-[バージョン番号]-[リリース番号].i386.rpm`

を実行してください。

インストールが開始されます。

CLUSTERPROは以下の場所にインストールされます。このディレクトリを変更するとアンインストールできなくなりますので注意してください。

インストールディレクトリ : /opt/nec/clusterpro

3. インストール後、インストールCD-ROM媒体をumountします。

4. インストールCD-ROM媒体を取り除いた後、サーバをリブートします。

CLUSTERPRO サーバRPMをインストールした後、「CLUSTERPRO for Linux Ver3.0 クラスタ生成編」の「インストール後の設定」をよくお読みになり、必要な設定をおこなってください。

なお、共有ディスクの設定、クラスタ生成、ライセンス登録については、本書でも説明します。

2.4 PowerGres Plus のインストール

次に、PowerGres Plus を、以下の手順で両フェイルオーバノード (SERVER1, SERVER2) にインストールします。

PowerGres PlusのCD-ROMのREADMEに従って製品をインストールします。

インストールは、root 権限を待つユーザで行ってください。なお、以降、この文書のコマンド例では、コマンドプロンプトを、root 実行の場合には #、ユーザ実行の場合には \$ で記述します。

以下のとおりコマンドを入力してインストールします。

```
# mount /mnt/cdrom
# cd /mnt/cdrom
# sh ./install.sh
```

インストールが正常に終了すると、/usr/local/pgsqlplusに製品がインストールされます。

第3章 フェイルオーバクラスタの構築

各ノードへ CLUSTERPRO サーバ および PowerGres Plus をインストールし、管理端末での CLUSTERPRO トレッキングツールの準備が完了したら、PowerGres Plus のフェイルオーバクラスタの構築を行ないます。

PowerGres Plus のフェイルオーバクラスタの構築は、以下の手順で行います。

手順	項目	説明箇所
1	共用ディスクの設定 パーティションの確保 ファイルシステムの作成 マウントポイントの作成	3.1
2	共有ディスク上への PowerGres Plus データベース構築 セットアップ前の準備 セットアップ (*1) インスタンスの起動 (*2) データベースの定義 (*2) スキーマ、表、インデックスの定義 (*3) PowerGres Plus の停止 (*2) と共有ディスクアンマウント	3.2
3	CLUSTERPRO トレッキングツールによるフェイルオーバクラスタ設定 クラスタ情報の生成 PowerGres Plus起動・停止用 EXECリソースの追加	3.3
4	クラスタの生成 PowerGres Plusグループの起動	3.4

*1 : PostgreSQL Plus Setupでの操作

*2 : PostgreSQL Plus Administratorでの操作

*3 : PostgreSQL Plus Administratorでの操作 (psqlなどでも操作可能)

3.1 共有ディスクの設定

最初に共有ディスクの設定を行います。

PowerGres Plusでは、インスタンスデータ用、ユーザデータベース用、バックアップ用の3つのディスクを使用することを推奨しています。

以降の説明は、3つのディスクを使用した場合について説明します。

ディスク	パーティション	内容
sda	/dev/sda1	CLUSTER用
	/dev/sda2	インスタンスデータ用（システムカタログや動作環境ファイルなどを格納）
sdb	/dev/sdb1	ユーザデータベース用
sdc	/dev/sdc1	バックアップ用（復旧に必要なデータのバックアップを格納）

備考：PowerGres Plusでは、PowerGres Plusを運用するために必要な環境の全体をインスタンスと呼びます。インスタンスごとにセットアップして、起動や停止を行うことができます。

3.1.1 パーティションの確保

片方のサーバから `fdisk` コマンドを使い、共有ディスクに対して以下のようなパーティション設定を行います。

```
# fdisk -l /dev/sda

ディスク /dev/sda: ヘッド 255, セクタ 63, シリンダ 4290
ユニット = シリンダ数 of 16065 * 512 バイト

デバイス ブート 始点 終点 ブロック ID システム
/dev/sda1      1      2    16033+  83  Linux
/dev/sda2      3    512  4096575  83  Linux
```

`/dev/sda1` は、CLUSTER パーティションとして、10 MB 程度以上を確保します。

各サーバは、この CLUSTER パーティションに対して Raw I/O でアクセスし、サーバ間の情報交換を行います。

インスタンスデータを配置する`/dev/sda2` は、PowerGres Plusの切り替えパーティションとして使用します。

また、ユーザデータベース用のパーティション`/dev/sdb1`、バックアップ用のパーティション`/dev/sdc1`を、それぞれ別のディスクに設定します。

3.1.2 ファイルシステムの作成

片方のサーバから、インスタンスデータ用のパーティション /dev/sda2 に対して、EXT3 ファイルシステムを作成します。

```
# mkfs.ext3 /dev/sda2
```

次のコマンドを実行します。

```
# tune2fs -l /dev/sda2
```

mkfs.ext3 コマンドを実行して、EXT3 ファイルシステムを構築しただけでは、そのパーティションに対して、マウントカウント到達時点 (Maximum mount count:) もしくは起動後の経過時間到達 (Check interval:) により fsck の自動実行制御が設定されています。

