

- OSS による統合 DB を介した基幹システムと業務システム連携の実証 -

「運用手順書」

Ver 1.0 (2008/02/25)

(株)野村総合研究所
オープンソースソリューションセンター

変更履歴

版	日付	担当者	変更内容	備考
0.1	2007/12/25	NRI	ドラフト版作成	
0.2	2008/02/11	NRI	<p>環境構築手順書との整合性をとる為、下記の修正を行いました。</p> <p>2 本書の対象範囲</p> <ul style="list-style-type: none"> ・全サーバに UPS デバイス管理ミドルウェア「APCUPSD」を追加 <p>3 ハードウェア構成</p> <ul style="list-style-type: none"> ・HP 製「HP Proliant DL360 G5」のスペックに修正 <p>4.1 ソフトウェア構成図</p> <ul style="list-style-type: none"> ・全サーバに UPS デバイス管理ミドルウェア「APCUPSD」を追加 <p>4.2 ソフトウェア一覧</p> <ul style="list-style-type: none"> ・全サーバに UPS デバイス管理ミドルウェア「APCUPSD」を追加 <p>全体</p> <ul style="list-style-type: none"> ・全サーバの仮ホスト名、IP アドレスを実際のホスト名、IP アドレスに修正 	
1.0	2008/02/25	BSN アイネット	初版作成	

目次

1. はじめに	5
2. 本書の対象範囲	6
3. ハードウェア構成.....	7
4. ソフトウェア構成.....	8
4.1. ソフトウェア構成図.....	8
4.2. ソフトウェア一覧.....	9
5. 運用手順	10
5.1. ロードバランサ(Heartbeat/httpd).....	10
5.1.1. 手動フェイルオーバー手順.....	10
5.1.2. 障害復旧手順.....	12
5.2. DB サーバ(Heartbeat/pgpool・PostgreSQL).....	18
5.2.1. 手動フェイルオーバー手順.....	18
5.2.2. 障害復旧手順.....	23
6. 補足資料	42
6.1. Heartbeat 主要コマンドリスト.....	42
6.1.1. Heartbeat 起動.....	42
6.1.2. Heartbeat 停止.....	42
6.1.3. Heartbeat 稼動状況確認.....	43
6.1.4. Heartbeat ノード Active/Standby 切り替え.....	44
6.1.5. Heartbeat リソース制御.....	45
6.2. pgpool 主要コマンドリスト.....	46
6.2.1. pgpool 起動.....	46
6.2.2. pgpool 停止.....	47
6.2.3. pgpool 稼動状態確認.....	48
6.2.4. pgpool スイッチオーバー.....	50

6.3. PostgreSQL 主要コマンドリスト.....	5 1
6.3.1. PostgreSQL 起動.....	5 1
6.3.2. PostgreSQL 停止.....	5 2
6.3.3. 全データベースダンプ.....	5 3
6.3.4. 不要領域削除.....	5 4

1. はじめに

本書ではオープンソースソフトウェアによる統合データベースを介した基幹システムと業務システム間の連携についての実証実験における基盤運用手順について示す。

手順内で示している各コマンドの詳細に関しては、6補足資料にて説明を行っているので、不明なオプション等があれば、適宜参照のこと。

2. 本書の対象範囲

本書の対象範囲について図 2-1 に示す。

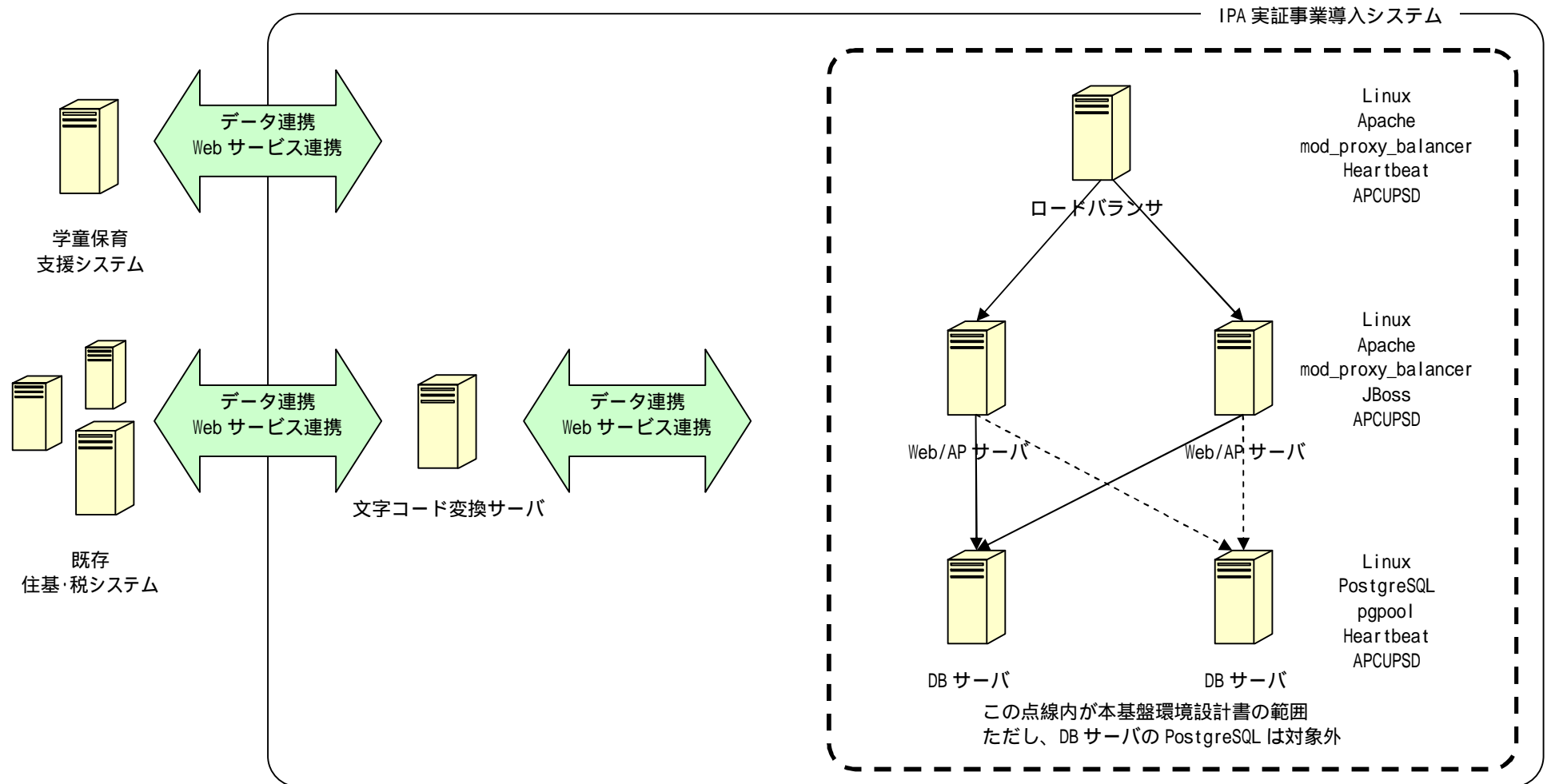


図 2-1 対象範囲

3. ハードウェア構成

ハードウェア一覧を表 3-1 に示す。

表 3-1 ハードウェア一覧

#	サーバ名称	スペック	数量	製品名
1.	ロードバランサ	CPU : Dual Core Xeon 5160 3GHz メモリ: 2GB Disk : 73GB SAS × 2(RAID1)	1	HP ProLiant DL360 G5
2.	Web/AP サーバ	CPU : Dual Core Xeon 5160 3GHz × 2 メモリ: 2GB Disk : 73GB SAS × 2(RAID1)	2	HP ProLiant DL360 G5
3.	DB サーバ	CPU : Dual Core Xeon 5050 3GHz メモリ: 4GB Disk : 73GB SAS × 2(RAID1)	2	HP ProLiant DL360 G5

