

# IBM Power Systems 上での CLUSTERPRO 動作検証結果

2010/01/15

## 1. はじめに

IT 環境では TCO 削減が求められており、サーバーの設置スペースや消費電力の削減、ハードウェア資源の有効利用によりコスト削減を実現する手法の 1 つとしてサーバー仮想化技術を活用したサーバー統合が挙げられます。IBM Power Systems は、複数の OS を 1 台の筐体で稼働させるサーバー仮想化機能(PowerVM)を標準搭載しており、数の少ないサーバーに業務を集約することができます。

一方、OS についてもコスト削減を目的として Linux を採用するケースが増えています。企業や部門を最適化するための統廃合が進む中、Linux はノウハウの再利用や技術者の確保を容易にしています。また、Linux は様々なプラットフォームに対応しているため、導入するシステム毎に最適なプラットフォームを選択することができます。

Linux の用途として旧来は WEB フロントエンドやメールサーバなどが中心でしたが、現在はシステムの基幹部であるリレーショナルデータベース(RDBMS)を安定稼働できるインフラとしての役割を担うまでになっています。

このように利用範囲が広まっている Power System 上の Linux ですが、課題もありました。IBM Power Systems はプロセッサやメモリー、PCI アダプター等の各コンポーネントの高可用性を維持する機能を提供しますが、各論理区画(LPAR)上の業務アプリケーションの障害には対応することができません。また、筐体の電源が全面的に停止する障害には対応することができません。これは RDBMS のインフラとして使用するためには解決しなければならない課題です。RDBMS はデータ保全の要であるため、その可用性を高める事は極めて重要です。

この課題を解決するにはシステム異常を発見した場合に待機系に業務を引き継ぐ HA クラスターソフトウェアを利用する事が一般的ですが、これまでは IBM Power Systems の機能を完全に引き出せる HA クラスターソフトウェアは存在しませんでした。

IBM Power Systems が提供する仮想化機能(PowerVM)の 1 つに Virtual I/O Server(VIOS)があります。VIOS を利用することで複数の LPAR で PCI デバイスを共有利用することが可能になり、必要最小限の PCI デバイスでネットワークや外部ディスクを共有利用することができます。しかし既存の HA クラスターソフトウェアは物理的なディスク環境に依存しているため、VIOS 構成環境での外部ディスクの制御が出来ませんでした。

NEC CLUSTERPRO はハードウェア環境に依存せずに稼働できるという特徴を持っており、VIOS 構成環境においても使用可能です。

本稿では IBM Power Systems 上に VIOS 環境、Linux、RDBMS (DB2 及び Oracle)、CLUSTERPRO を組み合わせた環境を構築し、クラスターとして正しく動作することを検証した結果を報告します。

## 2. 環境構成

### 2.1. ハードウェア構成

本検証では、筐体単位でのメンテナンスを含め、お客様環境を想定して物理的に 2 筐体を使用し、各筐体上に論理区画(Logical Partition, 以下 LPAR)を構成して筐体間での引き継ぎを行うクラスター構成としています。

各 LPAR には、OS、アプリケーションが使用するネットワークやディスク I/O 用に物理アダプター、仮想アダプターを構成、更には物理/仮想を共存させることも可能ですが、今回は柔軟な構成を可能にする仮想アダプターを構成しています。

クラスターを構成するノード間の共有ディスクには Storage Area Network (SAN) ディスクを使用し、SAN スイッチ経由で接続しています。

#### 2.1.1. 仮想アダプターと Virtual I/O Server

IBM Power Systems が提供する仮想化機能(PowerVM)の 1 つに、仮想アダプターがあります。仮想アダプターは、LPAR 上でネットワークならびにディスク I/O 用のインターフェースを提供し、名称のとおり各 LPAR には物理アダプターの割り当てを必要としません。一方、外部ネットワークとの通信や SAN ディスクのような物理ディスクへのアクセスには物理アダプターでの接続が必要です。各 LPAR 上に構成する仮想アダプターと外部ネットワークやディスクとの接続、つまり仮想と物理の橋渡しを図 1 のように Virtual I/O Server (以下 VIOS) が担います。

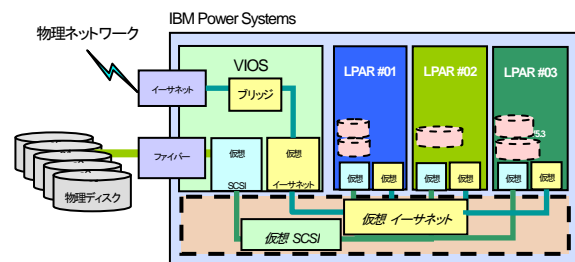


図 1 Virtual I/O Server 概要

VIOS 区画にはネットワークやディスクと接続に使用する物理アダプターを構成し、その物理アダプターと各 LPAR 上に構成した仮想アダプター間の橋渡しを VIOS が行います。可用性ならびに保守性を考慮して VIOS に構成する物理アダプターを冗長化することが可能であり、更には VIOS 区画自体を 1 筐体内に冗長構成とすることが可能です。

### 2.1.2. ディスク構成

本検証におけるディスク構成を図 2 に示します。

2 台の Power Systems 上に各々 LPAR を構成し、各 LPAR には OS 領域用とデータエリア用のディスクをそれぞれ割り当てます。外部 SAN ディスクに RAID5 を構成し、OS 領域 / データエリア共に RAID5 上に作成した LUN を使用します。

OS 領域用には LPAR 毎に独立した形態でディスクを割り当てます。この OS 領域用ディスクは CLUSTERPRO による引き継ぎ対象とはなりません。一方、データエリアは引き継ぎ対象とするため、クラスターを構成する LPAR 間で共有ディスクとして構成します。本検証では、この共有ディスク上にデータベースが使用するデータを配置し、移動系から待機系への引き継ぎ後にも同一データにアクセス可能な環境としています。データベースの構成に関しては、後述 2.3, 2.4 章 データベース構成 を参照ください。

次に、仮想化環境における LPAR へのディスクの割り当てに関して記します。各筐体には可用性/保守性を考慮して VIOS を冗長構成としています。クラスターを構成する LPAR は、この冗長化した VIOS が各々提供するパスを経由してディスク(LUN)を multi-path ディスクとして構成します。このように 1 筐体内に VIOS を冗長化し、各 VIOS から同一 LUN に対してパスを提供する multi-path 構成にすることにより、片側の VIOS が停止中であっても LPAR はもう片側の VIOS 経由でディスクにアクセスする事を可能し、可用性を高めています。

(例) 図中の“VIOS1-1”が停止中であっても“VIOS1-2”経由で RHEL5.4 およびデータエリアにアクセス可能

また、各 VIOS から外部 SAN ディスクへのアクセスに使用する物理アダプター(本検証ではファイバー・チャネル(FC)アダプター、図中の F)を冗長化し、物理アダプターの障害にも耐え得る構成としています。

LPAR の視点からは、同一 LUN に対して 2 VIOS を経由した 4 パスが確保されている構成となり、LPAR 上に 2 物理 FC ポートを構成した環境より高い可用性レベルが提供されることとなります。当構成では VIOS に物理的に構成している 4 ポートの接続が全て切れるまで LPAR はディスク障害を検知することなく継続稼働可能な構成となります。

### 2.1.3. ネットワーク構成

本検証におけるネットワーク構成を図 3 (次ページ)に示します。ディスク構成と同様、2 台の Power Systems 上の各々 LPAR を作成し、各 LPAR にはパブリックとインタコネクットの 2 系統の LAN を構成します。パブリック LAN はサービス用、インタコネクット LAN はクラスターを構成する LPAR 間の死活監視の 1 つであるネットワーク経由のハートビート用に使います。