以下のコマンドを実行し、fsck の自動実行制御を抑制してください。

```
# tune2fs -c -1 -i 0 /dev/sda2
```

同様に、ユーザデータベース用のパーティション /dev/sdb1、バックアップ用のパーティション /dev/sdc1 についても、ファイルシステムの作成を行います。

```
# mkfs.ext3 /dev/sdb1
# tune2fs -l /dev/sdb1
# tune2fs -c -1 -i 0 /dev/sdb1
# mkfs.ext3 /dev/sdc1
# tune2fs -l /dev/sdc1
# tune2fs -c -1 -i 0 /dev/sdc1
```

上記設定を行うことで、OS からの自動的な fsck 実行制御を抑制することができますが、定期的なメンテナンスのタイミングで fsck を実行することをお勧めします。

3.1.3 マウントポイントの作成

両方のサーバで、切り替えパーティションのマウントポイントを作成します。

パーティション	マウントポイント名
/dev/sda2 (インスタンスデータ用)	/PlusData
/dev/sdb1 (ユーザデータベース用)	/PlusDbspace
/dev/sdc1 (バックアップ用)	/PlusBackup

```
# mkdir /PlusData
```

同様に、ユーザデータベース用のパーティション /dev/sdb1、バックアップ用のパーティション /dev/sdc1 についても、マウントポイントの作成を行います。

```
# mkdir /PlusDbspace
# mkdir /PlusBackup
```

ここまで作業完了後に、両方のサーバを一旦リブートします。

3.2 共有ディスク上への PowerGres Plus データベース構築

次に共有ディスク上に PowerGres Plus データベースを構築し、各ノードの PowerGres Plus から共有ディスク上データベースにアクセスするための設定を行います。

3.2.1 セットアップ前の準備

セットアップを開始する前に、以下の作業を行います。

【ファイルシステムのマウント】

セットアップを行うサーバで、共有ディスクをマウントします。
この作業は、root 権限を待つユーザで実行します。

```
# mount /dev/sda2 /PlusData
# mount /dev/sdb1 /PlusDbspace
# mount /dev/sdc1 /PlusBackup
```

【インスタンス管理者用ユーザの登録】

セットアップは、root 権限を待たないユーザで実行する必要があります。
あらかじめ、インスタンス管理者用のユーザをシステム（OS）に登録し、そのユーザでログインしてから作業してください。

ここでは、ユーザ名に“postgres”、グループ名に“postgres”を追加したと仮定して以降の手順を説明します。
ユーザの追加は、root 権限を待つユーザで、useraddコマンド、または、OSに添付のツールにより追加することができます。

システムに登録するインスタンス管理者用ユーザのユーザIDと、インスタンス管理者用グループのグループIDはすべてのサーバで同じIDを使用してください。

【ディレクトリの作成】

セットアップでは、インスタンスを格納するディレクトリおよびバックアップディレクトリにファイルを作成します。また、セットアップ後に行うデータベースの定義では、ユーザデータベース用ディレクトリにデータベースを作成します。

あらかじめ、以下のディレクトリを作成してください。

項目	ディレクトリ
インスタンスを格納するディレクトリ	/PlusData/data1
ユーザデータベース用ディレクトリ	/PlusDbspace/data1
バックアップディレクトリ	/PlusBackup/data1

ディレクトリの作成は、root 権限を待つユーザで実行します。

【ディレクトリの権限の設定】

作成した各ディレクトリは、インスタンス管理者用ユーザ “postgres” だけが書き込み可能な権限にしておきます。ディレクトリの権限の設定は、root 権限を待つユーザで実行します。

ファイルマネージャから、以下のように設定します。



コマンドで行う場合は、以下のように入力します。

```
# chown postgres /PlusData/data1 /PlusDbspace/data1 /PlusBackup/data1
# chgrp postgres /PlusData/data1 /PlusDbspace/data1 /PlusBackup/data1
# chmod 700 /PlusData/data1 /PlusDbspace/data1 /PlusBackup/data1
```

【環境変数の設定】

PowerGres Plus および man コマンドを使用するためには、以下の環境変数の設定が必要です。インスタンス管理者用ユーザ (postgres) でログインしなおし、以下の環境変数を設定します。

- 環境変数PATHに、/usr/local/pgsqlplus/binを追加
- 環境変数MANPATHに、/usr/local/pgsqlplus/manを追加

環境変数の設定例を以下に示します。

sh、bashの場合

```
$ PATH=/usr/local/pgsqlplus/bin:$PATH ; export PATH
$ MANPATH=/usr/local/pgsqlplus/man:$MANPATH ; export MANPATH
```

csh、tcshの場合

```
% setenv PATH /usr/local/pgsqlplus/bin:$PATH
% setenv MANPATH /usr/local/pgsqlplus/man:$MANPATH
```

注意

環境変数LANGに、ja_JP.eucJP(日本語)、およびC(英語)以外を指定しないでください。PostgreSQL Plus Setupのファイル選択画面で、正しく表示されない場合があります。

【カーネルパラメタの変更】

OS のカーネルパラメタを変更する必要があります。

変更が必要なパラメタは、以下のものです。

パラメタ名	適切な値
kernel.shmmmax	PowerGres Plus のセットアップで指定するデータベースバッファの大きさ以上の値。バイト単位で指定します。 データベースバッファの大きさは、デフォルトでは実メモリ量の約半分の値となります。(注1)
kernel.shmmni	(11 * インスタンス数) 以上の値。(注2)
kernel.sem	このパラメタには、空白で区切った4つの値を指定します。 1番目 30 以上の値。ただし、CPU 数が 16 を超えるときには、CPU数 16 個に対し、1ずつ値を増やしてください。(注1) 2番目 (1200 * インスタンス数)以上の値。(注2) 3番目 32 以上の値。(注1) 4番目 (350 * インスタンス数)以上の値。(注2)