4. ソフトウェア構成

4.1. ソフトウェア構成図

ホンシステムにおけるソフトウェア構成図を以下の図 4-1 に示す。

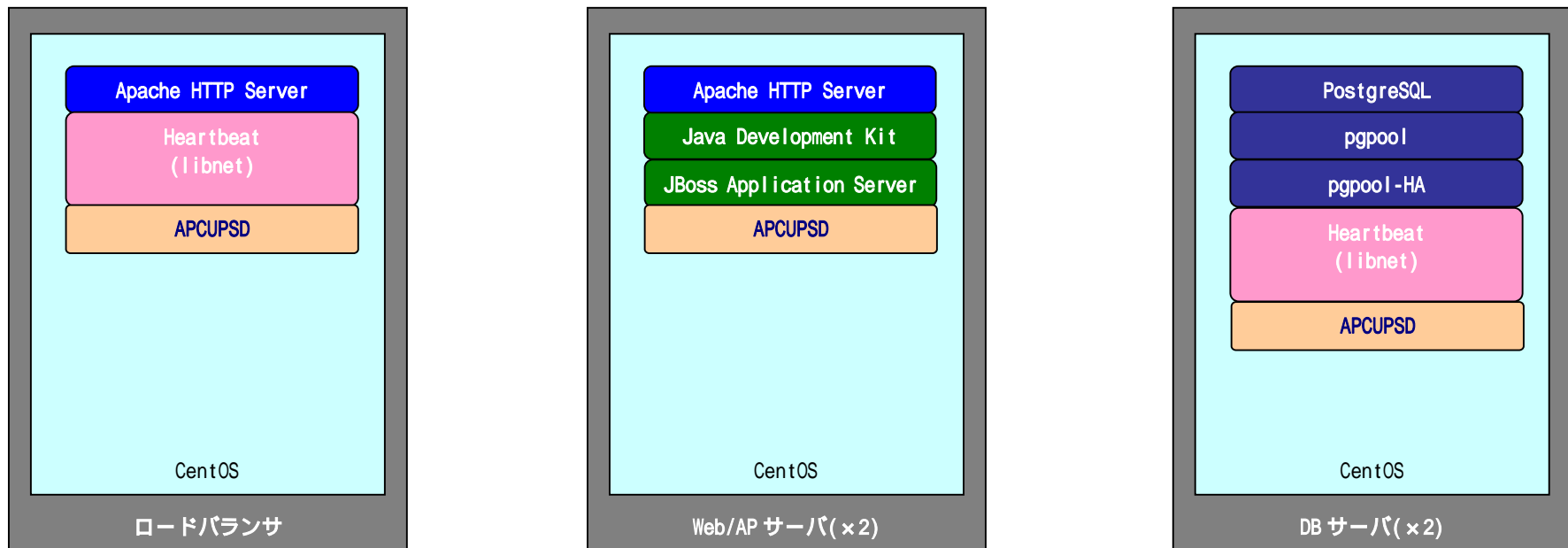


図 4-1 ソフトウェア構成図

4.2. ソフトウェア一覧

本システムにおけるソフトウェアおよび、そのバージョンの一覧を以下の表 4-1 に示す。

表 4-1 ソフトウェア一覧

No	ミドルウェア名	バージョン	内容	備考
1.	CentOS	4.5	コミュニティ版 OS	RHEL 4 Update 5 に対応
2.	Apache HTTP Server	2.2.6	http サーバ	
3.	HeartBeat (libnet)	2.1.2 (1.1.2.1)	冗長性・可用性向上ミドルウェア	
4.	Java Development Kit	1.5.0_12	Java VM	
5.	JBoss Application Server	4.2.1.GA	Web アプリケーションサーバ	
6.	PostgreSQL	8.2.5	RDBMS	
7.	pgpool	3.4.1	PostgreSQL の冗長化・可用性向上ミドルウェア	
8.	pgpool-HA	1.1.0	pgpool と Heartbeat の連携ミドルウェア	
9.	APCUPSD	3.14.2-1.e14	UPS 制御ミドルウェア	

5. 運用手順

5.1. ロードバランサ(Heartbeat/httpd)

5.1.1. 手動フェイルオーバー手順

5.1.1.1. 実施条件

手動フェイルオーバーパターン

- ✓ ロードバランサ(Heartbeat/httpd)を手動でフェイルオーバーさせるパターンとしては、以下の2つが考えられる。
 - システム運用中の片系ロードバランサのリポート・シャットダウンを行う場合()
 - システム運用中の片系ロードバランサが動作不安定時のリソース移動を行う場合()

現在のロードバランサは、1台のHeartbeat構成である為、手動フェイルオーバーは実施できない。以下は、将来的にロードバランサをスケールアウトした際の手順を示している。

5.1.1.2. 実施手順

- ✓ 実施条件により手順が若干異なる。以下のに従い、対応するフェイルオーバーおよび、復旧手順を実行する。

表 5-1 ロードバランサ(Heartbeat/httpd)の手動フェイルオーバー実施手順

実施条件	フェイルオーバー手順		復旧手順
	-(ア)	-(イ)	-(ア)
	Heartbeat フェイルオーバー	リポート/ シャットダウン	Heartbeat フェイルバック
リポート・シャットダウン			
動作不安定時のリソース移動			

は、Heartbeat のフェイルオーバーだけでは、現象が解消しない場合に実施する。

フェイルオーバー手順

(ア) Heartbeat フェイルオーバー

- ✓ 現在リソースが稼働しているロードバランサ(#1)を、crm_standby コマンドよりスタンバイ状態に設定する。

```
# crm_standby U LB01 v true
```

- ✓ crm_mon コマンドより、ロードバランサ(#1)の Node が「standby」状態になり、ロードバランサ (#2)側に httpd リソースが、移動したことを確認する。

```
# crm_mon -i 3  
Refresh in 3s...
```

```

=====
Last updated: Tue Dec 18 09:20:22 2007
Current DC: lb02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf)
2 Nodes configured.
1 Resources configured.
=====

Node: lb02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf): online
Node: lb01 (80389bde-c07a-454c-8735-98b0e3ede432): standby

Resource Group: group_01
  resource_ip (heartbeat::ocf:IPAddr):      Started lb02
  resource_ap (heartbeat::ocf:apache):      Started lb02
  resource_ml (heartbeat::ocf:MailTo):      Started lb02

```

(イ) リブート/シャットダウン

- ✓ メンテナンス対象のロードバランサをリブート、またはシャットダウンする。

```
# shutdown r now
```

or

```
# shutdown h now
```

復旧手順

(ア) Heartbeat フェイルバック

- ✓ ロードバランサの Heartbeat は自動フェイルバック設定を行っている為、ロードバランサ(#1)を「online」状態に戻すことで、ロードバランサ(#2)から自動的にフェイルバックが行われる。
- ✓ crm_standby コマンドより、ロードバランサ (#1)側を「online」状態に戻す。

```
# crm_standby -U lb01 -v false
```

```
crm_standby[10113]: 2007/12/18_09:23:42 info: Invoked: crm_standby -U lb01 -v false
```

この後、crm_mon コマンドより、ロードバランサ(#1)に正しくフェイルバックが行われていることを確認する。

5.1.2. 障害復旧手順

5.1.2.1. 実施条件

障害パターン

- ✓ Heartbeat(httpd)の障害のパターンとしては、以下の4つが考えられる。
 - マシン電源障害()
 - ネットワーク障害()
 - Heartbeat プロセスダウン()
 - httpd 起動失敗()

現在のロードバランサの Heartbeat は、1 台の構成である為、いずれの障害が発生してもシングルポイントフェイルが発生する。対応できる障害としては、唯一、仮想 IP アドレスの停止や、httpd プロセスが停止した場合に Heartbeat によって再始動が行われる。再始動が失敗した場合は、同様にシングルポイントフェイルが発生する。以下は、将来的にロードバランサをスケールアウトした際の手順を示している。

障害状況確認(マシン電源障害、ネットワーク障害、Heartbeat プロセスダウン)

- ✓ Heartbeat の現在の動作状況を crm_mon コマンドより確認する。

```
# crm_mon i 3
```

- ✓ 「Node」の片方が、「OFFLINE」となっている場合、マシンの電源断、ネットワーク遮断、Heartbeat プロセスダウンのいずれかが原因として考えられる。

```
Refresh in 2s...

=====
Last updated: Mon Dec 17 18:58:06 2007
Current DC: lb02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf)
2 Nodes configured.
1 Resources configured.
=====

Node: lb02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf): online
Node: lb01 (80389bde-c07a-454c-8735-98b0e3ede432): OFFLINE

Resource Group: group_01
  resource_ip (heartbeat::ocf:IPAddr):      Started lb02
  resource_ap (heartbeat::ocf:apache):      Started lb02
  resource_ml (heartbeat::ocf:MailTo):      Started lb02
```

(ア) マシン電源確認

- ✓ マシンの電源が切れていないことを確認する。

(イ) ネットワーク接続確認

- ✓ ping コマンドの応答があることを確認する。疎通している場合は、下記のようなメッセージが表示される。

```
# ping lb01
PING LB01 (192.168.100.11) 56(84) bytes of data.
64 bytes from LB01 (192.168.100.11): icmp_seq=0 ttl=64 time=0.816 ms
64 bytes from LB01 (192.168.100.11): icmp_seq=1 ttl=64 time=0.187 ms
64 bytes from LB01 (192.168.100.11): icmp_seq=2 ttl=64 time=0.432 ms
64 bytes from LB01 (192.168.100.11): icmp_seq=3 ttl=64 time=0.782 ms
. . .
```

(ウ) Heartbeat プロセス確認

- ✓ 「OFFLINE」側の DB サーバにログインを行い、service コマンドより、heartbeat プロセスの起動を確認する。起動している場合は、下記のようなメッセージが表示される。

```
# service heartbeat status
heartbeat OK [pid 11456 et al] is running on lb01 [lb01]...
```

障害状況確認(httpd 起動確認)

- ✓ Heartbeat の現在の動作状況を crm_mon コマンドより確認する。

```
# crm_mon i 3
```

- ✓ 「Failed actions」に「resource_ap_start_0」が表示され、httpd の稼働のホストが、「lb02」になっている場合、httpd の起動が失敗したと考えられる。

```
Refresh in 3s...

=====
Last updated: Mon Dec 17 21:30:23 2007
Current DC: lb02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf)
2 Nodes configured.
1 Resources configured.
=====
```