仮想化環境における LPAR 上のネットワークの構成に関してですが、各筐体に冗長化した VIOS 経由により複数経路が確保されます。これにより、片側 VIOS が停止中の場合も LPAR 上に構成した仮想アダプターはもう片側の VIOS を経由して外部ネットワークと通信する事が可能です。物理アダプターの耐障害性を考慮して VIOS 上に割り当てるアダプターを冗長化し、EtherChannel もしくは EtherChannel Backup を構成することも可能ですが、本検証においては EtherChannel/EtherChannel Backup 共に構成していません。

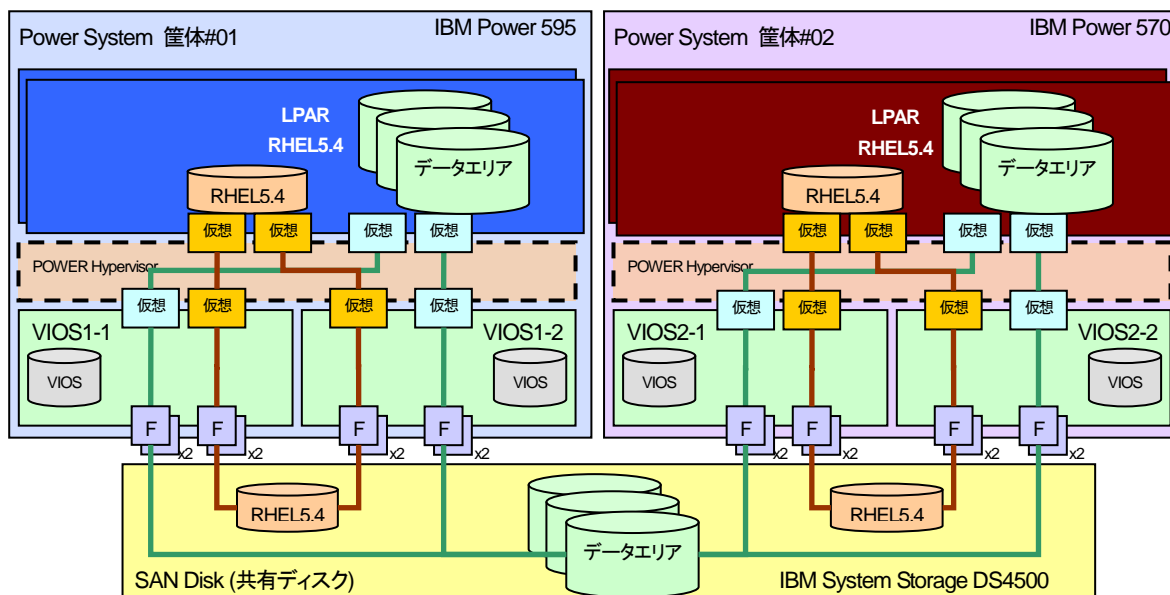


図 2 検証環境 ディスク構成

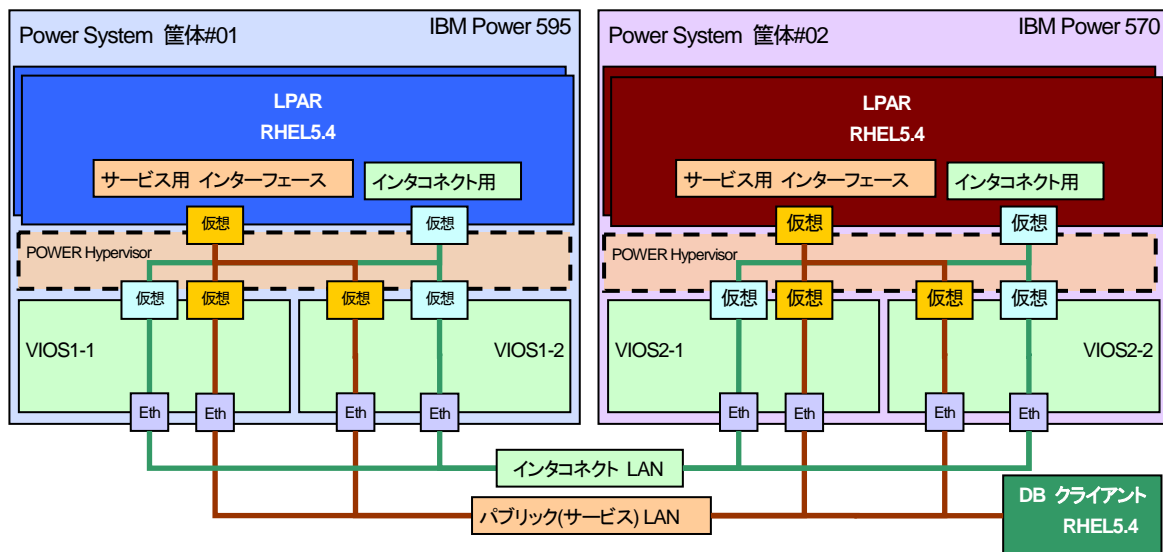


図 3 検証環境 ネットワーク構成

なお、本検証環境である VIOS 冗長構成において LPAR からの経路を 2 経路提供する機能は SEA Failover により実装しています。VIOS に Primary と Backup を定義し、Primary 側の VIOS が停止した場合には Backup 側の VIOS 経由での通信に自動的に切り替わります。SEA Failover 機能は VIOS が提供し、冗長 VIOS の死活監視は VIOS が行います。

## 2.2. OS構成

LPAR へ Red Hat Enterprise Linux 5.4 (以降、RHEL5.4) for IBM POWER を導入し、共有ディスクでわりあてた LUN へマルチパスアクセスするためのドライバー構成、パーティション作成、RAW デバイス定義を行います。

### 2.2.1. Linux導入

2 つの VIOS により冗長化されたストレージ接続パスを使用して、外部ディスクストレージの LUN に RHEL5.4 を導入します。通常の RHEL5.4 の導入手順と比べて主に次の 2 点が異なります。

- ・ インストール時には、RHEL 5.4 に同梱されている Device Mapper Multipath(以降、DMMP)を使用します。
- ・ 同一ネットワークセグメント上の PC にて VNC Viewer を使用し、ネットワーク経由で GUI のインストーラーを操作してインストールします。

また、VIOS においては次の設定を行います。

- ・ 2 つの VIOS においては、RHEL5.4 をインストールする LUN の hdisk の属性「reserve\_policy」を予め「no\_reserve」にします。
- ・ RHEL5.4 導入後は、LPAR の SMS にてブートリストに 2 つの vSCSI を指定します。

インストール作業は次の手順で実施します。

#### 1) メディア準備

インストール対象の LPAR に DVD ドライブを接続し、RHEL 5.4 のインストール・メディアをセットします。

#### 2) LPAR 起動

LPAR を起動して、SMS メニューより、DVD ドライブを指定して Normal Mode Boot を実行します。

#### 3) インストーラー起動

コンソールに、インストーラーの起動メッセージが表示されますので、boot: プロンプトに続けて、次のオプションを入力します。マルチパスドライバー DMMP を使用するために、「mpath」を指定します。また、VNC を使用しますので、vnc 関連のオプションを指定します。IP アドレスやパスワードは実際に使用するものを指定します。

入力例：

```
linux vnc vncpassword=xxx ip=10.7.10.123
netmask=255.255.0.0 mpath
```

#### 4) VNC 接続

VNC Viewer からの接続を要求するメッセージが表示されますので、同一ネットワーク上の PC の VNC Viewer から、GUI のインストーラー画面に接続します。認証画面が表示されますので、先に vncpassword で指定したパスワードを入力すると、GUI インストーラーである Anaconda の画面が表示されます。