注1) 他の製品でチューニングが指示されている場合は、最大の値を設定してください。

OS のデフォルト値がここで指示した値よりも大きい場合には、OS のデフォルト値を指定してください。

注2) 現在設定されている値に、ここで指定している値を加算してください。

ここでは、Red Hat Linux での設定方法の例を示します。
 /etc/sysctl.conf にパラメタを記述します（値は一例です）

```
kernel.shmmmax = 536870912
kernel.shmmni = 4096
kernel.sem = 250 32000 100 478
```

OS の root 権限を持つユーザで、以下のコマンド実行します。

```
# sysctl -p
```

3.2.2 セットアップ

セットアップは、PostgreSQL Plus Setupで行います。

PostgreSQL Plus Setupは、GUIベースの簡単な入力によってデータベースの初期化を行うツールです。起動する前には、X-Windowsシステムが使用可能な環境であることを必ず確認してください。すでに、Windowマネージャなどを利用している環境では、そのまま使用することができます。

PostgreSQL Plus Setupは、セットアップ前の準備で登録したインスタンス管理者用ユーザ（postgres）で行います。

【PostgreSQL Plus Setupの実行】

PostgreSQL Plus Setupは、pgx_setupコマンドを実行して起動します。

pgx_setupの起動は、ファイルマネージャから、以下のコマンドを起動します。

```
/usr/local/pgsqlplus/bin/pgx_setup
```

ファイルマネージャからコマンドを起動したときの画面を以下に示します。



pgx_setupコマンドを実行すると、PostgreSQL Plus SetupのTOP画面が表示されます。
【次へ】ボタンを押して、次の画面に進みます。

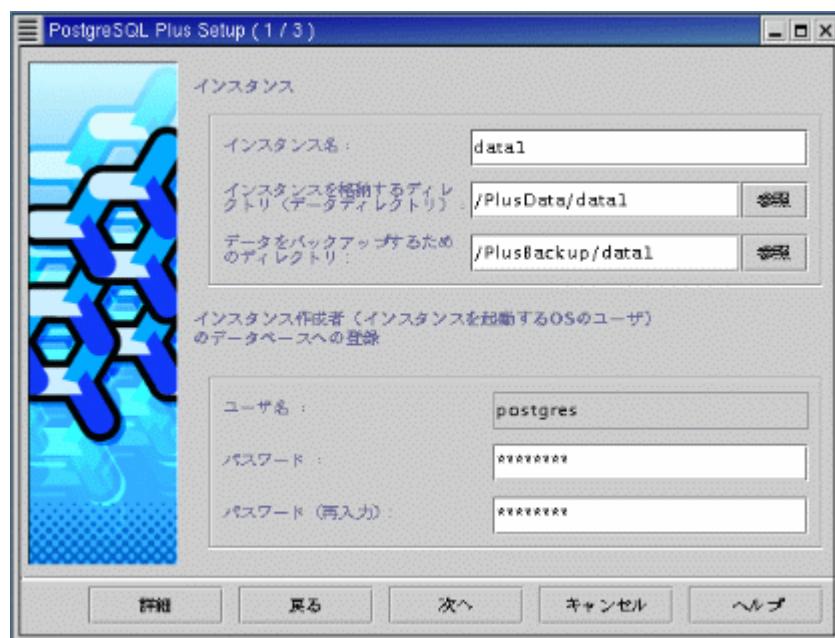
セットアップの実行手順について、以下に説明します。

【インスタンス情報の設定】

ここでは、インスタンス名として data1 を指定したとして説明します。
データディレクトリには、インスタンスデータ用のディスクのパス（/PlusData/data1）を、バッカアップするためのディレクトリには、バックアップ用のディスクのパス（/PlusBackup/data1）を指定します。

PostgreSQL Plus SetupのTOP画面で、【次へ】ボタンを押すと、【PostgreSQL Plus Setup (1/3)】画面が表示されます。

この画面では、インスタンスを格納するディレクトリ、データをバックアップするためのディレクトリ（バックアップディレクトリ）およびデータベースシステムに登録する管理者の情報を設定します。