Node: lb02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf): online
Node: lb01 (80389bde-c07a-454c-8735-98b0e3ede432): online

Resource Group: group_01

resource_ip (heartbeat::ocf:IPAddr):	Started lb02
resource_ap (heartbeat::ocf:pgpool):	Started lb02
resource_ml (heartbeat::ocf:MailTo):	Started lb02

Failed actions:

resource_ap_start_0 (node=lb01, call=15, rc=-1): Timed Out

5.1.2.2. 実施手順

障害復旧手順

- ✓ 実施条件により手順が若干異なる。以下の表 5-2 に従い、対応する復旧手順を実行する。

表 5-2 ロードバランサ障害(Heartbeat/httpd)の復旧手順

実施条件	復旧手順				
	(ア)	(イ)	(ウ)	(エ)	(オ)
	電源復旧	ネットワーク復旧	Heartbeat復旧	httpd復旧	Heartbeatフェイルバック
マシン電源障害		-	-	-	自動
ネットワーク障害	-		-	-	自動
Heartbeat プロセスダウン	-	-		-	自動
httpd 起動失敗	-	-	-		自動

フェイルバックは、自動で行われる。

(ア) 電源復旧

- ✓ マシンの電源を On にし、OS の起動を確認する。
- ✓ 電源が On にできない場合や、OS が起動しない場合は、HW トラブルの可能性があるので、システム管理者に連絡を行う。

(イ) ネットワーク復旧

- ✓ LAN ケーブルの断線有無、ネットワーク中継機器の正常確認、ファイアウォールの設定を確認し、ロードバランサ (#1)と、ロードバランサ(#2)が相互に ping 通信が可能であることを確認する。
- ✓ 上記を確認した上で、ネットワークが復旧しない場合は、HW トラブル、NW トラブルの可能性があるので、システム管理者に連絡を行う。

(ウ) Heartbeat 復旧

- ✓ 通常、Heartbeat は OS 起動時に自動起動する設定になっているが、なにかしらの原因により、プロセスがダウンしてしまった場合は、service コマンドより、Heartbeat の手動起動を行う。

```
# service heartbeat start
Starting High-Availability services:
[ OK ]
```

- ✓ コマンド上は、「OK」と即座に表示されてしまうので、実際の起動確認は、crm_mon コマンドにて、停止していた側の Node が「online」になることを確認する。

```
# crm_mon -i 3
Refresh in 2s...

=====
Last updated: Tue Dec 18 08:58:14 2007
Current DC: lb02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf)
2 Nodes configured.
1 Resources configured.
=====

Node: lb02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf): online
Node: lb01 (80389bde-c07a-454c-8735-98b0e3ede432): online

Resource Group: group_01
  resource_ip (heartbeat::ocf:IPaddr):      Started lb02
  resource_ap (heartbeat::ocf:apache):      Started lb02
  resource_ml (heartbeat::ocf:MailTo):      Started lb02
```

起動までに、若干時間がかかることがある。

- ✓ 上記手順を経ても、Heartbeat が起動しない場合は、Heartbeat ログ(/opt/Openstandia/heartbeat/logs/ha-log)を確認し、ERROR や FATAL レベルのログが表示されていないかどうか確認し、原因が分かれば対処を行う。

```
# egrep "(ERROR|FATAL)" /opt/Openstandia/heartbeat/logs/ha-log
```

- ✓ Heartbeat ログに特に問題がない場合は、HWトラブルもしくは、OSトラブルの可能性があるので、システム管理者に連絡を行う。

(エ) httpd 復旧

- ✓ httpd が起動失敗したホスト側で、以下の apachectl コマンドを実行し、起動可能有無をチェックする。

```
# apachectl start
Syntax error on line 423 of /opt/Openstandia/apache2.2/conf/httpd.conf:
Invalid command 'hoge', perhaps misspelled or defined by a module not included in the server configuration
```

- ✓ 上記のように起動時に時に、「Syntax error」の文言が表示される場合は、httpd が起動できない状態であることが分かる。
- ✓ エラーログに表示されているパラメータを修正し、再度、apachectl コマンドを実行し、正常起動できることを確認する。
- ✓ 上記手順を経ても、httpd が起動しない場合は、HWトラブルもしくは、OSトラブルの可能性があるので、システム管理者に連絡を行う。

- ✓ 正常に起動できる状態になった場合は httpd を停止し、crm_resource コマンドより、エラー状態となったリソースのクリーンアップを行う。

```
# apachectl stop
# crm_resource -H lb01 -r resource_ap -C
crm_resource[9301]: 2007/12/18_09:32:43 info: Invoked: crm_resource -H lb01 -r resource_ap C
```

- ✓ crm_mon コマンドにて、「Failed actions」がクリアされていることを確認する。

```
# crm_mon -i 3
Refresh in 2s...

=====
Last updated: Tue Dec 18 09:33:10 2007
Current DC: lb02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf)
2 Nodes configured.
1 Resources configured.
=====

Node: lb02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf): online
Node: lb01 (80389bde-c07a-454c-8735-98b0e3ede432): online

Resource Group: group_01
  resource_ip (heartbeat::ocf:IPAddr):      Started lb01
  resource_ap (heartbeat::ocf:apache):      Started lb01
  resource_ml (heartbeat::ocf:MailTo):      Started lb01
```

httpd の場合、リソースのエラー状態をクリアした時点で、自動フェイルバックが実行されるので、httpd は、ロードバランサ(#1)に移動する。

(オ) Heartbeat フェイルバック

- ✓ リソースのクリーンアップ後は、自動でフェイルバックが行われる為、手順としては不要。

5.2. DB サーバ(Heartbeat/pgpool・PostgreSQL)

5.2.1. 手動フェイルオーバー手順

5.2.1.1. 実施条件

手動フェイルオーバーパターン

- ✓ Heartbeat(pgpool)・PostgreSQL を手動でフェイルオーバーさせるパターンとしては、以下の4つが考えられる。
 - システム運用中の片系 DB サーバのリポート・シャットダウンを行う場合
 - システム運用中の片系 DB サーバが高負荷/不安定な時のノード切り離しを行う場合
 - システム運用中の片系 PostgreSQL の再始動/停止を行う場合
 - システム運用中の片系 PostgreSQL が高負荷/不安定な時のノード切り離しを行う場合

5.2.1.2. 実施手順

- ✓ 実施条件により手順が若干異なる。以下の表 5-3 に従い、対応するフェイルオーバーおよび、復旧手順を実行する。

表 5-3 DB サーバ(Heartbeat/pgpool)の手動フェイルオーバー実施手順

実施条件	フェイルオーバー手順				復旧手順	
	-(ア)	-(イ)	-(ウ)	-(エ)	-(ア)	-(イ)
	Heartbeat フェイルオーバー	pgpool スイッチオーバー	リポート/ シャットダウン	PostgreSQL 再始動/ 停止	データ 同期	Heartbeat フェイルバック
リポート・シャットダウン				-		
高負荷/不安定時のノード切り離し				-		
PostgreSQL の再始動/停止	-		-			-
PostgreSQL の高負荷/不安定時のノード切り離し	-		-			-

のフェイルバックに関しては、必須な手順ではない。
Heartbeat のフェイルオーバーで、現象が解消しない場合は実施する。

フェイルオーバー手順

(ア) Heartbeat フェイルオーバー

- ✓ 現在リソースが稼働している DB サーバ(#1)を、crm_standby コマンドより、スタンバイ状態に設定する。

```
# crm_standby U DB01 v true
```

- ✓ crm_mon コマンドより、DB サーバ(#1)の Node が「standby」状態になり、DB サーバ(#2)側に pgpool リソースが、移動したことを確認する。

```
# crm_mon -i 3
Refresh in 3s...

=====
Last updated: Tue Dec 18 09:20:22 2007
Current DC: db02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf)
2 Nodes configured.
1 Resources configured.
=====

Node: db02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf): online
Node: db01 (80389bde-c07a-454c-8735-98b0e3ede432): standby

Resource Group: group_01
  resource_ip (heartbeat::ocf:IPAddr):      Started db02
  resource_pg (heartbeat::ocf:pgpool):     Started db02
  resource_ml (heartbeat::ocf:MailTo):     Started db02
```

- ✓ crm_standby コマンドより、DB サーバ(#1)側を「online」状態に戻す。

```
# crm_standby -U db01 -v false
crm_standby[10113]: 2007/12/18_09:23:42 info: Invoked: crm_standby -U db01 -v false
```

DB サーバの Heartbeat は自動フェイルバック設定を行っていない為、DB サーバ(#1)を「online」に戻しても、リソースは移動しない。

(イ) pgpool スイッチオーバー

- ✓ 片側の DB サーバのリブートおよび、シャットダウンを行う際には、PostgreSQL とのレプリケーションが停止する為、事前に「pgpool」コマンドより、スイッチオーバー(pgpool の片側縮退運転)を実行する。

```
# pgpool s master switch
```

