

CLUSTERPRO MC ProcessSaver クイックリファレンス

**HP-UX 版
Linux 版**

第 2 版

2012年 11 月

日本電気株式会社

目次

- 1 はじめに
- 2 pfile の設定パラメータ
 - (1) 基本パラメータ
 - (2) オプションパラメータ
- 3 再起動スクリプトの記述方法
- 4 pcheck の起動・停止
 - (1) コマンドラインからの起動
 - (2) rc からの起動
- 5 運用管理
 - (1) 監視の一時停止・再開コマンド
 - (2) 監視状態の表示コマンド
 - － 監視コマンドの一覧表示
 - － 監視ルール表示
 - － 監視状態表示
 - (3) 運用中に出力されるメッセージ
 - (4) リトライオーバー時の対処
 - (5) オンライン保守手順
- [付録] パラメータ設定シート

1 はじめに

本マニュアルは ProcessSaver の基本的な運用に最低限必要な設定を簡易的に記載したものです。

本マニュアルでは基本的なプロセス監視機能

【プロセス監視】→【消滅検知】→【自動再開】

を運用する際に必要となるファイルおよびパラメータの意味や設定方法をご説明いたします。

なお、詳細な設定につきましては、同梱の「ユーザズガイド」をご覧ください。

※ 本書は、HP-UX版/Linux版共通のドキュメントです。

2 pfileの設定パラメータ

(1)基本パラメータ

(2)オプションパラメータ

※pfile とは pcheck(1M) プロセス単位に定義される設定ファイルです。

(1)基本パラメータ

PARAM・・・pcheck 全体の動作を規定する共通部情報

```
##### PARAM #####  
IPCKEY                0x1f000501      ①  
MSG_CHECK_INTERVAL    5                ②  
MONITOR_INTERVAL      10              ③  
SHM_DUMP_FILE         /var/opt/HA/PS/log/pcheck_dump ④
```

- ① IPCKEY.....共有メモリのkeyを指定します。
システム上で一意となる 0x00000001～0x7fffffff の数値を設定して下さい。
- ② MSG_CHECK_INTERVAL....内部イベントをスケジュールするタイマ値を指定します。
指定値は MONITOR_INTERVAL より小さい値を設定して下さい。
- ③ MONITOR_INTERVALプロセス監視を一定間隔で行う場合のタイマ値を指定します。
MSG_CHECK_INTERVAL の整数倍の値を設定して下さい。
- ④ SHM_DUMP_FILE.....共有メモリのイメージを出力するファイル名を絶対パスで指定します。
ファイル名は128文字未満で設定して下さい。

(続き)

PENT・・・プロセス単位の管理情報を規定する個別部情報

```
##### PENT #####  
## pname:restart shell:grace:retry_count_max:retry_over_action  
sampleproc:/home2/bin/restart.sh:86400:3:continue
```

各パラメータは
「:(コロン)」で
区切ります。

- ① 監視対象プロセス名.....ps コマンド(ps -ef)で表示されたプロセス名をそのまま設定して下さい。
- ② 再起動スクリプト名.....監視対象プロセスを起動するシェルスクリプトを絶対パスで設定します。
省略時は 0 または - (ハイフン) を指定してください。
また監視プロセス起動用の rc スクリプトを直接指定することも可能です。
rc スクリプトを指定した例 : /sbin/init.d/sample start
- ③ GRACE 値.....リトライ回数に達するまでの監視時間を指定します。
- ④ リトライ回数.....GRACE 値内に実行するプロセスの最大再開回数を指定します。
- ⑤ リトライオーバーアクション.....リトライ回数に達した場合の動作を設定します。
以下の動作を指定できます。



continue : 対象プロセス以外のプロセスを継続して監視
exit : クラスタ環境の設定により、サーバ切り替えにつながる終了処理
shutdown : pcheck コマンドを終了

(2) オプションパラメータ

オプションパラメータを指定することで、対象プロセスを特定するためのキーを指定することができます。
オプション情報を「, (カンマ)」で区切ることで複数のオプションを指定することができます。