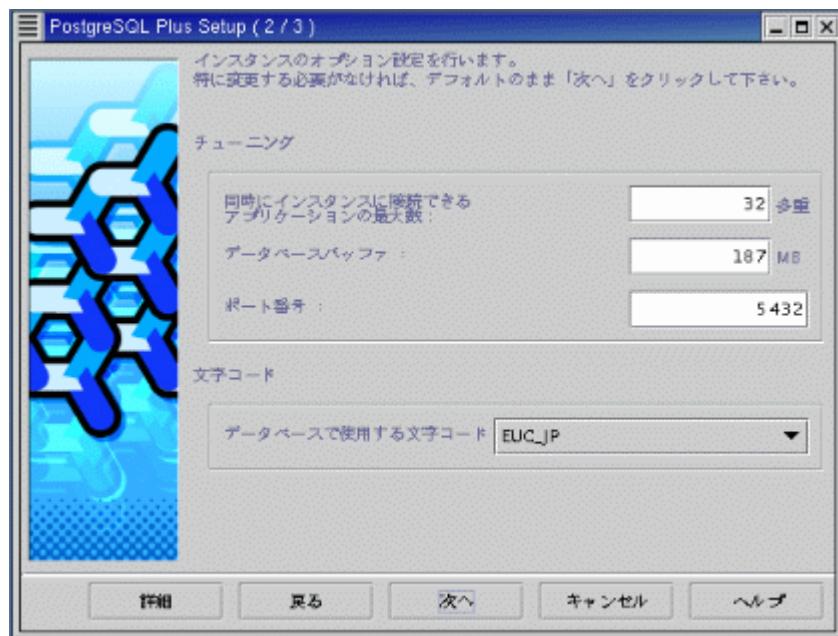


この画面の項目は、すべて入力必須です。
すべての情報を入力して【次へ】ボタンを押します。

【インスタンスのオプション情報の設定】

【PostgreSQL Plus Setup(1/3)】画面で、【次へ】ボタンを押すと、【PostgreSQL Plus Setup(2/3)】画面が表示されます。

この画面では、インスタンスのオプション情報を設定します。



この画面の項目は、通常変更する必要はありません。
使用する環境に合わせて変更する場合は、すべての項目を入力して【次へ】ボタンを押します。

【セットアップの実行】

【PostgreSQL Plus Setup(2/3)】画面で、【次へ】ボタンを押すと、【PostgreSQL Plus Setup(3/3)】画面が表示されます。

この画面では、入力した情報で、セットアップ処理を行います。



【実行】ボタンを押して、セットアップ処理を実行します。

実行の状態は、画面の下のリストに表示されます。異常が発生して処理が中断した場合は、リストに表示されたメッセージを参照し、原因を取り除いて、再度セットアップを行ってください。すでに、インスタンスを格納するディレクトリ、バックアップディレクトリが作成されている場合は、削除してから再実行してください。

セットアップが正常に終了したあと、PostgreSQL Plus Administratorから環境設定の情報を変更できます。

【セットアップ設定ファイルの待機システムへの複写】

セットアップによって作成された環境ファイルを、もう一方のサーバに複写します。
ftpコマンド等により、以下のファイルをもう一方のサーバに複写してください。

```
/usr/local/pgsqlplus/package/pgxadmin/webapps/admin/data1.html
/usr/local/pgsqlplus/package/pgxadmin/webapps/admin/data1_en.html
```

上記ファイル名の“data1”は、PostgreSQL Plus Setupで指定したインスタンス名です。

もう一方のサーバでは、PostgreSQL Plus Setupを実行しないでください。

3.2.3 インスタンスの起動

PostgreSQL Plus Administratorを利用して、以下の手順でインスタンスを起動します。
PostgreSQL Plus Administratorでデータベースを操作する場合は、必ずインスタンス作成者でログインする必要があります。

PostgreSQL Plus Administratorを利用するには、以下のブラウザが必要です。

- Linuxの場合
Mozilla 0.9.9以降
- Windows(R)の場合
Internet Explorer 6.0以降、またはNetscape Navigator 7.0以降

手順

1. ブラウザで以下のURLを指定してPostgreSQL Plus Administratorを起動します。

```
http://localhost:13000/admin/data1.html      : 日本語ページ
http://localhost:13000/admin/data1_en.html    : 英語ページ
```

上記URLの“localhost”は、PostgreSQL Plus Setupを実施したホスト名です。
また、“data1”は、PostgreSQL Plus Setupで指定したインスタンス名です。

Windowsシステムで実行する場合など、localhostがホスト名として使用できない場合は、環境に合わせて、localhostを使用可能なホスト名に変更してください。
最後のHTMLファイル名は、使用するインスタンス名にあわせて変更してください。

2. インスタンス作成者ログイン

PostgreSQL Plus Administratorを起動すると、インスタンス作成者ログイン画面が表示されます。

インスタンス作成者名“postgres”とパスワードを入力して【ログイン】ボタンを押します。

3. インスタンスの起動

メニューの【インスタンス起動停止】を選択し、【インスタンス起動停止】画面で、【起動】ボタンを押します。



注意

同一マシンで2つ以上のPostgreSQL Plus Administratorを起動しないでください。

3.2.4 データベースの定義

PostgreSQL Plus Administratorを利用して、以下の手順でデータベースを定義します。

詳細は、「PostgreSQL Plus 管理者ガイド」を参照してください。

1. インスタンス作成者ログイン

インスタンス作成者名 “postgres” とパスワードを入力して、【ログイン】ボタンを押します。

2. データベースの作成

【データベース操作】画面の、【データベースの追加・削除】を押します。

インスタンスの作成直後は、データベースとして、template1とtemplate0が定義されています。

【新規作成】ボタンを押し、【データベース新規追加】画面を表示します。

【データベース名】に必要な項目を入力して、【実行】ボタンを押します。

例えば、データベース db1 に、100M バイトのデータベーススペースを /PlusDbspace/data1/db1sp というファイルに作成する場合は、以下のように指定します。

