NEC

安定した業務を実現　　High Availability Software

HA/SystemResourceMonitor R4.2
ユーザーズガイド　基本編

- □ 本製品の概要について
- □ 本製品の機能
- □ 本製品の運用手順
- □ SG の設定
- □ ログメッセージ
- □ 注意・制限事項
- □ 付録

はしがき

（1）本製品は以下のオペレーティングシステムに対応します。

HP-UX　11i v2 (Itanium)

HP-UX　11i v3 (Itanium)

（2）プロダクト型番について

本書で説明しているすべての機能は、プログラムプロダクトであり
次の表のプロダクト型番およびプロダクト名に対応します。

OS 名	プロダクト型番	プロダクト名	プロダクトリリース
HP-UX	UQ5218	HA/SystemResourceMonitor	R4.2
HP-UX	UQ5218M	HA/SystemResourceMonitor メディア	R4.2

（3）本書について

本マニュアルは HP-UX 版の HA/SystemResourceMonitor の基本機能について記載したものです。

（4）本リリースでの強化点について

SystemResourceMonitor R4.2 (2011.10 月出荷版) では以下の機能を強化しています。

・拡張機能としてディスク容量監視機能を強化しました。

- ディスク空き容量監視機能を追加

詳細は、『HA/SystemResourceMonitor ユーザーズガイド　応用編』を参照してください。

（5）これまでの改版履歴について

SystemResourceMonitor R4.1 (2011.4 月出荷版) は R4.1 へリビジョンアップしました。
R4.1 より製品名が HA/SystemMonitor　から HA/SystemResourceMonitor へ変更になりました。
R4.1 では以下の機能を強化しています。

・統計情報編集コマンド（pssmview）の強化

- カーネルパラメータ統計系情報ファイルの整形、編集機能の追加

  カーネルパラメータ統計情報ファイルの整形、編集が可能となりました。
  これにより、カーネルパラメータ採取情報を数値のみによる確認だけではなく、視覚的に確認することが可能となりました。

- WebSAM Invariant Analyzer 連携機能の追加

  システム性能分析ソフトウェア WebSAM Invariant Analyzer(以降 IA)との連携が可能となりました。
  これにより、SystemResourceMonitor で行っている閾値監視では検出できないサイレント障害を検出することが可能となりました。

SystemMonitor R3.1e (2010.9 末出荷版) では以下の機能を強化しています。

・カーネルパラメータ情報収集機能を追加

以下のカーネルパラメータ情報収集機能を追加しました。

- カーネルパラメータ統計情報収集機能

  オーバーフローする可能性があるカーネルパラメータの情報を定期的に収集する機能

- 全カーネルパラメータ詳細情報収集機能

  システムの全カーネルパラメータ詳細情報を起動時に収集する機能

SystemMonitor R3.1d (2010.3 末出荷版) では以下の機能を強化しています。

・拡張機能として、監視するリソース情報を追加

拡張機能として、以下のリソース監視機能を追加しました。
詳細は、『HA/SystemMonitor ユーザーズガイド　応用編』を参照してください。

- ディスク容量監視機能

  ファイルシステムのマウントポイント単位にディスク容量の使用率を定期的に収集し、診断する機能

- ネットワーク状態監視機能

  プロセスが使用するソケット情報を定期的に収集する機能

SystemMonitor R3.1c-P2 (2009.12 末出荷版) では以下の機能を強化しています。

・プロセス名が 64 文字の長さのプロセスのリソース監視を改善

プロセス名の長さが 64 文字のプロセスで、リソース監視異常を検出した場合に正しく検出できないケースがありましたが、これを改善しました。

SystemMonitor R3.1c-P1 (2009.7 末出荷版) では以下の機能を強化しています。

・統計情報に CPU 個別情報を追加

統計情報ファイルに CPU 個別情報を追加しました。
これにより CPU ごとのリソース使用状況の確認が可能となりました。

・プロセスリソース監視機能の可否選択を強化
以下の監視について、個々に監視の可否を選択できるようになりました。
- ファイルリーク監視
- プロセス毎のオープンファイル数上限監視
- スレッドリーク監視
- プロセス毎のスレッド数上限監視
これにより、さまざまな状況に応じた監視選択が可能となりました。
設定方法については、「4章 SGの設定」を参照してください。

SystemMonitor R3.1c (2009.3 末出荷版) では以下の機能を強化しています。
・統計情報編集コマンド（pssmview）の強化
pssmview(1M) を時刻指定で実行することが可能となりました。
これまで年・月・日単位でしか統計情報ファイルの情報を解析、編集できませんでしたが、
時・分・秒単位まで解析、編集することが可能となりました。
これにより短時間内でのリソース使用状況の確認が可能となりました。

・システムリソース監視機能の判定条件を強化
システムリソース監視の異常判定条件に連続超過時間を追加しました。
これまではシステムリソースの異常を検出する閾値を一度でも超えると、異常と判定していましたが、
連続超過時間を指定することで指定された時間連続して閾値を超えるまで異常と判断しないようになりました。
これにより負荷等で一時的に増加するようなシステムリソースについて、異常を検出することがなくなったため、より正確な判定を行うことが可能となりました。

SystemMonitor R3.1a (2008.5 末出荷版) では以下の機能を強化しています。
・スワップメモリ監視の強化
スワップメモリ監視の対象としてスワップメモリ使用量に加え、スワップメモリ予約量についても監視が可能となりました。
これまでなかったスワップメモリ予約量監視の実現によりスワップメモリ監視の精度が高まり、より正確なスワップメモリ監視異常の検出が可能となりました。

・メモリ使用量監視の強化
R3.1 以前のバージョンでは、ユーザメモリ使用量を監視対象としていましたが、
R3.1a より、システム全体のメモリ使用量を監視するという観点から、ユーザメモリ使用量に加え、システムメモリ使用量の監視を実現しました。
これによりメモリ使用量監視の精度が高まり、より正確なメモリ使用量監視異常の検出が可能となりました。※1
※1 システムによっては、これまでの設定値について再検討が必要となる場合があります。

・システムリソース監視の強化
システムリソース監視の可否およびリソース異常と判定される閾値を、各リソース単位に設定することが可能となりました。
設定方法については、「4章 SGの設定」を参照してください。

・統計情報ファイル出力先指定機能の追加
SG ファイルの設定を変更することで、統計情報ファイルの出力先を任意の場所に変更可能となりました。
デフォルトで運用した場合は、これまでのバージョンと同様に **/var/opt/HA/PSSM/log/** 配下に出力されます。
設定方法については、「4章 SGの設定」を参照してください。

・統計情報編集コマンド（pssmview）のサポート
pssmview(1M)は、統計情報を解析、編集するコマンドです。
収集した統計情報の推移を簡易的に表示することができ、日々のリソース使用状況の
確認や、障害解析時に有効です。
統計情報ファイル編集コマンドの詳細については、「統計情報編集コマンド
ユーザーズガイド」を参照してください。※2
※2　pssmview(1M)は、PA-RISC には対応していません。

・データファイルのディスク使用量の削減
本リリースより、統計情報ファイルおよび解析結果ファイルのフォーマットが
変更となりました。これにより、ファイルサイズが削減され、より効率的に統計情報の
収集を行うことが可能となりました。
過去の統計情報ファイルについても R3.1a で使用することは可能ですので、
特に意識する必要はありません。
また、ファイルサイズの削減にともない、SG ファイルで指定される、統計情報ファイルの
サイズおよびバックアップ数の指定値が変更となりました。
デフォルトで運用した場合、30MByte のファイルを 10 個までバックアップとして保存します。
指定可能な値の範囲については、「 4 章 SG の設定」を参照してください。

・テキストログファイルの提供
リソース異常検出時のメッセージ及び、システムログ出力レベルのメッセージを出力するための
テキストログファイル（pssm_syslog.log）を追加しました。
リソース異常検出時のメッセージをシステムログに出力したくない場合に、本ファイルを
システムログと同レベルで扱うことができます。
また、R3.1 以前のテキストログファイル(pssm_trace.log)は、本製品の動作ログを記録するための
内部ログファイルとなりましたので、本ファイルは特に意識して確認する必要はありません。

・リソース監視異常検出時のアクション強化について
リソース監視異常を検出した場合のアクション指定の１つとして、プロセス停止アクションを
提供してきましたが、これまでのプロセス KILL アクション除外定義ファイル（psact_notstop.conf）
に加え、プロセス KILL アクション定義ファイル（psact_stop_allow.conf）を追加しました。
これにより、より柔軟なアクション定義が可能となりました。
異常検出時に、該当プロセスの停止を行いたい場合、本ファイルを **/var/opt/HA/PSSM/conf/** 配下
に作成し、停止したいプロセスを指定することで、本ファイルに記載されたプロセスについてのみ、
異常検出時に kill(2) システムコールによって停止されます。
設定方法については、「 4 章 SG の設定」を参照してください。

HA/SystemMonitor（2007 年 9 月末出荷版）は R3.1 へリビジョンアップしました。
主な機能は以下のとおりです。

- プロセスリソース監視機能
  以下のプロセスリソースの監視が可能です。
  - ・ゾンビプロセス（defunct）監視
  - ・メモリリーク監視
  - ・ファイルリーク監視
  - ・プロセス毎のオープンファイル数上限監視（カーネルパラメータ maxfiles_lim 相当）
  - ・高 CPU 使用率プロセス監視
  - ・スレッドリーク監視
  - ・プロセス毎のスレッド数上限監視（カーネルパラメータ max_thread_proc 相当）
  - ・プロセスの多重度監視

- システムリソース監視機能
  以下のシステムリソースの監視が可能です。
  - ・プロセス数上限監視（カーネルパラメータ nproc 相当）
  - ・メモリ量上限監視（物理メモリ量）
  - ・オープンファイル数上限監視（カーネルパラメータ nfile 相当）
  - ・ロックファイル数上限監視（カーネルパラメータ nflocks 相当）
  - ・スワップメモリ量上限監視（使用可能スワップ領域）
    ※セカンダリスワップ領域およびスワップ予約量は監視対象外です。
  - ・スレッド数上限監視（カーネルパラメータ nkthread 相当）
    ※ PA-RISC は対象外です。
  - ・CPU 使用率監視（CPU 数×100(%)）
  - ・ユーザプロセス数上限監視（カーネルパラメータ maxuprc 相当）

- リソース異常検出時のアクション実行機能
  リソース監視異常が検出された場合のアクションとして、以下を指定することが可能です。
  - ・テキストログ通知（デフォルト）
    本製品規定のテキストログへ、リソース監視異常検出のメッセージを出力します。
  - ・システムログ通知
    システムログファイルへ、リソース監視異常検出のメッセージを出力します。
  - ・該当プロセスの停止
    プロセスリソース監視異常検出時に、該当のプロセスを停止します。
    ※ゾンビプロセス(defunct)監視、プロセスの多重度監視およびシステムリソース監視は対象外です。
  - ・指定したスクリプトの実行
    任意に指定したスクリプトを実行することが可能です。
    本製品では、デフォルトでリソース監視異常検出時の情報収集スクリプト（**/var/opt/HA/PSSM/bin/collect_info.sh**）を提供します。

- ユーザ組み込み情報収集スクリプト定期実行機能
  規定のディレクトリ（**/var/opt/HA/PSSM/scripts/**）配下のスクリプトを定期的に実行します。
  ユーザが任意に実行させたいスクリプトおよび実行間隔を指定することが可能です。
  本製品では、デフォルトで全カーネルパラメータ情報収集スクリプト（**/var/opt/HA/PSSM/scripts/S00get_kernelparam.sh**）を提供します。

（6） 商標・登録商標について

- ✓ HP-UX、PA-RISC 、Serviceguard は、米国における米国 Hewlett-Packard Company の登録商標です。
- ✓ Itanium は、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。
- ✓ その他、本書に登場する会社名および商品名は各社の商標または登録商標です。
- ✓ なお、本書では®、TM マークを明記しておりません。

目 次

1 本製品の概要について

１．１　本製品が提供する主な機能について

本製品は、プロセスが使用するリソースの統計情報を継続的に収集し解析することで、リソース枯渇等の原因となっているプロセスによって引き起こされる様々な問題を早期検出する機能を提供します。
本機能を利用することで、OSの異常動作や業務プロセスの障害等の業務停止を伴うような致命的な障害を未然に防止することができます。

本製品の主な機能は以下のとおりです。

１．プロセス個別リソース監視機能
個々のプロセスのリソース情報を継続的に収集し解析します。
リソースの状態が異常の場合は検出し、特定のアクションを行います。

２．システムリソース監視機能
システムのリソース状態を継続的に収集し解析します。あらかじめ設定した閾値を超えた場合は異常と判断し、特定のアクションを行います。

３．リソース監視異常検出時アクション実行機能
リソース監視の異常検出時に、特定のアクションを実行します。
指定可能なアクションには、テキストログ通知、システムログ通知、対象プロセスの停止、スクリプト実行があります。

４．ユーザ組み込み情報採取スクリプトの定期実行機能
任意の情報収集スクリプトを定期的に実行し、必要な情報を採取することができます。
本製品では、全カーネルパラメータ情報、システム動作情報、仮想メモリ統計情報等を採取するスクリプトを提供します。

５．カーネルパラメータ情報収集機能
オーバーフローする可能性があるカーネルパラメータの情報を継続的に収集します。
また、本製品の起動時に一度だけ全カーネルパラメータ詳細情報を収集します。

１．２　本製品が収集・監視する情報について

本製品では、プロセス個別の情報、システム全体の情報およびCPU 個別情報、カーネルパラメータ情報を統計的に収集します。
収集した情報をもとに解析を行い、リソース監視を行います。
本機能で収集、監視する情報は以下となります。

- 収集する情報

プロセス個別の情報として以下を収集することができます。

種別	取得項目	
プロセス基本情報	ユーザ名(UNAME)	プロセスステータス(STATUS)
	プロセスID(PID)	プロセスグループID(PGID)
	親プロセスID(PPID)	プロセス名(COMMAND)
メモリ関連	メモリ使用量(MEMORY)	
ファイル関連	オープンファイル数(FILE)	ロックファイル数(LOCKF)
	オープンファイル数上限値にしめる割合(%FILE)	
CPU関連	CPU使用率(CPU)	1CPUに対するCPU使用率(CPU/CPU数)
	CPU使用時間(CPUTIME)	
スレッド関連	スレッド数(THREAD)	スレッド数上限値にしめる割合(%THREAD)

システム全体の情報として以下を収集することができます。

種別	取得項目	
プロセス／スレッド関連	現在の総プロセス数	現在の総スレッド数
	現在プロセスを最も多く起動しているユーザ名およびプロセス数	
メモリ関連	現在の総メモリ使用量	現在の総スワップメモリ使用量
	現在の総スワップメモリ予約量	
ファイル関連	現在の総オープンファイル数	現在の総ロックファイル数
CPU関連	現在の総CPU使用率	コンテキストスイッチの回数

CPU 個別情報として以下を収集することができます。

種別	取得項目	
CPU 基本情報	CPU 番号	
run queue 関連	収集時点から過去 1 分間の平均の run queue の長さ	収集時点から過去 5 分間の平均の run queue の長さ
	収集時点から過去 15 分間の平均の run queue の長さ	
システムコール関連	exec()システムコールの回数	

※CPU 個別情報の収集機能はR3.1c-P1 以降のバージョンで用意されています。

カーネルパラメータの情報として以下を収集することができます。

種別	取得項目	
カーネル パラメータ 基本情報	カーネルパラメータ名(NAME)	モジュール名(MODULE)
	概要(DESC)	デフォルト値(DEFVALUE)
	OS 起動時の値(BOOTVALUE)	現在の設定値(CURRENT)
	設定可能な最小値(MIN)	設定可能な最大値(MAX)
	現在の使用量(USAGE)	

※カーネルパラメータ情報の収集機能は R3.1e 以降のバージョンで用意されています。

- 監視する項目

収集したプロセス個別の情報から、以下の状態を監視することができます。

種別	監視項目
プロセス基本情報	ゾンビプロセス（defunct）監視
	プロセスの多重度監視
メモリ関連	メモリリーク監視
ファイル関連	ファイルリーク監視
	プロセス毎のオープンファイル数上限監視（カーネルパラメータmaxfiles_lim相当）
CPU関連	高CPU使用率プロセス監視
スレッド関連	スレッドリーク監視
	プロセス毎のスレッド数上限監視（カーネルパラメータmax_thread_proc相当）

収集したシステム全体の情報から、以下の状態を監視することができます。

種別	監視項目
プロセス／ スレッド関連	プロセス数上限監視（カーネルパラメータnproc相当）
	ユーザプロセス数上限監視（カーネルパラメータmaxuprc相当）
	システム全体のスレッド数上限監視（カーネルパラメータnkthread相当）
メモリ関連	メモリ量上限監視（物理メモリ量）
	スワップメモリ量上限監視（使用可能なスワップ領域）
ファイル関連	オープンファイル数上限監視（カーネルパラメータnfile相当）
	ロックファイル数上限監視（カーネルパラメータnflocks相当）
CPU関連	CPU使用率監視（CPU数 × 100(%)）

１．３　本製品の構成について

本製品で使用するディレクトリ及びファイル構成は以下のとおりです。

（1）ディレクトリ構成

本製品は、以下のディレクトリを使用します。

・ 実行形式格納ディレクトリ	**/opt/HA/PSSM/bin/**
・ SG ファイル管理ディレクトリ	**/var/opt/HA/PSSM/conf/**
・ 実行結果管理ディレクトリ	**/var/opt/HA/PSSM/log/**
・ ユーティリティ関連ディレクトリ	**/var/opt/HA/PSSM/bin/**
・ 定期実行スクリプト管理ディレクトリ	**/var/opt/HA/PSSM/scripts/**
・ 定期実行スクリプト設定ファイル管理ディレクトリ	**/var/opt/HA/PSSM/scripts/conf/**
・ 定期実行スクリプト実行結果管理ディレクトリ	**/var/opt/HA/PSSM/scripts/log/**
・ rc 関連ディレクトリ	**/sbin/init.d/** **/sbin/rc3.d/** **/sbin/rc2.d/** **/etc/rc.config.d/**

（2）ファイル構成

本製品は、以下のファイルを使用します。

・ **/opt/HA/PSSM/bin/**

pssmd
SystemResourceMonitor 管理デーモン
psaction
アクションコマンド
psanalyzer
統計情報解析コマンド
psmonitor
統計情報収集コマンド
psscriptexec
統計情報収集スクリプト定期実行コマンド
pssmview
統計情報編集コマンド
pssmkcview
カーネルパラメータ統計情報編集コマンド
pssmdatcp
統計情報ファイルコピーコマンド
pssmkcmonitor
カーネルパラメータ情報収集コマンド

・ **/var/opt/HA/PSSM/conf/**

pssm.conf
本製品全体の動作を既定するシステム定義ファイル
psaction.conf
異常検出時の動作を既定するアクション定義ファイル
psact_notstop.conf
プロセス KILL アクション対象から除外するプロセスを定義するプロセス KILL アクション除外定義ファイル
pssm.conf.default
デフォルトのシステム定義ファイル
psaction.conf.default
デフォルトのアクション定義ファイル
psact_notstop.conf .default
デフォルトのプロセス KILL アクション除外定義ファイル

・ **/var/opt/HA/PSSM/log/**

psmonitor.dat
リソースの統計情報を記録するファイル
psanalyzer.dat
統計情報の解析結果を記録するファイル
pssmkcmonitor.csv
カーネルパラメータの統計情報を記録するファイル
pssmkcmonitor_static.csv
全カーネルパラメータの詳細情報を記録するファイル
analyzedinfo.dat
統計情報の解析が終了した日時を記録するファイル
psanalyzer.view.dat
統計情報編集コマンドの解析結果を記録するファイル
pssm_syslog.log
テキストログを記録するファイル
pssm_trace.log
内部ログを記録するファイル
pssmview_trace.log
統計情報編集コマンドの内部ログを記録するファイル
pssmkcmonitor.log
カーネルパラメータ情報収集コマンドの内部ログを記録するファイル

- **/var/opt/HA/PSSM/bin/**
  collect_info.sh
  障害解析用の情報を採取するスクリプトファイル

- **/var/opt/HA/PSSM/scripts/**
  s00get_kernelparam.sh
  全カーネルパラメータ情報定期採取スクリプト
  s01get_sardata.sh
  システム動作情報(sar コマンド)定期採取スクリプト
  s02get_ipcsdata.sh
  プロセス間通信情報(ipcs コマンド)定期採取スクリプト
  s03get_vmstatdata.sh
  仮想メモリ統計情報(vmstat コマンド)定期採取スクリプト
  s04get_glancedata.sh
  システムパフォーマンス情報(glance コマンド)
  定期採取スクリプト
  shared_lib
  定期実行スクリプト共通ライブラリ

- **/var/opt/HA/PSSM/scripts/log/**
  scripts_trace.log
  定期実行スクリプトのテキストログを記録するファイル
  scheduled_kernelparam.log
  全カーネルパラメータ情報を定期的に記録するファイル
  scheduled_sardata.log
  システム動作情報(sar コマンド)を定期的に記録する
  ファイル
  scheduled_ipcsdata.log
  プロセス間通信情報(ipcs コマンド)を定期的に記録する
  ファイル
  scheduled_vmstatdata.log
  仮想メモリ統計情報(vmstat コマンド)を定期的に記録する
  ファイル
  scheduled_glancedata.log
  システムパフォーマンス情報(glance コマンド)を定期的に
  記録するファイル

- **/sbin/init.d/**
  pssmd
  本製品の起動・停止を行うファイル

- **/sbin/rc3.d/**
  S999pssmd
  rc から自動起動するリンクファイル

- **/sbin/rc2.d/**
  K111pssmd
  rc から自動停止するリンクファイル

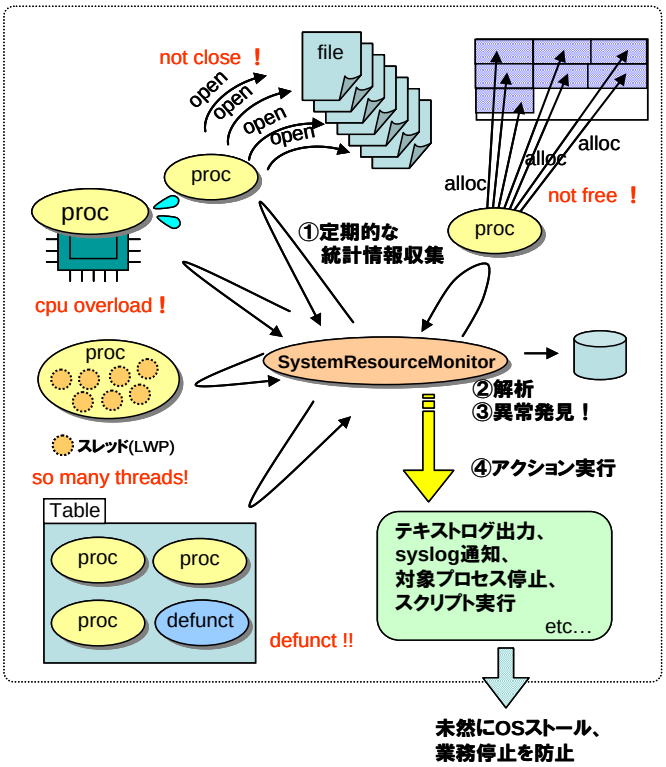
- **/etc/rc.config.d/**
  pssmconf
  rc からの自動起動の可否を設定するファイル

2 本製品の機能

2．1 本製品で提供する機能とは

(1) プロセス単位のリソース監視について

システム上で動作するプロセスの統計情報の
収集／解析を行います。
解析結果から以下の項目について、異常と
判断すると既定アクションを実行します。



- メモリリーク監視
  メモリ使用量を継続的に収集し、
  一定時間以上増加傾向にあるとき、
  メモリリークと判定します。

- ファイルリーク監視
  オープンファイル数を継続的に収集し、
  一定時間以上増加傾向にあるとき、
  ファイルリークと判定します。

- スレッドリーク監視
  プロセスが使用しているスレッド数を監視し、
  生成されたままの残留スレッドが一定時間
  以上増加傾向にあるとき、スレッドリークと
  判定します。

- 高 CPU 使用率プロセス監視
  CPU 使用率を継続的に収集し、一定時間以上連続して一定の閾値を超えた場合に
  高 CPU 使用率プロセス監視異常と判定します。

- プロセス多重度監視
  同名プロセスがどれだけ存在するか(多重度)を監視し、閾値を超えた場合にプロセス多重度監視異常
  と判定します。

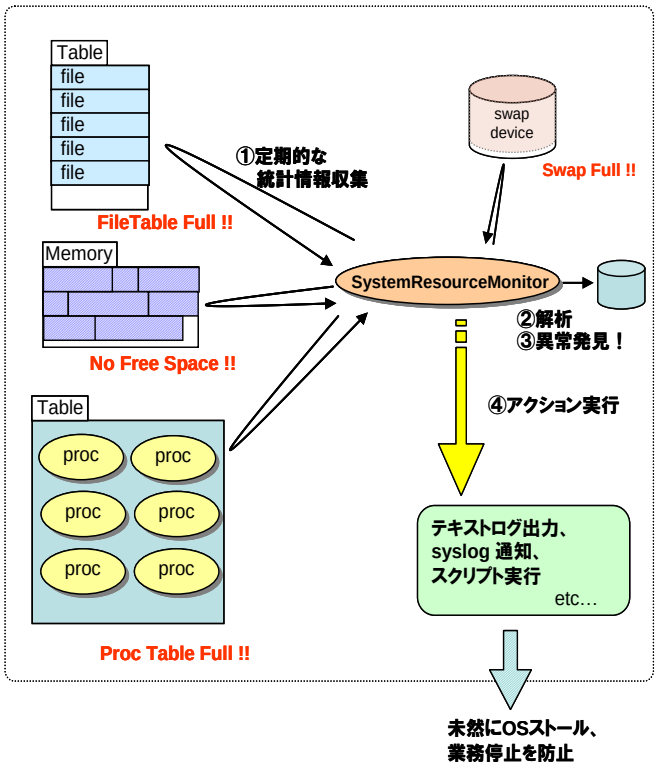
- ゾンビプロセス（defunct）監視
  プロセスのステータスが、指定した一定時間以上連続してゾンビ状態（defunct）であった場合に、
  ゾンビプロセス（defunct）監視異常と判定します。