※オプションパラメータは省略可能です。

● 同一名プロセスを特定する場合に有効なオプション

```
sampleproc:/home2/bin/restart.sh:86400:3:continue:uid=503
```

uid=xxx.....xxxには対象プロセスのuidを指定します。
対象プロセスが複数存在しuidが異なる場合に指定します。

```
sampleproc:/home2/bin/restart.sh:86400:3:continue:uname=user1
```

uname=xxx.....xxxには対象プロセスのユーザー名を指定します。
対象プロセスが複数存在し、ユーザー名が異なる場合に指定します。

```
sampleproc:/home2/bin/restart.sh:86400:3:continue:include_strings=keyword
```

include_strings=xxx または include_strings=xxx&yyy&zzz
.....xxx、yyy、zzzにはプロセス名の検索条件となる文字列を指定します。

```
sampleproc:/home2/bin/restart.sh:86400:3:continue:min_proc_count=3
```

min_proc_count=xxx.....xxxには起動するプロセス数を指定します。
指定された数未満となった場合にプロセス消滅を検出します。

(続き)

● プロセスをグルーピングする場合に有効なオプション

```
sampleproc1:/home2/bin/restart.sh:86400:3:continue:grouptag=group1  
sampleproc2:/home2/bin/restart.sh:86400:3:continue:grouptag=group1
```

grouptag=xxx.....xxxにはグループ監視を行う場合のグループ名を指定します。
注)グループ監視の詳細については、ユーザズガイドを参照してください。

● その他のオプション

```
sampleproc:/home2/bin/restart.sh:86400:3:continue:clear_cmd=/home2/bin/clear.sh
```

clear_cmd=xxx.....xxxにはリトライオーバー時に実行するシェルスクリプトファイル名を絶対パスで指定します。

```
sampleproc:/home2/bin/restart.sh:86400:3:continue:restart_timeout=30
```

restart_timeout=xxx.....xxxには再起動スクリプトの実行に時間がかかる場合に、監視対象プロセス毎に再起動スクリプトの実行を待ち合わせる時間を設定します。
注) MONITOR_INTERVALの整数倍の値を設定してください。

```
sampleproc:/home2/bin/restart.sh:86400:3:continue:restart_waittime=60
```

restart_waittime=xxx....xxxには監視対象プロセス毎に、プロセスの再起動がエラー終了した場合の待ち合わせ時間を設定します。

3 再起動スクリプトの記述方法

再起動スクリプトは、pcheck(1M)がプロセスの消滅を検出した場合に、プロセスを再開するための手続きを記述したファイルです。

監視対象プロセスが終了した場合に必要な処理と再起動のために必要な処理を記述してください。フォーマットは通常のシェルスクリプトです。

基本的な記述方式は以下となります。

① 再起動スクリプトには、プロセス停止に伴う後処理を記述します。ここでは stop を指定します。
念のため、後処理が完了するまで sleep によって待ち合わせます。

② プロセス起動の前処理と実際の起動処理を記述します。ここでは start を指定します。
念のため、開始処理が完了するまでsleepによって待ち合わせます。

③ exit 0を記述します(ProcessSaverは再起動スクリプトの戻り値によって正常に再起動できたかどうかを判断しているため、必ずexit0で終了します)。

再起動スクリプト（例）inetd の場合

```
#!/bin/sh

/sbin/init.d/inetd stop          →①
/usr/bin/sleep 5

/sbin/init.d/inetd start        →②
/usr/bin/sleep 5

exit 0                           →③
```

(注)上記は HP-UX の例となります。

Linux の場合は、/usr/bin/sleep を /bin/sleep に
/sbin/init.d/ を /etc/init.d/ に読み替えてください。

4 pcheckの起動・終了

(1) コマンドラインからの起動

(2) rcからの起動

(1) コマンドラインからの起動

コマンドラインからのProcessSaver起動、停止手順は以下の通りです。

注) 下記コマンドのpfile名には絶対パスを指定してください。

● 起動

コマンドのフォーマット: `/opt/HA/PS/bin/pcheck -f <pfile> &`

例: `# /opt/HA/PS/bin/pcheck -f /var/opt/HA/PS/conf/bin/pfile &`



pcheck(1M)は監視を開始すると終了するまで、フォアグラウンドにプロンプトが返ってこないため、「&」を付けてバックグラウンドでコマンドを実行します。

● 終了

コマンドのフォーマット: `/opt/HA/PS/bin/padmin -f <pfile> -c shutdown`

例: `# /opt/HA/PS/bin/padmin -f /var/opt/HA/PS/conf/bin/pfile -c shutdown`

(2)rcからの起動（HP-UXの場合）

rcファイルからpcheck(1M)を呼び出すことで、ノードの立ち上げ・立ち下げと連動して起動、終了を自動化できます。
rcファイルは、サンプルを参考にして作成します。