【新規にデータベーススペース名を作成し、割り当てる】をチェック

? データベース名 : db1

? データベーススペース名 : db1sp

? パス名 : /PlusDbspace/data1/db1sp

? サイズ : 100

データベースの追加が終わると、正常終了のメッセージが表示されます。

【終了】ボタンを押して、データベース操作画面に戻ります。

3.2.5 スキーマ、表、インデックスの定義

スキーマ、表、およびインデックスは、psqlなどの各種ツールや、アプリケーションから定義する方法と、PostgreSQL Plus Administratorで定義する方法があります。

PostgreSQL Plus Administratorを利用して、スキーマ定義、および表定義を行う例を、以下に説明します。詳細は、「PostgreSQL Plus 管理者ガイド」を参照してください。

1. インスタンス作成者ログイン

インスタンス作成者名 “ postgres ”とパスワードを入力して【ログイン】ボタンを押します。

2. スキーマ、表の作成

【データベース操作】画面の【SQL文実行】を押します。

データベース名に操作するデータベース名、SQL文にCREATEコマンドを入力して【実行】ボタンを押します。

ここでは、“ [3.2.4 データベースの定義](#) ”で作成したデータベース名 “ db1 ” を指定し、SQL文に以下のように入力し、【実行】ボタンを押します。

```
create schema sch1;
create table sch1.table1(col1 int not null, col2 int, primary key(col1));
```

処理が終了すると実行結果が表示されます。

3.2.6 PowerGres Plus の停止と共有ディスクアンマウント

スキーマ、表、およびインデックスの作成まで完了したなら、PowerGres Plusのインスタンスを停止します。

インスタンスの停止は、“ [3.2.3 インスタンスの起動](#) ”と同様の手順でPostgreSQL Plus Administratorを起動して行います。

注意

PostgreSQL Plus Administratorでデータベースを操作する場合は、必ずインスタンス作成者でログインする必要があります。

PowerGres Plusインスタンスの停止が完了したら、PostgreSQL Plus Administratorを終了させたあと、共有ディスクをアンマウントします。

共有ディスクのアンマウントは、マウントしているサーバ上で、root権限を持つユーザにて行います。

```
# umount /PlusData
# umount /PlusDbspace
# umount /PlusBackup
```

3.3 CLUSTERPRO トレッキングツールによるフェイルオーバクラスタ設定

ここからは、トレッキングツールでの設定となります。

3.3.1 クラスタ情報の生成

共有ディスク型の2ノードクラスタ環境を作成するための設定値を以下の表に用意しました。
この文書では以下の表の条件でクラスタ構成情報を作成し、クラスタ生成することを想定して記載しています。

この表の設定値をもとに管理端末にインストールしたトレッキングツールでクラスタ情報のFDを作成します。

トレッキングツールを使用したクラスタ情報のFDの作成方法については、以下のマニュアルを参照してください。

「CLUSTERPRO for Linux Ver3.0 クラスタ生成編」

「CLUSTERPRO for Linux Ver3.0 トレッキングツール編」

	設定パラメータ	設定値
クラスタ構成	クラスタ名	cluster
	サーバ数	2
	フェイルオーバグループ数	2
	モニタリソース数	6
ハートビートリソース	LANハートビート数	2
	COMハートビート数	1
	DISKハートビート数	1
1台目のサーバの情報 (マスターサーバ)	サーバ名	server1
	インタコネクトのIPアドレス (専用)	192.168.0.1
	インタコネクトのIPアドレス (バックアップ)	192.168.4.21
	パブリックのIPアドレス	192.168.4.21
	COMハートビートデバイス	/dev/ttyS0
	DISKハートビートデバイス	/dev/sda1 /dev/raw/raw1
2台目のサーバの情報	サーバ名	server2
	インタコネクトのIPアドレス (専用)	192.168.0.2
	インタコネクトのIPアドレス (バックアップ)	192.168.4.22
	パブリックのIPアドレス	192.168.4.22
	COMハートビートデバイス	/dev/ttyS0
	DISKハートビートデバイス	/dev/sda1 /dev/raw/raw1
1つ目のグループ (Webマネージャ用)	タイプ	フェイルオーバ
	グループ名	WebManager
	起動サーバ	server1 server2
	グループリソース数	1
1つ目のグループリソース *1	タイプ	floating ip resource
	グループリソース名	WebManagerFIP1
	IPアドレス	192.168.4.11
2つ目のグループ	タイプ	フェイルオーバ

設定パラメータ	設定値
(PowerGres Plus 起動用)	グループ名 PowerGres Plus
	起動サーバ server1 server2
	グループリソース数 5
1つ目のグループリソース	タイプ floating ip resource
	グループリソース名 fip1
	IPアドレス 192.168.4.12
2つ目のグループリソース	タイプ disk resource
	グループリソース名 PlusData
	デバイス名 /dev/sda2
	マウントポイント /PlusData
	ファイルシステム ext3
	ディスクタイプ Disk
3つ目のグループリソース	タイプ Disk resource
	グループリソース名 PlusDbspace
	デバイス名 /dev/sdb1
	マウントポイント /PlusDbspace
	ファイルシステム ext3
	ディスクタイプ Disk
4つ目のグループリソース	タイプ Disk resource
	グループリソース名 PlusBackup
	デバイス名 /dev/sdc1
	マウントポイント /PlusBackup
	ファイルシステム ext3
	ディスクタイプ Disk
5つ目のグループリソース (PowerGres Plus 起動・停止用リソース) 詳細な設定については、次節を参照	タイプ execute resource
	グループリソース名 db-start
	スクリプト 標準スクリプト
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 disk monitor
	モニタリソース名 PlusData-mon
	監視デバイス /dev/sda2
	監視方法 Dummy Read
	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 disk monitor
	モニタリソース名 PlusDbspace-mon
	監視デバイス /dev/sdb1
	監視方法 Dummy Read
	異常検出時 クラスタデーモン停止とOSシャットダウン
4つ目のモニタリソース	タイプ disk monitor
	モニタリソース名 PlusBackup-mon
	監視デバイス /dev/sdc1
	監視方法 Dummy Read
	異常検出時 クラスタデーモン停止とOSシャットダウン
5つ目のモニタリソース	タイプ ip monitor
	モニタリソース名 ipw1
	監視IPアドレス 192.168.4.254 (ゲートウェイ)
	異常検出時 "WebManager" グループのフェイルオーバ *2
6つ目のモニタリソース	タイプ ip monitor
	モニタリソース名 ipw2
	監視IPアドレス 192.168.4.254 (ゲートウェイ)