これ以降の手順は通常の RHEL と同じ手順ですが、インストール先デバイスがマルチパスデバイスになっていることを確認します。

- ・ ディスクパーティションレイアウト作成時に、確認画面を表示させ、デバイス名が「/dev/mapper/mpath0」になっていることを確認します。
- ・ デバイス名が「/dev/sda」等になっている場合は、マルチパス環境が正しく構成されていません。

次に、Dynamic LPAR 等の PowerVM 仮想化機能や、信頼性・可用性・保守性(RAS)の機能を RHEL5.4 で利用するために、IBM から提供される Service and productivity tools を追加導入します。

#### 1) rpm パッケージ準備

次のサイトから該当するシステムのリンクを選択し、

「RHEL 5」のタブにある rpm パッケージファイルをダウンロードします。

<http://www14.software.ibm.com/webapp/set2/sas/f/lopdiag/s/home.html>

今回の構成では、「Red Hat」カテゴリの「on HMC-or IVM-managed servers」を選択します。

## 2) rpm パッケージ導入

ダウンロードした RPM ファイルを導入対象の LPAR にファイル転送し、rpm コマンドで導入します。

実行例：

```
# rpm -Uvh*.rpm
```

以上で導入作業は終了ですが、マルチパスドライバー DMMP を使用して導入した RHEL5.4 環境は、運用するにあたり、次の点に注意してください。

インストーラーにより、下記の3つのファイルに、ストレージ LUN のユニークな ID である wwid 番号が自動的に書き込まれます。システムバックアップ・リストア時などに新しい LUN へ導入する必要がある場合は、この wwid を新しい wwid に書き換える必要があります。

### 1) /var/lib/multipath/bindings ファイル

マルチパスデバイス mpathX と wwid の関連付けが定義されています。LUN の異なる環境へリストアする際はリストア直後に新しい wwid へ変更してください。

### 2) /etc/multipath.conf ファイル

DMMP にて管理する対象 LUN の wwid が定義されています。LUN の異なる環境へリストアする際は「blacklist\_exceptions」の wwid を新しいものへ変更してください。

### 3) /boot/initrd-XXX ファイル

Linux 起動時に DMMP がマルチパスデバイスとして設定する LUN の wwid が、起動スクリプトファイルで定義されています。LUN の異なる環境へリストアする際は、上記の2ファイルを書き換えてから、mkinitrd コマンドでこのファイルを再作成してください。

## 2.2.2. データベース領域マルチパスドライバー構成

Linux でのデータベース領域のマルチパス構成は、2つの VIOS により冗長化されたストレージ接続パスを使用するため、RHEL5.4 のインストール時と同様に、マルチパスドライバー DMMP を使用して構成します。

構成は以下の手順で実施します。

### 1) wwid 確認

LUN 毎の wwid を scsi\_id コマンドで調査します。

実行例：

```
# /sbin/scsi_id -g -u -s /block/sdc
```

### 2) DMMP 設定ファイル編集

wwid を DMMP 設定ファイルに記述します。

設定ファイル「/etc/multipath.conf」の「blacklist\_exceptions」に wwid を追記

### 3) デバイスファイル生成

設定ファイルの変更を反映させます。

「multipath」コマンドを実行することで、DMMP に LUN を認識させ、デバイスファイル (例: /dev/mapper/mpath1) を生成させます。

データベースソフトウェアや OS 上のアプリケーション (fdisk 等) からは、ここで生成したデバイスファイルを指定して LUN へアクセスします。

尚、VIOS を使用しない構成では、マルチパスドライバーは DMMP ではなく、RDAC を使用する必要があります。RDAC の入手については次のサイトをご参照ください。

<http://www.lsi.com/rdac/>

クラスター環境で RDAC を使用する場合は、メインパス障害の時に次のようなパスの切り替えを繰り返す現象が発生します。

1. メインパス障害の発生ノードにマッピングされた全ての LUN でコントローラーの Ownership が切り替わります。これは、他ノードが使用中の LUN も含みます。
2. 他のノードが Preferred Path 経由での該当 LUN への接続を検知し、Ownership を Preferred Path に戻します。
3. メインパス障害ノードが再度 Ownership を切り替えます。

以後、2. と 3. を繰り返す現象が発生するため、約1分毎に一定量のエラーメッセージがシステムログに出力されます。切り替え自体はわずかな時間で行われますのでパフォーマンス上の影響はほとんどありませんが、繰り返す現象を発生させないように、RDAC の設定変更が可能です。

RDAC の設定項目である Disable LUN Rebalance を設定し、優先パスの復旧時に Ownership の切り戻しを発生しないようにする手順です。VIOS を使用しない環境においては、必要に応じて設定してください。

- 1) /etc/mpp.conf をエディターで開き、DisableLUNRebalance の値を 3 に変更します。
- 2) mppUpdate コマンドで initrd イメージを作り直します。
- 3) システムをリブートし、新しい initrd イメージで起動します。

## 2.2.3. 共有ディスクパーティション構成

CLUSTERPRO を用いた HA クラスター環境では、クラスターサーバー間での死活監視に使用するクラスタパーティションが必要となります。また、今回の構成では Oracle のデータベース領域として多くのパーティションが必要となります。

これらのパーティションは、必要なパーティション数の LUN を共有ディスクに作成し、マッピング、マルチパス構成することが可能ですが、1つの LUN を複数パーティションにわけて構成することも可能です。今回は、1つの LUN を複数パーティションにわけて構成する方法を用います。

パーティションの作成は RHEL5.4 の `fdisk` コマンドを使いますが、マルチパス環境では、マルチパスデバイスに対して `fdisk` コマンドを実行します。

実行例

```
# fdisk /dev/mapper/mpath1
```

今回作成した各パーティションの用途の詳細については、「2.5.3 ディスク構成」に後述しています。

### 2.2.4. RAWデバイス定義

前述の手順で作成した、CLUSTERPRO の構成に必要なクラスタパーティションは、RHEL5.4 にて RAW デバイスとして定義しておく必要があります。また、今回の構成では、Oracle のデータベース領域についても RAW デバイスを使用するため、同様に RHEL5.4 にて RAW デバイスとして定義しておく必要があります。

Linux 上での RAW デバイスの定義は、RHEL5.4 に同梱される `rawdevices` サービス機能を使用します。構成は以下の手順で実施します。

#### 1) デバイス定義

`rawdevices` 設定ファイルに実際のデバイス名と RAW デバイス名との関係を定義します。

設定ファイル「`/etc/sysconfig/rawdevices`」に、「`/dev/raw/raw1 /dev/mapper/mpath1`」のように指定します。必要な RAW デバイスの数だけ定義します。

#### 2) デバイスファイル生成

「`service rawdevices restart`」コマンドを実行することで設定ファイルの変更を `kernel` に認識させ、デバイスファイル（例：「`/dev/raw/raw1`」）を生成させます。このファイルは OS を再起動すると消えてしまいます。

#### 3) 自動生成設定

「`chkconfig rawdevices on`」コマンドを実行し、OS 再起動後も自動的にデバイスファイルを生成するようにします。

ここで実施した `rawdevices` サービス機能による定義方法の他に、`udev` 機能でも実施することができます。詳細についてはこちらの Web ページをご参照ください。

<http://kbase.redhat.com/faq/docs/DOC-10164>

CLUSTERPRO や Oracle においては、ここで生成したデバイスファイルを指定します。

## 2.3. データベース構成 (DB2)

この章では今回の検証で IBM DB2 (以下 DB2) を CLUSTERPRO 上に導入した際の情報をまとめます。

### 2.3.1. 導入ソフトウェア

導入したソフトウェアは以下の通りです。

- DB2 本体
  - DB2 v9.7 Enterprise Server Edition for POWER Linux (64bit)
- DB2 が必要とするコンポーネント