このコマンドの例では、マスタ側のレプリケーションを停止している。「-s」で指定した側を停止するという動作を行うので注意が必要。

このコマンドで指定するマスタ、センドリを混乱しないためにも、pgpool の設定ファイルで指定するバックエンドサーバは、双方で同一の設定を行っておくのがよい。

(ウ) リブート/シャットダウン

- ✓ メンテナンス対象の DB サーバをリブート、またはシャットダウンする。

```
# shutdown r now
```

or

```
# shutdown h now
```

(エ) PostgreSQL の再始動 / 停止

- ✓ メンテナンス対象の PostgreSQL を再始動、または停止する。

```
# service postgres restart
```

or

```
# service postgres stop
```

復旧手順

(ア) データ同期

- ✓ pgpool のスイッチオーバーを実行した場合は、データ同期が必要となる。
- ✓ データ同期の詳細手順については、次節の「5.2.2.2(カ)データ同期」で説明を行うので、そちらを参考のこと。

(イ) Heartbeat フェイルバック

- ✓ pgpool の稼動が DB サーバ(#2)にフェイルオーバーされた後に、DB サーバ(#1)側に pgpool の戻す手順を以下に示す。

【注意事項】

フェイルバックは必須な手順ではありません。DB サーバ(#1)に処理能力の高いマシンを配置しているなどの理由があれば実施する。
フェイルバック中や直後は DB サーバ接続の瞬断が発生する為、定期メンテナンスや、アプリケーションが稼動していない時間帯に実施する。

- ✓ 双方の Heartbeat が正常稼動(Node が「online」)していることを、crm_mon コマンドより確認する。

```
# crm_mon -i 3
```

```
Refresh in 2s...
```

```
=====
```

```
Last updated: Tue Dec 18 08:58:14 2007
```

```
Current DC: db02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf)
```

```
2 Nodes configured.
```

```
1 Resources configured.
```

```

=====
Node: db02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf): online
Node: db01 (80389bde-c07a-454c-8735-98b0e3ede432): online

Resource Group: group_01
  resource_ip (heartbeat::ocf:IPAddr):      Started db02
  resource_pg (heartbeat::ocf:pgpool):      Started db02
  resource_ml (heartbeat::ocf:MailTo):      Started db02

```

- ✓ crm_standby コマンドより、pgpool が稼働中の DB サーバ(#2)側を強制的に待機状態に設定する。

```

# crm_standby -U db02 -v true
crm_standby[8941]: 2007/12/18_09:20:34 info: Invoked: crm_standby -U db02 -v true

```

- ✓ crm_standby コマンドより、DB サーバ(#2)の Node が「standby」状態になり、DB サーバ(#1)側に pgpool リソースが、移動したことを確認する。

```

# crm_mon -i 3
Refresh in 3s...

=====
Last updated: Tue Dec 18 09:20:22 2007
Current DC: db02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf)
2 Nodes configured.
1 Resources configured.
=====

Node: db02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf): standby
Node: db01 (80389bde-c07a-454c-8735-98b0e3ede432): online

Resource Group: group_01
  resource_ip (heartbeat::ocf:IPAddr):      Started db01
  resource_pg (heartbeat::ocf:pgpool):      Started db01
  resource_ml (heartbeat::ocf:MailTo):      Started db01

```

- ✓ crm_standby コマンドより、DB サーバ(#2)側を「online」状態に戻す。

```
# crm_standby -U db02 -v false
crm_standby[10113]: 2007/12/18_09:23:42 info: Invoked: crm_standby -U db02 -v false
```

✓ crm_standby コマンドより、DB サーバ(#2)の Node が「online」状態に戻ったことを確認する。

```
# crm_mon -i 3
Refresh in 3s...

=====
Last updated: Tue Dec 18 09:25:06 2007
Current DC: db02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf)
2 Nodes configured.
1 Resources configured.
=====

Node: db02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf): online
Node: db01 (80389bde-c07a-454c-8735-98b0e3ede432): online

Resource Group: group_01
  resource_ip (heartbeat::ocf:IPAddr):      Started db01
  resource_pg (heartbeat::ocf:pgpool):      Started db01
  resource_ml (heartbeat::ocf:MailTo):      Started db01
```

5.2.2. 障害復旧手順

5.2.2.1. 実施条件

障害パターン

- ✓ Heartbeat(pgpool)・PostgreSQL の障害のパターンとしては、以下の4つが考えられる。
 - マシン電源障害
 - ネットワーク障害
 - Heartbeat プロセスダウン
 - pgpool 起動失敗
 - PostgreSQL 障害
 - pgpool スイッチオーバー運用

障害状況確認(マシン電源障害、ネットワーク障害、Heartbeat プロセスダウン)

- ✓ Heartbeat の現在の動作状況を crm_mon コマンドより確認する。

```
# crm_mon i 3
```

- ✓ 「Node」の片方が、「OFFLINE」となっている場合、マシンの電源断、ネットワーク遮断、Heartbeat プロセスダウンのいずれかが原因として考えられる。

```
Refresh in 2s...

=====
Last updated: Mon Dec 17 18:58:06 2007
Current DC: db02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf)
2 Nodes configured.
1 Resources configured.
=====

Node: db02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf): online
Node: db01 (80389bde-c07a-454c-8735-98b0e3ede432): OFFLINE

Resource Group: group_01
  resource_ip (heartbeat::ocf:IPAddr):      Started db02
  resource_pg (heartbeat::ocf:pgpool):      Started db02
  resource_ml (heartbeat::ocf:MailTo):      Started db02
```

(ア) マシン電源確認

- ✓ マシンの電源が切れていないことを確認する。

(イ) ネットワーク接続確認

- ✓ ping コマンドの応答があることを確認する。疎通している場合は、下記のようなメッセージが表示される。

```
# ping db01
PING DB01 (192.168.100.31) 56(84) bytes of data.
64 bytes from DB01 (192.168.100.31): icmp_seq=0 ttl=64 time=0.816 ms
64 bytes from DB01 (192.168.100.31): icmp_seq=1 ttl=64 time=0.187 ms
64 bytes from DB01 (192.168.100.31): icmp_seq=2 ttl=64 time=0.432 ms
64 bytes from DB01 (192.168.100.31): icmp_seq=3 ttl=64 time=0.782 ms
. . .
```

(ウ) Heartbeat プロセス確認

- ✓ 「OFFLINE」側の DB サーバにログインを行い、service コマンドより、heartbeat プロセスの起動を確認する。起動している場合は、下記のようなメッセージが表示される。

```
# service heartbeat status
heartbeat OK [pid 11456 et al] is running on db01 [db01]...
```

障害状況確認(pgpool 起動確認)

- ✓ Heartbeat の現在の動作状況を crm_mon コマンドより確認する。

```
# crm_mon i 3
```

- ✓ 「Failed actions」に「resource_pg_start_0」が表示され、pgpool の稼働のホストが、「db02」になっている場合、pgpool の起動が失敗したと考えられる。

```
Refresh in 3s...

=====
Last updated: Mon Dec 17 21:30:23 2007
Current DC: db02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf)
2 Nodes configured.
1 Resources configured.
=====
```



```
Node: db02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf): online
Node: db01 (80389bde-c07a-454c-8735-98b0e3ede432): online

Resource Group: group_01
  resource_ip (heartbeat::ocf:IPAddr):      Started db02
  resource_pg (heartbeat::ocf:pgpool):      Started db02
  resource_ml (heartbeat::ocf:MailTo):      Started db02

Failed actions:
  resource_pg_start_0 (node=db01, call=13, rc=1): Error
```

障害状況確認(PostgreSQL 障害、pgpool スイッチオーバー運用)

- ✓ postgres ユーザに su し、pgpool に接続を行う。

```
# su postgres
$ psql -h db -p 9999
```

- ✓ “show pool_status” コマンドより、現在の動作状況を psql コマンドより確認する。

```
postgres=# show pool_status;
```

item	value
listen_addresses	*
port	9999
socket_dir	/tmp
backend_host_name	DB01
backend_port	5432
secondary_backend_host_name	DB02
secondary_backend_port	5432
num_init_children	70
child_life_time	300
connection_life_time	0
child_max_connections	0
max_pool	4
logdir	/tmp

backend_socket_dir	/tmp	
replication_mode	1	
replication_strict	1	
replication_timeout	5000	
load_balance_mode	1	
weight_master	0.500000	
weight_secondary	0.500000	
replication_stop_on_mismatch	0	
replicate_select	0	
reset_query_list	ABORT; RESET ALL; SET SESSION AUTHORIZATION DEFAULT;	
print_timestamp	1	
master_slave_mode	0	
connection_cache	1	
health_check_timeout	20	
health_check_period	30	
health_check_user	apadmin	
insert_lock	0	
ignore_leading_white_space	0	
current_backend_host_name	DB02	
current_backend_port	5432	
replication_enabled	0	
master_slave_enabled	0	
num_reset_queries	3	
log_statement	1	
log_connections	0	
log_hostname	0	
enable_pool_hba	1	
server_status	master(DB01 on 5432) down secondary(DB02 on 5432) up	
(41 rows)		

- ✓ 下記のどちらかに該当する場合、pgpool が縮退運転となっている為、データの同期化が必要となる。
 - “replication_enabled” の値が、“0” になっている場合
 - “server_status” の値が、片側の DB サーバ側が “down”、もう片方の DB サーバが “up” になっている場合