設定パラメータ	設定値
異常検出時	"PowerGres Plus"グループのフェイルオーバ *2

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

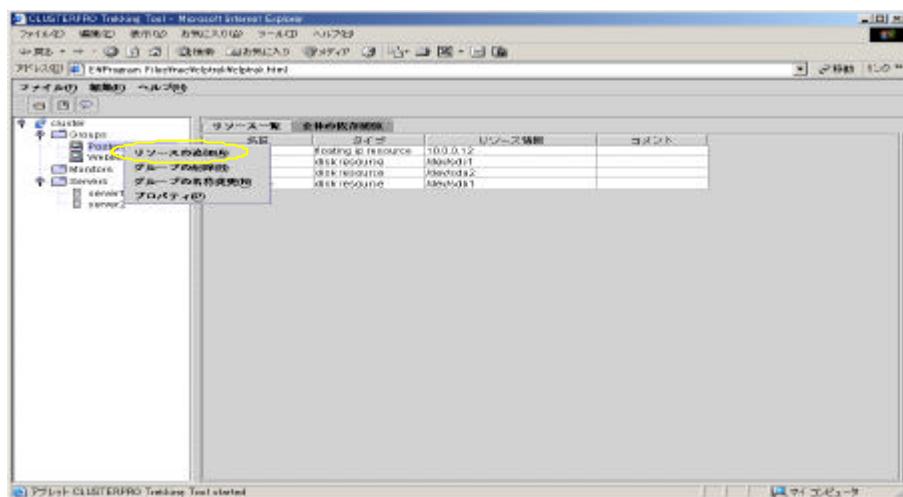
*2: 全てのインタコネクトLANが切断された状態でのフェイルオーバを試行させる設定は「CLUSTERPRO for Linux Ver3.0 トレッキングツール編 モニタリソース」を参照してください。

3.3.2 PowerGres Plus起動・停止用 EXECリソースの追加

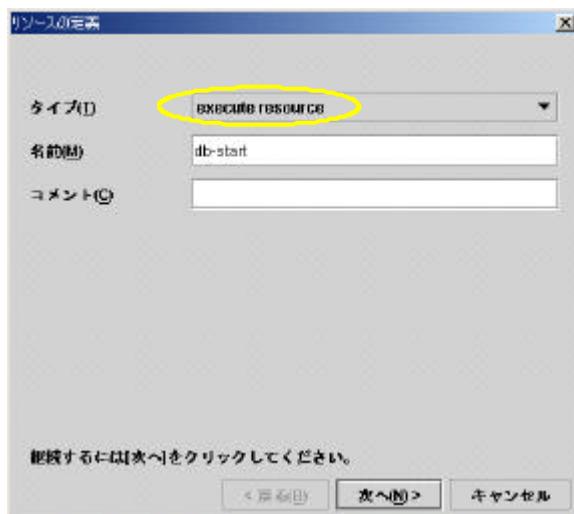
ここでは、トレッキングツールを使用してPowerGres Plus起動・停止用のEXECリソースを追加する方法について記載します。

前節に記載している設定値をもとにPowerGres Plus起動・停止用リソース以外のグループ/リソースがすべて設定されていることを前提として記載します。

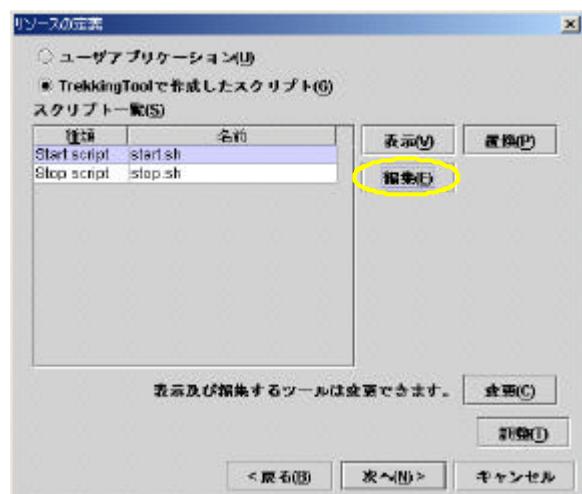
トレッキングツールでPowerGres Plus起動用のグループ (PowerGres Plus) を選択し「リソースの追加」を行ないます。