- Runtime for XL C/C++ Advanced Edition for Linux, V9.0 (C++ランタイム)
- `libaio`, `pdksh` (OS の CD-ROM より導入)

DB2 for Linux を動作させるために必要な追加ソフトウェアは、以下の URL にまとめられています。

<http://www.ibm.com/db2/linux/validate>

### 2.3.2. 導入計画 (物理配置)

今回の DB2 検証環境を図 4 に示します。

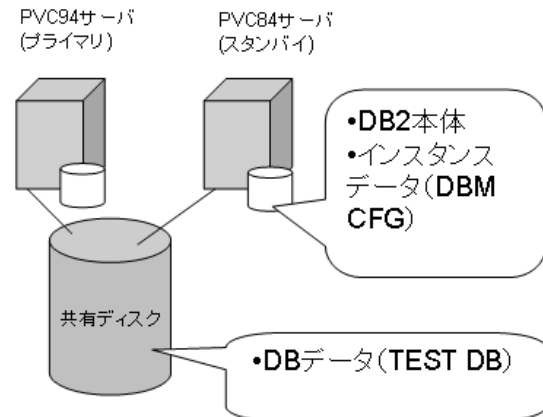


図 4 DB2 検証環境

DB2 を共有ディスク形式のクラスタ環境に導入する際には、まず以下のコンポーネントをローカルディスクと、共有ディスクのどちらに導入するべきかを決定する必要があります。

DB2 本体とインスタンスデータはローカルディスクにも共有ディスクにも格納可能です。DB データは必ず、共有ディスク上に格納します。

DB2 本体とインスタンスデータの両方をローカルディスクに導入すると、共有ディスクをマウントしない状態で DB2 インスタンスを起動したり、パッチを適用したりすることができます。

これにより稼働系を業務で使用しながら、待機系にパッチ (Fix Pack) を適用する事が可能になります。一般的にクラスタシステムを構築する要件ではできる限り停止時間を短くするのが望ましいので、今回は表 1 のようにインスタンスデータをローカルディスクに配置しました。

表 1 ディレクトリ構成・配置

コンポーネント	ディスク装置	パス
DB2 本体	ローカル	<code>/opt/ibm/db2/V9.7/</code>
インスタンスデータ	ローカル	<code>/home/ユーザ名/</code>
DB データ	共有	<code>/db2data/sharedb/</code>

この構成では以下の2つの注意点があります。

- 1) DBデータを保存するディレクトリは完全に空(1つもファイルが無い状態)である必要があります。  
Linuxではデバイスのトップディレクトリ(今回は/db2dataとしてマウントしたディレクトリ)にはLOST+FOUNDディレクトリが自動的に作られているケースが多いので、そこには直接データが格納できません。今回はその下にsharedb/ディレクトリを作る事で対応しています。
- 2) インスタンスレベルでパラメータを変更した場合は、両系統で更新する必要があります。  
今回はインスタンスデータをローカルディスクに別々に保存しているため、インスタンスレベルでパラメータを変更した場合は、もう片方でも同様の操作をしてパラメータを合わせる必要があります。

インスタンスレベルのパラメータ変更は、DBM CFG(データベース・マネージャ・コンフィグ)の変更と、データベース作成や削除がそれに当たります。

### 2.3.3. 導入計画 (DB2のユーザID)

導入前にDB2のユーザIDについて決定しておく必要があります。DB2では一般的に3つのユーザを使用します。

- インスタンスユーザ (インスタンスオーナー)
- 分離ユーザ (フェンスユーザ)
- 管理サーバユーザ(Admin ユーザ)

インスタンスユーザは、インスタンス(DB2のプログラムそのもの)を管理するためのユーザです。分離ユーザはストアプロシージャやユーザ定義関数を実行する際に使用されるユーザです。

管理サーバユーザは管理サーバの起動と停止のために必要ですが、DB2の動作に必須ではありませんので、今回は作成していません。そのためインスタンスユーザと分離ユーザを以下のように計画しました。

	ユーザ(ID番号)	パスワード	グループ(ID番号)
インスタンスユーザ	db2inst1 (500) ※ポート番号:50000	password	db2group (500)
分離ユーザ	db2fenc1 (501)	password	db2group (500)

値は任意ですが、ID番号まで決めておく事が大切です。ID番号を両系統で統一しておく事で正しくディスクパーミッションの設定が可能になります。

また、インスタンスユーザにはTCP/IPのポート番号が一つ必要になります。1,024より大きい値で、他のデーモン(サービス)が使用していないポート番号を選択してください。今回はデフォルト値の50,000を計画しました。

### 2.3.4. プライマリへの導入手順

今回の構成では、DB2本体をローカルディスクに導入する

ため、プライマリとサーバー、計2回導入を行う必要があります。まずプライマリ(PVC94)に導入します。

- 1) 前提ソフトウェアの導入  
rpmコマンドでC++ Runtimeやlibaio, pdkshを導入します。

- 2) DB2本体の導入  
DB2のインストーラーはGUIのdb2setupとコマンドライン形式のdb2\_installがあります。今回はどんな環境でも使用できるdb2\_installを使用して導入しました。

DB2本体のファイル(DB2\_ESE\_97\_Linux\_ipSeries.tar.gz)をrootユーザで展開し、中にあるdb2\_installを実行します。今回はインストールディレクトリを/opt/ibm/db2/V9.7(デフォルト値)とし、導入プロダクトにESE(Enterprise Server Edition)を選択しました。

- 3) インスタンスユーザ・分離ユーザの作成  
計画に沿ってまずインスタンスユーザが使用するTCP/IPポートを登録します。/etc/servicesファイルに以下のエントリを追加します。

```
db2c_db2inst1 50000/tcp
(db2c_db2inst1は任意の名前)
```

次にOS上にインスタンスユーザと分離ユーザを作成します。

```
# groupadd -g 500 db2group
# useradd -g db2group -u 500 db2inst1
# passwd db2inst1

# useradd -g db2group -u 501 db2fenc1
# passwd db2fenc1
```

OS上のユーザを使ってDB2のインスタンスを作成します。

```
#chmod 777 /tmp (※/tmpにdb2inst1がアクセス権限を持っていないと警告が出るため)
#/opt/ibm/db2/V9.7/instance/db2icrt -p 50000 -u db2fenc1 db2inst1
```

- 4) インスタンスの設定  
インスタンスに最低限必要な設定を行います。

```
# /opt/ibm/db2/V9.7/instance/db2iauto -off db2inst1
(自動起動の停止)
# su - db2inst1
> db2start
> db2set DB2COMM=TCP/IP
> db2 update dbm cfg using SVCENAME db2c_db2inst1
(TCP/IP通信が出来るように設定)
> db2stop
> db2start
(インスタンスを再起動しパラメータを反映)
> exit
```

- 5) 共有ディスクをプライマリ側にマウント  
rootユーザで/db2dataとしてマウントしました。マウントの情報を/etc/fstabに記述するとOSが自動的にマウントして

しまい、CLUSTERPRO が制御できなくなってしまうため /etc/fstab に記述はしないでください。

#### 6) データを保存するディレクトリを作成

root ユーザでディレクトリを作成し、そのオーナーをインスタンスユーザ (db2inst1:db2group) に変更します。これで DB2 からデータを読み書き出来るようになります。

```
# mkdir /db2data/sharedb/  
# chown db2inst1:db2group /db2data/sharedb/
```