※ rcファイルを作成したら、/sbin/init.d/ 配下に保存し、指定するrcのディレクトリ (/sbin/rc2.d/ または /sbin/rc3.d/) 配下にリンクします。

rcファイルを使用したコマンド実行による起動と停止の手順は以下のとおりです。

● 起動

```
# /sbin/init.d/rcfile start
```

● 終了

```
# /sbin/init.d/rcfile stop
```

注)マシン起動・終了と同時にrcファイルが実行されることを確認する場合はご使用のマシンをリブートする必要があります。

サンプル:/var/opt/HA/PS/conf/src/RC_sample/rcfile

```
#!/bin/sh
#
## for rc3.d
# trap "" 1 2 3 15 ★3

# Environment
PCHECK=/opt/HA/PS/bin/pcheck
PADMIN=/opt/HA/PS/bin/padmin
PFILE=/var/opt/HA/PS/conf/bin/pfile ★1
:      :      :
'start')
    ## You must execute on background.
    ## for rc2.d
★2 #    ${PCHECK} -f ${PFILE} &
    ## for rc3.d
★3 #    /usr/bin/nohup ${PCHECK} -f ${PFILE} &
    ;;
'stop')
    ${PADMIN} -f ${PFILE} -c shutdown
:      :      :
```

rc3からの起動を指定する場合 #を外す

該当のpfileを指定

rc2からの起動を指定する場合 #を外す

rc3からの起動を指定する場合 #を外す

(2)rcからの起動（Linuxの場合）

rcファイルからpcheck(1M)を呼び出すことで、ノードの立ち上げ・立ち下げと連動して起動、終了を自動化できます。rcファイルは、サンプルを参考にして作成します。

※ rcファイルを作成したら、/etc/init.d/ 配下に保存し、指定する rc のディレクトリ (/etc/rc2.d/ および /etc/rc3.d/、/etc/rc5.d) 配下にリンクします。

rcファイルを使用したコマンド実行による起動と停止の手順は以下のとおりです。

● 起動

```
# /etc/init.d/rcfile start
```

● 終了

```
# /etc/init.d/rcfile stop
```

注)マシン起動・終了と同時にrcファイルが実行されることを確認する場合はご使用のマシンをリブートする必要があります。

サンプル:/var/opt/HA/PS/conf/src/RC_sample/rcfile

```
#!/bin/sh
#
trap "" 1 2 3 13 15

# Environment
PCHECK=/opt/HA/PS/bin/pcheck
PADMIN=/opt/HA/PS/bin/padmin
PFILE=/var/opt/HA/PS/conf/bin/pfile ★1
:      :      :
'start')
    ## You must execute on background.
★2 # /usr/bin/nohup ${PCHECK} -f ${PFILE} &
    ;;
'stop')
    ${PADMIN} -f ${PFILE} -c shutdown
    :      :      :
```

該当のpfileを指定

#を外す

5 運用管理

(1)監視の一時停止・再開コマンド

(2)監視状態の表示コマンド

- － 監視コマンドの一覧表示
- － 監視ルール表示
- － 監視状態表示

(3)運用中に出力されるメッセージ

(4)リトライオーバー時の対処

(5)オンライン保守手順

(1)監視の一時停止・再開コマンド

● プロセス監視の停止コマンド

コマンドのフォーマット: `/opt/HA/PS/bin/padmin -f <pfile> -c stop`

例: `# /opt/HA/PS/bin/padmin -f /var/opt/HA/PS/conf/bin/pfile -c stop`

● pfileの再読み込みコマンド

コマンドのフォーマット: `/opt/HA/PS/bin/padmin -f <pfile> -c reload`

例: `# /opt/HA/PS/bin/padmin -f /var/opt/HA/PS/conf/bin/pfile -c reload`

● 停止中のプロセス監視の再開コマンド

コマンドのフォーマット: `/opt/HA/PS/bin/padmin -f <pfile> -c start`

例: `# /opt/HA/PS/bin/padmin -f /var/opt/HA/PS/conf/bin/pfile -c start`

(2)監視状態の表示コマンド

ー監視コマンドの一覧表示

`/opt/HA/PS/bin/padmin -l`

- ipckey.....共有メモリのkey
(system上で一意の値)
- pfile.....監視対象のpfile名
- mypid.....pcheckのプロセスID
- myname.....実行中のコマンド名
- message.....処理中のメッセージ(イベント)

```
# /opt/HA/PS/bin/padmin -l
ipckey      = 0x1f000303
mypid       = 27913
myname      = pcheck
pfile       = /var/opt/HA/PS/conf/bin/pfile
message     = start
```