PowerGres Plus起動・停止用のEXECリソースを追加します。(以下の例では、db-start) リソースタイプとして、「execute resource」を選択します。



「編集」ボタンをクリックすると、エディタが開くので、環境に合わせてスクリプトを修正します。修正ポイントについては、以下に記載します。



start.shにPowerGres Plusの起動(以下、追加部分)を付け加えてください。

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



追加部分
pidfile_check()
{
    if [ -f /PlusData/data1/postmaster.pid ]
    then
        echo "Evacuation of a postmaster.pid file /tmp/postmaster.pid.$$"
        mv /PlusData/data1/postmaster.pid /tmp/postmaster.pid.$$
        fi
    #
    # PowerGres Plus ユーザ名、データベース領域ディレクトリ、ポート番号を適切な値
    # に修正します。
    #
    su - postgres -c "export PGPORT=5432;/usr/local/pgsqlplus/bin/pg_ctl
start -D /PlusData/data1 -w -l /dev/null -o '-i -p 5432'"
        if [ $? -ne 0 ]
        then
            if [ ! -f /PlusData/data1/postmaster.pid ]
            then
                echo "A postmaster.pid file does not exist."
                if [ -f /tmp/postmaster.pid.$$ ]
                then
                    echo "Restoration of a postmaster.pid file /tmp/postmaster.pid.$$"
                    mv /tmp/postmaster.pid.$$ /PlusData/data1/postmaster.pid
                    fi
                fi
                if [ -f /tmp/postmaster.pid.$$ ]
                then
                    rm /tmp/postmaster.pid.$$
                fi
                return 1
            fi
            if [ ! -f /PlusData/data1/postmaster.pid ]
            then
                if [ -f /tmp/postmaster.pid.$$ ]
                then
                    mv /tmp/postmaster.pid.$$ /PlusData/data1/postmaster.pid
                    fi
                fi
                if [ -f /tmp/postmaster.pid.$$ ]
                then
                    rm /tmp/postmaster.pid.$$
                fi
                echo "pidfile_check EXIT(0)"
                return 0
            }
        fi
    
```

追加部分

```

if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
   追加部分
echo "PowerGres Plus start"
su - postgres -c "export PGPORT=5432;/usr/local/pgsqlplus/bin/pg_ctl start -D /PlusData/data1 -w -l /dev/null -o '-i -p 5432'"
if [ $? -ne 0 ]
then
    pidfile_check
    if [ $? -ne 0 ]
    then
        echo "START FAILED"
    fi
fi
追加部分
else
    echo "ERROR_DISK from START"
fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "FAILOVER1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "FAILOVER2"
        else
            echo "ON_OTHER2"
        fi
   追加部分
echo "PowerGres Plus start"
su - postgres -c "export PGPORT=5432;/usr/local/pgsqlplus/bin/pg_ctl start -D /PlusData/data1 -w -l /dev/null -o '-i -p 5432'"
if [ $? -ne 0 ]
then
    pidfile_check
    if [ $? -ne 0 ]
    then
        echo "START FAILED"
    fi
fi
追加部分
else

```

```
        echo "ERROR_DISK from FAILOVER"
    fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