- オープンファイル上限監視
  オープンファイル数を監視し、一定の閾値を超えた場合にオープンファイル上限監視異常と
  判定します。

- スレッド数上限監視
  スレッド数を監視し、一定の閾値を超えた場合にスレッド数上限監視異常と判定します。

(2)　システム単位のリソース監視について

システム上で動作するプロセス全体で利用している
リソースの統計情報の収集／解析を行います。
解析結果から以下の項目について、異常と
判断すると既定アクションを実行します。



- システム全体のリソース上限監視
  以下のシステム全体のリソースを監視し、
  指定した一定時間以上連続して、一定の閾値を
  超えた場合にカーネルリソース上限監視異常と
  判定します。
  - 総オープンファイル数
  - 総起動プロセス数
  - 総ロックファイル数
  - ユーザ毎の起動プロセス数
  - 総スレッド数

- メモリ使用量上限監視
  総メモリ使用量が、指定した一定時間以上連続して、
  一定の閾値を超えた場合にメモリ使用量上限監視
  異常と判定します。

- CPU 使用率上限監視
  CPU 使用率が、指定した一定時間以上連続して、
  一定の閾値を超えた場合に CPU 使用率上限監視
  異常と判定します。

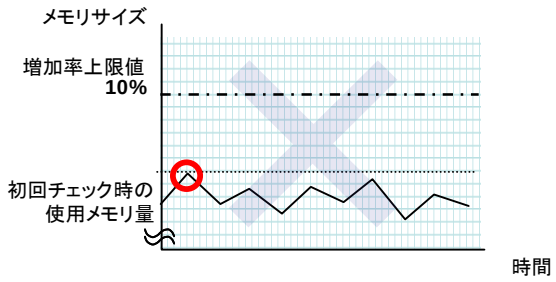
- スワップメモリ量上限監視
  総スワップメモリ使用量および総スワップメモリ予約量が、
  指定した一定時間以上連続して、一定の閾値を
  超えた場合にスワップメモリ使用量上限監視異常と判定します。

(3)　リソース監視異常判定方式の概要

①　プロセスリソース監視の判定は以下の方法によって行います。

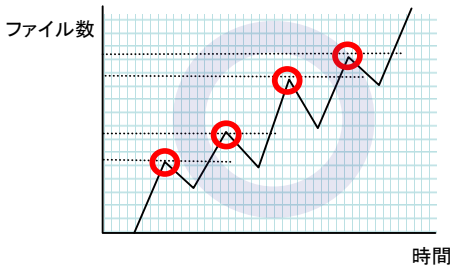
- メモリリーク監視、ファイルリーク監視およびスレッドリーク監視
  メモリ使用量、オープンファイル数、スレッド数を継続的に収集し、前回の数値と比較して増加していた場合に比較する値を更新します。
  更新回数が一定時間以上増加した場合にメモリリーク、ファイルリークおよびスレッドリークを検出します。

[リソース監視異常を検出しない]



メモリ使用量は一定の値を超えない範囲で増減しているため、更新回数はカウントされない。

[リソース監視異常を検出する]



オープンファイル数が規定回数を超えて更新されているため、ファイルリークと判定。

- 高 CPU 使用率プロセス監視
  CPU 使用率を継続的に収集し、一定時間以上連続して CPU 使用率の上限を超えていた場合に高 CPU 使用率プロセス監視異常と判定します。

- ゾンビプロセス（defunct）監視
  プロセスのステータスが一定時間以上連続してゾンビ状態（defunct）であった場合に、ゾンビプロセス（defunct）監視異常と判定します。

- オープンファイル数上限監視、スレッド数上限監視
  プロセスが使用しているオープンファイル数、スレッド数が閾値の一定のパーセンテージを超えていた場合に、オープンファイル数上限異常、スレッド数上限異常と判定します。

- プロセス多重度監視
  プロセスの多重度（同じプロセス名のプロセスが複数稼動している状態）が一定の閾値を超えていた場合に、プロセス多重度監視異常と判定します。

② システムリソース監視の判定は以下の方法によって行います。

- カーネルリソース上限監視
  各カーネルリソース（総オープンファイル数、総プロセス数、総ロックファイル数、ユーザごとの起動プロセス数、総スレッド数）の使用量を継続的に収集し、カーネルパラメータで設定される閾値の一定のパーセンテージを、一定時間以上連続して超えていた場合に、カーネルリソース上限監視異常と判定します。

- メモリ使用量上限監視
  総メモリ使用量が総メモリ使用量上限（物理メモリ量）の一定のパーセンテージを、一定時間以上連続して超えていた場合にメモリ使用量上限監視異常と判定します。

- CPU 使用率上限監視
  CPU 使用率が CPU 使用率上限（100(%)×CPU 数）の一定のパーセンテージを、一定時間以上連続して超えていた場合に CPU 使用率上限監視異常と判定します。

- スワップメモリ使用量上限監視
  総スワップメモリ使用量または総スワップメモリ予約量が総スワップメモリ使用量上限（使用可能なスワップ領域）の一定のパーセンテージを、一定時間以上連続して超えていた場合にスワップメモリ使用量上限監視異常と判定します。
  また、総スワップメモリ使用量と総スワップメモリ予約量の合計が総スワップメモリ使用量上限（使用可能なスワップ領域）の一定のパーセンテージを、一定時間以上連続して超えていた場合にもスワップメモリ使用量上限監視異常と判定します。

(4)　リソース監視異常通知方式

本製品のリソース監視異常通知にはテキストログ出力及びシステムログ出力による２種類の方式があります。
デフォルトはテキストログ出力のみとなっています。

① テキストログの設定

プロセスリソース監視の異常検出メッセージを、テキストログに出力します。
テキストログの構成は以下のとおりです。

ディレクトリ：**/var/opt/HA/PSSM/log/**
ファイル　　：**pssm_syslog.log**

ファイルのサイズは5MByte で、1 世代バックアップ（OLDpssm_syslog.log）されます。
テキストログは必ず出力され、設定の変更はできません。
バックアップファイルは上書きで更新されるため、更新ごとに前回のファイル内容が削除されます。

② システムログの設定

設定に応じて、システムログへ出力することも可能です。
システムログ通知を行う場合は、以下の手順で SG ファイルを変更してください。

/var/opt/HA/PSSM/conf/ 配下の **psaction.conf** をエディタ等で編集します。

・出力の対象となる監視項目の SYSLOG_REPORT 行のコメント（#）を外してください。
・メッセージの出力レベルを選択することも可能です。デフォルトは WARNING です。

以上で、システムログ通知の設定は終了です。
設定ファイルを変更後は、必ず本製品の再起動を行ってください。
再起動を行わない場合、SG ファイルの変更内容は反映されませんのでご注意ください。

（設定例）ゾンビプロセス検出時に WARNING レベルでシステムログ通知を行います。

DEFUNCT_ERROR_ACTION　SYSLOG_REPORT : WARNING

(5)　統計情報収集スクリプトの定期的な実行

本製品では、プロセスリソース監視、システムリソース監視、カーネルパラメータ情報の収集に加えて、ユーザが任意に作成した情報収集スクリプトを定期的に実行し、統計情報として収集する機能を提供します。

この機能を利用し、得られた統計情報を分析することで、システム状態を把握し、将来発生しうるリソース不足などの障害の未然検出やリソース増設の必要性の判断材料などに利用できます。

本製品は、以下の条件に合致するスクリプトを設定した時間間隔毎に実行します。

- スクリプト配置ディレクトリ

  /var/opt/HA/PSSM/scripts 配下

- スクリプト命名規則

  S（大文字）＋ 二桁の数字で始まるファイル名を持つスクリプトファイル

  例）S00getinfo.sh、S99disk_workload.sh など

スクリプトファイルは最大 100 個まで指定可能です。
また、スクリプトファイルの実行順序は S の後の二桁の数字が 00 から 99 まで順番に実行されますので、この数字を任意に変更することで制御できます。

ファイル名が大文字 S で開始していないスクリプトファイルは実行されません。また、二桁の数字が付与されていない場合は正しく動作しない場合があります。

実行間隔は SG ファイルにて変更可能です。デフォルト値は 1 日になっています。
設定値の詳細は後述の SG ファイルの説明を参照してください。

デフォルトでは、統計情報収集スクリプトは設定されておりませんが、本製品ではあらかじめ統計情報収集スクリプトを用意していますので、簡単にスクリプトの定期実行機能を利用することができます。
本製品規定の統計情報収集スクリプトの使用方法や設定手順については、「 7 章 付録」を参照してください。

(6)　カーネルパラメータ情報の収集について

カーネルパラメータの収集方法は2種類あります。

① カーネルパラメータ統計情報の収集方法
オーバーフローする可能性があるカーネルパラメータの使用量を定期的に収集します。
OS バージョンの違いにより、収集する情報は以下のとおりになります。

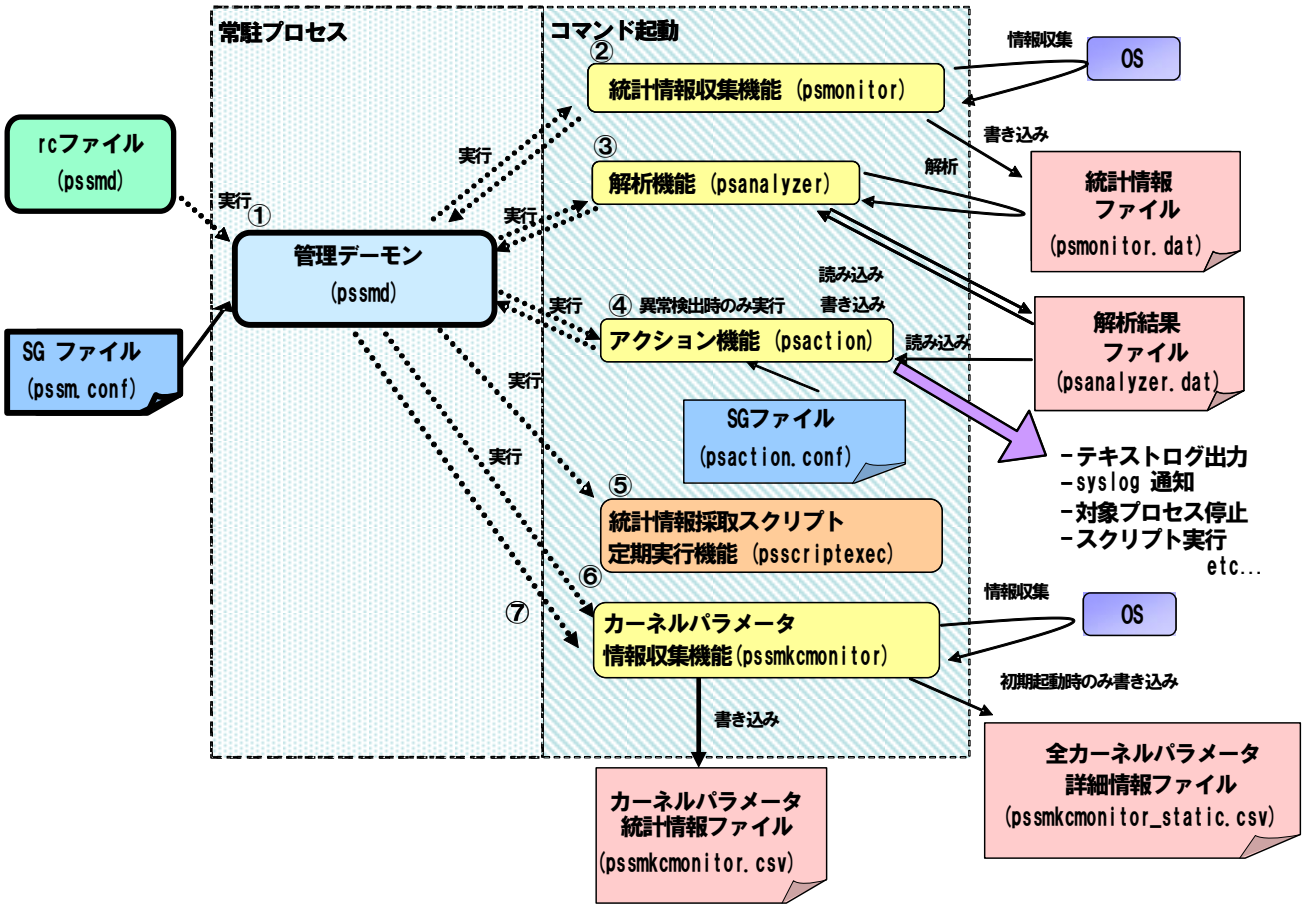
カーネルパラメータ名	OS Version ○：収集する ×：収集しない	
	B.11.23	B.11.31
dbc_max_pct	○	×
filecache_max	×	○
maxdsiz	○	○
maxdsiz_64bit	○	○
maxssiz	○	○
maxssiz_64bit	○	○
maxtsiz	○	○
maxtsiz_64bit	○	○
maxuprc	○	○
max_thread_proc	○	○
maxvgs	○	×
msgmbs	×	○
msgmni	○	○
msgseg	○	×
msgtql	○	○
nfile	○	×
nflocks	○	○
ninode	○	○
nkthread	○	○
nproc	○	○
npty	○	○
nswapdev	○	○
nswapfs	○	○
semmni	○	○
semmns	○	○
shmmni	○	○
shmseg	○	○
vx_ninode	○	○

② 全カーネルパラメータ詳細情報の収集方法
本製品を起動したときに一度だけ全カーネルパラメータ情報の収集を行います。
※　使用量の収集は行いません。

(7)　本製品の動作概要

本製品の動作フローは以下のようになります。

① 管理デーモンプロセス（**pssmd**）を起動します。
② **pssmd** デーモンが一定間隔で **psmonitor** コマンドを実行し、プロセスの利用するリソースの統計情報を継続的に収集し、統計情報ファイル（**psmonitor.dat**）に書き出します。
統計情報ファイル（**psmonitor.dat**）はデフォルトで 10 世代までバックアップされます。
③ **pssmd** デーモンが **psanalyzer** コマンドを実行し、収集した統計情報を一定間隔で解析し、解析結果を解析結果ファイル（**psanalyzer.dat**）へ書き込みます。
④ 解析の結果、リソース監視異常と判定すると、**psaction** コマンドを実行し異常検出の通知および SG ファイルで指定した既定処理（対象プロセス停止等）が実行されます。
⑤ 統計情報採取用の任意のスクリプトを定期的に実行することができます。
⑥ **pssmd** デーモンが起動時に **pssmkcmonitor** コマンドを実行し、全カーネルパラメータの詳細情報を一度だけ収集し、全カーネルパラメータ詳細情報ファイル（**pssmkcmonitor_static.csv**）に書き出します。全カーネルパラメータ詳細情報ファイル（**pssmkcmonitor_static.csv**）は、1 世代のみバックアップします。
⑦ **pssmd** デーモンが一定間隔で **pssmkcmonitor** コマンドを実行し、カーネルパラメータの統計情報を継続的に収集し、カーネルパラメータ統計情報ファイル（**pssmkcmonitor.csv**）に書き出します。
カーネルパラメータ統計情報ファイル（**pssmkcmonitor.csv**）は、デフォルトで 3 世代までバックアップします。



２．２　本製品の導入手順

本製品の導入手順は以下のようになります。

(1)　インストール

インストール方法については、同梱のリリースメモを参照してください。

(2)　SG の初期設定

基本的に SG 設定はデフォルト値のままご利用頂けます。
監視ポリシーを変更する場合は「 4 章 SG の設定」を参照してください。

(3)　起動

１．コマンドにより起動します。

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

２．下記コマンドにより本製品が正しく起動されたことを確認します。

```
# ps -ef | grep pssmd
root     481       1   0 15:41:35 pts/tc       0:00    /opt/HA/PSSM/bin/pssmd
```

（注）マシンの再起動は不要です。

以上で本製品の導入は完了です。

（注）本製品をデフォルトで運用した場合、リソース監視異常検出時のメッセージはテキストログにのみ出力され、システムログには出力されません。
リソース監視異常検出時のメッセージをシステムログへ出力するためには SG ファイルの内容を変更する必要があります。
設定方法については「 4 章 SG の設定」を参照してください。

２．３　rc からの自動起動について

rc からの自動起動は、インストール時にデフォルトで設定されます。
自動起動を停止する場合は、以下の手順を行ってください。

(1)　設定ファイルの変更

１．設定ファイル（**/etc/rc.config.d/pssmconf**）をエディタ等で以下のように変更してください。

PSSMD_START の値を 1 から 0 に変更します。

```
PSSMD_START=0
```

上記設定変更により、rc からの自動起動は停止されます。

3 本製品の運用手順

３．１　本製品の起動・停止

本製品の運用手順は以下のようになります。

（1） 起動

１．コマンドにより起動します。

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

２．下記コマンドにより本製品が正しく起動されたことを確認します。

```
# ps -ef | grep pssmd
root     481       1   0 15:41:35 pts/tc       0:00 /opt/HA/PSSM/bin/pssmd
```

（2） 停止

１．コマンドにより停止します。

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

２．下記コマンドにより正しく停止されたことを確認します。

```
# ps -ef | grep pssmd
```

停止されていれば、何も表示されません。

（注） 本製品による処理を実行中の場合は、処理の完了を待ち合わせるため、
停止にしばらく時間がかかる場合があります。

（3） 再起動

本製品を再起動する場合は、下記コマンドを実行してください。
（再起動には数秒かかります。）

```
# /sbin/init.d/pssmd restart
```

（注）各 SG ファイルの値を変更した後は、本製品を再起動する必要があります。
再起動を行わない場合、SG ファイルの変更内容は反映されません。

３．２　本製品によって作成されるファイル

本製品を起動すると**/var/opt/HA/PSSM/log/** 配下に下記のファイルが作成されます。

psmonitor.dat	（統計情報ファイル）
psanalyzer.dat	（解析結果ファイル）
pssmkcmonitor.csv	（カーネルパラメータ統計情報ファイル）
pssmkcmonitor_static.csv	（全カーネルパラメータ詳細情報ファイル）
pssm_syslog.log	（テキストログファイル）
pssm_trace.log	（内部ログファイル）
pssmkcmonitor.log	（カーネルパラメータ収集内部ログファイル）

（1）統計情報ファイルでは以下の情報を得ることができます。

指定時間間隔（デフォルトでは5分）で収集したプロセスの利用するリソースの統計情報、
システム全体のリソースの統計情報、CPU 個別のリソースの統計情報を継続的に記録します。
SG ファイルで指定されたサイズ（デフォルトでは30MByte）のファイルを
指定されたバックアップ数分（デフォルトでは10個）保存されます。
統計情報ファイルは以下のファイル名で保存されます。

psmonitor.dat,　psmonitor.dat.save1,　psmonitor.dat.save2　・・・

統計情報ファイルの出力先は、必要に応じて任意のディレクトリに変更が可能です。
設定の詳細については「 4 章 SG の設定」を参照してください。
（注）ファイル名を変更することはできません。

[プロセス個別情報]

１．　ユーザ名（UNAME）
２．　プロセス ID（PID）
３．　親プロセス ID（PPID）
４．　プロセスグループ ID（PGID）
５．　プロセスステータス（STATUS）
６．　メモリ使用量（MEMORY）
７．　オープンファイル数（FILE）
８．　ロックファイル数（LOCKF）
９．　プロセス毎のオープンファイル数上限に対する割合（%FILE）
１０．CPU 使用率（CPU）
１１．1CPU に対する使用率（CPU/CPU 数）
１２．CPU 使用時間（CPUTIME）
１３．プロセスの多重度（PNUM）
１４．プロセス毎のスレッド数（THREAD）
１５．プロセス毎のスレッド数上限値に対する割合（%THREAD）
１６．プロセス名（COMMAND）

[システム情報]

1. ホスト名
2. 情報取得日時
3. 総情報取得回数
4. 現在の総プロセス数
5. 現在の総メモリ使用量
6. 現在の総オープンファイル数
7. 現在の総ロックファイル数
8. 現在の CPU 使用率
9. コンテキストスイッチの回数
10. 現在の総スレッド数
11. 現在の総スワップメモリ使用量
12. 現在の総スワップメモリ予約量
13. 現在のユーザごとの起動プロセス数最大値（プロセス起動数が最大のユーザ名）

[CPU 個別情報]

1. CPU 番号
2. 収集時点から過去 1 分間の平均の run queue の長さ
3. 収集時点から過去 5 分間の平均の run queue の長さ
4. 収集時点から過去 15 分間の平均の run queue の長さ
5. exec()システムコールの回数

統計情報ファイルのフォーマットは以下のとおりです。

```
================================================================================
HOST    : [ホスト名]
DATE    : [情報取得日時]
COUNT : [情報取得回数]
INDEX  :
   PROCESS   [プロセス個別情報 INDEX]
   SYSTEM   [システム情報 INDEX]
--------------------------------------------------------------------------------
PROCESS   [ユーザ名]  [プロセス ID]  [親プロセス ID]  [プロセスグループ ID]  [プロセスステータス]  [メモリ使用量]
[オープンファイル数]  [ロックファイル数]  [オープンファイル数上限に対する割合]  [CPU 使用率]
[1CPU に対する CPU 使用率]  [CPU 使用時間]  [プロセス多重度]  [スレッド数]  [スレッド数上限に対する割合]
[プロセス名]
--------------------------------------------------------------------------------
SYSTEM   [現在の総プロセス数]  [総プロセス数上限]  [現在の総メモリ使用量]  [総メモリ使用量上限]
[現在の総オープンファイル数]  [総オープンファイル数上限]  [現在の総ロックファイル数]  [総ロックファイル数上限]
[現在の CPU 使用率]  [CPU 使用率上限]  [コンテキストスイッチの回数]  [現在の総スレッド数]  [総スレッド数上限]
[現在の総スワップメモリ使用量]  [現在の総スワップメモリ予約量]  [総スワップメモリ量上限]
[現在のユーザごとの最大起動プロセス数]  [ユーザごとのプロセス起動数上限]   [最大起動プロセス数のユーザ名]
--------------------------------------------------------------------------------
CPU   [CPU 番号]  [収集時点から過去 1 分間の平均の run queue の長さ]
[収集時点から過去 5 分間の平均の run queue の長さ]  [収集時点から過去 15 分間の平均の run queue の長さ]
[exec()システムコールの回数]  [参照情報 1][1]  [参照情報 2][1]
================================================================================
```

以下は統計情報ファイルの例です。(**/var/opt/HA/PSSM/log/psmonitor.dat**)

```
================================================================================
HOST   : host1                              ← ホスト名
DATE   : 2008/04/19 14:27:32                ← 情報取得日時
COUNT : 1                                   ← 情報取得回数
INDEX :                                     ← 出力情報 INDEX
 PROCESS UNAME:16 PID:6 PPID:6 PGID:6 STATUS:8 MEMORY:10 FILE:6 LOCKF:6 %FILE:6 %CPU: ・・・
 SYSTEM TOTAL_PROC_NUM:8 PROC_NUM_MAX:8 TOTAL_PROC_MEM:8 TOTAL_MEM_MAX:8 TOTAL_OPEN_FILE:8 ・・・
--------------------------------------------------------------------------------
TYPE UNAME PID PPID PGID STATUS MEMORY(KB) FILE LOCKF %FILE %CPU %CPU/2 CPUTIME PNUM THREAD %THREAD (COMMAND)
PROCESS root 0 0 0 SLEEP 64 0 0 0.00 0.02 0.01 0:17 1 1 0.09 (swapper)      ← プロセス個別情報
PROCESS root 1 0 0 RUNNING 656 0 0 0.00 0.05 0.02 0:00 1 2 0.18 (init)
               : :                         : :                 : :
--------------------------------------------------------------------------------
TYPE PROCESS MEMORY(MB) FILE LOCKFILE CPU(%) CSWITCH THREAD SWAP(KB) MAXUPROC
SYSTEM 211 4200 1963 2037 5094 NOVALUE 24 4096 21.62 200 97959439 617 8416 253100 770072 4194304 3 256 guest1  ← システム情報
--------------------------------------------------------------------------------
TYPE CPUID RUNQ_1MIN RUNQ_5MIN RUNQ_15MIN SYSCALL SUBINF01 SUBINF02
CPU 0 0.0286 0.0377 0.0421 998244 92758 79122                          ← CPU 個別情報
CPU 1 0.0242 0.0366 0.0407 996580 92791 78606
================================================================================
```

[1] 内部処理で使用する情報のため、特に意識する必要はありません。

（2）解析結果ファイルでは以下の解析結果を得ることができます。

指定時間間隔（デフォルトでは 60 分）で統計情報を解析します。
最新の解析結果は **psanalyzer.dat** に、前回の解析結果は **psanalyzer.dat.bak** として保存されます。
本ファイルは特に意識して確認する必要はありません。

[システム解析]

1．総プロセス数上限監視（カーネルパラメータの nproc 相当）
2．総メモリ使用量上限監視
3．総オープンファイル数上限監視（カーネルパラメータの nfile 相当）
4．総ロックファイル数上限監視（カーネルパラメータの nflocks 相当）
5．CPU 使用率上限監視（100(%)×CPU 数）
6．総スワップメモリ量上限監視（使用可能なスワップ領域）
7．ユーザプロセス数上限監視（カーネルパラメータの maxuprc 相当）
8．総スレッド数上限監視（カーネルパラメータ nkthread 相当）

[プロセス個別解析]

1．メモリリーク検出
2．ファイルリーク検出
3．ゾンビプロセス検出
4．高 CPU 使用率プロセス検出
5．スレッドリーク検出
6．プロセス毎のスレッド数上限監視
（カーネルパラメータ max_thread_proc 相当）
7．プロセス毎のオープンファイル数上限監視
（カーネルパラメータ maxfiles_lim 相当）
8．プロセス多重度監視