message の種類

start プロセス監視の実行
resume startと同じ
stop プロセス監視の停止
suspend stopと同じ
reload pfile再読み込み中

(続き)

— 監視ルール表示

```
padmin -f <pfile> -c show param
```

- MSG_CHECK_INTERVAL
.... 処理中のメッセージを確認する間隔
- MONITOR_INTERVAL
.... 監視間隔
- PFILE
.... 監視対象pfile名
- MESSAGE_BOX
.... 処理中のイベントを表す
メッセージ
- MONITOR_STOP_COUNT
.... 監視停止中のプロセス数
- FAIL_PROC_COUNT
.... リトライオーバーとなっているプロセス数
- ALL_PROC_COUNT
.... すべての監視プロセス数

```
# /opt/HA/PS/bin/padmin -f /var/opt/HA/PS/conf/bin//pfile
-c show param
MSG_CHECK_INTERVAL           = 5
MONITOR_INTERVAL             = 10
MONITOR_TRY_COUNT            = 2
SHM_DUMP_FILE                = /tmp/st/dump/proc_dump
PFILE                        = /var/opt/HA/PS/conf/bin
                               /pfile
MESSAGE_BOX                   = start
MONITOR_STOP_COUNT           = 0
FAIL_PROC_COUNT               = 0
ALL_PROC_COUNT                = 1
```

(続き)

— 監視状態表示

```
padmin -f <pfile> -c show pent
```

- pname.....監視対象のプロセス名
- pid.....監視対象プロセスID
- retry_count.....プロセスを再起動した回数
- restart_count.....プロセスを再起動した累計
- proc_sts.....プロセスの監視状態
- retry_over_act.....リトライオーバーとなった場合の処理
- rerun_time.....再起動日時（GRACE値時間内での初回再起動の日時）
- pent_id.....pfileのpent部分に指定したプロセスの通番
- monitor_sts.....pent単位の監視状態

```
# /opt/HA/PS/bin/padmin -f /var/opt/HA/PS/conf/bin/pfile  
-c show pent  
  
pname           = sampleproc  
pid             = 28073  
retry_count     = 0  
restart_count   = 0  
proc_sts        = AVAIL  
retry_over_act  = continue  
rerun_time      = -----  
pent_id         = 1  
monitor_sts     = on
```

(3)運用中に出力されるメッセージ

プロセスの消滅を検知すると、syslogに以下のようなメッセージが出力されますので、障害の原因を解析願います。

● プロセスの消滅を検知したメッセージ

Process[<プロセス名>, pid=<xxx>] Down

Jan 18 13:34:43 xxx pcheck[7750]: Process [sampleproc ,pid=16133] Down

● プロセスの再起動を開始したメッセージ

Try to restart (<プロセス名>)

Jan 18 13:34:43 xxx pcheck[7750]: Try to restart (sampleproc)

プロセスの消滅を検知すると、pfileに指定された再起動スクリプト等を実行しプロセスの再起動を開始します。

● プロセスの監視を開始したメッセージ

(<プロセス名>, pid=<xxx>) Up

Jan 18 13:35:04 xxx pcheck[7750]: (sampleproc ,pid=8107) Up

プロセスが起動していることを確認した後、pcheck(1M)によってプロセス監視を開始します。
※pfileに再起動スクリプト等を指定していない場合、プロセスの再起動は行われません。

● プロセスが動作していないことを検知したメッセージ

Set pid fail (<プロセス名>)

Jan 18 13:35:04 xxx pcheck[7750]: Set pid fail (sampleproc)

● リトライオーバーを検知したメッセージ

(<プロセス名>) RETRY_OVER:<リトライオーバーアクション>

Jan 18 13:39:25 xxx pcheck[7750]: (sampleproc) RETRY_OVER:CONTINUE

(4)リトライオーバーバ時の対処

pcheckが、対象プロセスの再起動にpfileで指定されたリトライ回数失敗するとリトライオーバーとなり監視を停止します。プロセス監視の再開は以下の手順で行います。