stop.shにPowerGres Plusの停止(以下、追加部分)を付け加えてください。

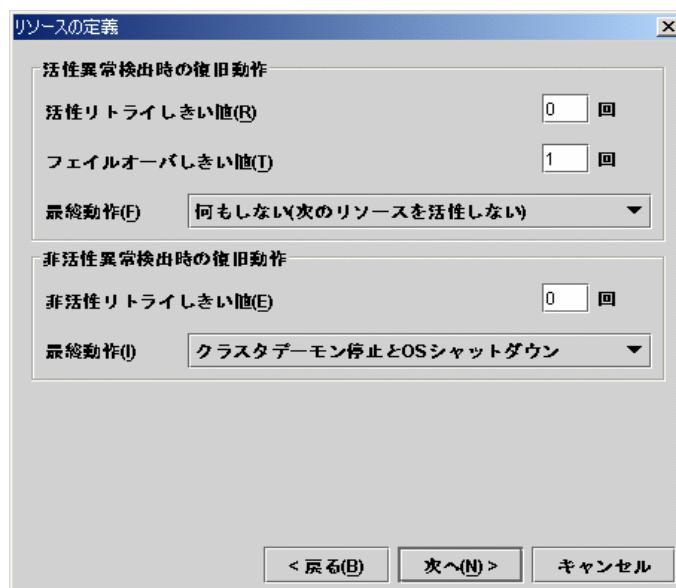
```
#!/bin/sh
#####
#*          stop.sh          *
#####
if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
   追加部分
    echo "PowerGres Plus stop"
    su - postgres -c '/usr/local/pgsqlplus/bin/pg_ctl stop -D /PlusData/data1 -m fast'
    追加部分
    else
        echo "ERROR_DISK from START"
    fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "FAILOVER1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "FAILOVER2"
        else
            echo "ON_OTHER2"
        fi
    追加部分
    echo "PowerGres Plus stop"
    su - postgres -c '/usr/local/pgsqlplus/bin/pg_ctl stop -D /PlusData/data1 -m fast'
    追加部分

    else
        echo "ERROR_DISK from FAILOVER"
    fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

「調整」ボタンをクリックし、開始スクリプト、終了スクリプトとも「同期」になっていることを確認します。(既定値は、同期)

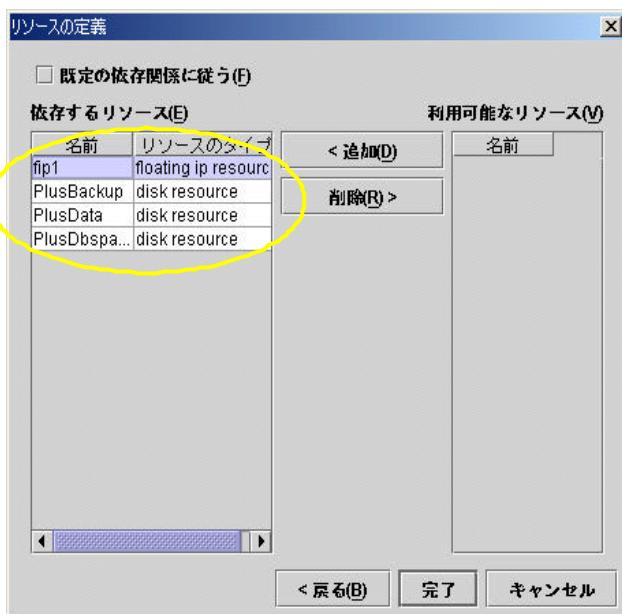


この画面では環境に合わせて設定してください。特に変更しなくても構いません。



この画面では、依存関係を作成します。

依存するリソースの欄にグループ内のすべてのディスクリソースとFIPリソースを設定してください。



「完了」ボタンをクリックすると、PowerGres Plus起動・停止用EXECリソースが作成されます。

3.4 クラスタの生成

トレッキングツールで作成したクラスタ情報のFDをもとにクラスタの生成を行い、ライセンスを登録し、ライセンス登録が終了したら両サーバをリブートしてください。

クラスタの生成、ライセンスの登録については、以下のマニュアルを参照してください。
「CLUSTERPRO for Linux Ver3.0 クラスタ生成編」
(クラスタ生成、ライセンス登録)

3.4.1 PowerGres Plus グループの起動

管理端末上のWebブラウザから、以下の手順にてWebマネージャに接続します。

1. Webブラウザを起動します。
2. WebブラウザのURLにWebマネージャ用グループのFIPアドレスとWebマネージャのポート番号を入力します。
`http://192.168.4.11:29003/`

Webマネージャのポート番号はトレッキングツールの「クラスタのプロパティ」->「ポート番号」->「WebマネージャHTTPポート番号」で設定するポート番号です。

詳細は「CLUSTERPRO for Linux Ver3.0 Webマネージャ編」(接続)を参照してください。

デフォルト設定では、ファイルオーバーグループが「自動起動」になっているため、両サーバの起動後、Webマネージャに接続するとWebマネージャの画面が表示されます。

PowerGres Plusのグループがオンラインになっていることを確認してください。

注意

PowerGres Plus ファイルオーバーグループ設定後に、PostgreSQL Plus Administratorを使用する場合には、“[付録A クラスタ設定後のPostgreSQL Plus Administrator使用時の注意事項](#)”に注意してご利用ください。

第4章 PowerGres Plus フェイルオーバクラスタの確認

それでは、実際に作成した PowerGres Plus フェイルオーバクラスタを確認してみましょう。

SERVER1でPowerGres Plusグループが起動している状態で、PostgreSQL クライアントへ `postgres` ユーザでログインし、以下のコマンドを実行します。

```
$ psql -h 192.168.4.12 -d db1 -c "select * from sch1.table1"
c1
--
1
(1 row)
```

先程作成したデータベースおよびテーブルが、フローティング IP アドレス (192.168.4.12) を経由して参照できます。

次に、`table1` テーブルに 1 ローデータを追加します。

```
$ psql -h 192.168.4.12 -d db1 -c "insert into sch1.table1 values(2)"
```

ここで、SERVER1 のノードがダウンしたと想定し、SERVER1 のコンソールから SERVER1をシャットダウンさせます。

```
# shutdown -h now
```

しばらくすると、SERVER2 は、SERVER1 のダウンを検出し、PowerGres Plus グループの引き継ぎを行います。管理端末のWebブラウザからWebマネージャを監視し、PowerGres Plus の引き継ぎが正しく完了したのを確認し、もう一度 PostgreSQL クライアントから以下のコマンド実行してみてください。

```
$ psql -h 192.168.4.12 -d db1 -c "select * from sch1.table1"
c1
--
1
2
(2 rows)
```

先程フェイルオーバさせる前に追加したローが正しく反映されており、今度は SERVER2 を経由して結果が返されています。

今回は、PostgreSQL クライアントから `psql` コマンドを使って確認しましたが、通常はアプリケーションサーバなどからアクセスすることになるはずです。これで、CLUSTERPRO を使った PowerGres Plus フェイルオーバの解説は終わりです。

付録A クラスタ設定後のPostgreSQL Plus Administrator使用時の注意事項

PowerGres Plus フェイルオーバグループ設定後に、PostgreSQL Plus Administratorを使用する場合には、以下の点に注意してご利用ください。

- PowerGres Plus フェイルオーバグループが、起動した状態で使用してください。
- Webブラウザに指定するURL内のサーバ名には、PowerGres PlusグループのフローティングIP アドレスを指定してください。