#### 7) テスト用データベースの作成

作成しておいた共有ディスク上のディレクトリ内にデータベースを作成し、接続可能かを確認します。

```
# su - db2inst1  
> db2 CREATE DB TEST1 ON /db2data/sharedb/  
> db2 CONNECT TO TEST1  
> db2 TERMINATE
```

これでプライマリ側の DB2 導入が完了しました。インスタンスを停止しておきます。

```
> db2stop  
> exit
```

### 2.3.5. スタンバイへの導入手順

スタンバイ側への導入手順は、プライマリと同じです。プライマリ側の手順 1)~5)を実行してください。6)以降はすでに共有ディスク上にデータがあるため実行しないでください。

プライマリ側でデータベースを作成しましたが、スタンバイ側では作成していないので、スタンバイ側でデータベースが存在する事を CATALOG コマンドで認識させる必要があります。

```
> db2 CATALOG DATABASE TEST1 ON /db2data/sharedb/
```

これでスタンバイ側からも DB に接続できるようになります。

```
> db2 CONNETC TO TEST1  
> db2 TERMINATE  
> db2stop  
> exit
```

### 2.3.6. DB2 クライアントの設定

DB2 クライアントからは、CATALOG コマンドでリモートサイトのデータベースを認識させます。CLUSTERPRO で使用する場合は、一般的にフローティング IP アドレスがサーバー側に用意されるので、CATALOG でもフローティング IP を指定します。

#### 1) クライアント用のインスタンス作成

クライアントマシン上でクライアント用のインスタンスを作成します。

(DB2 クライアントを導入後)

```
# chmod 777 /tmp  
# groupadd -g 500 db2group  
# useradd -g db2group -u 500 db2inst1  
# passwd db2inst1  
# /opt/ibm/db2/V9.7/instance/db2icrt -s client db2inst1
```

#### 2) リモート DB をカタログ

CATALOG TCPIP コマンドでリモート DB の IP アドレスとポート番号を指定する。フローティング IP アドレスとして 172.16.11.4 を使用して設定した。

```
# su - db2inst1  
> db2 CATALOG TCPIP NODE DB2SVR REMOTE 172.16.11.4  
SERVER 50000  
> db2 CATALOG DB TEST1 AT NODE DB2SVR  
> db2 CONNECT TO TEST1 USER db2inst1 USING password  
> db2 TERMINATE
```

### 2.3.7. DB2 をクラスターから制御するコマンド

DB2 には、ユーザが通常使用するコマンド以外にクラスターソフトウェアから DB2 を制御するための db2gcf コマンドが用意されています。

これらはタイムアウトが指定でき、環境変数を設定せずに直接利用できるという特徴がありますので、クラスターソフトウェアの制御スクリプトをカスタマイズする際はこれらのコマンドを利用すべきです。

インスタンスの開始	db2gcf -u -i <インスタンス名> -t <タイムアウト秒数> -L
インスタンスの停止	db2gcf -d -i <インスタンス名> -t <タイムアウト秒数> -L
インスタンスの強制停止	db2gcf -k -i <インスタンス名> -t <タイムアウト秒数> -L
インスタンスの状態表示	db2gcf -s -i <インスタンス名> -t <タイムアウト秒数> -L

db2gcf コマンドの詳細は DB2 のマニュアルに記載されています。

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.cmd.doc/doc/r0010986.html>

### 2.3.8. CLUSTERPRO からの DB2 の監視

CLUSTERPRO からの DB2 を監視する場合は以下の 2 つの方法が考えられます。

#### 1) データベース接続の試行

データベースに実際に接続し、表作成などの SQL を定期的に行うことで正常稼働しているかどうかを確認します。CLUSTERPRO の DB2 モニタリソースは自動的にこの接続試行を行います。

#### 2) インスタンスの状況監視

インスタンス全体の状況を監視するには、インスタンス全

体の親プロセスである db2sysc (Windows 環境では db2syscs.exe) の存在を ps コマンド等で監視するか、db2gcf コマンドでステータスを監視する方法があります。上記の監視を実施して正常であれば 0、異常であればそれ以外の値を返却するシェルスクリプトを作成し、CLUSTERPRO のカスタムモニタリソースに登録することで、インスタンスの状況監視を CLUSTERPRO に組み込むことができます。

今回は 1) を設定しました。DB2 モジュールの設定では DB2 導入ディレクトリ (ライブラリへのパス) 指定と、データベースの文字コードを設定する必要があります。

デフォルト設定で導入した場合、ディレクトリは /opt/ibm/db2/V9.7/ になり、ライブラリはその直下にあるため、モジュールに指定するのは /opt/ibm/db2/V9.7/lib32/ もしくは /opt/ibm/db2/V9.7/lib64/ になります。CLUSTERPRO の POWER Linux 用モジュールは 64bit パイナリですので、/opt/ibm/db2/V9.7/lib64/ を指定しています。

文字コードは、DB2 でデータベースを作成(CREATE DATABASE)した際に指定したものと同一ものを指定します。指定しないで CREATE DATABASE した場合のデフォルト値は Unicode(UTF-8) です。本検証では UTF-8 を使用しています。

## 2.4. データベース構成 (Oracle)

この章では本検証における Oracle 環境に関する情報をまとめます。

### 2.4.1. 導入ソフトウェア

使用したソフトウェアは以下のとおりです。

- Oracle プロダクト
  - Oracle Database 10g Release 2 Enterprise Edition for Linux on POWER
    - PSR 10204 を適用

### 2.4.2. 導入計画 (物理配置)

Oracle環境においてもDB2と同様、各種設定ファイルをローカルディスクもしくは共有ディスクのどちらに配置するかを検討する必要があります。本検証環境では、Oracleプロダクトならびに各種設定ファイル(init.ora, listener.ora, tnsnames.ora)をローカルディスク、DBデータおよびアーカイブ・ログを共有ディスク上に配置しています(図 5)。なお、DBデータ領域としてはRAWデバイス構成としています。

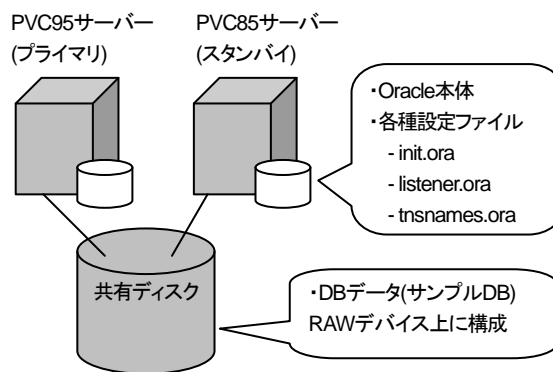


図 5 Oracle 検証環境

## 2.4.3. CLUSTERPROからのOracle の監視

CLUSTERPRO からの Oracle を監視する場合は以下の 2 つの方法が考えられます。

### 1) データベース接続の試行

データベースに実際に接続し、表作成などの SQL を定期的に行うことで正常稼働しているかどうかを確認します。CLUSTERPRO のモニタリソースは自動的にこの接続試行を行います。モニタリソースについては「2.5.6 モニタリソース」を参照ください。

### 2) Oracle プロセスの監視

Oracle がデータベースとして稼働するために必要なプロセスの存在を ps コマンド等で確認し、データベースの死活を監視します。プロセスの存在を確認する監視を実施して正常であれば 0、異常であればそれ以外の値を返却するシェルスクリプトを作成し、CLUSTERPRO のカスタムモニタリソースに登録することで、インスタンスの状況監視を CLUSTERPRO に組み込むことができます。

なお、本検証においては 1) を設定しました。

## 2.5. クラスタ構成

本検証では、各 Power Systems の各論理区画(LPAR)上に CLUSTERPRO をインストールし、両方向スタンバイ構成 (但しデータベースは片方向構成で実施しました)。