以下は解析結果ファイルの例です。（**/var/opt/HA/PSSM/log/psanalyzer.dat**）

```
INDEX{
SYSTEMINFO{
PROC,PROC_ERR_FLAG,OLD_PROC_ERR_FLAG,PROC_NUM_MAX,TOTAL_PROC_NUM,PROC_NUM_RATE,SYSPROC_,…
}
PROCINFO{
ERR_FLAG,OLD_ERR_FLAG,PID,DEFUNCT_COUNT,MEMSIZE,MAX_MEM_SIZE,FIRST_MEM_SIZE,MEM_OVER_COUNT, …
}
}
SYSTEMINFO{
PROC,0,0,15000,206,1,90
MEM,0,0,2032,304,14,90
FILE,0,0,65048,2889,4,90
LOCKF,0,0,4200,5,14,90
CPU,0,0,200,5.47,2,90
THREAD,0,0,16000,495,3,90
SWAP,0,0,4194304,857908,656736,20,15,36,90
MAXUPROC,0,0,256,6,2,90,guest1
}
PROCINFO{
0,0,0,0,64,64,64,0,0,0,0,0,0,1,1,1,1,0,(swapper)
0,0,1,0,544,544,544,0,0,0,0,0,0,1,2,2,2,0,(init)
0,0,2,0,64,64,64,0,0,0,0,0,0,1,1,1,1,0,(vhand)
                : :                    : :
}
```

（3）カーネルパラメータ統計情報ファイルでは以下の情報を得ることができます。

指定時間間隔（デフォルトでは 10 分）で収集したカーネルパラメータ統計情報を継続的に記録します。SG ファイルで指定されたサイズ（デフォルトでは 5MByte）のファイルを指定されたバックアップ数分（デフォルトでは 3 個）Z 形式で圧縮した状態で保存します。
カーネルパラメータ統計情報ファイルは以下のファイル名で保存します。

pssmkcmonitor.csv,　　pssmkcmonitor.csv.YYYYMMDDhhmmss.Z, ･･･

バックアップファイルは **pssmkcmonitor.csv.YYYYMMDDhhmmss.Z** の形式で保存されます。
YYYYMMDDhhmmss にはバックアップファイルが作成された年月日と時刻が設定されます。

例）バックアップファイルが 2010 年 9 月 1 日 12 時 00 分 00 秒に作成された場合
pssmdiskmonitor.csv.20100901120000.Z

バックアップファイルは、/usr/bin/compress を利用して Z 形式で圧縮されています。
展開する場合には、以下のコマンドを利用してください。

/usr/bin/uncompress　pssmdiskmonitor.csv.YYYYMMDDhhmmss.Z
または、
/usr/bin/compress　-d　pssmdiskmonitor.csv.YYYYMMDDhhmmss.Z

バックアップファイルが SG ファイルの設定値の個数より多くなった場合は、日付が一番古いファイルから削除されます。そのため、マシン時刻を変更した場合は、バックアップファイルを削除する順番が前後する場合があります。

カーネルパラメータ統計情報ファイルの出力先は、必要に応じて任意のディレクトリに変更が可能です。設定の詳細については「 4 章 SG の設定」を参照してください。
（注）ファイル名を変更することはできません。

[カーネルパラメータ情報]

1．　カーネルパラメータ名（NAME）
2．　モジュール名（MODULE）
3．　概要（DESC）
4．　デフォルト値（DEFVALUE）
5．　OS 起動時の値（BOOTVALUE）
6．　現在の設定値（CURRENT）
7．　設定可能な最小値（MIN）
8．　設定可能な最大値（MAX）
9．　現在の使用量（USAGE）

カーネルパラメータ統計情報ファイルのフォーマットは以下のとおりです。

```
================================================================================
HOST,[ホスト名]
OSVERSION,[OS バージョン情報]
DATE,[情報取得日時]
TICKTIME, [情報取得日時]
--------------------------------------------------------------------------------
[カーネルパラメータ名],[モジュール名],[概要],[デフォルト値],[OS 起動時の値],[現在の設定値],[設定可能な最小値],
[設定可能な最大値],[現在の使用量]
================================================================================
```

以下はカーネルパラメータ統計情報ファイルの例です。（**/var/opt/HA/PSSM/log/pssmkcmonitor.csv**）

```
================================================================================
==========================
HOST,host1
OSVERSION,B.11.31
DATE,2010/09/01 10:43:49
TICKTIME,1283305429
--------------------------------------------------------------------------------
--------------------------
NAME,MODULE,DESC,DEFVALUE,BOOTVALUE,CURRENT,MIN,MAX,USAGE
filecache_max,fs_bufcache,"Maximum amount of physical memory to be used for caching file I/O
data",1013227520,1013227520,1013227520,,,133263360
maxdsiz,vm,"Maximum size of the data segment of a 32-bit process
(bytes)",1073741824,1073741824,1073741824,262144,4294963200,46415872
              : :                    : :
```

（4）全カーネルパラメータ詳細情報ファイルでは以下の情報を得ることができます。

本製品の初回起動時に一度だけ全カーネルパラメータ詳細情報を記録します。
バックアップファイルは1 世代のみ保存します。
全カーネルパラメータ詳細情報ファイルは以下のファイル名で保存されます。
pssmkcmonitor_static.csv, pssmkcmonitor_static.csv.bak
カーネルパラメータ統計情報ファイルの出力先は、必要に応じて任意のディレクトリに
変更が可能です。設定の詳細については「 4 章 SG の設定」を参照してください。
（注）ファイル名を変更することはできません。

[カーネルパラメータ情報]

1. カーネルパラメータ名（NAME）
2. モジュール名（MODULE）
3. 概要（DESC）
4. デフォルト値（DEFVALUE）
5. OS 起動時の値（BOOTVALUE）
6. 現在の設定値（CURRENT）
7. 設定可能な最小値（MIN）
8. 設定可能な最大値（MAX）
9. 現在の使用量（USAGE）※ 使用量の収集は行わないため、常に0 になります。

カーネルパラメータ統計情報ファイルのフォーマットは以下のとおりです。

```
==========================================================================================
HOST,[ホスト名]
OSVERSION,[OS バージョン情報]
DATE,[情報取得日時]
TICKTIME, [情報取得日時]
------------------------------------------------------------------------------------------
[カーネルパラメータ名],[モジュール名],[概要],[デフォルト値],[OS 起動時の値],[現在の設定値],[設定可能な最小値],
[設定可能な最大値],[現在の使用量]
==========================================================================================
```

以下は全カーネルパラメータ詳細情報ファイルの例です。(**/var/opt/HA/PSSM/log/pssmkcmonitor_static.csv**)

```
================================================================================
========================
HOST,host1
OSVERSION,B.11.31
DATE,2010/09/01 10:43:47
TICKTIME,1283305427
--------------------------------------------------------------------------------
------------------------
NAME,MODULE,DESC,DEFVALUE,BOOTVALUE,CURRENT,MIN,MAX,USAGE
maxtsiz_64bit,vm,"Maximum size of the text (code) segment of a 64-bit process
(bytes)",1073741824,1073741824,1073741824,262144,4398046511103,0
maxtsiz,vm,"Maximum size of the text (code) segment of a 32-bit process
(bytes)",100663296,100663296,100663296,262144,1073741824,0
              : :                     : :
```

（5）ログファイルでは以下の情報を得ることができます。

本製品には、3 種類のログファイルがあり、リソース監視異常検出時のメッセージやコマンドの実行結果等を出力します。
設定に応じてシステムログへのメッセージ出力も可能ですが、テキストログ出力の可否についての変更はできません。

① **pssm_syslog.log**（テキストログファイル）
pssm_syslog.log はリソース監視異常検出時のメッセージ及び、システムログ出力レベルのメッセージをシステムログと同じ表示形式で出力します。
リソース異常検出時のメッセージをシステムログへ出力しない場合に、本ファイルを使用してメッセージ監視を行うことが可能です。

pssm_syslog.log は、5MByte で 1 世代バックアップ（**OLDpssm_syslog.log**）されます。
更新ごとに前回のバックアップファイルは上書きされ、削除されます。
フォーマットは以下のとおりです。

[検知日時] [ホスト名] [コマンド名[PID]] [メッセージ]

以下は、テキストログファイルの例です。（**/var/opt/HA/PSSM/log/pssm_syslog.log**）

```
Thu Apr  3 23:39:27 2008 host1 psaction[17205]: Find a sign of process resource failure
(type=fileleak   ,procA   ,pid=2443)
Thu Apr  3 23:39:27 2008 host1 psaction[17205]: Script execute
(/var/opt/HA/PSSM/bin/collect_info.sh)
Fri Apr  4 13:13:56 2008 host1 psaction[25874]: Find a sign of process resource failure
(type=cpu   ,value=7.72   ,maxvalue=200)
Fri Apr  4 13:13:56 2008 host1 psaction[25874]: Find a sign of process resource failure
(type=nkthread   ,value=2147463568   ,maxvalue=2147459472)
Fri Apr  4 13:13:56 2008 host1 psaction[25874]: Find a sign of process resource failure
(type=swchunk   ,value=61280   ,maxvalue=4194304)
```

（注）pssm_syslog.log は R3.1a 以降のバージョンで用意されています。R3.1 以前のバージョンの場合は、pssm_trace.log でメッセージの確認を行ってください。

② **pssm_trace.log**（内部ログファイル）
pssm_trace.log は内部ログファイルです。
pssm_trace.log は実行されたコマンドの結果及び、リソース監視異常検出時のメッセージを出力します。本ファイルは特に意識して確認する必要はありません。
内部ログ出力の可否についての変更はできません。

pssm_trace.log は、30MByte で 1 世代バックアップ（**pssm_trace.log.save1**）されます。
更新ごとに前回のバックアップファイルは上書きされ、削除されます。

③ **pssmkcmonitor.log**（カーネルパラメータ情報収集内部ログファイル）
pssmkcmonitor.log は内部ログファイルです。
pssmkcmonitor.log は実行されたコマンドの結果を出力します。
本ファイルは特に意識して確認する必要はありません。
内部ログ出力の可否についての変更はできません。

pssmkcmonitor.log は、30MByte で 1 世代バックアップ（**pssmkcmonitor.log.save1**）されます。更新ごとに前回のバックアップファイルは上書きされ、削除されます。

３．３　本製品がリソース監視異常と判断する条件

（1）　プロセスリソース監視の検出パターン

本製品では、「最大値更新回数」、「増加率」という２つのパラメータを組み合わせて検出を行います。
ファイルリークは最大値更新回数が閾値を超えた場合に検出します。
メモリリークは最大値更新回数が閾値を超え、増加率も閾値以上の場合に検出します。
ゾンビプロセス（defunct）は最大値更新回数が連続して閾値を超えた場合に検出します。
CPU 使用率の高いプロセスは、CPU 使用率上限値を超過した回数が連続して閾値を超えた場合に検出します。
パラメータの意味とデフォルトの閾値は以下のようになります。

パラメータの意味	閾値（デフォルト）
初期値からの増加率	10%
最大値を更新した回数（増加回数）/ CPU 使用率上限値を超えた回数	288回
CPU 使用率上限値	90%
モニタ間隔	5分

常にメモリ、オープンファイル数が増加するプロセス、常にゾンビプロセス（defunct）状態のプロセスおよび常に CPU 使用率上限値を超えているプロセスの場合、デフォルト値（モニタ間隔 5 分、閾値 288 回）で運用すると、約 1 日でこれらのリソース監視異常を検出します。
計算式は以下のとおりです。

統計情報を収集する間隔のデフォルト（モニタ間隔）＝ 5 分
24(時間)＊60(分)／5(分) ＝ 288(回)

上記の設定（デフォルト）で運用した場合の検出パターンは以下のようになります。

	メモリリーク監視		ファイルリーク／スレッドリーク監視	ゾンビプロセス（defunct）監視	高CPU 使用率プロセス監視
増加率 / 最大値更新回数	10%未満	10%以上	条件なし	条件なし	条件なし
288回未満	×	×	×	×	×
288回以上	×	○	○	○	○

※「○」･･･リソース監視異常を検出します。
「×」･･･リソース監視異常を検出しません。

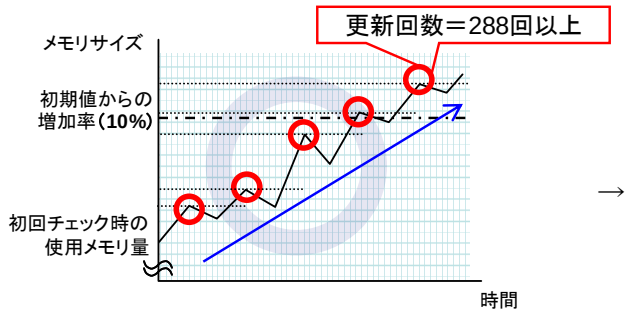
（2）プロセスリソース監視の判定条件

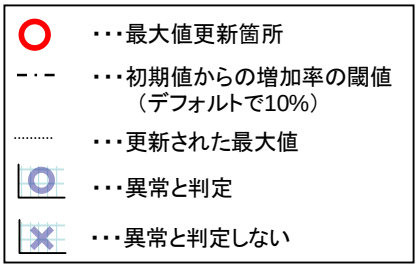
プロセスリソース監視の判定は以下の方法によって行います。

- メモリリーク監視
  ① 初回チェック時のメモリ使用量を最大メモリサイズに設定します。
  ② 現在のメモリ使用量と前回のメモリ使用量を比較し、前回より増加していれば最大メモリサイズと比較します。
  ③ 最大メモリサイズよりも現在のメモリ使用量の方が大きい場合、最大メモリサイズを現在のメモリ使用量に更新し、更新回数をカウントします。
  ④ 最大メモリサイズが既定された回数（デフォルトでは 288 回（=1 日））を超えて更新された場合、初回チェック時からの増加率と増加率上限値（デフォルトでは初回チェック時の 10%増）と比較し、増加率上限値を超えている場合にメモリリークと判定します。
  ※ 増加率は以下の計算式によって算出します。
  増加率 ＝（現在のメモリ使用量 - 初回のメモリ使用量）/ 初回のメモリ使用量 ＊ 100

次に、デフォルト値で運用した場合のメモリリークの具体的な例を示します。

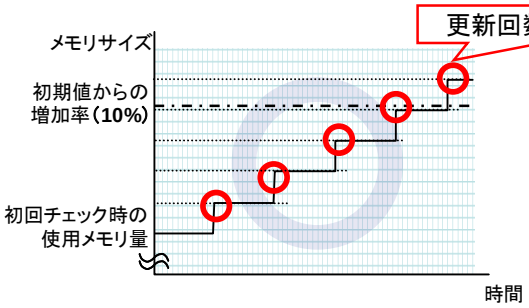
● **メモリ使用量が経過時間と共に増減しながら、規定回数以上最大値を更新し、増加率が初期値の 10%を超えた。**





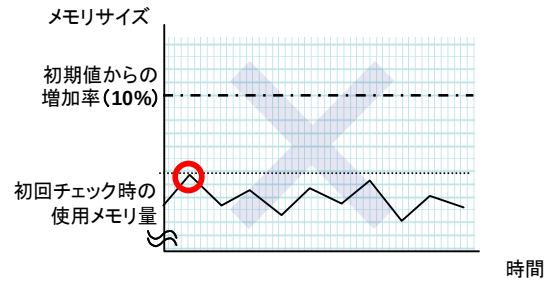
→ 最大値更新回数が 288 回（デフォルト）を超え、増加率も初期値の 10%を上回っているため、メモリリークと判定する。

● **メモリ使用量があるタイミングごと（ユーザ、サービス等を追加時）に一定量増加**



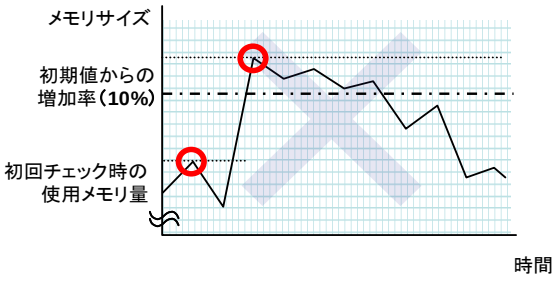
→ ユーザ、サービス等の追加により、一定のメモリ使用量が増加し、そのつど最大値更新回数はカウントされる。
最大値更新回数が 288 回（デフォルト）を超え、増加率も初期値の 10%以上を上回っているため、メモリリークと判定する。

● **メモリ使用量が経過時間と共に一定の範囲内で増減**



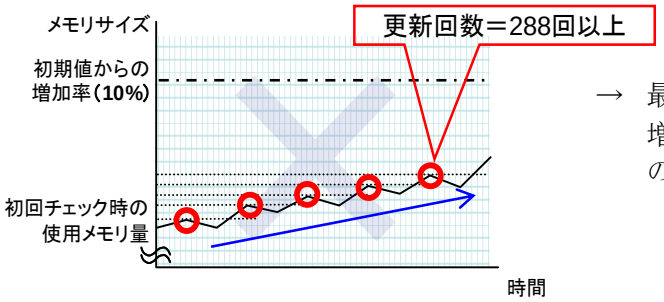
→ メモリ使用量は、一定の値を超えない範囲で増減しているため、更新回数はカウントされずメモリリークと判定しない。

● **メモリ使用量は一時的に増加率上限値を超えて増加し、その後下降線をたどりながら増減**



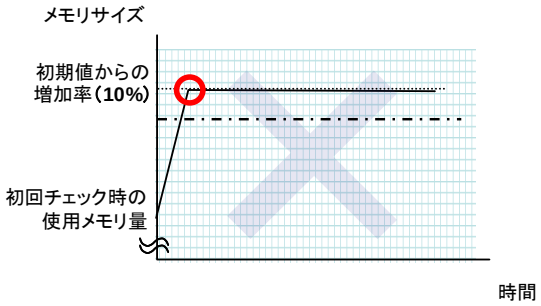
→ 一時的に増加率が初期値の 10%を超えているが、一定範囲内での一時的なメモリ使用量の増減のため、メモリリークと判定しない。

● **メモリ使用量は経過時間と共に増加しているが、起動時のメモリサイズからみるとわずかな増加率であり、初期値の 10%を超えていない**



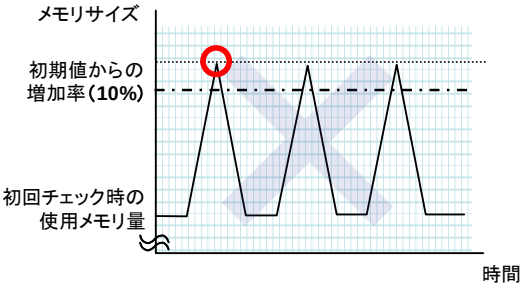
→ 最大値更新回数が 288 回を超えているが、増加率が初期値の 10%を超えていないので、メモリリークと判定しない。

● **メモリ使用量が初期値の 10%を超えてある値まで増加したのち、増減しない**



→ メモリ使用量は、初期値の 10%を超えて急激に増加しているが、その後のメモリ使用量に変動はなく、最大値更新回数はカウントされないため、メモリリークと判定しない。

● **ある時期にだけメモリ使用量が初期値の 10%を超えて増加するが、一定の値は超えていない**

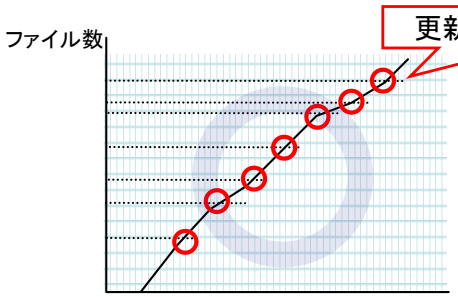


→ メモリ使用量は、ある期間（月次処理等の場合）になると初期値の 10%を超えて増加するが、ある範囲内での増加にとどまり、その後は最大値が更新されないため、メモリリークと判定しない。

- ファイルリーク監視
  ① 初回チェック時のオープンファイル数を最大オープンファイル数に設定します。
  ② 現在のオープンファイル数と前回のオープンファイル数を比較し、現在のオープンファイル数が前回よりも増加していた場合は、最大オープンファイル数と比較します。
  ③ 現在のオープンファイル数が最大オープンファイル数よりも多い場合、最大オープンファイル数を更新して更新回数をカウントします。
  ④ 更新回数が既定された回数（デフォルトでは 288 回）を超えて増加していた場合に、ファイルリークと判定します。

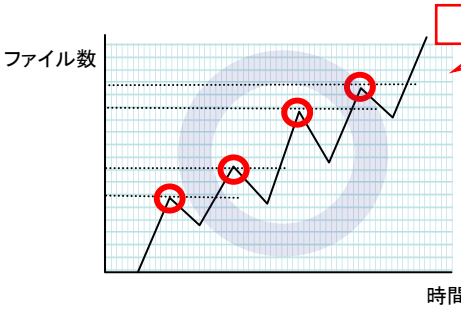
  次に、デフォルト値で運用した場合のファイルリークの具体的な例を示します。

● **オープンファイル数が経過時間と共に常に増加しつづけ、更新回数の上限値を超えた**



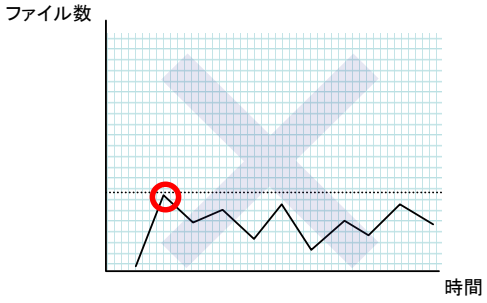
→ オープンファイル数の最大値は毎回更新され、規定回数（288 回）を超えたためファイルリークと判定する。

● **オープンファイル数が経過時間と共に増減を繰り返し、更新回数の上限値を超えた**



→ オープンファイル数の最大値が規定回数を超えて更新されているため、ファイルリークと判定する。

● **オープンファイル数は経過時間と共に一定の範囲内で増減している**



→ オープンファイル数は一定量を超えない範囲内で増減しているため、更新回数はカウントされない。そのため、ファイルリークとは判定しない。

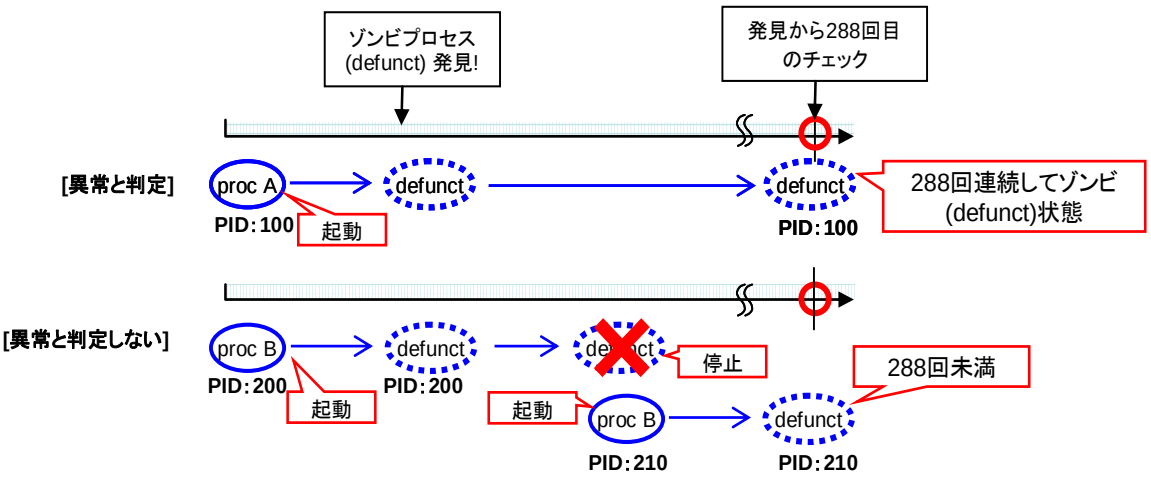
- スレッドリーク監視

スレッドリーク監視についての判定条件はファイルリーク監視と同様です。

- ゾンビプロセス（defunct）監視

  プロセスのステータスが既定回数(デフォルトでは 288 回(=1 日))以上連続してゾンビ状態(defunct)であった場合にゾンビプロセス（defunct）と判定します。

  以下に、デフォルトで運用した場合のゾンビプロセス検出の例を示します。



  常に起動・停止を繰り返すプロセスの場合、ゾンビプロセス（defunct）と判定しない。
  停止時にゾンビプロセス（defunct）となるが、しばらくすると消滅するようなプロセスの場合はゾンビプロセス（defunct）と判定しない。

- プロセス毎のオープンファイル数上限監視

  プロセス毎のオープンファイル数が上限（カーネルパラメータ maxfiles_lim 相当）の一定のパーセンテージ（デフォルトでは 90%）を超えていた場合にプロセス毎のオープンファイル数上限監視異常と判定します。

- プロセス毎のスレッド数上限監視

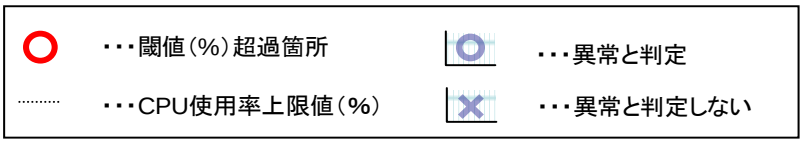
  プロセス毎のスレッド数が上限（カーネルパラメータ max_thread_proc 相当）の一定のパーセンテージ（デフォルトでは 90%）を超えていた場合にプロセス毎のスレッド数上限監視異常と判定します。

- プロセス多重度監視

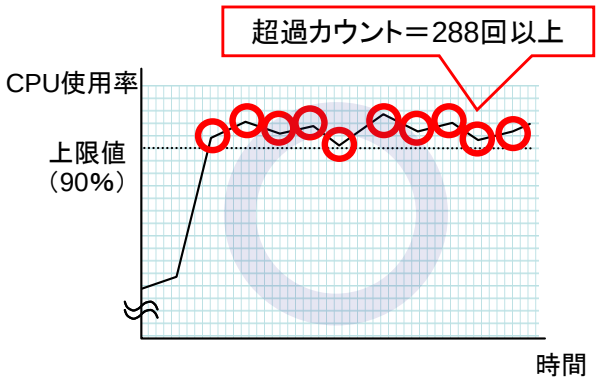
  現在のプロセスの多重度（同じプロセス名のプロセスが複数稼動している状態）が一定の閾値（デフォルトでは 100 個）を超えていた場合にプロセス多重度監視異常と判定します。

- 高 CPU 使用率プロセス監視
  プロセスの CPU 使用率が CPU 使用率上限値（デフォルトでは 90%）以上の状態が連続して規定回数（デフォルトでは 288 回（=1 日））を超えた場合に、高 CPU 使用率プロセスと判定します。