- ✓ **【重要】データの同期化が必要な場合は、“current_backend_host_name”の値を控えておく。(この値のサーバがデータ同期元となる為)**
復旧の手順で、pgpoolを停止させた後は、どちらが最新版のデータを持っているDBサーバであるかが、判断する材料がなくなってしまう。

(ア) PostgreSQL 接続確認

- ✓ psql コマンドから、双方の PostgreSQL に接続が行えることを確認する。疎通している場合は、下記のようなメッセージが表示される。

```
# su postgres
$ psql -h db01
Welcome to psql 8.2.5, the PostgreSQL interactive terminal.

Type:  ¥copyright for distribution terms
       ¥h for help with SQL commands
       ¥? for help with psql commands
       ¥g or terminate with semicolon to execute query
       ¥q to quit
postgres=# ¥q
$ psql -h db02
Welcome to psql 8.2.5, the PostgreSQL interactive terminal.

Type:  ¥copyright for distribution terms
       ¥h for help with SQL commands
       ¥? for help with psql commands
       ¥g or terminate with semicolon to execute query
       ¥q to quit
postgres=# ¥q
```

(イ) pgpool スイッチオーバー運用確認

- ✓ postgres ユーザに su し、pgpool に接続を行う。

```
# su postgres
$ psql -h db -p 9999
```

- ✓ “show pool_status” コマンドより、現在の動作状況を psql コマンドより確認する。

```
postgres=# show pool_status;
      item      | value |
-----+-----+

```

listen_addresses	*		
port	9999		
socket_dir	/tmp		
backend_host_name	DB01		
backend_port	5432		
secondary_backend_host_name	DB02		
secondary_backend_port	5432		
num_init_children	70		
child_life_time	300		
connection_life_time	0		
child_max_connections	0		
max_pool	4		
logdir	/tmp		
backend_socket_dir	/tmp		
replication_mode	1		
replication_strict	1		
replication_timeout	5000		
load_balance_mode	1		
weight_master	0.500000		
weight_secondary	0.500000		
replication_stop_on_mismatch	0		
replicate_select	0		
reset_query_list	ABORT; RESET ALL; SET SESSION AUTHORIZATION DEFAULT;		
print_timestamp	1		
master_slave_mode	0		
connection_cache	1		
health_check_timeout	20		
health_check_period	30		
health_check_user	apadmin		
insert_lock	0		
ignore_leading_white_space	0		
current_backend_host_name	DB02		
current_backend_port	5432		
replication_enabled	0		

master_slave_enabled	0	
num_reset_queries	3	
log_statement	1	
log_connections	0	
log_hostname	0	
enable_pool_hba	1	
server_status	master(DB01 on 5432) down secondary(DB02 on 5432) up	
(41 rows)		

- ✓ pgpool のスイッチオーバーを行った際も、下記のような縮退運転の状態として表示される。
 - “replication_enabled” の値が、“0” になっている場合
 - “server_status” の値が、片側の DB サーバ側が “down”、もう片方の DB サーバが “up” になっている場合
- ✓ **【重要】データの同期化が必要な場合は、“current backend host name” の値を控えておく。(この値のサーバがデータ同期元となる為)**
 復旧の手順で、pgpool を停止させてしまうと、どちらが最新版のデータを持っている DB サーバであるかが、判断する材料がなくなってしまうので注意。

5.2.2.2. 実施手順

障害復旧手順

- ✓ 実施条件により手順が若干異なる。以下の表 5-4 に従い、対応する復旧手順を実行する。

表 5-4DB サーバ障害(Heartbeat(pgpool)・PostgreSQL)の復旧手順

障害の種類	復旧手順						
	(ア)	(イ)	(ウ)	(エ)	(オ)	(カ)	(キ)
	電源 復旧	ネットワーク 復旧	Heartbeat 復旧	pgpool 復 旧	PostgreSQL 復旧	データ 同期	Heartbeat ファイルバ ック
マシン電源障害		-	-	-	-		
ネットワーク障害	-		-	-	-		
Heartbeat プロセスダウン	-	-		-	-	-	
pgpool 起動失敗	-	-	-		-	-	
PostgreSQL 障害	-	-	-	-			
pgpool スイッチ オーバー運用	-	-	-	-	-		

のフェイルバックに関しては、必須な手順ではない。

(ア) 電源復旧

- ✓ マシンの電源を On にし、OS の起動を確認する。
- ✓ 電源が On にできない場合や、OS が起動しない場合は、HW トラブルの可能性があるので、システム管理者に連絡を行う。

(イ) ネットワーク復旧

- ✓ LAN ケーブルの断線有無、ネットワーク中継機器の正常確認、ファイアウォールの設定を確認し、DB サーバ(#1)と DB サーバ(#2)が相互に ping 通信が可能であることを確認する。
- ✓ 上記を確認した上で、ネットワークが復旧しない場合は、HW トラブルの可能性があるので、システム管理者に連絡を行う。

(ウ) Heartbeat 復旧

- ✓ 通常、Heartbeat は OS 起動時に自動起動する設定になっているが、なにかしらの原因により、プロセスがダウンしてしまった場合は、service コマンドより、Heartbeat の手動起動を行う。

```
# service heartbeat start
Starting High-Availability services:
[ OK ]
```

- ✓ コマンド上は、「OK」と即座に表示されてしまうので、実際の起動確認は、crm_mon コマンドにて、停止していた側の Node が「online」になることを確認する。

```
# crm_mon -i 3
Refresh in 2s...

=====
Last updated: Tue Dec 18 08:58:14 2007
Current DC: db02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf)
2 Nodes configured.
1 Resources configured.
=====

Node: db02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf): online
Node: db01 (80389bde-c07a-454c-8735-98b0e3ede432): online

Resource Group: group_01
  resource_ip (heartbeat::ocf:IPAddr):      Started db02
  resource_pg (heartbeat::ocf:pgpool):      Started db02
  resource_ml (heartbeat::ocf:MailTo):      Started db02
```

起動までに、若干時間がかかることがある。

- ✓ 上記手順を経ても、Heartbeat が起動しない場合は、Heartbeat ログ(/opt/Openstandia/heartbeat/logs/ha-log)を確認し、ERROR や FATAL レベルのログが表示されていないかどうか確認し、原因が分かれば対処を行う。

```
# egrep "(ERROR|FATAL)" /opt/Openstandia/heartbeat/logs/ha-log
```

- ✓ Heartbeat ログに特に問題がない場合は、HWトラブルもしくは、OSトラブルの可能性があるので、システム管理者に連絡を行う。

(工) pgpool 復旧

- ✓ pgpool が起動失敗したホスト側で、以下の pgpool コマンドを実行し起動可能有無をチェックする。

```
# pgpool n
2007-12-18 09:06:32 ERROR: pid 9043: pool_config: invalid value **** for log_statement
2007-12-18 09:06:32 ERROR: pid 9043: Unable to get configuration. Exiting...
```

- ✓ 上記のように起動時に、「pgpool successfully started」の文言が表示されない場合は、pgpool が起動できない状態であることが分かる。
- ✓ エラーログに表示されているパラメータを修正し、再度、pgpool コマンドを実行し、正常起動できることを確認する。
- ✓ 上記手順を経ても、pgpool が起動しない場合は、HWトラブルもしくは、OSトラブルの可能性があるので、システム管理者に連絡を行う。
- ✓ 正常に起動できる状態になった場合は、crm_resource コマンドより、エラー状態となったリソースのクリーンアップを行う。

```
# crm_resource -H db01 -r resource_pg -C
crm_resource[9301]: 2007/12/18_09:32:43 info: Invoked: crm_resource -H db01 -r resource_pg C
```

- ✓ crm_mon コマンドにて、「Failed actions」がクリアされていることを確認する。

```
# crm_mon -i 3
Refresh in 2s...

=====
Last updated: Tue Dec 18 09:33:10 2007
Current DC: db02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf)
2 Nodes configured.
1 Resources configured.
=====

Node: db02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf): online
Node: db01 (80389bde-c07a-454c-8735-98b0e3ede432): online

Resource Group: group_01
  resource_ip (heartbeat::ocf:IPAddr):      Started db02
  resource_pg (heartbeat::ocf:pgpool):      Started db02
  resource_ml (heartbeat::ocf:MailTo):      Started db02
```

リソースのエラー状態をクリアしても、自動フェイルバックは実行されない設定になっているので、pgpool は、同一の DB サーバ上で稼働し続けます。

(オ) PostgreSQL 復旧

- ✓ 通常、PostgreSQL は OS 起動時に自動起動する設定になっているが、なにかしらの原因により、プロセスがダウンしてしまった場合は、service コマンドより、PostgreSQL の手動起動を行う。

```
# service postgres start
Starting PostgreSQL: ok
```

- ✓ 上記コマンドを実行後、即座に「Starting PostgreSQL: ok」と表示されるが、このコマンドでは起動していないことがある為、実際に psql コマンドより、双方の PostgreSQL に接続を行い、起動していることを確認する。

```
# su postgres
$ psql -h db01
Welcome to psql 8.2.5, the PostgreSQL interactive terminal.

Type:  ¥copyright for distribution terms
       ¥h for help with SQL commands
       ¥? for help with psql commands
       ¥g or terminate with semicolon to execute query
       ¥q to quit
postgres=# ¥q
$ psql -h db02
Welcome to psql 8.2.5, the PostgreSQL interactive terminal.

Type:  ¥copyright for distribution terms
       ¥h for help with SQL commands
       ¥? for help with psql commands
       ¥g or terminate with semicolon to execute query
       ¥q to quit
postgres=# ¥q
```