### 2.5.1. 導入ソフトウェア

本検証では、クラスタリングソフトウェアとして次の製品を使用しました。

- CLUSTERPRO 本体
  - CLUSTERPRO X 2.1 for Linux
    - ◇ アップデート(CPRO-XL050-05)適用
- データベース監視用オプション
  - CLUSTERPRO X Database Agent 2.1 for Linux<sup>1</sup>

<sup>1</sup> ppc 版のデータベース監視は、本検証に合わせて開発し、対応完了したものを使用しました。入手方法については末尾の NEC 連絡先にお問い合わせください。

## 2.5.2. ネットワーク構成

CLUSTERPRO では、ハートビート経路に冗長性を持たせるため、下記インタコネクタ LAN を 1 系統以上含む、計 2 系統以上のネットワーク経路を使用することを推奨しています。

- インタコネクタ LAN  
クラスターを構成するサーバ間で、互いの死活監視やクラスター情報の交換に使用。他用途での通信には使用しないことを推奨。(ハートビート LAN とも呼称します)
- パブリック LAN  
クライアントとの通信に使用する業務 LAN。インタコネクタ LAN 異常時の、バックアップ用としても使用。

本検証では、インタコネクタ LAN およびパブリック LAN をそれぞれ 1 系統ずつ、計 2 系統を使用しました。

## 2.5.3. ディスク構成

ストレージ内の同一ボリュームに対し、用途別に以下のパーティションに区切って使用しました。

なお、これらのパーティションは OS によって自動マウントされないよう、CLUSTERPRO のインストール前に設定しておく必要があります。

- クラスタパーティション  
クラスターを構成するサーバ間での、互いの死活監視(ディスクハートビート)に使用。  
共有ディスクを使用する場合は、切替パーティションを含む LUN 毎に 1 つ以上の作成を推奨します。  
本パーティションは CLUSTERPRO のディスクハートビートリソースが RAW で Read/Write のアクセスを行いますので、ファイルシステムを作成する必要はありません。
- 切替パーティション(ext3 形式)  
業務データの格納に使用します。  
この領域は、移動系側において CLUSTERPRO のディスクリソースによりアクセス制御および mount され、アクセスが可能になります。  
ext3, xfs など主要なファイルシステムに対応。(ただしジャーナリングファイルシステムを推奨)  
本検証では、DB2 のデータベース格納用領域として使用しました。
- 切替パーティション(RAW 形式)  
RAW 形式でアクセスされる業務データの格納に使用します。  
この領域は、移動系側において CLUSTERPRO の RAW リソースにより OS の RAW デバイスにマップ(bind)され、アクセスが可能になります。  
本検証では、Oracle のデータベース格納用領域として使用しました。

## 2.5.4. ハートビートとネットワークパーティション解決

CLUSTERPRO では、サーバ間の死活監視のためハ-

ートビートを実施しています。

万一ハートビートが行えなくなると、スプリットブレイン状態になり、業務停止や、最悪の場合共有ディスクの二重マウントによるデータ破壊が発生しかねないため、以下の機構を採用することでそのような状況を防止しています。

- 複数の経路使用  
本構成では次の経路を使用することで、一部がハードウェア故障等により不通になってもハートビートが継続可能な構成としています。
  - LANハートビート(「2.5.2ネットワーク構成」に記載の 2 系統を使用)
  - ディスクハートビート(「2.5.3ディスク構成」に記載のクラスタパーティションを使用)
- Kernel LAN ハートビートの使用  
高負荷環境での LAN ハートビート送受信を確実に行うため、各系統とも User 空間のほか Kernel 空間でもハートビートを実施しています。

## サーバ間監視 (LAN, ディスクハートビート)

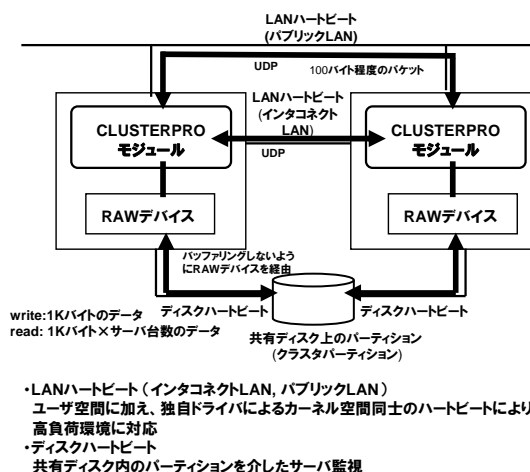


図 6: ハートビート経路

また、万一全てのハートビート経路で異常が発生した場合でも、スプリットブレインによるデータ破壊を防ぐため、次の機構を採用しています。

- Ping ネットワークパーティション解決リソース  
各サーバから通信可能で、常時起動している外部ネットワーク機器を対象として登録しておき、相手からのハートビート途絶時にそのネットワーク機器への通信ができない場合、強制的に自サーバをシャットダウンする機構。  
本機構により、全ハートビート経路断線によりスプリットブレインに陥った場合でも、登録した対象ネットワーク機器と通信できない側が自発的にシャットダウンするため、業務二重活性とその延長で発生し得るデータ破壊を防止します。

## 2.5.5. グループリソース

CLUSTERPRO では、切替対象の単位として各種グル-

プリソースを登録可能です。

本検証では、次のグループプリソースを登録しました。

- ディスクリソース  
稼働系側で共有ディスク上のデバイスをマウントするための機構を提供します。  
また本検証では、このリソースにより管理されるディスク領域を、DB2 のデータベース保管用に使用しました。
- Exec リソース  
DB2, Oracle など業務サービスを起動/停止するためのスクリプトを動作させる機構を提供します。  
なお、本リソースで起動/停止する業務サービスは、OS によって自動起動されない設定しておく必要があります。
- フローティング IP リソース  
仮想的な IP アドレスを提供。これによりクライアントからはどちらのサーバーが稼働系を意識せずに接続することが可能となります。
- RAW リソース  
稼働系側で共有ディスク上の RAW デバイスを使用可能とするための機構を提供します。  
また本検証では、このリソースにより管理されるディスク領域を、Oracle のデータベース保管用に使用しました。

## 2.5.6. モニタリソース

CLUSTERPRO では、監視対象に応じて各種モニタリソースを登録可能です。

本検証では、次のモニタリソースを登録しました。

- ARP モニタリソース  
フローティング IP リソース用の ARP パケットを送出する機構を提供します。
- DB2 モニタリソース  
DB2 データベースの監視機構を提供します。DB2 のインスタンスに対し、ダミーデータの読み書きを行う各種 SQL 文(create/drop/insert/update/select)を発行することで、データベースが正しく応答するか、サービス観点で監視を実施します。
- Oracle モニタリソース  
Oracle データベースの監視機構を提供します。Oracle のインスタンスに対し、ダミーデータの読み書きを行う各種 SQL 文(create/drop/insert/update/select)を発行することで、データベースが正しく応答するか、サービス観点で監視を実施します。
- ディスクモニタリソース  
ローカルディスクおよび共有ディスクの監視機構を提供します。
- IP モニタリソース  
外部ネットワーク機器への導通を監視する機構を提供します。
- RAW モニタリソース  
RAW デバイスの監視機構を提供します。
- ユーザ空間モニタリソース  
ユーザ空間のストール監視機構を提供します。