  以下に、デフォルトで運用した場合の高 CPU 使用率プロセス検出の例を示します。

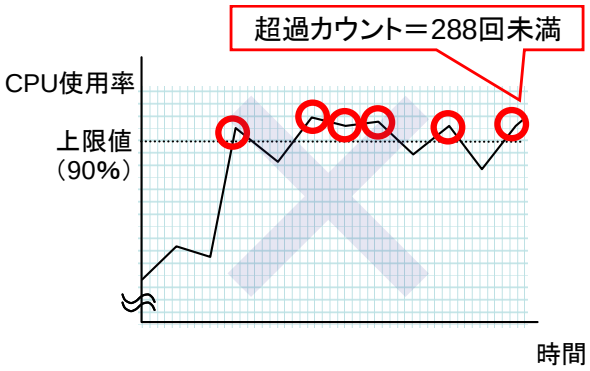


● **CPU 使用率が経過時間と共に CPU 使用率上限値を超え続け、閾値を超えた**



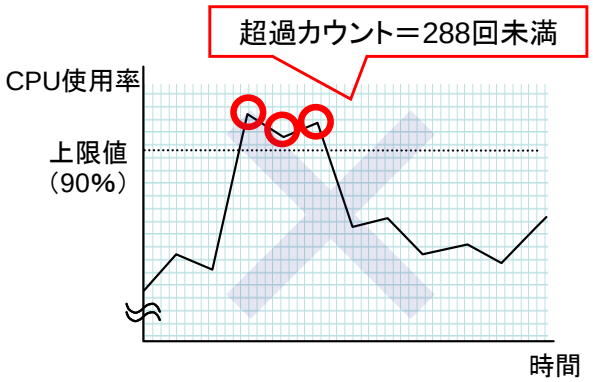
→ CPU 使用率は CPU 使用率上限値（90%）を超え続け、連続して閾値（288 回）を超えたため、高 CPU 使用率プロセスと判定する。

● **CPU 使用率が経過時間と共に CPU 使用率上限値の前後で増減している**



→ CPU 使用率が一度でも上限値を下回った場合、上限値超過回数は 0 にリセットされるため、規定回数（288 回）を超えない。そのため、高 CPU 使用率プロセスとは判定しない。

● **CPU 使用率が一時的に CPU 使用率上限値を超え、その後上限値を下回って推移**



→ 一時的に CPU 使用率上限値を超えるが、その後規定回数（288 回）を超える前に上限値以下で推移しているため、高 CPU 使用率プロセスとは判定しない。

（3）システムリソース監視の検出パターン

本製品では、「使用率上限値」、「連続超過時間」という２つのパラメータを組み合わせて検出を行います。
システム全体のリソース（オープンファイル数、プロセス数、ロックファイル数、ユーザプロセス数、スレッド数、メモリ使用量、CPU 使用率、スワップメモリ量）は、使用率上限値を連続超過時間、連続して超えた場合に検出します。

パラメータの意味とデフォルトの閾値は以下のようになります。

パラメータの意味	閾値（デフォルト）
使用率上限値	90%
連続超過時間	60分

常にシステム全体のリソースが使用率上限値を超えている場合、デフォルト値（使用率上限値 90%、連続超過時間 60 分）で運用すると、約 60 分後にこれらのリソース監視異常を検出します。

上記の設定（デフォルト）で運用した場合の検出パターンは以下のようになります。

連続超過時間	オープンファイル数上限監視	プロセス数上限監視	ロックファイル数上限監視	ユーザプロセス数上限監視	スレッド数上限監視	メモリ使用量上限監視	CPU 使用率上限監視	スワップメモリ量上限監視
60 分未満	×	×	×	×	×	×	×	×
60 分以上	○	○	○	○	○	○	○	○

※「○」･･･リソース監視異常を検出します。
「×」･･･リソース監視異常を検出しません。

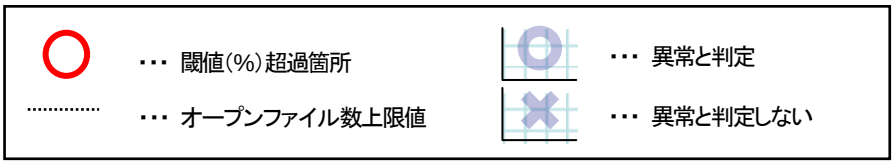
（4）システムリソース監視の判定条件

システムリソース監視の判定は以下の方法によって行います。

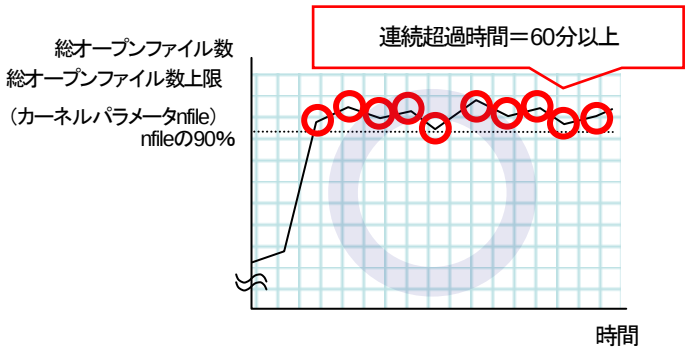
- オープンファイル数上限監視

  現在の総オープンファイル数が、総オープンファイル数上限値（カーネルパラメータ nfile 相当）の一定のパーセンテージ（デフォルトでは 90%）を、一定時間（デフォルトでは 60 分）以上連続して超えていた場合にオープンファイル数上限監視異常と判定します。

以下に、デフォルトで運用した場合のオープンファイル数上限監視異常検出の例を示します。

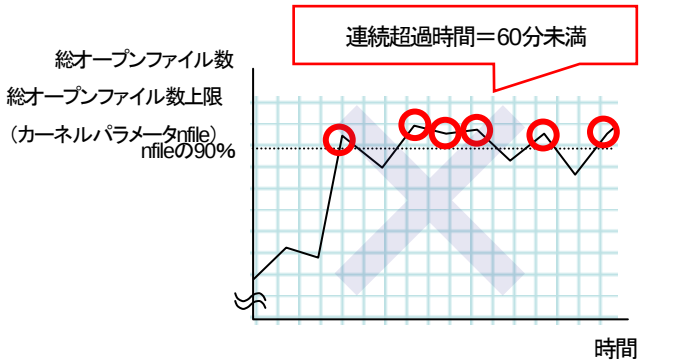


● **総オープンファイル数が経過時間と共に総オープンファイル数上限値を超え続け一定時間を超えた**



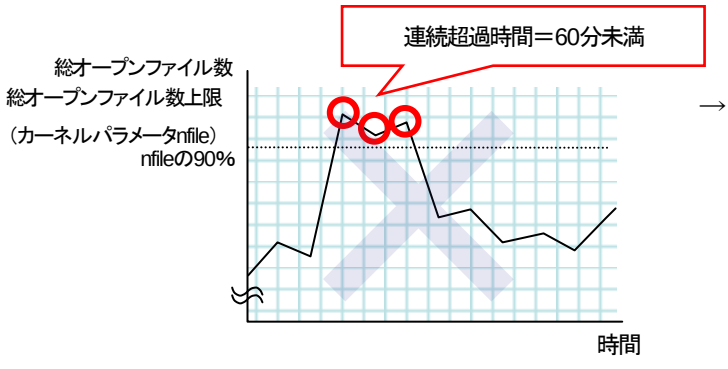
→ 総オープンファイル数は総オープンファイル数上限値（90%）を超え続け、連続して一定時間（60 分）を超えたため、オープンファイル数上限監視異常と判定する。

● **総オープンファイル数が経過時間と共に総オープンファイル数上限値の前後で増減し、連続して総オープンファイル数上限を超えない**



→ 総オープンファイル数は総オープンファイル数上限値（90%）を一時的に超えるが、指定された時間（60 分）連続して超える前に上限値以下で推移しているため、オープンファイル数上限監視異常とは判定しない。

● **総オープンファイル数が一時的に総オープンファイル数上限値を超えその後上限値を下回って推移**



→ 総オープンファイル数は総オープンファイル数上限値（90%）を一時的に超えるが、指定された時間（60 分）連続して超える前に上限値以下で推移しているため、オープンファイル数上限監視異常とは判定しない。

- プロセス数上限監視
  現在の総プロセス数が総プロセス数上限（カーネルパラメータ nproc 相当）の一定のパーセンテージ（デフォルトでは 90%）を、一定時間（デフォルトでは 60 分）以上連続して超えていた場合にプロセス数上限監視異常と判定します。
  判定条件はオープンファイル数上限監視と同様です。

- ロックファイル数上限監視
  現在の総ロックファイル数が総ロックファイル数上限（カーネルパラメータ nflocks 相当）の一定のパーセンテージ（デフォルトでは 90%）を、一定時間（デフォルトでは 60 分）以上連続して超えていた場合にロックファイル数上限監視異常と判定します。
  判定条件はオープンファイル数上限監視と同様です。

- ユーザプロセス数上限監視
  現在のユーザごとの起動プロセス数最大値がユーザごとの起動プロセス数上限（カーネルパラメータ maxuprc 相当）の一定のパーセンテージ（デフォルトでは 90%）を、一定時間（デフォルトでは 60 分）以上連続して超えていた場合にユーザプロセス数上限監視異常と判定します。
  判定条件はオープンファイル数上限監視と同様です。
  （注）ユーザプロセス数上限監視は、最も多くプロセスを起動しているユーザのプロセス数によって判定を行います。そのため、プロセス起動数が一番多いユーザ以外のユーザの起動プロセス数については上限監視は行いません。

- スレッド数上限監視
  現在の総スレッド数が総スレッド数上限（カーネルパラメータ nkthread 相当）の一定のパーセンテージ（デフォルトでは 90%）を、一定時間（デフォルトでは 60 分）以上連続して超えていた場合にスレッド数上限監視異常と判定します。
  判定条件はオープンファイル数上限監視と同様です。

- メモリ量上限監視
  現在の総メモリ使用量が総メモリ使用量上限（物理メモリ量）の一定のパーセンテージ（デフォルトでは 90%）を、一定時間（デフォルトでは 60 分）以上連続して超えていた場合にメモリ量上限監視異常と判定します。
  判定条件はオープンファイル数上限監視と同様です。

- CPU 使用率上限監視
  現在の CPU 使用率が CPU 使用率上限（100（%）×CPU 数）の一定のパーセンテージ（デフォルトでは 90%）を、一定時間（デフォルトでは 60 分）以上連続して超えていた場合に CPU 使用率上限監視異常と判定します。
  判定条件はオープンファイル数上限監視と同様です。

- スワップメモリ量上限監視
  現在の総スワップメモリ使用量または現在の総スワップメモリ予約量が総スワップメモリ量上限（使用可能なスワップ領域）の一定のパーセンテージ（デフォルトでは 90%）を、一定時間（デフォルトでは 60 分）以上連続して超えていた場合にスワップメモリ量上限監視異常と判定します。
  また、現在の総スワップメモリ使用量と現在の総スワップメモリ予約量の合計が総スワップメモリ量上限（使用可能なスワップ領域）の一定のパーセンテージ（デフォルトでは 90%）を、一定時間（デフォルトでは 60 分）以上連続して超えていた場合にもスワップメモリ量上限監視異常と判定します。
  判定条件はオープンファイル数上限監視と同様です。

SG ファイルはデフォルトで運用されることを推奨しますが、正しい運用状態にもかかわらず、リソース監視異常が頻繁に検出される場合にはシステムのリソース状態にあわせ、SG ファイルを変更する必要があります。

３．４　リソース監視異常検出時のスクリプト実行

リソース監視異常検出時のアクションとして、ユーザが指定したスクリプトを実行することが可能です。
スクリプト実行は以下のファイルで設定します。

/var/opt/HA/PSSM/conf/psaction.conf

（1）　リソース監視異常検出時のスクリプト実行手順

リソース監視異常検出時のアクションタイプに **SCRIPT_EXEC** を指定することで、
該当のリソース監視異常を検出した場合に指定したスクリプトを実行することが可能です。
リソース監視異常検出時にスクリプトを実行する場合、**SCRIPT_EXEC** のオプションに、
スクリプト名を絶対パスで記述します。

SCRIPT_EXEC 行は、各リソース監視異常単位で異常検出時に実行させたいスクリプトを
指定することが可能です。

SCRIPT_EXEC:<スクリプト名>

（設定例）メモリリーク検出時に/opt/WS/WebServer_restart.sh というスクリプトを実行する場合。
MEMORY_LEAK _ERROR_ACTION　SCRIPT_EXEC:/opt/WS/WebServer_restart.sh

※ 指定するスクリプトには必ず実行権を与えてください。実行権がない場合、スクリプトの実行に失敗します。
スクリプトは絶対パスで指定してください。
1 つのリソース監視異常について、**SCRIPT_EXEC** 行を複数指定することはできません。
2 つ以上のスクリプトを実行させたい場合は、**SCRIPT_EXEC** のオプションにコロン（:）区切りで実行するスクリプトを指定してください。

（設定例）リソース監視異常検出時に/opt/WS/WebServer_restart.sh というスクリプトと
/var/opt/HA/PSSM/bin/collect_info.sh というスクリプトを実行する場合。
… SCRIPT_EXEC:/opt/WS/WebServer_restart.sh:/var/opt/HA/PSSM/bin/collect_info.sh

指定可能な動作定義パラメータについては「 4 章 SG の設定」を参照してください。

以下に、設定ファイルの例を示します。

< /var/opt/HA/PSSM/conf/psaction.conf >

```
############################################################
# Action Information
############################################################

# The action when the defunct process is discovered is specified.
DEFUNCT_ERROR_ACTION                    TRACE_REPORT
DEFUNCT_ERROR_ACTION                    SYSLOG_REPORT:WARNING
DEFUNCT_ERROR_ACTION                    SCRIPT_EXEC:/var/opt/HA/PSSM/bin/collect_info.sh  ★

# The action when the memory leak process is discovered is specified.
MEMORY_LEAK_ERROR_ACTION                TRACE_REPORT
MEMORY_LEAK_ERROR_ACTION                SYSLOG_REPORT:WARNING
#MEMORY_LEAK_ERROR_ACTION                PROCESS_KILL:6
MEMORY_LEAK_ERROR_ACTION                SCRIPT_EXEC:/var/opt/HA/PSSM/bin/collect_info.sh  ★

# The action when the file descriptor leak process is discovered is specified.
FILE_LEAK_ERROR_ACTION                  TRACE_REPORT
FILE_LEAK_ERROR_ACTION                  SYSLOG_REPORT:WARNING
#FILE_LEAK_ERROR_ACTION                  PROCESS_KILL:6
FILE_LEAK_ERROR_ACTION                  SCRIPT_EXEC:/var/opt/HA/PSSM/bin/collect_info.sh  ★

# The action when the open file num over process is discovered is specified.
MAX_FILE_ERROR_ACTION                   TRACE_REPORT
MAX_FILE_ERROR_ACTION                   SYSLOG_REPORT:WARNING
#MAX_FILE_ERROR_ACTION                   PROCESS_KILL:6
MAX_FILE_ERROR_ACTION                   SCRIPT_EXEC:/var/opt/HA/PSSM/bin/collect_info.sh  ★

# The action when the high CPU utilization process is discovered is specified.
CPU_ERROR_ACTION                        TRACE_REPORT
CPU_ERROR_ACTION                        SYSLOG_REPORT:WARNING
#CPU_ERROR_ACTION                        PROCESS_KILL:6
CPU_ERROR_ACTION                        SCRIPT_EXEC:/var/opt/HA/PSSM/bin/collect_info.sh  ★

          :                   :                   :

# The action when the thread num orver process is discovered is specified.
MAX_THREAD_ERROR_ACTION                 TRACE_REPORT
MAX_THREAD_ERROR_ACTION                 SYSLOG_REPORT:WARNING
#MAX_THREAD_ERROR_ACTION                 PROCESS_KILL:6
MAX_THREAD_ERROR_ACTION                 SCRIPT_EXEC:/var/opt/HA/PSSM/bin/collect_info.sh  ★

# The action when the kernel parameter upper error is discovered is specified.
KERNEL_ERROR_ACTION                     TRACE_REPORT
KERNEL_ERROR_ACTION                     SYSLOG_REPORT:WARNING
KERNEL_ERROR_ACTION                     SCRIPT_EXEC:/var/opt/HA/PSSM/bin/collect_info.sh  ★
```

★リソース監視異常検出時にSCRIPT_EXECアクションを適用したい動作定義パラメータの
＃（コメント）を外し、実行するスクリプト名を絶対パスで指定してください。
対象となるリソース監視異常検出時に指定したスクリプトを実行します。

異常検出時のアクションにPROCESS_KILLまたはSCRIPT_EXECを指定した場合は、
リソース監視異常検出のメッセージについてもシステムログへ出力されるように、
SYSLOG_REPORTの指定も行ってください。

(注) その他のアクション指定については、「４章 SGの設定」を参照してください。

（2） リソース監視異常検出時の障害解析用情報収集スクリプトについて

SCRIPT_EXEC のオプションには障害解析用の情報を収集するスクリプト **/var/opt/HA/PSSM/bin/collect_info.sh** が指定されています。

本スクリプトの詳細については、「 7 章 付録」を参照してください。

（注）**SCRIPT_EXEC** はデフォルトで無効になっていますので、本スクリプトを実行する場合は、＃（コメント）を外してください。

（3） スクリプト実行に関する注意事項

- 1 つのリソース監視異常について、**SCRIPT_EXEC** 行を複数指定することはできません。
  2 つ以上のスクリプトを実行させたい場合は、**SCRIPT_EXEC** のオプションにコロン（:）区切りで実行するスクリプトを指定してください。

  （設定例）メモリリーク検出時に/opt/aaa.sh というスクリプトと/var/opt/bbb.sh というスクリプトを実行する場合。

  MEMORY_LEAK _ERROR_ACTION SCRIPT_EXEC:/opt/aaa.sh: /var/opt/bbb.sh

- 指定するスクリプトは絶対パスで指定してください。

- コロン（:）が含まれるスクリプトを指定することはできません。

- コマンドへのパスが正しく設定されていない場合、コマンドの実行に失敗します。
  スクリプト内で実行するコマンドは絶対パスで記述してください。

- 指定するスクリプトには必ず実行権を与えてください。実行権がない場合、スクリプトの実行に失敗します。

- アクション実行はデフォルトで 5 分のタイムアウト値をもっているため、実行する処理は 5 分以内に終了する必要があります。
  したがって、スクリプト内で長時間 sleep により待ち合わせを行う等、処理に時間がかかり 5 分を超えても終了しない場合、正常にスクリプト実行が行われない場合があります。
  また、スクリプト内でデーモン化されていないプロセスを起動する場合には、コマンドラインの最後に&を付与し、バックグラウンドで起動し、スクリプトが 5 分以内に終了するよう記述してください。
  5 分を超えるスクリプトを実行する必要がある場合は、直接開発元へお問い合わせください。

- スクリプトからプロセスを起動する場合に、起動するプロセスが環境変数に依存している場合は、その環境変数を設定してからプロセスを呼び出してください。

- 異常検出時のアクションに PROCESS_KILL または SCRIPT_EXEC を指定した場合は、リソース監視異常検出のメッセージについてもシステムログへ出力されるように、SYSLOG_REPORT の指定も行ってください。

３．５　ProcessSaver との連携について

ProcessSaver と連携することで、プロセスリソースの障害を早期に検出し問題が発生する前に該当プロセスの再起動を行うことが可能です。

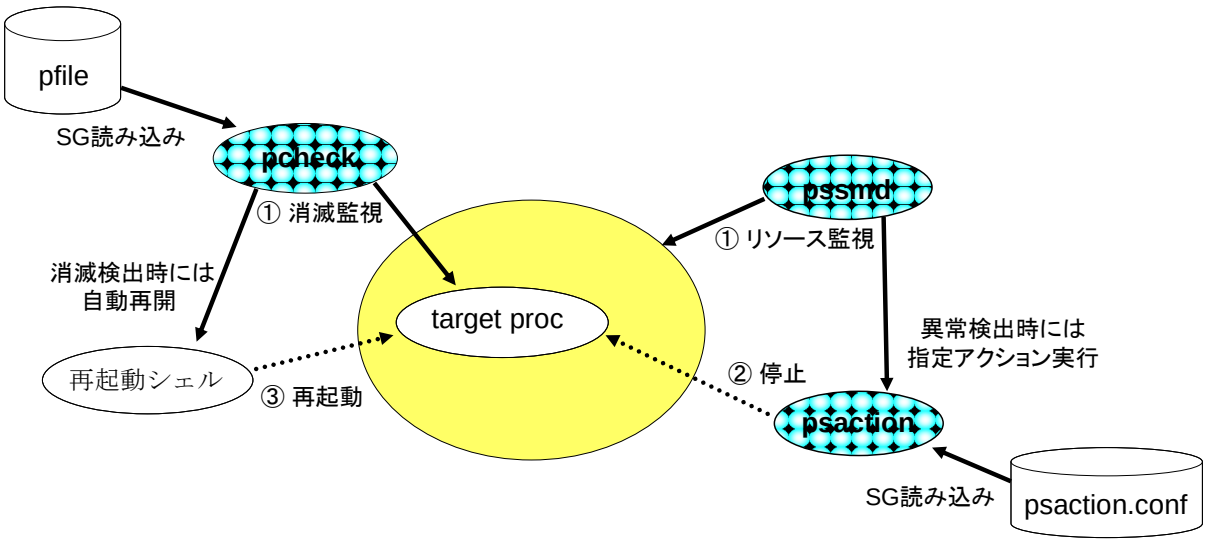
（1）　ProcessSaver との連携による該当プロセスの再起動の流れ

リソース監視異常と判定されたプロセスを停止することで、ProcessSaver が監視対象プロセスの消滅を検知し、プロセスの再起動を実行します。

（注）本機能は ProcessSaver で監視を行っており、消滅検出時に再起動を指定しているプロセスについてのみ有効です。

［ProcessSaver 連携の仕組み］

① pssmd によってプロセスリソースを監視します。
　pcheck によってプロセスの消滅を監視します。
② 異常検出時のアクションに PROCESS_KILL を指定することで、リソース監視異常が発生したプロセスを停止します。
　（注）init 等の停止することのできないプロセスや defunct 状態のプロセスは、kill(2) システムコールで停止することはできません。
③ 停止されたプロセスについて ProcessSaver で監視及びプロセス消滅検出時に再起動を指定している場合、再起動を行います。



（2） 連携可能な障害のパターン

再起動の対象となるリソース監視は以下のとおりです。

・メモリリーク監視

メモリ使用量が一定時間以上増加傾向にあるプロセス

・ファイルリーク監視

オープンファイル数が一定時間以上増加傾向にあるプロセス

・プロセス毎のオープンファイル数上限監視

オープンファイル数がカーネルパラメータ maxfiles_lim の一定のパーセンテージを超えているプロセス

・高 CPU 使用率プロセス監視

CPU 使用率が、一定時間以上連続して一定の閾値(%)を超えている状態にあるプロセス

・スレッドリーク監視

スレッド数が一定時間以上増加傾向にあるプロセス

・プロセス毎のスレッド数上限監視

スレッド数がカーネルパラメータ max_thread_proc の一定のパーセンテージを超えているプロセス

（注）本機能においてゾンビ（defunct）プロセス監視、プロセス多重度監視、システム全体のリソース監視は対象外です。

（3） ProcessSaver との連携設定手順

プロセスの停止及び再起動は以下のファイルで設定します。

・SystemResourceMonitor の設定ファイル

/var/opt/HA/PSSM/conf/psaction.conf

該当プロセスの停止が行われるよう設定してください。

（設定例）メモリリーク検出時に対象プロセスを kill(2) システムコールで停止する場合

MEMORY_LEAK_ERROR_ACTION　PROCESS_KILL:6

PROCESS_KILL のオプションには kill(2) システムコールが送信するシグナル番号を指定してください。
指定可能なシグナルは**15（SIGTERM）**、**9（SIGKILL）**、**6（SIGABRT）**です。
デフォルトは**6（SIGABRT）**です。
デフォルトのシグナル**6（SIGABRT）**を指定した場合、プロセスを停止する際に該当プロセスが動作するカレントディレクトリに core ファイルを作成するため、解析時に有効です。
該当のプロセス停止時に core ファイルを作成したくない場合は、任意にシグナル番号を変更してください。

また、PROCESS_KILL アクションを使用される場合は、SYSLOG_REPORT についても指定し、リソース監視異常検出時にシステムログへのメッセージ出力が行われるように設定を行ってください。

（設定例）メモリリーク検出時に、システムログにメッセージ WARNING で出力します。

MEMORY_LEAK _ERROR_ACTION　SYSLOG_REPORT : WARNING

・ProcessSaver の設定ファイル

/var/opt/HA/PS/conf/bin/pfile_XXX
/var/opt/HA/PS/conf/bin/restart_XXX.sh

対象プロセスを監視し、消滅を検知した場合にプロセスの再起動を実行します。

（注）ProcessSaver での監視手順については、ProcessSaver のユーザーズガイド等を参照してください。

以下に、各設定ファイルの例を示します。

・アクション定義ファイル

< /var/opt/HA/PSSM/conf/psaction.conf >

```
#############################################################
# Action Information
#############################################################

# The action when the defunct process is discovered is specified.
DEFUNCT_ERROR_ACTION                    TRACE_REPORT
#DEFUNCT_ERROR_ACTION                   SYSLOG_REPORT:WARNING
#DEFUNCT_ERROR_ACTION                   SCRIPT_EXEC:/var/opt/HA/PSSM/bin/collect_info.sh

# The action when the memory leak process is discovered is specified.
MEMORY_LEAK_ERROR_ACTION                TRACE_REPORT
MEMORY_LEAK_ERROR_ACTION                SYSLOG_REPORT:WARNING
MEMORY_LEAK_ERROR_ACTION                PROCESS_KILL:6  ★
#MEMORY_LEAK_ERROR_ACTION               SCRIPT_EXEC:/var/opt/HA/PSSM/bin/collect_info.sh

# The action when the file descriptor leak process is discovered is specified.
FILE_LEAK_ERROR_ACTION                  TRACE_REPORT
FILE_LEAK_ERROR_ACTION                  SYSLOG_REPORT:WARNING
FILE_LEAK_ERROR_ACTION                  PROCESS_KILL:6  ★
#FILE_LEAK_ERROR_ACTION                 SCRIPT_EXEC:/var/opt/HA/PSSM/bin/collect_info.sh

# The action when the open file num over process is discovered is specified.
MAX_FILE_ERROR_ACTION                   TRACE_REPORT
MAX_FILE_ERROR_ACTION                   SYSLOG_REPORT:WARNING
MAX_FILE_ERROR_ACTION                   PROCESS_KILL:6  ★
#MAX_FILE_ERROR_ACTION                  SCRIPT_EXEC:/var/opt/HA/PSSM/bin/collect_info.sh

          :                   :                   :

# The action when the thread num orver process is discovered is specified.
MAX_THREAD_ERROR_ACTION                 TRACE_REPORT
MAX_THREAD_ERROR_ACTION                 SYSLOG_REPORT:WARNING
MAX_THREAD_ERROR_ACTION                 PROCESS_KILL:6  ★
#MAX_THREAD_ERROR_ACTION                SCRIPT_EXEC:/var/opt/HA/PSSM/bin/collect_info.sh

# The action when the kernel parameter upper error is discovered is specified.
KERNEL_ERROR_ACTION                     TRACE_REPORT
#KERNEL_ERROR_ACTION                    SYSLOG_REPORT:WARNING
#KERNEL_ERROR_ACTION                    SCRIPT_EXEC:/var/opt/HA/PSSM/bin/collect_info.sh
```