- ✓ PostgreSQL が起動しない場合は、PostgreSQL エラーログ(/opt/Openstandia/pgsql/data/pg_log/postgresql-yyyy-mm-dd.log)を確認し、PostgreSQL の起動エラーの原因の対処を行う。
- ✓ 上記以外で、PostgreSQL が起動しない場合は、HWトラブルもしくは、OSトラブルの可能性があるので、システム管理者に連絡を行う。

(カ) データ同期

A) Heartbeat 停止

- ✓ データ同期が必要な場合、まず、双方の Heartbeat サービスを停止する。この手順により pgpool が停止し、アプリケーションからの pgpool/PostgreSQL へのデータアクセスは不可となる。

どちらの DB サーバが最新のデータを保持しているか、“current_backend_host_name” を控えた上で停止する。

データ同期実施時は、DB サーバ(pgpool)への接続が不可能となる為、定期メンテナンスや、アプリケーションが稼動していない時間帯に実施する。

```
# service heartbeat stop
Stopping High-Availability services:
[ OK ]
```

B) データ同期元ホストでの操作

- ✓ 同期元ホスト(“current_backend_host_name” で、設定されていたホスト)に postgres ユーザでログインを行う。

```
# ssh postgres@DB02
postgres@db02's password:
```

- ✓ 以後、他のユーザからの接続および、更新処理を受け付けないように、PostgreSQL の起動ポートを一時的に変更する。

```
$ pg_ctl stop
$ pg_ctl o “ -p 1234 ” start
```

- ✓ システム停止時間に余裕があれば、ダンプを実行する前に、フルバキューム&アナライズ処理を実行する。

```
$ TODAY=`date +%Y-%m-%d`
$ vacuumdb p 1234 a f z v 2> /tmp/db02_vacuumed.$TODAY.log
```

再利用領域が大きい場合や、大量のデータが存在する場合は、実行に長時間を要する場合がある。

- ✓ データ同期させる為のデータベースダンプ出力を行い、エラーログが出力されていないことを確認する。

```
$ TODAY=`date +%Y-%m-%d`
$ pg_dumpall p 1234 v c -o > /tmp/db02_synchronized.$TODAY.out 2> /tmp/db02_synchronized.$TODAY.log
$ more /tmp/db02_synchronized.$TODAY.log
```

C) データ同期先ホストでの操作

- ✓ 同期先ホストに postgres ユーザでログインを行う。

```
# ssh postgres@DB01
postgres@db01's password:
```

- ✓ データを同期する前に、念の為 DB サーバ(#1)のデータベースダンプ出力を行い、エラーログが出力されていないことを確認する。

```
$ TODAY=`date +%Y-%m-%d`
$ pg_dumpall v -c -o > /tmp/db01_backup.$TODAY.out 2> /tmp/db01_backup.$TODAY.log
$ more /tmp/db01_backup.$TODAY.log
```

- ✓ “root”ユーザに su を行い、PostgreSQL サーバを停止して、DB データディレクトリをリネームする。

```
$ su -
# service postgres stop
# cd /opt/Openstandia/pgsql
# TODAY=`date +%Y-%m-%d`
# mv data data.$TODAY
```

正常にデータが同期できたことを完全に確認できるまで、この “data.\$TODAY” ディレクトリの削除は行わない。

- ✓ 新たに PostgreSQL で使用するデータディレクトリを作成、所有権を設定する。

```
# mkdir p /opt/Openstandia/pgsql/data
# chown postgres:postgres /opt/Openstandia/pgsql/data
# exit
```

- ✓ postgres ユーザに戻り、以下のオプションにより PostgreSQL データディレクトリの初期化を行い、設定ファイルのみを上書きコピーする。
- ✓ 他のユーザからの接続および、更新処理を受け付けないように、PostgreSQL の起動ポートを一時的に変更し、PostgreSQL を起動する。

```
$ initdb D /opt/Openstandia/pgsql/data
$ cp /opt/Openstandia/pgsql/data.$TODAY/*.conf /opt/Openstandia/pgsql/data/
$ pg_ctl o “ -p 1234 ” start
```

- ✓ “/tmp” ディレクトリにデータ同期元ホスト側の同期データファイルの SCP コピー & PostgreSQL ヘインポートする。

```
$ scp postgres@DB02:/tmp/db02_synchronized.$TODAY.out /tmp
postgres@db02's password:
db02_synchronized.2007-12-18.out          100% 6026    5.9KB/s   00:00
$ psql p 1234 -e < /tmp/db02_synchronized.$TODAY.out 2> /tmp/db02_imported_error.$TODAY.log ¥
| tee /tmp/db02_imported.$TODAY.log
```

- ✓ システム停止時間に余裕があれば、ダンプを実行する前に、フルバキューム & アナライズ処理を実行する。

```
$ vacuumdb p 1234 a f z v 2> /tmp/db01_vacuumed.$TODAY.log
```

再利用領域が大きい場合や、大量のデータが存在する場合は、実行に長時間を要する場合がある。

- ✓ psql コマンドにより、ローカルの PostgreSQL に接続し、データが同期されたことを確認する。

```
$ psql p 1234 GYOM_DB
. . .
. . .
```

停止していた時間帯に作成されたレコードの存在チェックや、テーブルのレコード登録件数などの比較を行う。

D) Heartbeat 開始

- ✓ 双方の DB サーバの再起動を行う。

```
# shutdown r now
```

- ✓ 再起動後は、Heartbeat および、PostgreSQL が自動起動する。
- ✓ postgres ユーザに su し、pgpool に接続を行う。

```
# su postgres
$ psql -h db -p 9999
```

- ✓ “show pool_status” コマンドより、pgpool の状態を確認する。

```
postgres=# show pool_status;
```

item	value
listen_addresses	*
port	9999
socket_dir	/tmp
backend_host_name	DB01
backend_port	5432
secondary_backend_host_name	DB02
secondary_backend_port	5432
num_init_children	70
child_life_time	300
connection_life_time	0
child_max_connections	0
max_pool	4

logdir	/tmp	
backend_socket_dir	/tmp	
replication_mode	1	
replication_strict	1	
replication_timeout	5000	
load_balance_mode	1	
weight_master	0.500000	
weight_secondary	0.500000	
replication_stop_on_mismatch	0	
replicate_select	0	
reset_query_list	ABORT; RESET ALL; SET SESSION AUTHORIZATION DEFAULT;	
print_timestamp	1	
master_slave_mode	0	
connection_cache	1	
health_check_timeout	20	
health_check_period	30	
health_check_user	apadmin	
insert_lock	0	
ignore_leading_white_space	0	
current_backend_host_name	DB01	
current_backend_port	5432	
replication_enabled	1	
master_slave_enabled	0	
num_reset_queries	3	
log_statement	1	
log_connections	0	
log_hostname	0	
enable_pool_hba	1	
server_status	master(DB01 on 5432) up secondary(DB02 on 5432) up	
(41 rows)		

- ✓ pgpool の状態が下記のようにになっている場合は、復旧(データ同期)が完了したことになる。
 - “replication_enabled” の値が、“1”
 - “server_status” の値が、両方の DB サーバが “up”

E) 一時ファイル・ディレクトリ削除

- ✓ データ同期手順で作成した一時ファイルを削除する。(もしくは、バックアップサーバへの移動)
一時ファイルはバックアップとしても利用できる為、サーバのディスク容量が逼迫している場合のみ実施する。

1. データ同期元 DB サーバ側

```
# rm f /tmp/db*_synchronized.*.out /tmp/db*_synchronized.*.log
```

or

```
# mv /tmp/db*_synchronized.*.out /tmp/db*_synchronized.*.log /share/
```

2. データ同期先 DB サーバ側

```
# rm f /tmp/db*_synchronized.*.out
```

```
# rm f /tmp/db*_backup.*.out /tmp/db*_backup.*.log
```

```
# rm f /tmp/db*_imported*.*.log
```

or

```
# mv /tmp/db*_synchronized.*.out /tmp/db*_synchronized.*.log /share/
```

```
# mv /tmp/db*_backup.*.out /tmp/db*_backup.*.log /share/
```

```
# mv /tmp/db*_imported*.*.log /share/
```

(キ) Heartbeat フェイルバック

- ✓ pgpool の稼動が DB サーバ(#2)にフェイルオーバーされた後に、DB サーバ(#1)側に pgpool の戻す手順を以下に示す。
ここでは、DB サーバ(#2)で動作している pgpool を、DB サーバ(#1)にフェイルバックする例で説明している。

【注意事項】

フェイルバックは必須な手順ではありません。DB サーバ(#1)に処理能力の高いマシンを配置しているなどの理由があれば実施する。
フェイルバック中や直後は DB サーバ接続の瞬断が発生する為、定期メンテナンスや、アプリケーションが稼動していない時間帯に実施する。

- ✓ 双方の Heartbeat が正常稼動(Node が「online」)していることを、crm_mon コマンドより確認する。

```
# crm_mon -i 3
Refresh in 2s...

=====
Last updated: Tue Dec 18 08:58:14 2007
Current DC: db02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf)
2 Nodes configured.
1 Resources configured.
=====

Node: db02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf): online
Node: db01 (80389bde-c07a-454c-8735-98b0e3ede432): online

Resource Group: group_01
  resource_ip (heartbeat::ocf:IPAddr):      Started db02
  resource_pg (heartbeat::ocf:pgpool):      Started db02
  resource_ml (heartbeat::ocf:MailTo):      Started db02
```

- ✓ crm_standby コマンドより、pgpool が稼動中の DB サーバ(#2)側を強制的に待機状態に設定する。

```
# crm_standby -U db02 -v true
crm_standby[8941]: 2007/12/18_09:20:34 info: Invoked: crm_standby -U db02 -v true
```

- ✓ crm_standby コマンドより、DB サーバ(#2)の Node が「standby」状態になり、DB サーバ(#1)側に pgpool リソースが、移動したことを確認する。

```
# crm_mon -i 3
```

Refresh in 3s...