### 3. テスト項目および結果

#### 3.1. 正常状態チェック

以下の項目を実施し、正常状態においていずれの動作も問題ないことを確認しました。

##### ・WebManager 接続

項目	確認内容
WebManager 接続	WebManager(CLUSTERPRO の管理マネージャ)へ接続できることを確認

##### ・サーバー監視

項目	確認内容
サーバー	WebManager よりサーバーの状態が「正常動作中」となっていることを確認

##### ・ハートビート、ネットワークパーティション解決

項目	確認内容
LAN ハートビートリソース	WebManager より全ての LAN ハートビートリソースの状態が「正常動作中」となっていることを確認 (ハートビート専用 LAN、パブリック LAN とも)
ディスクハートビートリソース	WebManager よりディスクハートビートリソースの状態が「正常動作中」となっていることを確認
Ping ネットワークパーティション解決リソース	WebManager より Ping ネットワークパーティション解決リソースの状態が「正常動作中」となっていることを確認

##### ・グループ、グループリソース

項目	確認内容
グループ	WebManager より全てのグループの状態が「正常動作中」となっていることを確認
ディスクリソース	WebManager よりディスクリソースの状態が「正常動作中」となっていること、および対象マウントポイント先にマウントされていることを確認
EXEC リソース	WebManager より EXEC リソースの状態が「正常動作中」となっていること、およびリソースの起動/停止時に、それぞれ開始スクリプト/停止スクリプトが実行されることを確認
フローティング IP リソース	WebManager よりフローティング IP リソースの状態が「正常動作中」となっていること、およびクライアントからフローティング IP アドレスを使用して活性側サーバーと通信可能なことを確認
RAW リソース	WebManager より全ての RAW リソースの状態が「正常動作中」となっていることを確認

##### ・モニタリソース

項目	確認内容
ディスクモニタリソース	WebManager よりディスクモニタリソースの状態が「正常動作中」となっており、監視異常を検出しないことを確認
IP モニタリソース	WebManager より IP モニタリソースの状態が「正常動作中」となっており、監視異常を検出しないことを確認
NIC Link Up/Down モニタリソース	WebManager より NIC Link Up/Down モニタリソースの状態が「正常動作中」となっており、監視異常を検出しないことを確認
PID モニタリソース	WebManager より PID モニタリソースの状態が「正常動作中」となっており、監視異常を検出しないことを確認
RAW モニタリソース	WebManager より RAW モニタリソースの状態が「正常動作中」となっており、監視異常を検出しないことを確認
ユーザ空間モニタリソース	WebManager よりユーザ空間モニタリソースの状態が「正常動作中」となっており、監視異常を検出しないことを確認
ARP モニタリソース	WebManager より ARP モニタリソースの状態が「正常動作中」となっており、監視異常を検出しないことを確認
DB2 モニタリソース	WebManager より DB2 モニタリソースの状態が「正常動作中」となっており、監視異常を検出しないことを確認
Oracle モニタリソース	WebManager より Oracle モニタリソースの状態が「正常動作中」となっており、監視異常を検出しないことを確認

### 3.2. 異常状態チェック

ハードウェアや OS 障害を想定して以下のコンポーネントに対し擬似障害を発生させ、いずれの動作も問題ないことを確認しました。

#### ・共有ディスク装置

項目	操作	確認内容
共有ディスク装置 SCSI/FCパス	サーバー側 FC ケーブル を抜線	<ul style="list-style-type: none"> <li>・ディスクモニタリソースが異常検出すること</li> <li>・ディスクハートビートリソースが停止すること</li> <li>・グループが待機系にフェイルオーバーすること</li> </ul>

#### ・ネットワーク経路

インタコネク ト LAN	インタコネク ト LAN を抜 線	<ul style="list-style-type: none"> <li>・インタコネク ト LAN に対応するハートビートリソースが非活性になること</li> <li>・グループのフェイルオーバーは発生せず元のサーバーで継続動作すること(ハートビートはパブリック LAN をバックアップ用ハートビート経路として使用し継続)</li> </ul>
パブリック LAN	パブリック LAN を抜線	<ul style="list-style-type: none"> <li>・パブリック LAN に対応するハートビートリソースが非活性になること</li> <li>・IP モニタリソースが異常を検出しグループが待機系へフェイルオーバーすること</li> </ul>

#### ・VIOS

VIOS	同一筐体内に 存在する 2 つ の VIOS のう ち片方を停止	<ul style="list-style-type: none"> <li>・もう一方の VIOS により I/O は正常に動作継続すること</li> <li>・CLUSTERPRO は異常を検出することなくグループが稼働系で活性し続けること</li> </ul>
------	---	---

#### ・グループリソース、モニタリソース

項目	操作	確認内容
ディスクリ ソース	ディスクをマ ウントした状 態でグループ を起動する	・ディスクリソース活性 に失敗し、待機系へフ ェイルオーバーするこ と
EXEC リソー ス	「exit 1」で終 了するスクリ プトに設定 し、グループ を起動する	・EXEC リソース活性に 失敗し、待機系へフェ イルオーバーすること
フローティン グ IP リソー ス	ネットワー ク内の他サー バーで同一の アドレスを付 加した状態で グループを起 動する	・フローティング IP リ ソース活性に失敗し、 待機系へフェイルオ ーバすること
RAW リソー ス	存在しない RAW デバイ ス名を指定し てグループを 起動する	・RAW リソース活性に 失敗し、待機系へフェ イルオーバーすること
PID モニタ リソース	EXEC リソー スの常駐プロ セスを kill コ マンドで外部 から強制終了	・PID 監視異常によりグ ループが待機系へフ ェイルオーバーするこ と

### 3.3. フェイルオーバーグループ遷移チェック

以下の状態遷移を実施し、各サーバーのステータスおよびグループの状態が正しく遷移することを確認しました。

シーケンス	操作	確認内容
1	サーバー1、サーバー2を起動	サーバー1、サーバー2が「起動済」状態になること グループがサーバー1で起動すること
2	グループをサーバー2へ手動でフェイルオーバー	グループがサーバー2へフェイルオーバーすること
3	グループをサーバー1へ手動でフェイルオーバー	グループがサーバー1へフェイルオーバーすること
4	サーバー1をコマンドからシャットダウン	グループがサーバー2へフェイルオーバーすること
5	WebManager からクラスタシャットダウン操作	グループが停止しサーバー2がシャットダウンすること
6	サーバー1、サーバー2を起動	サーバー1、サーバー2が「起動済」状態になること グループがサーバー1で起動すること
7	サーバー1をコマンドからシャットダウン	サーバー1が「停止」状態になること グループがサーバー2へフェイルオーバーすること
8	サーバー2をコマンドからシャットダウン	グループが停止状態となること
9	サーバー1、サーバー2を起動	サーバー1、サーバー2が「起動済」状態になること グループがサーバー1で起動すること
10	WebManager からクラスタシャットダウン操作	グループが停止し、サーバー1、サーバー2がシャットダウンすること
11	サーバー2を起動 [相手サーバー起動待ち(5分)経過後]	サーバー2が「起動済」状態になること グループがサーバー2で起動すること
12	サーバー1を起動	サーバー1が「起動済」状態になること
13	サーバー2をコマンドからシャットダウン	サーバー2が「停止」状態になること グループがサーバー1へフェイルオーバーすること
14	サーバー2を起動	サーバー2が「起動済」状態になること

15	WebManager からクラスタシャットダウン	グループが停止し、サーバー1、サーバー2がシャットダウンすること
16	サーバー1を起動 [相手サーバー起動待ち(5分)経過後]	サーバー1が「起動済」状態になること グループがサーバー1で起動すること
17	サーバー2を起動	サーバー2が「起動済」状態になること
18	サーバー1をコマンドからシャットダウン	サーバー1が「停止」状態になること グループがサーバー2へフェイルオーバーすること
19	サーバー1を起動	サーバー1が「起動済」状態になること
20	WebManager からクラスタシャットダウン	グループが停止し、サーバー1、サーバー2がシャットダウンすること
21	サーバー1、サーバー2を起動	サーバー1、サーバー2が「起動済」状態になること グループがサーバー1で起動すること