★リソース監視異常検出時に PROCESS_KILL アクションを適用したい動作定義パラメータの
＃（コメント）を外してください。
対象となるリソース監視異常検出時に該当のプロセスを kill(2) システムコールで停止します。

PROCESS_KILL のオプション（kill(2) システムコールが送信するシグナル番号）は、デフォルトで **6（SIGABRT）**が指定されていますが **9（SIGKILL）**または **15（SIGTERM）**を指定することも可能です。
シグナル番号を変更する場合は、直接ファイルを編集してください。

異常検出時のアクションに PROCESS_KILL を指定した場合は、リソース監視異常検出の
メッセージについてもシステムログへ出力されるように、SYSLOG_REPORT の指定も行ってください。

（注）その他のアクション指定については、「 4 章 SG の設定」を参照してください。

・pfile の例

＜/var/opt/HA/PS/conf/bin/pfile_XXX＞

```
##### PARAM #####
IPCKEY                  0x1f000001
MSG_CHECK_INTERVAL      5
MONITOR_INTERVAL        10
SHM_DUMP_FILE           /var/opt/HA/PS/log/pcheck_dump_procA

##### PENT #####
procA:/var/opt/HA/PS/conf/bin/restart_XXX.sh:86400:3:continue
★1    ★2
```

★1　監視対象プロセス名を指定してください。
★2　必要に応じてプロセスを再起動するスクリプトを指定してください。

・再起動スクリプトの例

＜/var/opt/HA/PS/conf/bin/restart_XXX.sh＞

```
#!/bin/sh
procA stop    ★1
/usr/bin/sleep 5
procA start   ★2
/usr/bin/sleep 5
exit 0
```

★1　監視対象プロセスの停止処理を記述してください。
　　プロセスの消滅を検出した場合、プロセスがいなくても関連するファイルの削除等の後処理を実行する場合があるため。
★2　監視対象プロセスの開始処理を記述してください。

（注）pfile 及び再起動スクリプトは ProcessSaver の設定となります。
　　設定手順の詳細については、ProcessSaver のユーザーズガイド等を参照してください。

（4） ProcessSaver 連携に関する注意事項

- 異常検出時のアクションに PROCESS_KILL を指定した場合、リソース監視異常と判定されたプロセスは ProcessSaver での監視の有無にかかわらず kill(2) システムコールで停止されますのでご注意ください。
  また、ProcessSaver で監視対象となっているプロセスについて、プロセス消滅検知時のアクションに再起動を指定していないプロセスは再起動されません。
  リソース監視異常と判定された場合に停止させたいプロセスは、すべてプロセス KILL アクション定義ファイル（psact_stop_allow.conf）に設定してください。
  または、リソース監視異常と判定されても停止させたくないプロセスは、すべてプロセス KILL アクション除外定義ファイル（psact_notstop.conf）に設定してください。

  ※ プロセス KILL アクション定義ファイル（psact_stop_allow.conf）およびプロセス KILL アクション除外定義ファイル（psact_notstop.conf）の設定手順については、「 4 章 SG の設定」を参照してください。

- PROCESS_KILL のオプションには kill(2) システムコールが送信するシグナル番号を指定してください。
  デフォルトのシグナル番号 **6（SIGABRT）** を指定した場合、プロセスを停止する際に該当プロセスが動作するカレントディレクトリに core ファイルを作成するため、解析時に有効です。
  該当のプロセス停止時に core ファイルを作成したくない場合は、任意にシグナル番号を変更してください。

- 異常検出時のアクションに PROCESS_KILL を指定した場合、プロセス KILL アクション定義ファイル（psact_stop_allow.conf）が存在する場合、プロセス KILL アクション除外定義ファイル（psact_notstop.conf）の内容は反映されません。
  両方のファイルに重複して指定されているプロセスについては、プロセス KILL アクション定義ファイル（psact_stop_allow.conf）が有効になります。

- init 等の停止することのできないプロセスや defunct 状態のプロセスは停止されません。

- ゾンビ（defunct）プロセス監視、プロセス多重度監視、システム全体のリソース監視異常検出時に該当プロセス等の停止を行うことはできません。

- 異常検出時のアクションに PROCESS_KILL を指定した場合は、リソース監視異常検出のメッセージについてもシステムログへ出力されるように、SYSLOG_REPORT の指定も行ってください。

３．６　Serviceguard との連携について

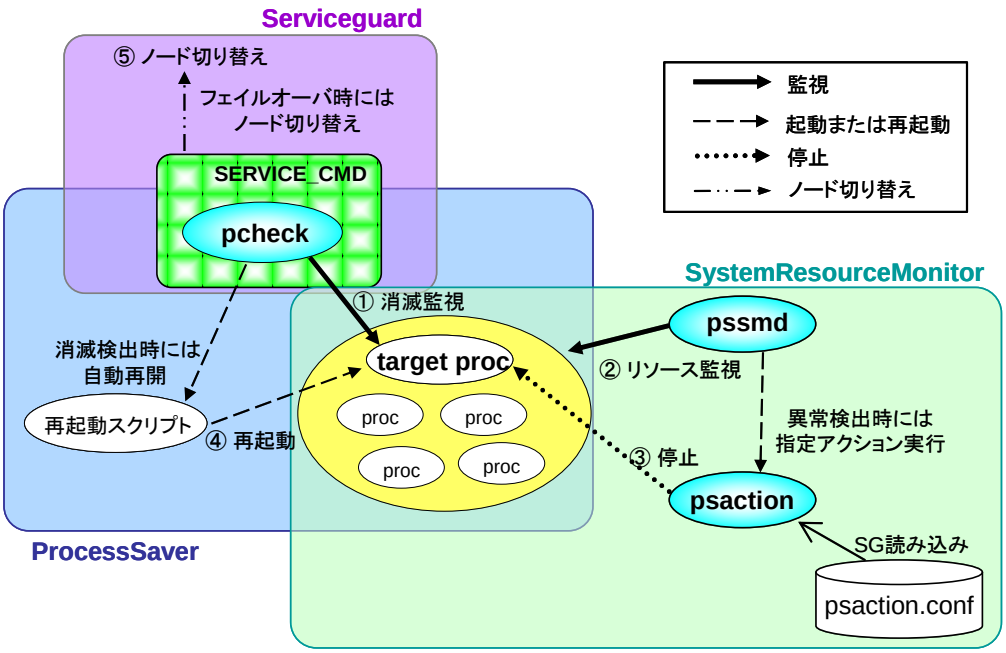
Serviceguard と連携することで、プロセスリソースの障害時に、クラスタシステムでのソフトウェアの可用性を向上させます。
SystemResourceMonitor の Serviceguard 連携は、ProcessSaver を利用して行うか、またはダミープロセスを使用して行います。
SystemResourceMonitor で障害を引き起こす原因となるプロセスの検出・停止を行い、ProcessSaver でプロセスの再起動を行うことで、無用なパッケージ切り替えやノード切り替えを防止します。
プロセス再開後にも障害の原因が改善されない場合には、Serviceguard 連携により、適切なパッケージ切り替えやノード切り替えを行い、業務を継続することが可能です。
（注）本機能は ProcessSaver との連携を行っている場合にのみ動作します。
ProcessSaver 連携の詳細については、「 3.5 章 ProcessSaver との連携について」を参照してください。

(1)　Serviceguard 連携の仕組み

本機能は、ProcessSaver との連携によって、Serviceguard 連携を可能とします。
ProcessSaver によって対象となるプロセスを監視し、プロセスの監視を行っている pcheck(1M)を Serviceguard のパッケージとして定義します。
SystemResourceMonitor が対象プロセスの異常検出→停止を行い、ProcessSaver が再起動を行うことで、自系再開を試みてシステムの復旧を図ります。自系再開を行っても復旧できない場合は、pcheck(1M)がリトライオーバとなり、Serviceguard によってパッケージ切り替えやノード切り替えを行います。

Serviceguard 連携は以下のような流れで動作します。

① Serviceguard の service.sh から pcheck(1M)を呼び出し、ターゲットのプロセスを監視します。
② SystemResourceMonitor が、プロセスリソースを監視します。
③ ターゲットプロセスの異常を SystemResourceMonitor が検出した場合、ターゲットプロセスを停止します。
④ ProcessSaver によってターゲットプロセスの停止を検知し、再起動を行います。
⑤ 自系再開を試みてもシステムが復旧しない場合は、pcheck(1M)が停止します。
⑥ Serviceguard が pcheck(1M)の停止を検知し、パッケージ移動やノード切り替えを行います。



(2)　Serviceguard との連携設定手順

Serviceguard のパッケージ制御スクリプトにサービスコマンドとなるシェルスクリプトを登録します。
そのサービスコマンドにおいて、対象プロセスおよびpcheck(1M)を起動するような設定を
行うことによって、Serviceguard 連携を行います。

Serviceguard と連携するためには、各ノードに本製品およびProcessSaver をインストールし、
以下のファイルを設定する必要があります。

① SystemResourceMonitor の設定ファイル
・アクション定義ファイル（psaction.conf）
② ProcessSaver の設定ファイル
・pcheck(1M)コマンドの SG ファイル（pfile）
③ Serviceguard の設定ファイル
・サービスコマンド（service.sh）
・パッケージ制御スクリプト（pkg.sh）

① SystemResourceMonitor の設定

・アクション定義ファイル（psaction.conf）
/var/opt/HA/PSSM/conf/psaction.conf

対象プロセスの異常を検出した場合に、プロセスを停止するため、プロセス KILL アクションを
設定します。
プロセス KILL アクションの設定方法の詳細については、「 3.5 章 ProcessSaver との連携について」
を参照してください。

プロセス KILL アクションの対象となるリソース監視は以下のとおりです。
・メモリリーク監視
・ファイルリーク監視
・プロセス毎のオープンファイル数上限監視
・高 CPU 使用率プロセス監視
・スレッドリーク監視
・プロセス毎のスレッド数上限監視

（注）本機能においてゾンビ（defunct）プロセス監視、プロセス多重度監視、システム全体の
リソース監視は対象外です。
プロセス KILL アクションを設定した場合、対象のリソース監視異常検出時には
プロセスを区別することなく停止処理を行います。
リソース監視異常と判定された場合に停止させたいプロセスは、すべてプロセス KILL
アクション定義ファイル（psact_stop_allow.conf）に設定してください。
または、リソース監視異常と判定されても停止させたくないプロセスをすべてプロセス KILL
アクション除外定義ファイル（psact_notstop.conf）に設定してください。
両方のファイルに重複して指定されているプロセスについては、プロセス KILL アクション定義
ファイル（psact_stop_allow.conf）が有効になります。

プロセス KILL アクション定義ファイル（psact_stop_allow.conf）およびプロセス KILL
アクション除外定義ファイル（psact_notstop.conf）の設定手順については、
「 4 章 SG の設定」を参照してください。

その他のアクション指定については、「 4 章 SG の設定」を参照してください。

② ProcessSaver の設定

・pcheck(1M)コマンドの SG ファイル（pfile）

/var/opt/HA/PS/conf/bin/pfile_XXX

Serviceguard と連携する場合、SG ファイルのプロセス監視のリトライオーバアクションに **exit** を指定します。
exit 以外を指定した場合の動作は保証いたしません。

対象となるプロセスが消滅し、SG の指定回数内で再起動できない場合、pcheck(1M)を終了し、Serviceguard の設定にしたがってパッケージ切り替えやノード切り替えを行います。

（注）ProcessSaver での監視手順については、ProcessSaver のユーザーズガイド等を参照してください。

・pfile の例

＜/var/opt/HA/PS/conf/bin/pfile_XXX＞

```
##### PARAM #####
IPCKEY                  0x1f000001
MSG_CHECK_INTERVAL      5
MONITOR_INTERVAL        10
SHM_DUMP_FILE           /var/opt/HA/PS/log/pcheck_dump_procA

##### PENT #####
procA:/var/opt/HA/PS/conf/bin/restart_XXX.sh:86400:3:exit
```

③ Serviceguard の設定

・サービスコマンド（service.sh）

サービスコマンドでは、対象プロセスを監視するために pcheck(1M)を起動します。
このサービスコマンドは運用環境に応じて作成する必要があります。作成したサービスコマンドを、Serviceguard のパッケージ制御スクリプトに登録し、Serviceguard からの起動および監視を行うことにより連携を実現します。

対象プロセスが再起動できない場合、サービスコマンドから実行した pcheck(1M)が検知し、SG ファイル（pfile）の exit 指定にしたがって終了します。
pcheck(1M)の終了によって、サービスコマンドが終了するため、サービスコマンドの終了を Serviceguard が検知し、パッケージ切り替えやノード切り替え等の処理を実行します。

・サービスコマンドの例

＜/etc/cmcluster/pkg1/service.sh＞

```
#!/sbin/sh

procA start        ★1

/usr/bin/sleep 10  ★2

/opt/HA/PS/bin/pcheck -f /var/opt/HA/PS/conf/bin/pfile_XXX  ★3
```

★1　対象プロセスの起動
　　対象プロセスの起動をパッケージ制御スクリプトで実行している場合は不要です。
★2　対処プロセスの起動処理が終了するまで待ち合わせします。
★3　pcheck の起動

（注）pcheck(1M)は、監視対象プロセスを起動した後で実行してください。

（注）複数の pcheck コマンドを呼び出す場合は、バックグラウンドで起動し、
　　最後の pcheck をフォアグラウンドで起動してください。

（注）プロセス監視の対象となるプロセスを起動する場合は、pcheck(1M)が実行できるようにバックグラウンド等で起動してください。
（注）サービスコマンドの設定方法の詳細については、Serviceguard のマニュアルを
　　参照してください。

・パッケージ制御スクリプト（pkg.sh）
Serviceguard において、サービスコマンドの起動および監視を行うために、Serviceguard のパッケージ制御スクリプトを設定します。

・パッケージ制御スクリプトの例
＜/etc/cmcluster/pkg1/pkg.sh＞

```
        :                   :
SERVICE_NAME[0]=" service1"                          ★1
SERVICE_CMD[0]=" /etc/cmcluster/pkg1/service.sh"     ★2
SERVICE_RESTART[0]=" "                               ★3
        :                   :
```

★1　サービス名
★2　サービスコマンドのパス名
★3　再起動回数

（注）パッケージ制御スクリプトの設定方法の詳細については、Serviceguard のマニュアルを参照してください。

（3）　システムリソース監視における Serviceguard 連携

システムリソース監視における Serviceguard 連携は、スクリプト実行アクションを利用することで可能です。

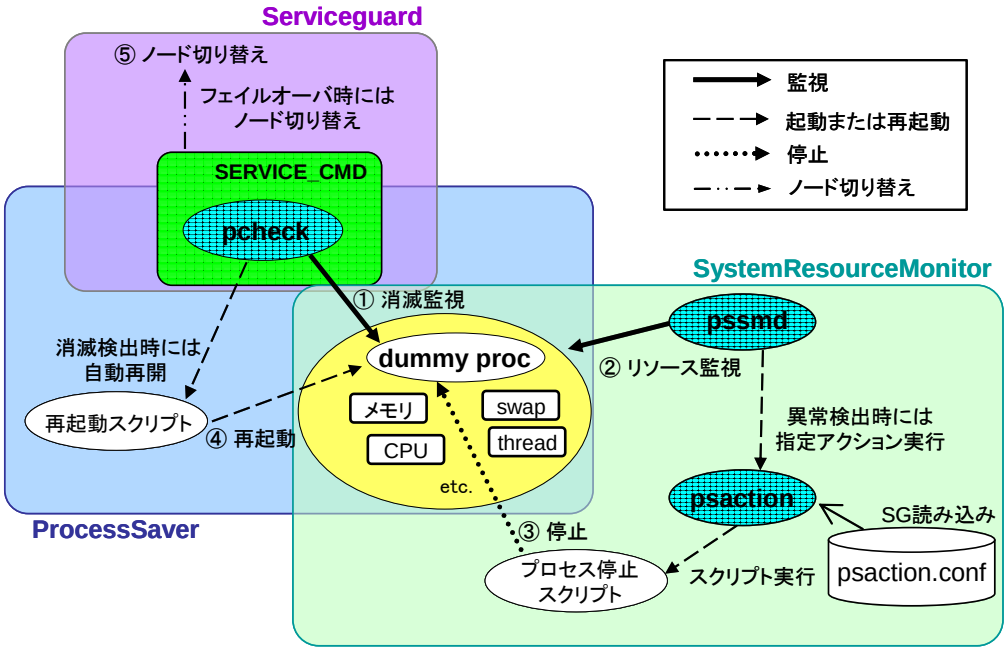
システムリソース監視は、異常発生時に原因となるプロセスの特定は行わないため、プロセス KILL アクションの対象外になります。前述のプロセス KILL アクションを利用した Serviceguard 連携の代わりに、スクリプト実行アクションを利用する方法によって Serviceguard 連携を実現します。

スクリプト実行アクションを利用した Serviceguard 連携を行うために、以下のファイルを用意します。

・ProcessSaver で監視を行うためのダミープロセス
　ProcessSaver の SG ファイル(pfile)に監視対象として設定します。
・ダミープロセスの停止処理を行うスクリプト
　SystemResourceMonitor のシステムリソース監視のスクリプト実行アクションに指定します。
　（注）スクリプト実行アクションの設定に関する詳細は「 3.4 章 リソース異常検出時のユーザ指定によるスクリプト実行」を参照してください。

システムリソース監視の Serviceguard 連携は以下の流れで動作します。

①Serviceguard の service.sh から pcheck(1M)を呼び出し、ダミープロセスを監視します。
②SystemResourceMonitor が、システムリソースを監視します。
③システムリソースの異常を SystemResourceMonitor が検出した場合、指定したスクリプトによって、ダミープロセスを停止します。
④ProcessSaver によってダミープロセスの停止を検知し、pcheck(1M)が停止します。
⑤Serviceguard が pcheck(1M)の停止を検知し、パッケージ移動やノード切り替えを行います。



ダミープロセスおよびプロセス停止スクリプトの例は以下のとおりです。

ダミープロセスの例：pssm_stat.c

```
#include <stdio.h>
#include <unistd.h>

main()
{
        while(1){
                sleep(600);
        }
}
```

スクリプトの例：/var/opt/HA/PSSM/bin/stop_pssm_stat.sh

```
#!/bin/sh

PS_CMD="/usr/bin/ps"
GREP_CMD="/usr/bin/grep"
AWK_CMD="/usr/bin/awk"
PROC_NAME=" pssm_stat"
pid=" "

pid=`${PS_CMD} -e | ${GREP_CMD} ${PROC_NAME} | ${GREP_CMD} -v "${GREP_CMD}" |
${AWK_CMD} '{printf("%s ",$1)}END{printf("¥n")}'`
    if [ -n "$pid" ]; then
        /usr/bin/kill -15 ${pid}
    fi

exit 0
```

(4) Serviceguard 連携に関する注意事項

- Serviceguard との連携には、ProcessSaver が必要となります。

- 異常検出時のアクションに PROCESS_KILL を指定した場合、リソース監視異常と判定されたプロセスは ProcessSaver での監視の有無にかかわらず kill(2) システムコールで停止されますのでご注意ください。
  また、ProcessSaver で監視対象となっているプロセスについて、プロセス消滅検知時のアクションに再起動を指定していないプロセスは再起動されません。
  リソース監視異常と判定された場合に停止させたいプロセスは、すべてプロセス KILL アクション定義ファイル（psact_stop_allow.conf）に設定してください。
  または、リソース監視異常と判定されても停止させたくないプロセスをすべてプロセス KILL アクション除外定義ファイル（psact_notstop.conf）に設定してください。

  プロセス KILL アクション定義ファイル（psact_stop_allow.conf）およびプロセス KILL アクション除外定義ファイル（psact_notstop.conf）の設定手順については、「 4 章 SG の設定」を参照してください。

- 異常検出時のアクションに PROCESS_KILL を指定した場合、プロセス KILL アクション定義ファイル（psact_stop_allow.conf）が存在する場合、プロセス KILL アクション除外定義ファイル（psact_notstop.conf）の情報は反映されません。
  両方のファイルに重複して指定されているプロセスについては、プロセス KILL アクション定義ファイル（psact_stop_allow.conf）が有効になります。

- PROCESS_KILL のオプションには kill(2) システムコールが送信するシグナル番号を指定してください。
  デフォルトのシグナル番号 **6（SIGABRT）**を指定した場合、プロセスを停止する際に該当プロセスが動作するカレントディレクトリに core ファイルを作成するため、解析に役立てることが可能です。
  該当のプロセス停止時に core ファイルを作成したくない場合は、任意にシグナル番号を変更してください。

- init 等の停止することのできないプロセスや defunct 状態のプロセスは停止されません。

- Serviceguard 連携は、以下のプロセスリソース監視についてのみ可能です。
  - ・メモリリーク監視
  - ・ファイルリーク監視
  - ・プロセス毎のオープンファイル数上限監視
  - ・高 CPU 使用率プロセス監視
  - ・スレッドリーク監視
  - ・プロセス毎のスレッド数上限監視

  システムリソース監視についての Serviceguard 連携は、スクリプト実行アクションを利用することで可能です。

- Serviceguard と連携して、pcheck(1M)によるプロセス監視を実施している場合、pcheck(1M)を kill(2) システムコール等によって終了しないようにしてください。
  pcheck(1M)の終了によって、パッケージ切り替えやノード切り替えを実行します。
  Serviceguard と連携している場合に、プロセス監視を停止・再開する場合の手順については、ProcessSaver のユーザーズガイド等を参照してください。

- Serviceguard と連携する場合、ProcessSaver によるプロセス監視のリトライオーバアクションには exit を指定してください。

- Serviceguard と ProcessSaver の連携に関する注意事項は、ProcessSaver のユーザーズガイドを参照してください。

- 異常検出時のアクションに PROCESS_KILL または SCRIPT_EXEC を指定した場合は、リソース監視異常検出のメッセージについてもシステムログへ出力されるように、SYSLOG_REPORT の指定も行ってください。

4 SG の設定

４．１　SG ファイルの記述

基本的には SG ファイルはデフォルト値のまま変更する必要はありません。
SG ファイルの内容は以下のようになります。

（1）SG ファイルの配置

SG ファイルは下記のディレクトリで管理されます。
ディレクトリにはデフォルトの SG ファイルが事前に用意されています。
基本的に SG の値を変更する必要はありませんが、システム要件等により
変更することは可能です。

SG ファイルディレクトリは以下のとおりです。

/var/opt/HA/PSSM/conf/

（2）SG ファイルの種類

SG ファイルは、本製品全体の動作を規定するシステム定義ファイル **(pssm.conf)**
と異常検出時の動作を規定するアクション定義ファイル **(psaction.conf)** があります。

用意されているファイルは以下のとおりです。

/var/opt/HA/PSSM/conf/pssm.conf
psaction.conf
psact_notstop.conf

pssm.conf.default
psaction.conf.default
psact_notstop.conf.default

pssm.conf.default、**psaction.conf.default** および **psact_notstop.conf.default** は
デフォルト値を記述しています。
各 SG の設定を変更する際の参考としてご利用ください。
実際にシステムに反映されるファイルは **pssm.conf**、**psaction.conf** および
psact_notstop.conf です。
pssm.conf.default、**psaction.conf.default** および **psact_notstop.conf.default** は
変更しないでください。

以下のファイルは、デフォルトでは用意されていませんが、ユーザが任意に作成することで、
その機能を利用することが可能です。

/var/opt/HA/PSSM/conf/psact_stop_allow.conf

（注）各ファイルの詳細については、「 4.2 章 SG ファイルの設定値」を参照してください。

（3）SG ファイルの変更手順

SG の各設定値の変更は、エディタ等で行ってください。
（注）SG ファイルを変更した後は必ず本製品の再起動を行う必要があります。
再起動を行わない場合、SG ファイルの変更内容は反映されません。
※ マシンの再起動は不要です。

４．２　SG ファイルの設定値

（1）pssm.conf の設定値について

pssm.conf は、本製品全体の動作を規定するシステム定義ファイルです。
pssm.conf に指定するパラメータは、監視を定義するパラメータと解析を定義するパラメータの 2 種類があります。

以下に pssm.conf に指定するパラメータを記述します。

監視定義パラメータ	
項目	説明
MONITOR_INTERVAL	統計情報を収集する間隔を分単位で指定します。 指定値は 1～1440（分=60*24 分=1 日）の範囲です。 デフォルト値は 5（分）です。 ***ANALYZE_INTERVAL*** より小さい値を指定してください。
ANALYZE_INTERVAL	統計情報を解析する間隔を分単位で指定します。 指定値は 1～10080（分=60*24*7 分=1 週間）の範囲です。 デフォルト値は 60（分）です。 ***MONITOR_INTERVAL*** より大きい値で、***MONITOR_INTERVAL*** の 100 倍以内の値を指定してください。
SCRIPTEXEC_INTERVAL	統計情報スクリプトを実行する間隔を分単位で指定します。 指定値は 1～10080（分=60*24*7 分=1 週間）の範囲です。 デフォルト値は 1440（分=1 日）です。
MONITOR_FILE_SIZE	統計情報ファイルの最大サイズを MByte 単位で指定します。 指定値は 1～100（MByte）です。 デフォルト値は 30 (MByte)です。
MONITOR_FILE_NUM	統計情報ファイルのバックアップ数を指定します。 指定値は 1～15（個）です。 デフォルト値は 10（個）です。 例）psmonitor.dat.save1、psmonitor.dat.save2
OUTPUT_DATA_FILE_PATH	統計情報ファイル、解析結果ファイルおよびカーネルパラメータ統計情報ファイル、全カーネルパラメータ詳細情報ファイルの出力先ディレクトリを指定します。 指定可能なディレクトリ名の長さは 512 文字以内です。 デフォルトは ***/var/opt/HA/PSSM/log*** です。 指定された出力先ディレクトリが存在しない場合はデフォルトのディレクトリ配下に出力されます。 出力先は絶対パスで指定してください。 ファイル名を指定することはできません。
CPU_MONITOR	CPU 個別情報の収集の可否を指定します。 ***ENABLE*** ：CPU 個別情報を収集し、統計情報ファイルに出力します。 デフォルトは ***ENABLE*** です。 ***DISABLE*** ：CPU 個別情報を収集しません。