- pfileに指定したリトライオーバーアクションがcontinueの場合はpcheckは起動状態で監視を停止していますので、以下のコマンドによりプロセス監視を再開することができます。

```
/opt/HA/PS/bin/padmin -f <pfile> -c reload
```

```
# /opt/HA/PS/bin/padmin -f /var/opt/HA/PS/conf/bin/pfile -c reload
```

- リトライオーバーアクションがshutdown又はexitの場合は、pcheckが消滅していますので、pcheckコマンドによりプロセス監視を再開して下さい。

```
/opt/HA/PS/bin/pcheck -f <pfile> &
```

```
# /opt/HA/PS/bin/pcheck -f /var/opt/HA/PS/conf/bin/pfile &
```

(5)オンライン保守手順

監視対象プロセスの停止、再起動は以下の手順で行います。

① プロセス監視の一時停止

コマンドのフォーマット: `/opt/HA/PS/bin/padmin -f <pfile> -c stop`

例: `# /opt/HA/PS/bin/padmin -f /var/opt/HA/PS/conf/bin/pfile -c stop`

② 対象プロセスを停止、再起動

実行形式の変更によるプロセスの再起動等のメンテナンス作業を行うことが可能です。

③ pfileの再読み込み(初期化)

コマンドのフォーマット: `/opt/HA/PS/bin/padmin -f <pfile> -c reload`

例: `# /opt/HA/PS/bin/padmin -f /var/opt/HA/PS/conf/bin/pfile -c reload`



ProcessSaverは対象プロセスのPIDによって監視を行っているため、保守によって再起動を行う場合は、pfileの再読み込みを行わなければなりません。

(続き)

④ プロセス監視の再開

コマンドのフォーマット: `/opt/HA/PS/bin/padmin -f <pfile> -c start`

例: `# /opt/HA/PS/bin/padmin -f /var/opt/HA/PS/conf/bin/pfile -c start`

★ 手順③、④の代わりに以下のコマンドによっても、プロセス監視の再開が行えます。

コマンドのフォーマット: `/opt/HA/PS/bin/padmin -f <pfile> -c restart`

例: `# /opt/HA/PS/bin/padmin -f /var/opt/HA/PS/conf/bin/pfile -c restart`

※ pfileを変更した場合には、reload を使用してpfileの再読み込みが必要となります。

[付録]パラメータ設定シート(例)

以下のパラメータシートによって、pfileのパラメータを容易に設定することができます。

※現在記入している値は例です。未記入のシートをプリントアウトしてご利用ください。

プロセス監視 pfile設定項目					
pfile名		/var/opt/HA/PS/conf/bin/pfile			
運用形態		pcheck起動			
PARAM部					
項目	パラメータ名	設定値		備考	
共有メモリー	IPCKEY	0x1f000501		共有メモリのキー(0x00000001～0x7fffffff)	
メッセージチェックインターバル	MSG_CHECK_INTERVAL	5		内部イベントチェック間隔(秒)	
モニタインターバル	MONITOR_INTERVAL	10		プロセス監視間隔(秒)	
共有メモリダンプファイル	SHM_DUMP_FILE	/var/opt/HA/PS/log/pcheck_dump		共有メモリイメージ出力ファイル	
PENT部					
監視対象プロセス名	再起動スクリプト名	グレース 値	リトライ 回数	リトライオーバ アクション	備考
sampleproc	/home2/bin/restart.sh	86400	3	continue	

パラメータ設定シート

プロセス監視 pfile設定項目					
pfile名					
運用形態					
PARAM部					
項目	パラメータ名	設定値			備考
共有メモリキー	IPCKEY				共有メモリのキー(0x00000001~0x7fffffff)
メッセージチェックインターバル	MSG_CHECK_INTERVAL				内部イベントチェック間隔(秒)
モニタインターバル	MONITOR_INTERVAL				プロセス監視間隔(秒)
共有メモリダンプファイル	SHM_DUMP_FILE				共有メモリイメージ出力ファイル
PENT部					
監視対象プロセス名	再起動スクリプト名	グレース 値	リトライ 回数	リトライオーバ アクション	備考

Empowered by Innovation

NEC