### 3.4. データベース動作チェック

DB2、Oracle それぞれのクラスターセットに対し以下の項目を実施し、いずれの動作も問題ないことを確認しました。(本章では、DB2 Agent、Oracle Agent を総称して“Database Agent”と記します)

項目	操作	確認内容
起動	WebManager から手動でグループ起動	<ul style="list-style-type: none"> <li>データベースが起動し、グループが起動状態となること</li> <li>クライアント端末からデータベースに接続しトランザクションを正常に処理できること</li> </ul>
正常系動作	同上	<ul style="list-style-type: none"> <li>Database Agent が監視開始状態となり、異常を検出しないこと</li> </ul>
停止	WebManager から手動でグループ停止	<ul style="list-style-type: none"> <li>データベースが停止し、グループが停止状態となること</li> </ul>
DB プロセス異常	DB のプロセスを kill コマンドで外部から強制終了	<ul style="list-style-type: none"> <li>Database Agent が異常を検出すること</li> <li>グループが待機系にフェイルオーバーされ、クライアントからトランザクションを継続して処理できること</li> </ul>
サーバー異常	稼動系サーバーで外部コマンドにより OS シャットダウン	<ul style="list-style-type: none"> <li>Database Agent が異常を検出すること</li> <li>グループが待機系にフェイルオーバーされ、クライアントからトランザクションを継続して処理できること</li> </ul>

### 3.5. テスト結果まとめ

上記のとおり、Power Systems 上において DB2、Oracle をクラスター化し、クライアントからのトランザクションが問題なく継続可能なことを確認しました。

また、ハードウェア、OS、上位アプリケーション(DB2、Oracle)それぞれの観点で擬似障害を発生させた結果、いずれも想定どおりフェイルオーバーが発生し、クライアントへのトランザクションを継続して提供することを確認しました。

## 4. おわりに

本稿では、IBM Power Systems 環境において、CLUSTERPRO の導入による DB2 および Oracle の二重化構成について検証を実施しました。

その結果、正常系および異常系ともに想定どおり動作し、ハードウェア、OS、アプリケーションの各種コンポーネントのいずれかで障害が発生した場合でもフェイルオーバーを実施す

ることにより、サービスの可用性が向上することを確認できました。

また、本構成の導入により、他のクラスタリングソフトとの組み合わせでは実現困難な次のメリットが受けられることを実証しました。

- VIOS 構成で外部ディスク制御機能を備えたクラスター化が可能
- Database Agent の導入により、データベースの動作を単純なプロセス生存監視に留まらず、サービスレベルで監視することが可能

## 5. 参考URL

- CLUSTERPRO  
<http://www.nec.co.jp/clusterpro/>
- IBM Power Systems  
<http://www.ibm.com/systems/jp/power/>
- IBM PowerVM  
<http://www.ibm.com/systems/jp/power/software/virtualization/>
- IBM Linux for Power Systems  
<http://www.ibm.com/systems/jp/power/software/linux/>
- IBM DB2  
<http://www.ibm.com/software/jp/data/db2/>

日本アイ・ビー・エム株式会社  
パワーシステム事業部

日本電気株式会社  
第二コンピュータソフトウェア事業部  
CLUSTERPRO グループ  
[info@clusterpro.jp.nec.com](mailto:info@clusterpro.jp.nec.com)

商標情報

IBM、Power Systems、PowerVM、DB2 は International Business Machines Corporation の米国およびその他の国における商標です。

CLUSTERPRO は日本電気株式会社の登録商標です。

Linux は Linus Torvalds 氏の米国およびその他の国における登録商標または商標です。

その他、文中の社名、商品名は、各社の商標または登録商標です。

## 付録

### CLUSTERPRO設定情報

#### ・ DB2 起動スクリプト(start.sh)

---

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

ulimit -s unlimited

if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"

# "-t"で指定する値は、インスタンス開始処理を待ち合わせる
#タイムアウト時間です。開始完了までの最大所要時間を上回る
#秒数を指定してください。
        /opt/ibm/db2/V9.7/bin/db2gcf -i db2inst1
-u -t 600
        if [ $? -ne 0 ]; then
            exit 1
        fi

        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
    else
        echo "ERROR_DISK from START"
    fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "FAILOVER1"

# "-t"で指定する値は、インスタンス開始処理を待ち合わせる
#タイムアウト時間です。開始完了までの最大所要時間を上回る
#秒数を指定してください。
        /opt/ibm/db2/V9.7/bin/db2gcf -i db2inst1
-u -t 600
        if [ $? -ne 0 ]; then
            exit 1
        fi

        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "FAILOVER2"
        else
            echo "ON_OTHER2"
        fi
    else
```

---

---

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

---

#### ・ DB2 停止スクリプト(stop.sh)

---

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

ulimit -s unlimited

if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"

# "-t"で指定する値は、インスタンス停止処理を待ち合わせる
#タイムアウト時間です。停止完了までの最大所要時間を上回る
#秒数を指定してください。
        /opt/ibm/db2/V9.7/bin/db2gcf -i db2inst1
-d -t 600
        if [ $? -ne 0 ]; then
            /opt/ibm/db2/V9.7/bin/db2gcf -i
db2inst1 -k -t 600
            exit 1
        fi

        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
    else
        echo "ERROR_DISK from START"
    fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "FAILOVER1"

# "-t"で指定する値は、インスタンス停止処理を待ち合わせる
#タイムアウト時間です。停止完了までの最大所要時間を上回る
#秒数を指定してください。
        /opt/ibm/db2/V9.7/bin/db2gcf -i db2inst1
-d -t 600
        if [ $? -ne 0 ]; then
            /opt/ibm/db2/V9.7/bin/db2gcf -i
db2inst1 -k -t 600
```

---

---

```

        exit 1
    fi

    if [ "$CLP_SERVER" = "HOME" ]
    then
        echo "FAILOVER2"
    else
        echo "ON_OTHER2"
    fi

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

```

---

• Oracle 起動スクリプト(start.sh)

---

```

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

ulimit -s unlimited

if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"

        su -l oracle -c 'lsnrctl start LISTENER'
        su -l oracle -c 'export
ORACLE_SID=clprodb;sqlplus /nolog
@/root/ora_start.sh'

        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
    else
        echo "ERROR_DISK from START"
    fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "FAILOVER1"

        su -l oracle -c 'lsnrctl start LISTENER'
        su -l oracle -c 'export
ORACLE_SID=clprodb;sqlplus /nolog
@/root/ora_start.sh'

```

---

```

        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "FAILOVER2"
        else
            echo "ON_OTHER2"
        fi
    else
        echo "ERROR_DISK from FAILOVER"
    fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0

```

---

• Oracle 停止スクリプト(stop.sh)

---

```

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

ulimit -s unlimited

if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"

        su -l oracle -c 'export
ORACLE_SID=clprodb;sqlplus /nolog
@/root/ora_stdn.sh'
        su -l oracle -c 'lsnrctl stop LISTENER'

        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
    else
        echo "ERROR_DISK from START"
    fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "FAILOVER1"

        su -l oracle -c 'export
ORACLE_SID=clprodb;sqlplus /nolog
@/root/ora_stdn.sh'
        su -l oracle -c 'lsnrctl stop LISTENER'

        if [ "$CLP_SERVER" = "HOME" ]

```

---

---

```
    then
        echo "FAILOVER2"
    else
        echo "ON_OTHER2"
    fi

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

---