KERNEL_MONITOR	カーネルパラメータ情報の収集の可否を指定します。 ***ENABLE*** ：カーネルパラメータ情報を収集し、カーネルパラメータ統計情報ファイル、全カーネルパラメータ詳細情報ファイルに出力します。デフォルトは***ENABLE*** です。 ***DISABLE***：カーネルパラメータ情報を収集しません。
KERNEL_MONITOR_INTERVAL	カーネルパラメータ統計情報の収集を実行する間隔を分単位で指定します。指定値は5～1440（分=60*24 分=1 日）の範囲です。 デフォルト値は10（分）です。
KERNEL_MONITOR_FILE_SIZE	カーネルパラメータ統計情報ファイルの最大サイズを MByte 単位で指定します。指定値は1～100（MByte）です。 デフォルト値は5 (MByte)です。
KERNEL_MONITOR_FILE_NUM	カーネルパラメータ統計情報ファイルのバックアップ数を指定します。指定値は1～15（個）です。 デフォルト値は3（個）です。 例）pssmkcmonitor.csv.YYYYMMDDhhmmss.Z

:

解析定義パラメータ（プロセスリソース関連）	
項目	説明
DEFUNCT_CHECK	ゾンビプロセス（defunct）の検出の可否を指定します。 ***ENABLE*** ：ゾンビプロセス（defunct）の検出を行います。 デフォルトは***ENABLE*** です。 ***DISABLE***：ゾンビプロセス（defunct）の検出を行いません。
DEFUNCT_CHECK_NUM	ゾンビプロセス（defunct）と判断する上限値（回）を指定します。 指定値は0～10000（回）の範囲です。 デフォルト値は288（回=1 日）です。 指定した回数、連続でゾンビプロセス（defunct）があった場合にゾンビプロセス（defunct）監視異常と判断します。 ***DEFUNCT_CHECK*** が***ENABLE*** の場合のみ有効です。 0 を指定した場合、***DEFUNCT_CHECK*** が ***ENABLE*** であってもゾンビプロセス（defunct）の検出を行いません。
MEMORY_LEAK_CHECK	プロセス単位のメモリリーク検出の可否を指定します。 ***ENABLE*** ：プロセス単位のメモリリークの検出を行います。 デフォルトは***ENABLE*** です。 ***DISABLE***：プロセス単位のメモリリーク検出を行いません。
MEMORY_LEAK_CHECK_NUM	メモリリークが発生していると判断する上限値（回）を指定します。 指定値は0～10000（回）の範囲です。 指定した回数メモリが増加し、初期値から増加率が***MEMORY_LEAK_CHECK_RATE*** で指定された率（%）を超えた場合にメモリリーク監視異常を検出します。 デフォルト値は288（回=1 日）です。 ***MEMORY_LEAK_CHECK*** が***ENABLE*** の場合のみ有効です。 0 を指定した場合、***MEMORY_LEAK_CHECK*** が ***ENABLE*** であってもプロセス単位のメモリリーク検出を行いません。
MEMORY_LEAK_CHECK_RATE	メモリリークが発生していると判断する上限値を指定します。 メモリが増加した回数が***MEMORY_LEAK_CHECK_NUM*** を超えて、 （現在のメモリ量 - 初回のメモリ量）/ 初回のメモリ量＊100 で算出した値が本パラメータで設定された値を超えた場合にメモリリーク監視異常を検出します。 指定値の範囲は0～1000（%）の範囲です。 デフォルト値は10（%）です。 ***MEMORY_LEAK_CHECK*** が***ENABLE*** の場合のみ有効です。 0 を指定した場合、***MEMORY_LEAK_CHECK*** が ***ENABLE*** であってもプロセス単位のメモリリーク検出を行いません。
FILE_LEAK_CHECK	プロセス単位のファイルリークおよびオープンファイル数上限監視の可否を指定します。 ***ENABLE*** ：プロセス単位のファイルリークの検出およびオープンファイル数上限監視を行います。 デフォルトは***ENABLE*** です。 ***DISABLE***：プロセス単位のファイルリークの検出およびオープンファイル数上限監視を行いません。

FILE_LEAK_CHECK_NUM	ファイルリークと判断する上限値（回）を指定します。 指定値は0～10000（回）の範囲です。 デフォルト値は288（回=1日）です。 指定された回数、オープンファイル数が増加した場合にファイルリーク監視異常を検出します。 ***FILE_LEAK_CHECK*** が ***ENABLE*** の場合のみ有効です。 0 を指定した場合、***FILE_LEAK_CHECK*** が ***ENABLE*** であってもファイルリーク検出を行いません。
FILE_LEAK_CHECK_RATE	プロセス単位のオープンファイル数上限に対してオープンファイル数上限監視異常と判断する上限値（%）を指定します。 指定値は0～100（%）の範囲です。 デフォルト値は90（%）です。 ***FILE_LEAK_CHECK*** が ***ENABLE*** の場合のみ有効です。 0 を指定した場合、***FILE_LEAK_CHECK*** が ***ENABLE*** であってもプロセス毎のオープンファイル数上限監視を行いません。
CPU_THRESHOLD_CHECK	高CPU使用率プロセス検出の可否を指定します。 ***ENABLE*** ：高CPU使用率プロセスの検出を行います。 デフォルトは***ENABLE*** です。 ***DISABLE***：高CPU使用率プロセスの検出を行いません。
CPU_CHECK_NUM	高CPU使用率プロセスと判断する上限値（回）を指定します。 指定値は0～10000（回）の範囲です。 デフォルト値は288（回=1日）です。 連続して指定された回数、***CPU_CHECK_RATE*** で指定された率（%）を超えた場合に高CPU使用率プロセス監視異常を検出します。 ***CPU_THRESHOLD_CHECK*** が ***ENABLE*** の場合のみ有効です。 0 を指定した場合、***CPU_THRESHOLD_CHECK*** が ***ENABLE*** であっても高CPU使用率プロセス検出を行いません。
CPU_CHECK_RATE	高CPU使用率プロセスと判断する上限値（%）を指定します。 指定値は0～100（%）の範囲です。 デフォルト値は90（%）です。 指定された上限値（%）を ***CPU_CHECK_NUM*** で指定した回数連続で超えた場合に高CPU使用率プロセス監視異常を検出します。 ***CPU_THRESHOLD_CHECK*** が ***ENABLE*** の場合のみ有効です。 0 を指定した場合、***CPU_THRESHOLD_CHECK*** が ***ENABLE*** であっても高CPU使用率プロセス検出を行いません。
PROC_COUNT_LIMITED_CHECK	プロセス多重度監視の可否を指定します。 ***ENABLE*** ：プロセス多重度監視を行います。 ***DISABLE***：プロセス多重度監視を行いません。 デフォルトは***DISABLE*** です。

PROC_COUNT_LIMITED_NUM	プロセス多重度の上限値（個）を指定します。 指定値は0～10000（個）の範囲です。 デフォルト値は100（個）です。 指定された個数以上、プロセスが多重起動されるとプロセス多重度監視異常を検出します。 ***PROC_COUNT_LIMITED_CHECK*** が ***ENABLE*** の場合のみ有効です。 0 を指定した場合、***PROC_COUNT_LIMITED_CHECK*** が ***ENABLE*** であってもプロセス多重度監視を行いません。
THREAD_COUNT_CHECK	プロセス単位のスレッドリークおよびスレッド数上限の監視の可否を指定します。 ***ENABLE*** ：プロセス単位のスレッドリークの検出およびスレッド数上限の監視を行います。 デフォルトは ***ENABLE*** です。 ***DISABLE***：プロセス単位のスレッドリークの検出およびスレッド数上限の監視を行いません。
THREAD_CHECK_NUM	スレッドリークと判断する上限値（回）を指定します。 指定値は0～10000（回）の範囲です。 デフォルト値は288（回=1 日）です。 連続して指定された回数、スレッド数が増加した場合にスレッドリーク監視異常を検出します。 ***THREAD_COUNT_CHECK*** が ***ENABLE*** の場合のみ有効です。 0 を指定した場合、***THREAD_COUNT_CHECK*** が ***ENABLE*** であってもスレッドリークの検出を行いません。
THREAD_CHECK_RATE	プロセス単位のスレッド数上限に対してスレッド数上限監視異常と判断する上限値（%）を指定します。 指定値は0～100（%）の範囲です。 デフォルト値は90（%）です。 ***THREAD_COUNT_CHECK*** が ***ENABLE*** の場合のみ有効です。 0 を指定した場合、***THREAD_COUNT_CHECK*** が ***ENABLE*** であってもプロセス毎のスレッド数上限監視を行いません。

解析定義パラメータ（システムリソース関連）	
説明	説明
KERNEL_LIMIT_CHECK	システムリソース（カーネルリソース）チェックの可否を指定します。 ***ENABLE***：システムリソースの上限検出を行います。 デフォルトは***ENABLE***です。 ***DISABLE***：システムリソースの上限検出を行いません。
SYS_PROC_CHECK_RATE	システム全体の起動プロセス数上限監視異常と判断する上限値（%）を指定します。 指定値は0～100（%）の範囲です。 デフォルト値は90（%）です。 カーネルパラメータ（nproc）が指定された上限値（%）を超えた場合にプロセス数上限監視異常を検出します。 0を指定した場合、***KERNEL_LIMIT_CHECK***が***ENABLE***であってもプロセス数上限監視を行いません。
SYS_MEMORY_CHECK_RATE	システム全体のメモリ量上限監視異常と判断する上限値（%）を指定します。 指定値は0～100（%）の範囲です。 デフォルト値は90（%）です。 システム全体のメモリ使用量が指定された上限値（%）を超えた場合にメモリ量上限監視異常を検出します。 0を指定した場合、***KERNEL_LIMIT_CHECK***が***ENABLE***であってもメモリ量上限監視を行いません。
SYS_FILE_CHECK_RATE	システム全体のオープンファイル数上限監視異常と判断する上限値（%）を指定します。 指定値は0～100（%）の範囲です。 デフォルト値は90（%）です。 カーネルパラメータ（nfile）が指定された上限値（%）を超えた場合にオープンファイル数上限監視異常を検出します。 0を指定した場合、***KERNEL_LIMIT_CHECK***が***ENABLE***であってもオープンファイル数上限監視を行いません。
SYS_LOCKF_CHECK_RATE	システム全体のロックファイル数上限監視異常と判断する上限値（%）を指定します。 指定値は0～100（%）の範囲です。 デフォルト値は90（%）です。 カーネルパラメータ（nflocks）が指定された上限値（%）を超えた場合にロックファイル数上限監視異常を検出します。 0を指定した場合、***KERNEL_LIMIT_CHECK***が***ENABLE***であってもロックファイル数上限監視を行いません。
SYS_CPU_CHECK_RATE	システム全体のCPU使用率上限監視異常と判断する上限値（%）を指定します。 指定値は0～100（%）の範囲です。 デフォルト値は90（%）です。 システムのCPU使用率が指定された上限値（%）を超えた場合にCPU使用率上限監視異常を検出します。 0を指定した場合、***KERNEL_LIMIT_CHECK***が***ENABLE***であってもCPU使用率上限監視を行いません。

SYS_THREAD_CHECK_RATE	システム全体のスレッド数上限監視異常と判断する上限値（%）を指定します。 指定値は0～100（%）の範囲です。 デフォルト値は90（%）です。 カーネルパラメータ（nkthread）が指定された上限値（%）を超えた場合にスレッド数上限監視異常を検出します。 0 を指定した場合、***KERNEL_LIMIT_CHECK*** が ***ENABLE*** であってもスレッド数上限監視を行いません。
SYS_SWAP_CHECK_RATE	システム全体のスワップメモリ量上限監視異常と判断する上限値（%）を指定します。 指定値は0～100（%）の範囲です。 デフォルト値は90（%）です。 システムのスワップ領域サイズが指定された上限値（%）を超えた場合にスワップメモリ量上限監視異常を検出します。 0 を指定した場合、***KERNEL_LIMIT_CHECK*** が ***ENABLE*** であってもスワップメモリ量上限監視を行いません。
SYS_MAXUPROC_CHECK_RATE	ユーザ起動プロセス数上限監視異常と判断する上限値（%）を指定します。 指定値は0～100（%）の範囲です。 デフォルト値は90（%）です。 カーネルパラメータ（maxuprc）が指定された上限値（%）を超えた場合にユーザプロセス数上限監視異常を検出します。 0 を指定した場合、***KERNEL_LIMIT_CHECK*** が ***ENABLE*** であってもユーザ起動プロセス数上限監視を行いません。
SYS_PROC_CHECK_TIME	システム全体の起動プロセス数上限監視異常と判断する連続超過時間を分単位で指定します。 指定値は1～1440（分=60*24 分=1 日）の範囲で、***MONITOR_INTERVAL*** で指定された値の正の整数倍である必要があります。正の整数倍でない値を指定した場合は、この値を MONITOR_INTERVAL で割ったときの商（小数点以下切捨て）+1 を MONITOR_INTERVAL にかけた値で動作します。 デフォルト値は60（分）です。 連続して指定された時間、***SYS_PROC_CHECK_RATE*** で指定された率（%）を超えた場合にプロセス数上限監視異常を検出します。 ***KERNEL_LIMIT_CHECK*** が ***ENABLE*** の場合、かつ ***SYS_PROC_CHECK_RATE*** に指定された値が 0 でない場合のみ有効です。

SYS_MEMORY_CHECK_TIME	システム全体のメモリ量上限監視異常と判断する連続超過時間を分単位で指定します。 指定値は1～1440（分=60*24 分=1 日）の範囲で、***MONITOR_INTERVAL*** で指定された値の正の整数倍である必要があります。正の整数倍でない値を指定した場合は、この値を MONITOR_INTERVAL で割ったときの商（小数点以下切捨て）+1 を MONITOR_INTERVAL にかけた値で動作します。 デフォルト値は60（分）です。 連続して指定された時間、***SYS_MEMORY_CHECK_RATE*** で指定された率（%）を超えた場合にメモリ量上限監視異常を検出します。 ***KERNEL_LIMIT_CHECK*** が ***ENABLE*** の場合、かつ ***SYS_MEMORY_CHECK_RATE*** に指定された値が0でない場合のみ有効です。
SYS_FILE_CHECK_TIME	システム全体のオープンファイル数上限監視異常と判断する連続超過時間を分単位で指定します。 指定値は1～1440（分=60*24 分=1 日）の範囲で、***MONITOR_INTERVAL*** で指定された値の正の整数倍である必要があります。正の整数倍でない値を指定した場合は、この値を MONITOR_INTERVAL で割ったときの商（小数点以下切捨て）+1 を MONITOR_INTERVAL にかけた値で動作します。 デフォルト値は60（分）です。 連続して指定された時間、***SYS_FILE_CHECK_RATE*** で指定された率（%）を超えた場合にオープンファイル数上限監視異常を検出します。 ***KERNEL_LIMIT_CHECK*** が ***ENABLE*** の場合、かつ ***SYS_FILE_CHECK_RATE*** に指定された値が 0 でない場合のみ有効です。
SYS_LOCKF_CHECK_TIME	システム全体のロックファイル数上限監視異常と判断する連続超過時間を分単位で指定します。 指定値は1～1440（分=60*24 分=1 日）の範囲で、***MONITOR_INTERVAL*** で指定された値の正の整数倍である必要があります。正の整数倍でない値を指定した場合は、この値を MONITOR_INTERVAL で割ったときの商（小数点以下切捨て）+1 を MONITOR_INTERVAL にかけた値で動作します。 デフォルト値は60（分）です。 連続して指定された時間、***SYS_LOCKF_CHECK_RATE*** で指定された率（%）を超えた場合にロックファイル数上限監視異常を検出します。 ***KERNEL_LIMIT_CHECK*** が ***ENABLE*** の場合、かつ ***SYS_LOCKF_CHECK_RATE*** に指定された値が0でない場合のみ有効です。

SYS_CPU_CHECK_TIME	システム全体の CPU 使用率監視異常と判断する連続超過時間を分単位で指定します。 指定値は 1～1440（分=60*24 分=1 日）の範囲で、***MONITOR_INTERVAL*** で指定された値の正の整数倍である必要があります。正の整数倍でない値を指定した場合は、この値を MONITOR_INTERVAL で割ったときの商（小数点以下切捨て）+1 を MONITOR_INTERVAL にかけた値で動作します。 デフォルト値は 60（分）です。 連続して指定された時間、***SYS_CPU_CHECK_RATE*** で指定された率（%）を超えた場合に CPU 使用率監視異常を検出します。 ***KERNEL_LIMIT_CHECK*** が ***ENABLE*** の場合、かつ ***SYS_CPU_CHECK_RATE*** に指定された値が 0 でない場合のみ有効です。
SYS_THREAD_CHECK_TIME	システム全体のスレッド数上限監視異常と判断する連続超過時間を分単位で指定します。 指定値は 1～1440（分=60*24 分=1 日）の範囲で、***MONITOR_INTERVAL*** で指定された値の正の整数倍である必要があります。正の整数倍でない値を指定した場合は、この値を MONITOR_INTERVAL で割ったときの商（小数点以下切捨て）+1 を MONITOR_INTERVAL にかけた値で動作します。 デフォルト値は 60（分）です。 連続して指定された時間、***SYS_THREAD_CHECK_RATE*** で指定された率（%）を超えた場合にスレッド数上限監視異常を検出します。 ***KERNEL_LIMIT_CHECK*** が ***ENABLE*** の場合、かつ ***SYS_THREAD_CHECK_RATE*** に指定された値が 0 でない場合のみ有効です。
SYS_SWAP_CHECK_TIME	システム全体のスワップメモリ量上限監視異常と判断する連続超過時間を分単位で指定します。 指定値は 1～1440（分=60*24 分=1 日）の範囲で、***MONITOR_INTERVAL*** で指定された値の正の整数倍である必要があります。正の整数倍でない値を指定した場合は、この値を MONITOR_INTERVAL で割ったときの商（小数点以下切捨て）+1 を MONITOR_INTERVAL にかけた値で動作します。 デフォルト値は 60（分）です。 連続して指定された時間、***SYS_SWAP_CHECK_RATE*** で指定された率（%）を超えた場合にスワップメモリ量上限監視異常を検出します。 ***KERNEL_LIMIT_CHECK*** が ***ENABLE*** の場合、かつ ***SYS_SWAP_CHECK_RATE*** に指定された値が 0 でない場合のみ有効です。

SYS_MAXUPROC_CHECK_TIME	システム全体のユーザ起動プロセス数上限監視異常と判断する連続超過時間を分単位で指定します。 指定値は1～1440（分=60*24 分=1 日）の範囲で、***MONITOR_INTERVAL*** で指定された値の正の整数倍である必要があります。正の整数倍でない値を指定した場合は、この値を MONITOR_INTERVAL で割ったときの商（小数点以下切捨て）+1 を MONITOR_INTERVAL にかけた値で動作します。 デフォルト値は60（分）です。 連続して指定された時間、***SYS_MAXUPROC_ CHECK_RATE*** で指定された率（%）を超えた場合にユーザ起動プロセス数上限監視異常を検出します。 ***KERNEL_LIMIT_CHECK*** が ***ENABLE*** の場合、かつ ***SYS_MAXUPROC_CHECK_RATE*** に指定された値が 0 でない場合のみ有効です。

pssm.conf の設定例は以下のとおりです。

</var/opt/HA/PSSM/conf/pssm.conf>

```
#################################################################
# System Config Area
#################################################################

# Monitor interval timer (minutes)
# minimum = 1, maximum = 1440(1day), default = 5
MONITOR_INTERVAL            5

# Analyze interval timer (minutes)
# This parameter is larger than MONITOR_INTERVAL.
# minimum = 1, maximum = 10080(1week), default = 60
ANALYZE_INTERVAL            60

# Monitor file size (MByte)
# minimum = 1, maximum = 100, default = 30
MONITOR_FILE_SIZE           30

# Backup num of monitor file (num)
# minimum = 1, maximum = 15, default = 10
MONITOR_FILE_NUM            10

# Output path of monitor file
# default = /var/opt/HA/PSSM/log
# The maximum length of path is 512 bytes.
OUTPUT_DATA_FILE_PATH       /var/opt/HA/PSSM/log

# Monitoring CPU information
# monitor = ENABLE : not monitor = DISABLE
CPU_MONITOR                 ENABLE

# Monitoring kernel parameter information
# monitor = ENABLE : not monitor = DISABLE
KERNEL_MONITOR              ENABLE

# Kernel monitor interval timer(minutes)
# minimum = 5, maximum = 1440(1day), default = 10
KERNEL_MONITOR_INTERVAL         10

# Kernel monitor file size(MByte)
# minimum = 1, maximum = 100, default = 5
KERNEL_MONITOR_FILE_SIZE        5

# Backup num of kernel monitor file(num)
# minimum = 1, maximum = 15, default = 3
KERNEL_MONITOR_FILE_NUM         3
```

```
#############################################################
# Analyze Information & Threshold of each process
#############################################################

########## DEFUNCT_CHECK ##########

# nonactivated process defunct check (Not used)
# check = ENABLE : not check = DISABLE
DEFUNCT_CHECK                   ENABLE

# Threshold of defunct process (num)
# minimum = 1, maximum = 10000, default = 288(1 day)
DEFUNCT_CHECK_NUM               288

######### MEMORY_LEAK_CHECK ########

# nonactivated process memory leak check (Not used)
# check = ENABLE : not check = DISABLE
MEMORY_LEAK_CHECK               ENABLE

# Threshold of memory leak process (num)
# minimum = 1, maximum = 10000, default = 288(1 day)
MEMORY_LEAK_CHECK_NUM           288

# Threshold of memory leak process (%)
# minimum = 1, maximum = 1000, default = 10
MEMORY_LEAK_CHECK_RATE          10

########## FILE_LEAK_CHECK ##########

# nonactivated process file descriptor leak check (Not used)
# check = ENABLE : not check = DISABLE
FILE_LEAK_CHECK             ENABLE

# Threshold of file descriptor leak process (num)
# minimum = 1, maximum = 10000, default = 288(1 day)
FILE_LEAK_CHECK_NUM         288

# Threshold of maxfiles_lim upper process (%)
# minimum = 1, maximum = 100, default = 90
FILE_LEAK_CHECK_RATE        90

######## PROCESS_COUNT_CHECK ########

# nonactivated process num limited check (Not used)
# check = ENABLE : not check = DISABLE
PROC_COUNT_LIMITED_CHECK    DISABLE

# Threshold of limited orver process (num)
# minimum = 1, maximum = 10000, default = 100
PROC_COUNT_LIMITED_NUM      100
```

```
########## THREAD_CHECK ##########

# nonactivated process thread orver check (Not used)
# check = ENABLE : not check = DISABLE
THREAD_COUNT_CHECK          ENABLE

# Threshold of thread num leak process (num)
# minimum = 1, maximum = 10000, default = 288(1 day)
THREAD_CHECK_NUM            288

# Threshold of max_thread_proc upper process (%)
# minimum = 1, maximum = 100, default = 90
THREAD_CHECK_RATE           90

######## CPU_THRESHOLD_CHECK ########

# nonactivated cpu utilization check (Not used)
# check = ENABLE : not check = DISABLE
CPU_THRESHOLD_CHECK         ENABLE

# Threshold of CPU utilization process (num)
# minimum = 1, maximum = 10000, default = 288(1 day)
CPU_CHECK_NUM               288

# Threshold of CPU utilization process (%)
# minimum = 1, maximum = 100, default = 90
CPU_CHECK_RATE              90

########## KERNEL_CHECK ##########

# nonactivated kernel parameter upper check (Not used)
# check = ENABLE : not check = DISABLE
KERNEL_LIMIT_CHECK          ENABLE


# Threshold of kernel parameter upper check (%)
# minimum = 1, maximum = 100, default = 90, disable = 0
SYS_PROC_CHECK_RATE         90
SYS_MEMORY_CHECK_RATE       90
SYS_FILE_CHECK_RATE         90
SYS_LOCKF_CHECK_RATE        90
SYS_CPU_CHECK_RATE          90
SYS_THREAD_CHECK_RATE       90
SYS_SWAP_CHECK_RATE         90
SYS_MAXUPROC_CHECK_RATE     90


# kernel parameter upper check timer (minutes)
# minimum = 1, maximum = 1440 (1day), default = 60
SYS_PROC_CHECK_TIME         60
SYS_MEMORY_CHECK_TIME       60
SYS_FILE_CHECK_TIME         60
SYS_LOCKF_CHECK_TIME        60
SYS_CPU_CHECK_TIME          60
SYS_THREAD_CHECK_TIME       60
SYS_SWAP_CHECK_TIME         60
SYS_MAXUPROC_CHECK_TIME     60
```

（2）psaction.conf の設定値について

psaction.conf は、異常検出時の動作を既定するアクション定義ファイルです。

以下に psaction.conf に指定するパラメータおよび指定可能なアクションタイプを記述します。

※複数指定することが可能です。

指定可能なアクションタイプ	
項目	説明
TRACE_REPORT	テキストログファイルへメッセージを出力します。 デフォルトで必ず出力され、設定を変更することはできません。
SYSLOG_REPORT	システムログファイルへメッセージを出力します。 ***SYSLOG_REPORT*** のオプションは以下を指定してください。 ***ERROR*** ：出力レベルを ***ERROR*** とします。 ***WARNING*** ：出力レベルを ***WARNING*** とします。 ***INFO*** ：出力レベルを ***INFO*** とします。 デフォルトは ***WARNING*** です。
PROCESS_KILL	対象プロセスに kill(2) システムコールを送り、停止します。 ***PROCESS_KILL*** のオプションは送信するシグナル番号を指定してください。 指定可能なシグナルは ***6(SIGABRT)***、***9(SIGKILL)***、および ***15(SIGTERM)***とします。 デフォルトは ***6(SIGABRT)***です。
SCRIPT_EXEC	指定されたスクリプトを実行します。 ***SCRIPT_EXEC*** のオプションは実行可能なスクリプトを指定してください。 指定可能なスクリプト名の長さは 1022 文字以内です。 デフォルトは ***/var/opt/HA/PSSM/bin/collect_info.sh*** です。 上記スクリプトは障害解析に有効な情報を収集します。 スクリプト指定の詳細については前述の章を参照してください。