=====

Last updated: Tue Dec 18 09:20:22 2007

Current DC: db02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf)

2 Nodes configured.

1 Resources configured.

=====

Node: db02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf): **standby**

Node: db01 (80389bde-c07a-454c-8735-98b0e3ede432): **online**

Resource Group: group_01

resource_ip (heartbeat::ocf:IPAddr): **Started db01**

resource_pg (heartbeat::ocf:pgpool): **Started db01**

resource_ml (heartbeat::ocf:MailTo): **Started db01**

✓ crm_standby コマンドより、DB サーバ(#2)側を「online」状態に戻す。

```
# crm_standby -U db02 -v false
```

```
crm_standby[10113]: 2007/12/18_09:23:42 info: Invoked: crm_standby -U db02 -v false
```


✓ crm_standby コマンドより、DB サーバ(#2)の Node が「online」状態に戻ったことを確認する。

```
# crm_mon -i 3
Refresh in 3s...

=====
Last updated: Tue Dec 18 09:25:06 2007
Current DC: db02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf)
2 Nodes configured.
1 Resources configured.
=====

Node: db02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf): online
Node: db01 (80389bde-c07a-454c-8735-98b0e3ede432): online

Resource Group: group_01
  resource_ip (heartbeat::ocf:IPAddr):      Started db01
  resource_pg (heartbeat::ocf:pgpool):      Started db01
  resource_ml (heartbeat::ocf:MailTo):      Started db01
```

6. 補足資料

6.1. Heartbeat 主要コマンドリスト

6.1.1. Heartbeat 起動

6.1.1.1. コマンド

- ✓ Heartbeat の起動自体は、OS 起動時に自動起動を行っている為、通常はこのコマンドを使用することはない。
- ✓ “root”ユーザから、Heartbeat の起動を行いたい場合、以下のコマンドから実行する。

```
# service heartbeat start
```

6.1.1.2. 実行例

- ✓ “root”ユーザで、Heartbeat を起動する。

```
# service heartbeat start
```

6.1.2. Heartbeat 停止

6.1.2.1. コマンド

- ✓ “root”ユーザで、Heartbeat の停止を行いたい場合、以下のコマンドから実行する。

```
# service heartbeat stop
```

6.1.2.2. 実行例

- ✓ “root”ユーザで、Heartbeat を停止する。

```
# service heartbeat stop
```

6.1.3. Heartbeat 稼働状況確認

6.1.3.1. コマンド

- ✓ Heartbeat の稼働状態、リソースの始動有無、エラーの状態などの確認時に使用する。

```
# crm_mon i [画面リフレッシュ秒数]
```

- ✓ “-i”オプションで指定された秒数毎に、画面がリフレッシュされるようになる。

6.1.3.2. 実行例

- ✓ Heartbeat の稼働状態を、3 秒ごとに表示したい場合

```
# crm_mon -i 3
Refresh in 2s...

=====
Last updated: Tue Dec 18 08:58:14 2007
Current DC: db02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf)
2 Nodes configured.
1 Resources configured.
=====

Node: db02 (b6587dd4-a24a-4759-82d1-8283ed09d2bf): online
Node: db01 (80389bde-c07a-454c-8735-98b0e3ede432): online

Resource Group: group_01
  resource_ip (heartbeat::ocf:IPAddr):      Started db01
  resource_pg (heartbeat::ocf:pgpool):      Started db01
  resource_ml (heartbeat::ocf:MailTo):      Started db01
```

モニタを終了する時は、「ctrl + c」で、終了する。。

6.1.4. Heartbeat ノード Active/Standby 切り替え

6.1.4.1. コマンド

- ✓ Heartbeat 動作中のノードの Active/Standby の切り替えを行う。
- ✓ Heartbeat でいう、Active とはリソース始動可能なノード、Standby とはリソース始動不可能なノードのことをさす。その為、正常動作時は両方のノードが Active(online)になっていなければならない。(Active/Standby でないことに注意が必要)

```
# crm_standby U [対象ホスト名] v [true|false]
```

- ✓ “-U”オプションには、Active/Standby の切り替えを行う対象のホスト名を指定する。
- ✓ “-v”オプションには、true(Standby 化)または、false(Active 化)を指定する。

6.1.4.2. 実行例

- ✓ DB サーバ(#2)を Standby 化し、動作しているリソースを DB サーバ(#1)へ手動フェイルオーバー(バック)する場合には、下記のコマンドを実行する。

```
# crm_standby U db02 -v true
crm_standby[8941]: 2007/12/18_09:20:34 info: Invoked: crm_standby -U db02 -v true
```

このコマンド実行後は、crm_mon コマンドで、Node の状態が切り替わったことを確認する。

- ✓ DB サーバ(#2)を再び Active 化し、リソース始動可能な状態に戻す場合には、下記のコマンドを実行する。

```
# crm_standby -U db02 -v false
crm_standby[10113]: 2007/12/18_09:23:42 info: Invoked: crm_standby -U db02 -v false
```

このコマンド実行後は、crm_mon コマンドで、Node の状態が切り替わったことを確認する。

6.1.5. Heartbeat リソース制御

6.1.5.1. コマンド

- ✓ Heartbeat 稼動中に発生したリソースのエラー状態を制御する。
- ✓ 基本的に、エラー状態となったリソースをクリーンアップする目的のみで、使用する。

```
# crm_resource H [ホスト名] r [リソース名] C
```

- ✓ “-H”オプションでは、リソースエラーが発生しているホスト名を指定する。
- ✓ “-r”オプションでは、crm_mon コマンド上で表示されている起動が失敗したリソース名を指定する。
- ✓ “-C”オプションでは、リソースのクリーンアップ処理の実行を指定している。

6.1.5.2. 実行例

- ✓ DB サーバ(#1)上で起動失敗したリソース(resource_pg)を、クリーンアップする場合には、下記のコマンドを実行する。

```
# crm_resource -H db01 -r resource_pg -C
```

```
crm_resource[9301]: 2007/12/18_09:32:43 info: Invoked: crm_resource -H db01 -r resource_pg C
```

このコマンド実行後は、crm_mon コマンドで、指定したリソースのクリーンアップが行われたかどうか確認する。

6.2. pgpool 主要コマンドリスト

6.2.1. pgpool 起動

6.2.1.1. コマンド

- ✓ pgpool の起動自体は Heartbeat が管理を行う為、通常は管理者がこのコマンドを使用することはない。
- ✓ pgpool が起動しない場合などのエラー確認のために使用することがある。

```
# pgpool [ f 設定ファイル] [ a 認証ファイル] n [-d]
```

- ✓ “-f”オプションを指定するのは、デフォルトの設定ファイル(/opt/Openstandia/pgpool/etc/pgpool.conf)以外を使用する場合のみ。
- ✓ “-a”オプションを指定するのは、デフォルトの認証ファイル(/opt/Openstandia/pgpool/etc/pool_hba.conf)以外を使用する場合のみ。
- ✓ “-n”オプションを指定した場合、pgpool は非デーモンモード(フォアグラウンド)で起動する。このモードで起動した場合は、「ctrl+c」により、pgpool を停止できる。
- ✓ “-d”オプションを指定した場合は、出力されるログレベルがデバッグレベルとなる。

6.2.1.2. 実行例

- ✓ pgpool が正常に起動できるかどうか確認する場合には、下記のコマンドを実行する。

```
# pgpool -n
2007-12-19 03:45:29 LOG: pid 3954: pgpool successfully started
2007-12-19 03:45:29 LOG: pid 3954: starting health checking
```

バックエンドの PostgreSQL の起動状態によっては、エラーログが表示されることがあるが、pgpool の起動自体は、1 行目の「pgpool successfully started」が表示されれば、OK となる。

6.2.2. pgpool 停止

6.2.2.1. コマンド

- ✓ pgpool の停止自体は Heartbeat が管理を行う為、通常は管理者がこのコマンドを使用することはない。
- ✓ pgpool を強制的に停止させる場合に使用する。

```
# pgpool [-m {s[mart]|f[ast]|i[mmediate]}] stop
```