動作定義パラメータ	
項目	説明
DEFUNCT_ERROR_ACTION	ゾンビプロセス（defunct）を検出した場合に実行するアクションを指定します。 デフォルトは***TRACE_REPORT*** です。 （注）本アクションに***PROCESS_KILL*** を指定することはできません。
MEMORY_LEAK_ERROR_ACTION	メモリリークと判断されるプロセスを検出した場合に実行するアクションを指定します。 デフォルトは***TRACE_REPORT*** です。
FILE_LEAK_ERROR_ACTION	ファイルリークと判断されるプロセスを検出した場合に実行するアクションを指定します。 デフォルトは***TRACE_REPORT*** です。
MAX_FILE_ERROR_ACTION	プロセス毎のオープンファイル数の上限監視異常のプロセスを検出した場合に実行するアクションを指定します。 デフォルトは***TRACE_REPORT*** です。
CPU_ERROR_ACTION	高 CPU 使用率プロセスと判断されるプロセスを検出した場合に実行するアクションを指定します。 デフォルトは***TRACE_REPORT*** です。
PROC_COUNT_ERROR_ACTION	プロセス多重度監視異常と判断されるプロセスを検出した場合に実行するアクションを指定します。 デフォルトは***TRACE_REPORT*** です。 （注）本アクションに***PROCESS_KILL*** を指定することはできません。
THREAD_LEAK_ERROR_ACTION	スレッドリークと判断されるプロセスを検出した場合に実行するアクションを指定します。 デフォルトは***TRACE_REPORT*** です。
MAX_THREAD _ERROR_ACTION	プロセス毎のスレッド数の上限監視異常を検出した場合に実行するアクションを指定します。 デフォルトは***TRACE_REPORT*** です。
KERNEL_ERROR_ACTION	システム全体のリソースの上限監視異常を検出した場合に実行するアクションを指定します。 デフォルトは***TRACE_REPORT*** です。 （注）本アクションに***PROCESS_KILL*** を指定することはできません。

設定例は以下のようになります。

- **DEFUNCT_ERROR_ACTION　SYSLOG_REPORT : ERROR**
  ゾンビプロセスを検出した場合に、システムログにメッセージ ERROR で出力します。
- **MEMORY_LEAK_ERROR_ACTION　PROCESS_KILL:9**
  メモリリークを検出した場合に、9（SIGKILL）のシグナルを送信し、該当のプロセスを停止します。

psaction.conf の設定例は以下のとおりです。

</var/opt/HA/PSSM/conf/psaction.conf>

```
##      Copyright (c) 2006-2011 NEC Corporation                 ##
##      NEC CONFIDENTIAL AND PROPRIETARY                        ##
##      All rights reserved by NEC Corporation.                 ##
##      This program must be used solely for the purpose for    ##
##      which it was furnished by NEC Corporation.   No part    ##
##      of this program may be reproduced or disclosed to       ##
##      others, in any form, without the prior written          ##
##      permission of NEC Corporation.    Use of copyright      ##
##      notice does not evidence publication of the program.    ##

# psaction.conf(HA/SystemResourceMonitor action Configuration)

#############################################################
# Action Information
#############################################################

# The action when the defunct process is discovered is specified.
DEFUNCT_ERROR_ACTION            TRACE_REPORT
#DEFUNCT_ERROR_ACTION           SYSLOG_REPORT:WARNING
#DEFUNCT_ERROR_ACTION           SCRIPT_EXEC:/var/opt/HA/PSSM/bin/collect_info.sh

# The action when the memory leak process is discovered is specified.
MEMORY_LEAK_ERROR_ACTION        TRACE_REPORT
#MEMORY_LEAK_ERROR_ACTION       SYSLOG_REPORT:WARNING
#MEMORY_LEAK_ERROR_ACTION       PROCESS_KILL:6
#MEMORY_LEAK_ERROR_ACTION       SCRIPT_EXEC:/var/opt/HA/PSSM/bin/collect_info.sh

# The action when the file descriptor leak process is discovered is specified.
FILE_LEAK_ERROR_ACTION          TRACE_REPORT
#FILE_LEAK_ERROR_ACTION         SYSLOG_REPORT:WARNING
#FILE_LEAK_ERROR_ACTION         PROCESS_KILL:6
#FILE_LEAK_ERROR_ACTION         SCRIPT_EXEC:/var/opt/HA/PSSM/bin/collect_info.sh

# The action when the open file num over process is discovered is specified.
MAX_FILE_ERROR_ACTION           TRACE_REPORT
#MAX_FILE_ERROR_ACTION          SYSLOG_REPORT:WARNING
#MAX_FILE_ERROR_ACTION          PROCESS_KILL:6
#MAX_FILE_ERROR_ACTION          SCRIPT_EXEC:/var/opt/HA/PSSM/bin/collect_info.sh

# The action when the high CPU utilization process is discovered is specified.
CPU_ERROR_ACTION                TRACE_REPORT
#CPU_ERROR_ACTION               SYSLOG_REPORT:WARNING
#CPU_ERROR_ACTION               PROCESS_KILL:6
#CPU_ERROR_ACTION               SCRIPT_EXEC:/var/opt/HA/PSSM/bin/collect_info.sh

# The action when the many same processes is discovered is specified.
PROC_COUNT_ERROR_ACTION         TRACE_REPORT
#PROC_COUNT_ERROR_ACTION        SYSLOG_REPORT:WARNING
#PROC_COUNT_ERROR_ACTION        SCRIPT_EXEC:/var/opt/HA/PSSM/bin/collect_info.sh

# The action when the thread leak process is discovered is specified.
THREAD_LEAK_ERROR_ACTION        TRACE_REPORT
#THREAD_LEAK_ERROR_ACTION       SYSLOG_REPORT:WARNING
#THREAD_LEAK_ERROR_ACTION       PROCESS_KILL:6
#THREAD_LEAK_ERROR_ACTION       SCRIPT_EXEC:/var/opt/HA/PSSM/bin/collect_info.sh

# The action when the thread num orver process is discovered is specified.
MAX_THREAD_ERROR_ACTION         TRACE_REPORT
#MAX_THREAD_ERROR_ACTION        SYSLOG_REPORT:WARNING
#MAX_THREAD_ERROR_ACTION        PROCESS_KILL:6
#MAX_THREAD_ERROR_ACTION        SCRIPT_EXEC:/var/opt/HA/PSSM/bin/collect_info.sh

# The action when the kernel parameter upper error is discovered is specified.
KERNEL_ERROR_ACTION             TRACE_REPORT
#KERNEL_ERROR_ACTION            SYSLOG_REPORT:WARNING
#KERNEL_ERROR_ACTION            SCRIPT_EXEC:/var/opt/HA/PSSM/bin/collect_info.sh
```

（3）psact_notstop.conf の設定値について

psact_notstop.conf は、***PROCESS_KILL*** アクションの対象から除外するプロセスを規定する
プロセス KILL アクション除外定義ファイルです。
本ファイルに記述されたプロセスは、異常検出時の動作に ***PROCESS_KILL*** を指定している場合でも、
停止されません。***PROCESS_KILL*** アクションを指定した場合、停止させたくないプロセスはすべて
本ファイルに指定してください。
本ファイルに指定可能なプロセス数は 256 個です。

以下の形式で指定したプロセスについて、プロセスの停止は行われません。

- **/sbin/init.d/process1**
  プロセスを絶対パスで指定。
- **process2**
  プロセス名の basename のみ指定。
- **/sbin/init.d/process3 -a**
  プロセスに引数をつけて指定。
  （注）引数によって指定するプロセスを区別する場合、プロセス名は絶対パスで引数まで
  指定してください。

（設定例）/usr/sbin/cron、/usr/sbin/syslogd を ***PROCESS_KILL*** アクションの対象外とする場合。

- /usr/sbin/cron ★1
- /usr/sbin/syslogd -D ★2

</var/opt/HA/PSSM/conf/psact_notstop.conf>

```
# psact_notstop.conf(HA/SystemResourceMonitor process doesn't stop list)
#########################################################################
# Please specify the process name.
#    The specified process name is excluded from an action stop object
#    (max 256 parameter)
#    ex. pssmd
#    ex. /opt/HA/PSSM/bin/pssmd
#########################################################################
/opt/HA/PSSM/bin/pssmd
/opt/HA/PSSM/bin/psmonitor
/opt/HA/PSSM/bin/psanalyzer
/opt/HA/PSSM/bin/psaction
/opt/HA/PSSM/bin/pssmdatcp
/opt/HA/PSSM/bin/pssmutil

/usr/sbin/cron          ★1
/usr/sbin/syslogd -D    ★2
      :           :
```

（注）プロセス KILL アクション除外定義ファイルにデフォルトで設定されている
以下の本製品のプロセスは削除しないでください。

/opt/HA/PSSM/bin/pssmd
/opt/HA/PSSM/bin/psmonitor
/opt/HA/PSSM/bin/psanalyzer
/opt/HA/PSSM/bin/psaction
/opt/HA/PSSM/bin/pssmdatcp
/opt/HA/PSSM/bin/pssmutil
/opt/HA/PSSM/bin/pssmkcmonitor
/opt/HA/PSSM/bin/pssmkcview

（4）psact_stop_allow.conf の設定値について

psact_stop_allow.conf は、***PROCESS_KILL*** アクションの対象とするプロセスを指定する
プロセス KILL アクション定義ファイルです。
PROCESS_KILL アクションを指定した場合に、停止するプロセスを指定したい場合は
本ファイルを **/var/opt/HA/PSSM/conf/** 配下に作成し、対象となるプロセスを指定してください。
本ファイルに記述されたプロセスは、異常検出時の動作に ***PROCESS_KILL*** を指定している場合に、
kill(2) システムコールによって停止されます。

（注）本ファイルにプロセス名を記述しない場合は、全てのプロセスが停止処理の対象外となります。
つまり異常検出時の動作に ***PROCESS_KILL*** を指定している場合であっても、対象となる
プロセスの停止処理は実行されません。
（注）本ファイルが存在する場合、前述のプロセス KILL アクション除外定義ファイル
（psact_notstop.conf）の内容は反映されません。

本ファイルに指定可能なプロセス数は 256 個です。

以下の形式で指定したプロセスについて、プロセスの停止を行います。

- **/sbin/init.d/process1**
  プロセス名を絶対パスで指定。
- **process2**
  プロセス名の basename のみ指定。
- **/sbin/init.d/process3 -a**
  プロセス名を絶対パスで引数をつけて指定。

（注）引数によって指定するプロセスを区別する場合、プロセス名は絶対パスで引数まで
指定してください。

（設定例）/usr/sbin/cron、/usr/sbin/syslogd を ***PROCESS_KILL*** アクションの対象とする場合。

- /usr/sbin/cron ★1
- /usr/sbin/syslogd -D ★2

</var/opt/HA/PSSM/conf/psact_stop_allow.conf >

```
/usr/sbin/cron          ★1
/usr/sbin/syslogd -D    ★2
      :           :
```

5 ログメッセージ

5．1　本製品が出力するメッセージの形式

本製品の異常通知にはテキストログ出力及びシステムログ出力による 2 種類の方式があります。
テキストログは必ず出力され、設定の変更はできません。
設定に応じてシステムログへの出力も可能です。設定方法については「 4 章 SG の設定」を参照してください。
テキストログ及びシステムログの出力形式は以下のとおりです。

（1） テキストログメッセージのフォーマットは以下のとおりです。

```
Mon Nov 20 10:14:19 2006: xxxx : msg
```

- xxxx　:ログレベル
- msg　:メッセージ

テキストログの level は以下のとおりです。

level　：　LOG_WARNING　または　LOG_ERR、LOG_INFO

（2） システムログメッセージのフォーマットは以下のとおりです。

```
May 24 09:35:26 hostname xxxx[yyyy]: msg
```

- xxxx　:デーモン名(pssmd)
  :コマンド名(psmonitor, psanalyzer, psaction, psscriptexec)
- yyyy　:pid
- msg　:メッセージ

システムログの facility と level は以下のとおりです。

facility　：　LOG_USER
level　：　LOG_WARNING　または　LOG_ERR、LOG_INFO

システムログへ出力するための設定については、「 4 章 SG の設定」を参照してください。

（3） リソースに関する問題が検出されたメッセージ

本製品が出力するメッセージには、リソース監視異常の検出を報告するものがあります。
これらのメッセージは、運用管理ソフト等により監視することをお勧めします。
ただしデフォルトの設定の場合、以下のメッセージはシステムログには出力されませんので、
システムログに出力する場合は、SG ファイルを変更する必要があります。
SG ファイルの変更手順につきましては、「 4 章 SG の設定」を参照してください。

対象となるメッセージは下記のとおりです。
なお、これら以外のメッセージについては特に監視する必要はありません。
出力レベルのデフォルトは LOG_WARNING です。

[プロセスリソースに関するメッセージ]

Find a sign of process resource failure (***xxx***)

※ ***xxx*** 内は、検出された事象により以下となります。

・ゾンビプロセス（defunct）を検出した場合
type=defunct ,<プロセス名> ,pid=<yyy>

・メモリリークプロセスを検出した場合
type=memleak ,<プロセス名> ,pid=<yyy>

・ファイルリークプロセスを検出した場合
type=fileleak ,<プロセス名> ,pid=<yyy>

・プロセス単位のオープンファイル数が maxfiles_lim の閾値を超えた場合
type=maxfile ,<プロセス名> ,pid=<yyy>

・高 CPU 使用率プロセスを検出した場合
type=cpu ,<プロセス名> ,pid=<yyy>

・スレッドリークプロセスを検出した場合
type=threadleak ,<プロセス名> ,pid=<yyy>

・プロセス単位のスレッド数が max_thread_proc の閾値を超えた場合
type=maxthread ,<プロセス名> ,pid=<yyy>

・同一名プロセス数が閾値を超えた場合
type=proc_count ,<プロセス名> ,pid=<yyy>

[システムリソースに関するメッセージ]

システムリソースの使用率が現在も閾値を超えている場合

Find a sign of process resource failure now value is over the threshold (***xxx***)

システムリソースの使用率が一度は閾値を超えたが現在は正常な値の場合

Find a sign of process resource failure but now value is not over the threshold (***xxx***)

※ ***xxx*** 内は、検出された事象により以下となります。

・総プロセス数が nproc の閾値を超えた場合
type=nproc ,value=<yyy> ,maxvalue=<zzz>

・総オープンファイル数が nfile の閾値を超えた場合
type=nfile ,value=<yyy> ,maxvalue=<zzz>

・総ロックファイル数が nflocks の閾値を超えた場合
type=nflocks ,value=<yyy> ,maxvalue=<zzz>

・総メモリ使用量が物理メモリ量の閾値を超えた場合
type=memsize ,value=<yyy> ,maxvalue=<zzz>

・CPU 使用率が 100(%)×CPU 数の閾値を超えた場合
type=cpu ,value=<yyy> ,maxvalue=<zzz>

・総スレッド数が nkthread の閾値を超えた場合
type=nkthread ,value=<yyy> ,maxvalue=<zzz>

・総スワップメモリ使用量が使用可能なスワップ領域の閾値を超えた場合
type=swchunk ,value=<yyy> ,maxvalue=<zzz>

・総スワップメモリ予約量が使用可能なスワップ領域の閾値を超えた場合
type=swreserve ,value=<yyy> ,maxvalue=<zzz>

・総スワップメモリ使用量と総スワップメモリ予約量の合計した値が使用可能なスワップ領域の閾値を超えた場合
type=swchunk&swreserve ,value=<yyy> ,maxvalue=<zzz>

・ユーザプロセス数が maxuprc の閾値を超えた場合
type=maxuprc ,value=<yyy> ,maxvalue=<zzz> , uname=<aaa>

これらのメッセージが出力された場合、リソースに関するなんらかの問題が検出されたことを意味しますので、原因を調査してください。

6 注意・制限事項

本製品には、以下の注意・制限事項があります。

- 本製品の設定値は、デフォルトで使用することを推奨します。

- SG ファイルの値を変更した場合は、必ず本製品を再起動してください。
  再起動を行わない場合、変更内容は反映されません。

- 統計情報ファイルに記録されるリソース情報は、統計情報収集間隔（MONITOR_INTERVAL）ごとの瞬間のリソース情報であり、統計情報収集間隔（MONITOR_INTERVAL）内の継続的なリソース情報ではありません。
  そのため、統計情報の収集時点以外のタイミングで発生したリソース異常は検出されません。

- 本製品は、デフォルトで運用した場合、リソース異常検出時のメッセージはシステムログへは出力されません。リソース異常検出のメッセージをシステムログへ出力させる場合は、SG ファイルを変更する必要があります。

- 統計情報解析間隔（ANALYZE_INTERVAL）は、統計情報収集間隔（MONITOR_INTERVAL）以上の値を設定してください。
  統計情報解析間隔（ANALYZE_INTERVAL）が統計情報収集間隔（MONITOR_INTERVAL）より小さい値を指定された場合、統計情報収集間隔（MONITOR_INTERVAL）と同じ値で動作します。

- 統計情報解析間隔（ANALYZE_INTERVAL）は、統計情報収集間隔（MONITOR_INTERVAL）の正の整数倍の値を設定してください。

- 統計情報解析間隔（ANALYZE_INTERVAL）と統計情報収集間隔（MONITOR_INTERVAL）の差が 100 倍以上の場合、正常に監視が行われないことがあります。

- 以下のような場合には、リソース監視異常を検出できないことがあります。
  ・プロセス ID が随時変更されている場合
  ・ファイルリーク、メモリリーク、スレッドリークが、一定量を超えない範囲で増減している場合
  ・プロセスの CPU 使用率が上限値の前後で増減を繰り返している場合
  ・システム全体のリソース監視で、上限値の前後で増減を繰り返している場合

- 統計情報ファイルおよび解析結果ファイルの内容が不正な場合、その情報についてはリソース異常監視に反映されません。

- 本製品で収集している CPU 使用率上限監視では、システム全体に占める CPU 使用率の収集および監視を行っています。一般的なプロセスの CPU 使用率を表示するコマンド（/usr/bin/top コマンド）で表示される CPU 使用率とは算出方法の違いにより、値が異なる場合があります。
  したがって、top(1M) で常に上位にいるようなプロセスでも、CPU 使用率上限監視異常とは判定されない場合があります。

- システムリソースの閾値（%）を個別のリソースごとに指定する機能は、R3.1a 以降のバージョンより可能です。
  R3.1 以前のバージョンの場合、個別のリソースごとに閾値を指定することはできません。

- システムリソースの監視で連続超過時間（分）を指定する機能は、R3.1c 以降のバージョンより可能です。
  R3.1c より前のバージョンの場合は、システムリソースの閾値（%）を一度でも超えた場合に異常と判定されます。

- システムリソース上限監視において、異常検出時のアクションを個別のリソースごとに指定することはできません。

- 本製品で出力されるデータファイルは、R3.1a 以降ファイルフォーマットが変更となったため、任意にデータファイルの情報を使用している場合は、修正が必要となります。
  また、ファイルフォーマットの変更に伴い、データファイルのバックアップ数およびファイルサイズのデフォルト値が変更になりました。
  デフォルトで運用した場合、30MByte のファイルを 10 個までバックアップとして保存します。
  必要に応じて、SG ファイルの設定を行ってください。

- 本製品のデーモン(pssmd)、コマンド類(psmonitor, psanalyzer, psaction, psscriptexec, pssmkcmonitor)を単独で実行した場合、定期実行で正しく情報の収集および解析ができない場合があります。本製品は、必ず以下のコマンドを使用して起動してください。
  ```
  # /sbin/init.d/pssmd start
  ```

- date(1) コマンドなどで動作中にマシンタイムを変更した場合、正しく動作しない場合があります。マシンタイムを変更した場合は、必ず本製品を再起動してください。

- スクリプト実行アクションを指定する場合、以下の制約があります。
  - 1 つのリソース監視異常について、SCRIPT_EXEC 行を複数指定することはできません。
    2 つ以上のスクリプトを実行させたい場合は、SCRIPT_EXEC のオプションにコロン（:）
    区切りで実行するスクリプトを指定してください。

  - コロン（:）が含まれるスクリプトを指定することはできません。

  - SCRIPT_EXEC のオプションには、スクリプトを絶対パスで指定してください。
    指定可能なスクリプト名の長さは、引数を含め 1023 文字未満です。

  - スクリプト内で実行するコマンドは絶対パスで呼び出してください。
    コマンドへのパスが設定されていないとコマンド実行に失敗する場合があります。

  - 指定するスクリプトには必ず実行権を与えてください。実行権がない場合、スクリプトの
    実行に失敗します。

  - アクション実行はデフォルトで 5 分のタイムアウト値をもっているため、実行する処理は
    5 分以内に終了する必要があります。
    したがって、スクリプト内で長時間 sleep により待ち合わせを行う等、処理に時間が
    かかり 5 分を超えても終了しない場合、スクリプト実行は停止されます。
    また、スクリプト内でデーモン化されていないプロセスを起動する場合には、コマンドライン
    の最後に&を付与してバックグラウンドで起動し、スクリプトが 5 分以内に終了するよう
    記述してください。
    5 分を超えるスクリプトを実行する必要がある場合は、直接開発元へお問い合わせください。

  - スクリプトからプロセスを起動する場合に、起動するプロセスが環境変数に依存している
    場合は、その環境変数を設定してからプロセスを呼び出してください。

  - 異常検出時のアクションに SCRIPT_EXEC を指定した場合は、SYSLOG_REPORT も
    指定を行ってください。

- プロセス KILL アクションを指定する場合、以下の制約があります。

  ・異常検出時のアクションに PROCESS_KILL を指定した場合、リソース監視異常と判定されたプロセスは ProcessSaver での監視の有無にかかわらず kill(2) システムコールで停止されます。
  また、ProcessSaver で監視対象となっているプロセスについて、プロセス消滅検知時のアクションに再起動を指定していないプロセスは再起動されません。

  ・リソース監視異常と判定された場合に停止させたいプロセスは、すべてプロセス KILL アクション定義ファイル（psact_stop_allow.conf）に設定してください。
  プロセス KILL アクション定義ファイル（psact_stop_allow.conf）に設定されたプロセスのみ停止されます。
  または、リソース監視異常と判定されても停止を行いたくないプロセスをすべてプロセス KILL アクション除外定義ファイル（psact_notstop.conf）に設定してください。
  プロセス KILL アクション除外定義ファイル（psact_notstop.conf）に設定されたプロセスは、異常検出時にプロセスの停止を行いません。

  ・異常検出時のアクションに PROCESS_KILL を指定し、かつプロセス KILL アクション定義ファイル（psact_stop_allow.conf）が存在する場合、プロセス KILL アクション除外定義ファイル（psact_notstop.conf）の情報は反映されません。
  両方のファイルに重複して指定されているプロセスについては、プロセス KILL アクション定義ファイル（psact_stop_allow.conf）が有効になります。

  ・異常検出時のアクションに PROCESS_KILL を指定した場合であっても、プロセス KILL アクション定義ファイル（psact_stop_allow.conf）が存在してかつ、プロセス名の記述が 1 つもない場合は、プロセスの停止を行いません。

  ・プロセス KILL アクション除外定義ファイル（psact_notstop.conf）またはプロセス KILL アクション定義ファイル（psact_stop_allow.conf）に、引数付きのプロセス名を指定する場合、プロセス名は必ず絶対パスで指定してください。
  絶対パスでないプロセスを引数によって区別することはできません。

  ・PROCESS_KILL のオプションには kill(2) システムコールが送信するシグナル番号を指定してください。
  デフォルトのシグナル番号 **6（SIGABRT）**を指定した場合、プロセスを停止する際に該当プロセスのワーキングディレクトリに core ファイルを作成するため、解析に役立てることが可能です。
  該当のプロセス停止時に core ファイルを作成したくない場合は、任意のシグナル番号に変更してください。

  ・init 等の停止することのできないプロセスや defunct 状態のプロセスは停止されません。

  ・ゾンビ（defunct）プロセス監視、プロセス多重度監視、システム全体のリソース監視の異常検出時に該当プロセス等の停止を行うことはできません。

  ・異常検出時のアクションに PROCESS_KILL を指定した場合は、SYSLOG_REPORT の設定を必ず行ってください。
  設定方法については、「 4 章 SG の設定」を参照してください。

- pssmview(1M)を使用する場合、以下の制約があります。
  ・pssmview(1M)で指定した読み込みファイル（統計情報ファイル）のサイズ（ファイルサイズ×ファイル数）が大きい場合、処理に時間がかかる場合があります。
  例）30（MByte）× 10（個） を指定して pssmview(1M) コマンドを実行した場合、約 30 分前後かかります。

  ・pssmview(1M)を実行する場合に、定期実行で使用している統計情報ファイルを指定することはできません。
  また、定期実行の統計情報ファイルと同じディレクトリ配下のファイルも指定することはできません。
  定期実行で使用している統計情報ファイル、または定期実行の統計情報ファイルと同じディレクトリ配下のファイルについて pssmview(1M)で解析を行う場合は、
  定期実行で使用していないディレクトリ配下にコピーし、コピーしたファイルを指定してください。

  ・pssmview(1M)の注意・制限事項の詳細については、同梱の「統計情報編集コマンドユーザーズガイド」を参照してください。

7　付録

7．1　本製品が提供する情報採取スクリプトについて

ここでは、本製品が提供する情報採取スクリプトの使用例について記述します。
本製品が提供するスクリプトには、以下の 2 種類があります。

・定期実行スクリプト
　定期的に実行され多種多様な統計情報を採取するスクリプト
・リソース監視異常検出時情報採取スクリプト
　障害検出時に一度だけ実行され、障害発生時点での情報を採取するスクリプト

７．２　定期実行スクリプトについて

（1）定期実行スクリプトの概要

プロセスリソース、システムリソース監視に加えて、ユーザが任意に作成した情報採取スクリプトを定期的に実行し、統計情報として採取する機能を提供します。
本製品規定の定期実行用の情報採取スクリプトは以下のディレクトリに用意しています。

/var/opt/HA/PSSM/scripts/　配下

統計情報スクリプトの定期実行の詳細については、前述の「２．１　本製品で提供するリソース監視とは」の （5）統計情報採取スクリプトの定期的な実行 の章を参照してください。

本製品では、以下の情報採取スクリプトを用意しています。
これらの情報採取スクリプトを使用する場合には、スクリプト名の先頭の小文字 “s” を大文字 “S” に変更する必要があります。
なおスクリプト内で使用している各コマンドの詳細については、HP-UX のマニュアルページ等を参照してください。

- 全カーネルパラメータ情報定期採取スクリプト（s00get_kernelparam.sh）
  kctune(1M) コマンドを実行し、全カーネルパラメータ情報を採取しログファイルに保存します。
  ログファイルは、デフォルト 1MByte のサイズでバックアップ数は 3 世代となります。
  以下のファイル名で保存されます。
  /var/opt/HA/PSSM/scripts/log/scheduled_kernelparam.log
  バックアップファイル名は、以下の名前で保存されます。
  /var/opt/HA/PSSM/scripts/log/scheduled_kernelparam.log.save1
  /var/opt/HA/PSSM/scripts/log/scheduled_kernelparam.log.save2

- システム動作情報(sar コマンド)定期採取スクリプト（s01get_sardata.sh）
  sar (1M) コマンドを実行し、OS の動作状況に関する情報を採取しログファイルに保存します。
  ログファイルは、デフォルト 1MByte のサイズでバックアップ数は3 世代となります。
  以下のファイル名で保存されます。
  /var/opt/HA/PSSM/scripts/log/scheduled_sardata.log
  バックアップファイル名は、以下の名前で保存されます。
  /var/opt/HA/PSSM/scripts/log/scheduled_sardata.log.save1
  /var/opt/HA/PSSM/scripts/log/scheduled_sardata.log.save2

- プロセス間通信情報(ipcs コマンド)定期採取スクリプト（s02get_ipcsdata.sh）
  ipcs(1) コマンドを実行し、プロセス間通信のアクティブな機能に関する情報を採取しログファイルに保存します。
  ログファイルは、デフォルト 1MByte のサイズでバックアップ数は 3 世代となります。
  以下のファイル名で保存されます。
  /var/opt/HA/PSSM/scripts/log/scheduled_ipcsdata.log
  バックアップファイル名は、以下の名前で保存されます。
  /var/opt/HA/PSSM/scripts/log/scheduled_ipcsdata.log.save1
  /var/opt/HA/PSSM/scripts/log/scheduled_ipcsdata.log.save2

・ 仮想メモリ統計情報(vmstat コマンド)定期採取スクリプト（s03get_vmstatdata.sh）
vmstat(1) コマンドを実行し、プロセス、仮想メモリ、トラップ、CPU アクティビティの正確な統計情報を採取しログファイルに保存します。
ログファイルは、デフォルト 1MByte のサイズでバックアップ数は 3 世代となります。
以下のファイル名で保存されます。

/var/opt/HA/PSSM/scripts/log/scheduled_vmstatdata.log

バックアップファイル名は、以下の名前で保存されます。

/var/opt/HA/PSSM/scripts/log/scheduled_ipcsdata.log.save1
/var/opt/HA/PSSM/scripts/log/scheduled_ipcsdata.log.save2

・ システムパフォーマンス情報(glance コマンド)定期採取スクリプト（s04get_glancedata.sh）
glance (1) コマンドを実行し、システムリソース、動作中のプロセス、CPU、メモリ、ディスク I/O などの情報を採取しログファイルに保存します。
ログファイルは、デフォルト 1MByte のサイズでバックアップ数は 3 世代となります。
以下のファイル名で保存されます。

/var/opt/HA/PSSM/scripts/log/scheduled_glancedata.log

バックアップファイル名は、以下の名前で保存されます。

/var/opt/HA/PSSM/scripts/log/scheduled_glancedata.log.save1
/var/opt/HA/PSSM/scripts/log/scheduled_glancedata.log.save2

（2）定期実行スクリプトの導入手順

ここでは例として、全カーネルパラメータ情報定期採取スクリプト（s00get_kernelparam.sh）を使用する場合の手順について説明します。

① 定期実行スクリプトのパラメータ設定

定期実行スクリプトで情報を採取するログファイル名やログファイルサイズ等の設定を行います。
通常はデフォルト設定のまま特に変更する必要はありません。
ディスク容量など、環境に応じて変更する必要がある場合は、各スクリプトの先頭にあるパラメータの値を変更してください。

パラメータの設定例は以下のとおりです。

< /var/opt/HA/PSSM/scripts/s00get_kernelparam.sh >

```
    :
#### Start of Customer Define Parameters ####
# Log File Name
HASM_KCTUNE_LOG_FILE_NAME="/var/opt/HA/PSSM/scripts/log/scheduled_kernelparam.log"

# Log File Num (num)
# minimum = 1, maximum = 10, default = 3
HASM_KCTUNE_LOG_FILE_NUM=3

# Log File Size (MByte)
# minimum = 1, maximum = 256, default = 1
HASM_KCTUNE_LOG_FILE_SIZE=1

#### End of Customer Define Parameters ####
    :
```

ログファイルを保存するファイル名を指定します。

ログファイルを保存する個数を設定します。デフォルト 3 の場合、最大 3 ファイルまで保存されます。

ログファイル 1 つあたりの最大サイズを MByte 単位で設定します。
デフォルトは 1MByte となります。
サイズが設定された値を超えた場合にローテートされます。

（注） #### Start of Customer Define Parameters #### から
End of Customer Define Parameters #### の範囲以外の箇所については
変更しないでください。
スクリプトが正しく動作しなくなる可能性があります。

② スクリプトファイル名の変更

定期実行を行うためにファイル名を変更します。
情報採取スクリプトの定期実行を行う場合には、スクリプト名の先頭の小文字 “s” を大文字 “S” に変更する必要があります。

```
# cd   /var/opt/HA/PSSM/scripts/
# mv   s00_kernelparam.sh   S00_kernelparam.sh
```

③ スクリプトの定期実行間隔の設定

本製品の SG ファイル（pssm.conf）のパラメータ ***SCRIPTEXEC_INTERVAL*** で
スクリプトの定期実行間隔を設定します。
デフォルトでは、1440 分（1 日）が設定されており、指定された時間間隔でスクリプトが
実行され情報が定期的に保存されます。
通常はデフォルト設定のままで特に変更する必要はありません
定期実行の間隔を変更したい場合のみ、設定を変更してください。

pssm.conf の設定例は以下のとおりです。
</var/opt/HA/PSSM/conf/pssm.conf>

```
#############################################################
# System Config Area
#############################################################

# Monitor interval timer (minutes)
# minimum = 1, maximum = 10080(1week), default = 5
MONITOR_INTERVAL                1

# Analyze interval timer (minutes)
# This parameter is larger than MONITOR_INTERVAL.
# minimum = 1, maximum = 10080(1week), default = 60
ANALYZE_INTERVAL                2

# Script execution interval timer (minutes)
# minimum = 1, maximum = 10080(1week), default = 1440 (1day)
SCRIPTEXEC_INTERVAL             10

# Monitor file size (MByte)
# minimum = 1, maximum = 100, default = 30
MONITOR_FILE_SIZE               1
        :
```

スクリプトの定期実行間隔を設定します。（単位は分）
10 分間隔で実行する場合、10 を指定します。

④ 本製品の再起動

```
# /sbin/init.d/pssmd  restart
```

（注）pssm.conf を変更した場合は、再起動が必要となります。

⑤ スクリプトが実行されたことを確認する。

再起動後に設定されたスクリプトが実行され 1. で設定したログファイルと定期実行スクリプト
のテキストログファイルが作成されることを確認します。

```
# ls  /var/opt/HA/PSSM/scripts/log
scheduled_kernelparam.log    scripts_trace.log
```

1. で指定されたファイルが作成されることを確認。

定期実行スクリプトのテキストログファイルが作成されることを確認。

この後 3. の ***SCRIPTEXEC_INTERVAL*** で設定された時間間隔でスクリプトが実行され
情報を定期的に採取します。

（3）各スクリプトで作成されるログファイルの例

各スクリプトによって作成されるファイルの例は以下となります。

全カーネルパラメータ情報定期採取スクリプト（s00get_kernelparam.sh）によって作成されるファイルの例

< /var/opt/HA/PSSM/scripts/log/scheduled_kernelparam.log>

```
================================================
 HOST  : host1
 DATE  : 2008/04/24 13:42:59
 COUNT : 1
------------------------------------------------
Tunable             NSTREVENT
Description         Maximum number of concurrent Streams bufcalls
Module              hpstreams
Current Value       50 [Default]
Value at Next Boot  50 [Default]
Value at Last Boot  50
Default Value       50
Can Change          At Next Boot Only

Tunable             NSTRPUSH
Description         Maximum number of Streams modules in a stream
Module              hpstreams
Current Value       16 [Default]
Value at Next Boot  16 [Default]
Value at Last Boot  16
Default Value       16
Can Change          At Next Boot Only
       :
================================================
 HOST  : host1
 DATE  : 2008/04/24 13:52:59
 COUNT : 2
------------------------------------------------
Tunable             NSTREVENT
       :
```

システム動作情報定期採取スクリプト（s01get_sardata.sh）によって作成されるファイルの例
</var/opt/HA/PSSM/scripts/log/scheduled_sardata.log>

```
================================================================

HOST   : host1
DATE   : 2008/04/24 13:43:03
COUNT  : 1
----------------------------------------------------------------

    iget/s    namei/s     dirbk/s
         3       3094          26

   bread/s    lread/s    %rcache    bwrit/s    lwrit/s    %wcache     pread/s    pwrit/s
         0         23        100          0          9        100           0          0

   scall/s    sread/s    swrit/s     fork/s     exec/s    rchar/s     wchar/s
      5061         25         14       0.99       0.99     186550       40554

     msg/s      sema/s    select/s
      0.00       0.00       2.97

      pset        cpu     runq-sz    %runocc     swpq-sz    %swpocc
         0          0        2.0         99
         0          1        3.0         99
                system        2.5         99        0.0          0

      pset        cpu        %usr        %sys        %wio       %idle
         0          0          99           0           0           0
         0          1          98           1           0           0
                system          99           1           0           0

   text-sz         ov     proc-sz         ov     inod-sz         ov     file-sz         ov
       N/A        N/A  535/15000          0  1093/162144          0  1380/65048          0

   rawch/s    canch/s    outch/s    rcvin/s    xmtin/s    mdmin/s
         0          0          0          0          0          0

================================================================

HOST   : host1
DATE   : 2008/04/24 18:05:20
COUNT  : 2
----------------------------------------------------------------

    iget/s    namei/s     dirbk/s
         4       3225          18
         :          :           :
```

プロセス間通信情報定期採取スクリプト（s02get_ipcsdata.sh）によって作成されるファイルの例
</var/opt/HA/PSSM/scripts/log/scheduled_ipcsdata.log>

```
================================================================
 HOST  : host1
 DATE  : 2008/04/24 13:43:05
 COUNT : 1
----------------------------------------------------------------
IPC status from /dev/kmem as of Thu Apr 24 13:43:05 2008
T ID     KEY        MODE       OWNER  GROUP CREATOR CGROUP CBYTES QNUM QBYTES LSPID LRPID STIME   RTIME    CTIME
Message Queues:
q  0 0x3c1c2e04 -Rrw--w--w-  root   root   root   root      0    0  16384     0     0 no-entry no-entry 10:09:26
q  1 0x3e1c2e04 --rw-r--r--  root   root   root   root      0    0    264     0     0 no-entry no-entry 10:09:26
T ID     KEY        MODE       OWNER  GROUP CREATOR CGROUP NATTCH  SEGSZ CPID LPID  ATIME   DTIME    CTIME
Shared Memory:
m  0 0x411c2e06 --rw-rw-rw-  root   root   root    root     0     348  773  773 10:09:33 10:09:33 10:09:26
m  1 0x4e0c0002 --rw-rw-rw-  root   root   root    root     1   61760  773  773 10:09:29 10:09:33 10:09:26
m  2 0x412051eb --rw-rw-rw-  root   root   root    root     1    8192  773  785 10:09:29 10:09:26 10:09:26
m  3 0x440c0639 ----------  root   root   root    root     1       1 1354 1355 10:09:48 no-entry 10:09:48
          :
T ID     KEY        MODE       OWNER GROUP CREATOR CGROUP NSEMS   OTIME    CTIME
Semaphores:
s  0 0x4f1c013a --ra-------  root  root   root   root     1 10:09:26 10:09:26
s  1 0x411c2e06 --ra-ra-ra-  root  root   root   root     1 10:09:33 10:09:26
s  2 0x4e0c0002 --ra-ra-ra-  root  root   root   root     2 10:09:29 10:09:26
s  3 0x412051eb --ra-ra-ra-  root  root   root   root     2 no-entry 10:09:26
s  4 0x00446f6e --ra-r--r--  root  root   root   root     1 no-entry 10:09:46
s  5 0x00446f6d --ra-r--r--  root  root   root   root     1 no-entry 10:09:46
s  6 0x01090522 --ra-r--r--  root  root   root   root     1 no-entry 10:09:46
s  7 0x00a5c581 --ra-------  sfmdb users  sfmdb  users    17 10:09:55 10:09:55
s  8 0x00a5c582 --ra-------  sfmdb users  sfmdb  users    17 10:09:55 10:09:55
s  9 0x00a5c583 --ra-------  sfmdb users  sfmdb  users    17 10:09:55 10:09:55
s 10 0x00a5c584 --ra-------  sfmdb users  sfmdb  users    17 10:09:55 10:09:55
            :
================================================================
 HOST  : host1
 DATE  : 2008/04/24 18:05:21
 COUNT : 2
----------------------------------------------------------------
IPC status from /dev/kmem as of Thu Apr 24 13:43:05 2008
T ID     KEY        MODE       OWNER GROUP CREATOR CGROUP CBYTES  QNUM QBYTES LSPID LRPID  STIME   RTIME   CTIME
Message Queues:
q  0 0x3c1c2e04 -Rrw--w--w-  root  root   root   root      0     0  16384     0     0 no-entry no-entry 10:09:26
q  1 0x3e1c2e04 --rw-r--r--  root  root   root   root      0     0    264     0     0 no-entry no-entry 10:09:26
T ID     KEY        MODE       OWNER GROUP CREATOR CGROUP NATTCH      SEGSZ  CPID  LPID   ATIME    DTIME    CTIME
Shared Memory:
            :
```

仮想メモリ統計情報定期採取スクリプト（s03get_vmstatdata.sh）によって作成されるファイルの例
</var/opt/HA/PSSM/scripts/log/scheduled_vmstatdata.log>

```
================================================
 HOST  : host1
 DATE  : 2008/04/24 13:43:07
 COUNT : 1
------------------------------------------------

        procs           memory                  page                              faults       cpu
    r     b     w      avm    free   re   at    pi   po    fr   de    sr     in     sy    cs  us sy id
    4     1     0   906067   29041  291   64    18    0     0    0     0    687  13039   772  97  3  0

Disk Transfers
  device    xfer/sec
  c2t0d0         7
  c2t1d0         5
c10t12d3         0
 c10t2d5         0
 c10t0d1         0
 c10t0d2         0
 c10t0d3         0
c10t12d0         0
c12t12d0         0
c10t12d1         0
c12t12d1         0
c10t11d1         0
c10t11d2         0
#### Report on the number of forks and the number of pages of virtual memory involved since boot-up. ####
623773 forks, 22015330100 pages, average=  35293.82

#### Print the total number of several kinds of paging-related events. ####
255 swap ins
255 swap outs
335 pages swapped in
6140 pages swapped out
57771657 total address trans. faults taken
13769034 page ins
57433 page outs
552552 pages paged in
     :
================================================
 HOST  : host1
 DATE  : 2008/04/24 18:05:24
 COUNT : 2
------------------------------------------------

        procs           memory                  page                              faults       cpu
     :
```

システムパフォーマンス情報定期採取スクリプト（s04get_glancedata.sh）によって作成されるファイルの例

</var/opt/HA/PSSM/scripts/log/scheduled_glancedata.log>

```
================================================
 HOST  : host1
 DATE  : 2008/04/24 13:43:07
 COUNT : 1
------------------------------------------------

#################### Global Metrics ###################
GBL_IPC_QUEUE                6.9
GBL_LAN_QUEUE                0.0
GBL_MEM_PAGEIN               0
     :

#################### Table Metrics ####################
TBL_BUFFER_CACHE_AVAIL       40kb
TBL_BUFFER_CACHE_HIGH        40kb
TBL_BUFFER_CACHE_MAX         973kkb
     :

#################### By Swap Metrics ####################
BYSWP_SWAP_SPACE_NAME        /dev/vg00/lvol2
BYSWP_SWAP_TYPE              device
BYSWP_SWAP_PRI               1
     :

#################### By CPU Metrics ####################
BYCPU_ID                      0
BYCPU_STATE                   Enable
BYCPU_CPU_INTERRUPT_TIME      0.00
     :

#################### By Disk Metrics ####################
BYDSK_DEVNAME                 64000/0xfa00/0x3
BYDSK_FS_IO_RATE              0.0
BYDSK_LOGL_IO_RATE            na
     :
BYDSK_DEVNAME                 64000/0xfa00/0x2
BYDSK_FS_IO_RATE              0.0
BYDSK_LOGL_IO_RATE            na
     :
================================================
 HOST  : host1
 DATE  : 2008/04/25 13:43:07
 COUNT : 2
------------------------------------------------
     :
```

７．３　リソース監視異常検出時情報採取スクリプト

（1）異常検出時情報採取スクリプトの概要

リソース監視異常検出時に、一度だけ実行し障害発生時のサーバー情報を採取するスクリプトについて記載します。

情報採取スクリプトのサンプルは、以下となります。

/var/opt/HA/PSSM/bin/collect_info.sh

本スクリプトは、各 HP-UX の標準コマンドを実行し、障害発生時のシステム情報を採取して以下のログファイルに保存します。

/var/opt/HA/PSSM/log/collect_info_<実行年月日時刻>

ログファイルのサイズは、環境によって異なりますが 100Kbyte 程度となります。

（2）異常検出時情報採取スクリプトの導入手順

リソース監視異常検出時のスクリプト実行機能についての詳細、設定手順については、前述の「3．4 リソース監視異常検出時のスクリプト実行」を参照してください。

本スクリプト実行時に実行されるコマンドおよび採取する情報は以下となります。
各 HP-UX コマンドの詳細および出力結果については、HP-UX のマニュアルページ等を参照してください。

- **/usr/bin/top**（システム上の最上位のプロセスに関する情報の表示）
  CPU 使用率が上位のプロセス情報を表示します。
  プロセスのステータス、CPU 使用率、プロセスの常駐サイズ等の情報を採取します。

- **/usr/sbin/sar**（システム動作状況のレポート）
  CPU の利用状況（モードごとの実行時間）について採取します。

- **/usr/bin/vmstat**（仮想メモリ統計情報のレポート）
  プロセス、仮想メモリ、トラップ、CPU アクティビティの正確な統計情報を採取します。
  - ・実行待ち行列にあるプロセス
  - ・アクティブな仮想ページ
  - ・ページインされたページ
  - ・デバイス割込み/秒
  - ・通常および低い優先順位プロセスのユーザ時間　etc…

- **/usr/sbin/swapinfo**（システムのページングスペース情報）
  デバイスおよびファイルシステムのページングスペースに関する情報を採取します。
  予備のページングスペース、メモリページングスペース等の情報を採取します。

- **/usr/bin/netstat**（ネットワークステータスの表示）
  ネットワークインタフェースおよびプロトコルに関する統計情報、並びに各種ネットワーク関係のデータ構造の内容について採取します。

- **/usr/bin/ipcs**（プロセス間通信機能のステータスの報告）
  プロセス間通信のアクティブな機能に関する情報を採取します。
  - ・機能種別（共有メモリセグメント、メッセージ待ち行列、セマフォ）
  - ・機能アクセスモードおよびフラグ
  - ・機能項目の識別子　etc…

- **/usr/bin/ps**（プロセスステータスのレポート）
  起動している、すべてのプロセス情報を採取します。
  - ・ユーザ ID
  - ・プロセス ID
  - ・プロセスの開始時刻　etc…

- **/usr/bin/tail -n 20 /var/adm/syslog/syslog.log**（システムログ情報）
  本スクリプト実行時のシステムログの末尾 20 行を採取します。

リソース監視異常検出時情報採取スクリプトの実行によって作成されるファイルの例は以下のとおりです。

</var/opt/HA/PSSM/log/ collect_info_070724132027>

```
######## Command result (/usr/bin/top) ############################
System: host1 Tue Jul 24 13:20:32 2007
Load averages: 0.01, 0.02, 0.02
224 processes: 179 sleeping, 45 running
Cpu states:
CPU    LOAD   USER   NICE    SYS    IDLE    BLOCK  SWAIT   INTR   SSYS
 0     0.01   0.0%   0.0%   0.0%  100.0%     0.0%   0.0%   0.0%   0.0%
          :                                :
CPU   TTY   PID  USERNAME  PRI  NI  SIZE   RES  STATE  TIME  %WCPU  %CPU  COMMAND
 0      ?  2659  root      152  20  938M  188M  run    37:08  1.32  1.32  java
          :                                :                            :
######## Command result (/usr/sbin/sar) ############################
HP-UX host1 B.11.23 U ia64    07/24/07
19:20:32    %usr    %sys    %wio   %idle
19:20:33       0       0       0     100
          :                    :
Average        0       0       0     100
######## Command result (/usr/bin/vmstat) ############################
      procs           memory                  page                    faults       cpu
 r    b    w      avm    free    re   at    pi   po  fr   de   sr   in    sy    cs    us  sy  id
 1    1    0   1698948   17115   43   21    1    0    0    0    3   571  6140   928   2   1   98
          :                               :                            :
######## Command result (/usr/sbin/swapinfo) ############################
             Kb      Kb      Kb   PCT  START/      Kb
TYPE      AVAIL    USED     FREE   USED   LIMIT  RESERVE  PRI  NAME
dev     4194304  965060  3229244    23%       0       -     1  /dev/vg00/lvol2
          :                               :
######## Command result (/usr/bin/netstat) ############################
Active Internet connections
Proto Recv-Q Send-Q  Local  Address          Foreign Address        (state)
tcp        0      0    ost  1.49398          host1.49399            ESTABLISHED
          :                               :                            :
######## Command result (/usr/bin/ipcs) ############################
IPC status from /dev/kmem as of Tue Jul 24 13:20:47 2007
T         ID     KEY         MODE         OWNER     GROUP
Message Queues:
q       4200     0x3c1c2e04  -Rrw--w--w-  root      root
          :                               :
######## Command result (/usr/bin/ps) ############################
     UID   PID  PPID    C    STIME    TTY   TIME    COMMAND
    root     0     0    0    Jul 19     ?   0:21    swapper
          :                               :
######## Command result (/var/adm/syslog/syslog.log) ############################
Jul 24 13:20:27 host1 psaction[21805]: Find a sign of process resource failure
(type=swchunk  ,value=965060 ,maxvalue=4194304)
 :                              :                            :
```

７．４　統計情報ファイルコピーコマンドについて

（1）　統計情報ファイルコピーコマンドの概要

HA/SystemResourceMonitor（pssmd）で作成する統計情報ファイルを別のディレクトリにバックアップする機能を提供します。

統計情報ファイルコピーコマンドの構成は以下のとおりです。

ディレクトリ：**/opt/HA/PSSM/bin/**
コマンド　　：**pssmdatcp**

（2）　利用の機会について

以下のような場合にご利用ください。

・統計情報ファイル(psmonitor.dat)のバックアップを取りたい場合

例）/Data/psmonitor.dat.save1 ファイルを /SyM/SaveData 配下にバックアップしたい場合、以下のコマンドを実行します。

/opt/HA/PSSM/bin/pssmdatcp –f /Data/psmonitor.dat.save1 –d /SyM/SaveData

・統計情報編集コマンド（pssmview）を HA/SystemResourceMonitor の監視を停止することなく、利用したい場合

例）HA/SystemResourceMonitor で使用しているディレクトリ配下のファイル psmonitor.dat.save2 を統計情報編集コマンドの引数に指定し、CSV 形式でファイルを出力させたい場合、以下の手順で可能です。

① 統計情報ファイルコピーコマンドを利用し、psmonitor.dat.save2 を HA/SystemResourceMonitor で使用していないディレクトリ配下（/SyM/SaveData/）にバックアップを取ります。

/opt/HA/PSSM/bin/pssmdatcp –f /var/opt/HA/PSSM/log/psmonitor.dat.save2 –d /SyM/SaveData

② バックアップしたファイルを指定して、統計情報編集コマンド（pssmview）を実行し、/tmp 配下に psmonitor.csv という CSV 形式ファイルに出力させます。
※ 統計情報編集コマンド（pssmview）については、
「統計情報編集コマンド　ユーザーズガイド」を参照してください。

/opt/HA/PSSM/bin/pssmview –i /SyM/SaveData/psmonitor.dat.save2 –o /tmp/psmonitor.csv

（3） コマンドリファレンス

名称

pssmdatcp(1M) － 統計情報ファイルコピーコマンド

構文

```
pssmdatcp  -f input_file -d directory [-h]
```

戻り値

正常時：0

異常時：0 以外

機能説明

pssmdatcp は統計情報ファイルを指定されたディレクトリ配下にコピーするコマンドです。

-f input_file	コピーするファイルを input_file に絶対パスで指定します。 複数指定可能です。 最大 15 ファイルまで指定できます。 複数指定する場合は、”“（スペース）区切りで指定します。 例）-f /tmp/psmonitor.dat　/tmp/psmonitor.dat.save1
-d directory	コピー先ディレクトリを directory に絶対パスで指定します。
-h	usage を表示します。

NX ソフトウェア

HA/SystemResourceMonitor R4.2
ユーザーズガイド　基本編

2011 年 12 月　第 2 版
日本電気株式会社
東京都港区芝五丁目７番１号
TEL (03) 3454-1111 (代表)

(P)

保護用紙