- ✓ “-f”オプションを指定するのは、デフォルトの設定ファイル(/opt/Openstandia/pgpool/etc/pgpool.conf)以外を使用する場合のみ。
- ✓ “-a”オプションを指定するのは、デフォルトの認証ファイル(/opt/Openstandia/pgpool/etc/pool_hba.conf)以外を使用する場合のみ。
- ✓ “-m”オプションで停止時のモードを指定する。“f[ast]” および、“i[mmediate]”を指定した場合は、pgpool に対して接続ユーザが存在しても強制的に pgpool を停止する。
- ✓ “-m”オプションを指定しない、もしくは、“-m s[mart]”で実行した場合は、接続ユーザが接続を切断するまで終了を待機する。
 コネクションプールからの接続が存在する場合は、いくら待機しても終了させられない為、強制終了オプションを使用する。

6.2.2.2. 実行例

- ✓ 稼働中の pgpool を強制停止する。

```
# pgpool m fast stop
stop request sent to pgpool. waiting for termination....done.
```

- ✓ 稼働中の pgpool を接続ユーザが居ないタイミングで停止する。

```
# pgpool m smart stop
stop request sent to pgpool. waiting for termination.....
.....done.
```

6.2.3. pgpool 稼働状態確認

6.2.3.1. コマンド

- ✓ pgpool に接続可能なユーザに su を行い、psql コマンドより、pgpool に接続を行う。
- ✓ "show pool_status" コマンドを実行することで、pgpool の状態を確認できる。

```
postgres=# show pool_status;
```

6.2.3.2. 実行例

- ✓ DB サーバ仮想 IP アドレス上で稼働している pgpool に接続を行い、状態を確認する場合には下記のコマンドを実行する。

```
# su postgres
$ psql h DB p 9999
Welcome to psql 8.2.5, the PostgreSQL interactive terminal.

Type:  ¥copyright for distribution terms
       ¥h for help with SQL commands
       ¥? for help with psql commands
       ¥g or terminate with semicolon to execute query
       ¥q to quit

postgres=# show pool_status;
```

item	value
listen_addresses	*
port	9999
socket_dir	/tmp
backend_host_name	DB01
backend_port	5432
secondary_backend_host_name	DB02
secondary_backend_port	5432
num_init_children	70
child_life_time	300
connection_life_time	0
child_max_connections	0
max_pool	4

logdir	/tmp	
backend_socket_dir	/tmp	
replication_mode	1	
replication_strict	1	
replication_timeout	5000	
load_balance_mode	1	
weight_master	0.500000	
weight_secondary	0.500000	
replication_stop_on_mismatch	0	
replicate_select	0	
reset_query_list	ABORT; RESET ALL; SET SESSION AUTHORIZATION DEFAULT;	
print_timestamp	1	
master_slave_mode	0	
connection_cache	1	
health_check_timeout	20	
health_check_period	30	
health_check_user	apadmin	
insert_lock	0	
ignore_leading_white_space	0	
current_backend_host_name	DB01	
current_backend_port	5432	
replication_enabled	1	
master_slave_enabled	0	
num_reset_queries	3	
log_statement	1	
log_connections	0	
log_hostname	0	
enable_pool_hba	1	
server_status	master(DB01 on 5432) up secondary(DB02 on 5432) up	
(41 rows)		

6.2.4. pgpool スイッチオーバー

6.2.4.1. コマンド

- ✓ 2つの PostgreSQL のうち片側へのレプリケーションを停止したいような場合、pgpool のスイッチオーバーコマンドを実行する。

```
# pgpool [ f 設定ファイル] [ a 認証ファイル] [-s m[aster]|s[econdary]] switch
```

- ✓ “-f”オプションを指定するのは、デフォルトの設定ファイル(/opt/Openstandia/pgpool/etc/pgpool.conf)以外を使用する場合のみ。
- ✓ “-a”オプションを指定するのは、デフォルトの認証ファイル(/opt/Openstandia/pgpool/etc/pool_hba.conf)以外を使用する場合のみ。
- ✓ “-s”オプションは、どちらの PostgreSQL へのレプリケーションを停止するかを指定する。“m[aster]”を指定した場合は、マスタ側の PostgreSQL のレプリケーションが停止する。“s[econdary]”を指定した場合は、セカンダリ側の PostgreSQL のレプリケーションが停止する。

6.2.4.2. 実行例

- ✓ マスタ側の PostgreSQL のレプリケーションを停止する場合、以下のコマンドを実行する。

```
# pgpool s master switch
```

このコマンド実行後は、pgpool 稼動確認コマンドより、スイッチオーバーが正しく実行されたかどうか確認を行う。

- ✓ セカンダリ側の PostgreSQL のレプリケーションを停止する場合、以下のコマンドを実行する。

```
# pgpool s secondary switch
```

このコマンド実行後は、pgpool 稼動確認コマンドより、スイッチオーバーが正しく実行されたかどうか確認を行う。

6.3. PostgreSQL 主要コマンドリスト

6.3.1. PostgreSQL 起動

6.3.1.1. コマンド

- ✓ postgres ユーザから、PostgreSQL の起動を行いたい場合、以下のコマンドから実行する。
- ✓ PostgreSQL の起動自体は、OS 起動時に自動起動を行っている為、通常は postgres ユーザがこのコマンドを使用することはない。

```
$ pg_ctl [-o “ -p 接続ポート ” ] start
```

- ✓ “-o”オプションを指定した場合、PostgreSQL を起動する際のオプションを指定できる。”-p 接続ポート”を指定した場合は、PostgreSQL に接続できるポートを指定したポートに設定できる。

or

- ✓ “root”ユーザから、PostgreSQL の起動を行いたい場合、以下のコマンドから実行する。

```
# service postgres start
```

これらのコマンドからは、起動ポートの指定等はできない。

6.3.1.2. 実行例

- ✓ “postgres”ユーザで、DB サーバの PostgreSQL をポート “1234” で起動する。

```
# su postgres
$ pg_ctl stop
```

6.3.2. PostgreSQL 停止

6.3.2.1. コマンド

- ✓ postgres ユーザから、PostgreSQL の停止を行いたい場合、以下のコマンドから実行する。

```
$ pg_ctl stop
```

or

- ✓ “root”ユーザから、PostgreSQL の停止を行いたい場合、以下のコマンドから実行する。

```
# service postgres stop
```

6.3.2.2. 実行例

- ✓ “postgres”ユーザで、DB サーバの PostgreSQL を停止する。

```
# su postgres
```

```
$ pg_ctl stop
```

6.3.3. 全データベースダンプ

6.3.3.1. コマンド

- ✓ データベースのバックアップおよび、PostgreSQL 間のデータ同期を行う際に必要なダンプファイルを生成する為に使用する。

```
$ pg_dumpall [ p ポート番号] c o v > [ダンプファイル名] 2> [ダンプ実行ログ]
```

- ✓ “-p”オプションを指定した場合、通常の PostgreSQL のポート(4321)以外のポート番号で接続する。
- ✓ “-c”オプションでは、データベースを再作成するコマンドの前に、データベースのクリーンアップ(削除)するコマンドを書き出す。ロールおよびテーブル空間用の DROP コマンドも同様に追加する。
- ✓ “-o”オプションでは、各テーブルのオブジェクト識別子(OID)をデータの一部としてダンプする。pgpool によるデータ同期を実行している場合は、このオプションの指定は必須となる。
- ✓ “-v”オプションでは、冗長モードを指定する。これを指定すると、pg_dumpall は開始時刻と終了時刻をダンプファイルに、進行メッセージを標準エラーに出力するようになる。

このコマンドを実行する際は、PostgreSQL が事前に起動していなければならない。

6.3.3.2. 実行例

- ✓ DB サーバ(#1)のデータベースフルバックアップおよび、バックアップ時のログを生成する。

```
# su postgres
$ TODAY=`date +%Y-%m-%d`
$ pg_dumpall -c -o v > /tmp/db01_backup.$TODAY.out 2> /tmp/db01_backup.$TODAY.log
```

6.3.4. 不要領域削除

6.3.4.1. コマンド

- ✓ データベースの不要領域の削除、および統計情報の更新を行う際に使用する。

```
$ vacuumdb [ p 接続ポート] a f z v 2> [バキューム実行ログ]
```

- ✓ “-p”オプションを指定した場合、通常の PostgreSQL のポート(4321)以外のポート番号で接続する。
- ✓ “-a”オプションでは、全てのデータベースに対してバキュームを行う。
- ✓ “-f”オプションでは、“完全な(full)”クリーンアップ(ディスク使用領域も削減)する。
- ✓ “-z”オプションでは、オプティマイザが使用するデータベースの統計情報を更新する。
- ✓ “-v”オプションでは、処理中に詳細な情報を表示する。

このコマンドを実行する際は、PostgreSQL が事前に起動していなければならない。

6.3.4.2. 実行例

- ✓ DB サーバ(#1)の不要領域削除および、統計情報の更新を行う。

```
# su postgres
$ TODAY=`date +%Y-%m-%d`
$ vacuumdb p 1234 a f z v 2> /tmp/db01_vacuumed.$TODAY.log
```