

Solaris 性能情報採取ツール sardo 解説書

2014 年 1 月 28 日 Rev.1.2h

株式会社エイ・シー・ティ

本書に掲載の製品名、会社名等は一般にそれぞれの会社の商標、または登録商標です。また、本文では TM、®、© の各マークは省略しています。本ドキュメントの内容は、ソフトウェアの改良、変更等で適宜改訂されることがあります。

目次

はじめに	6
第 1 章 sardoの簡単使用ガイド	8
第 2 章 sardoの仕組みと考え方	9
2-1. sardoのコンセプト	9
2-2. 構成	11
2-3. 性能情報採取と編集の流れ	12
2-4. 性能情報採取コマンド	13
2-4-1. 採取開始・終了時の採取情報	13
2-4-2. サンプリングインターバル単位に採取する情報	14
2-4-3. プロセスアカウンティング情報	15
2-4-4. 操作によって任意のタイミングで採取する情報	15
2-5. 性能情報採取・編集結果について	16
2-5-1. sardoの作成するファイル	16
2-5-2. sardoの作成するディレクトリ	18
第 3 章 sardoの操作	20
3-1. sardo.shの操作	21
3-1-1. 自動実行 auto	21
3-1-2. 情報採取ディスク容量の見積り estimate	23
3-1-3. メモリ使用状況の出力 memstat	24
3-1-4. プロセスマップの出力 pmap	24
3-1-5. 手動実行のためのセットアップ setup	25
3-1-6. 手動による採取開始 start	26
3-1-7. 強制停止 stop force	27
3-1-8. アーカイブの作成 tar	28
3-1-9. 展開 unpack	29
3-2. sardo.editの操作	30
3-2-1. 全データの編集 all	30
3-2-2. 部分データの抽出編集 select	31
3-3. sardo.evalの操作	32
3-4. グラフの描画 grapher.xls	33
第 4 章 sardoのファイル形式	38
4-1. sardoディレクトリ下のファイル形式	38
4-1-1. baseline ベースライン	38
4-1-2. cpu.minmax CPU性能値の最小・平均・最大	38
4-1-3. date.info 採取日付	39
4-1-4. disk.info ディスク容量の採取前と後	39
4-1-5. disk.minmax ディスク性能値の最小・平均・最大	40
4-1-6. memory.info メモリ消費ページとバイト	41
4-1-7. network.info ネットワーク性能値のまとめ	42
4-1-8. network.minmax ネットワーク性能値の最小・平均・最大	44
4-1-9. os.info OS各種性能値のまとめ	45
4-1-10. os.minmax OS性能値の最小・平均・最大	46
4-1-11. pacct.info プロセスアカウンティング情報のまとめ	47
4-1-12. pmap.info プロセスのメモリ使用状況	48
4-1-13. process.minmax プロセスCPU消費の最小・平均・最大	50
4-1-14. sardo.log sardo実行ログ	51
4-1-15. sardo.param sardoパラメタファイル	52
4-1-16. segment.info メモリセグメントの割当状況	52
4-2. begin 開始時の情報	53
4-3. data 採取中の情報	53
4-4. end 終了時の情報	54
4-5. memstat ファンクションmemstatの採取情報	54

4-6. misc システム情報	55
4-7. pacct プロセスアカウンティング情報(バイナリ)	55
4-8. pmap ファンクションpmapの採取情報	56
4-9. preserve 編集情報の保存ディレクトリ	56
4-10. sadc sadc(1M)出力ファイル(バイナリ)	58
4-11. speditdir 性能要素単位の編集結果	59
4-11-1. cpu性能要素ディレクトリ	59
4-11-2. disk性能要素ディレクトリ	60
4-11-3. network性能要素ディレクトリ	61
4-11-4. os性能要素ディレクトリ	62
4-11-5. process性能要素ディレクトリ	64
4-12. splistdir 性能要素単位の各コマンド出力	65
4-13. sptrendir 傾向データ	67
第5章 sardoの計算式	68
5-1. ネットワークスループットの計算	68
5-2. baselineレコードの計算	69
第6章 sardoの応用	70
6-1. 性能情報のファイルサイズを見積る	70
6-1-1. 見積り用ファイルestimateの作成	70
6-1-2. estimateファイルの見方	71
6-2. 性能値の評価	74
6-2-1. baselineレコード作成の仕組み	74
6-2-2. baselineを用いた性能データの評価	76
6-2-3. baselineファイルのカスタマイズ	77
6-3. 環境変数	78
6-3-1. 共通の環境変数	78
6-3-2. sardo.edit固有の環境変数	79
6-3-3. sardo.eval固有の環境変数	79
6-4. sardo.eventイベントチェックルーチン	80
6-5. cron(8)デーモンによるsardo.shの自動起動	81
付録A. メッセージ	83
付録A-1. sardo.shのメッセージ	84
付録A-2. sardo.editのメッセージ	89
付録A-3. sardo.evalのメッセージ	94
付録B. ツールのシステム性能に及ぼす影響	95
付録C. ツールのメモリ使用量	103
付録D. サンプリングタイムの違いによる測定値の変化	105
参考文献	108
ドキュメント更新履歴	109
連絡先	109

図目次

図 2-3-1. sardoの性能情報採取と編集のモデル	12
図 2-5-1-1. sardoの作成するファイル	16
図 2-5-2-1. sardoの作成するディレクトリ	18
図 3-4-1. グラフ描画エクセル	33
図 3-4-2. フォルダ選択画面	34
図 3-4-3. 性能要素の選択画面	34
図 3-4-4. フォルダ確認画面	35
図 3-4-5. グラフ名の確認メッセージボックス	35
図 3-4-6. グラフ日付確認のメッセージボックス	35
図 3-4-7. 日付入力のインプットボックス	35
図 3-4-8. 再度の日付確認メッセージボックス	36
図 3-4-9. グラフの出力例	37
図 4-1-1-1. baselineファイル	38
図 4-1-2-1. cpu.minmaxファイル	38
図 4-1-3-1. date.infoファイル	39
図 4-1-4-1. disk.infoファイル	39
図 4-1-5-1. disk.minmaxファイル	40
図 4-1-6-1. memory.infoファイル	41
図 4-1-7-1. network.infoファイル	42
図 4-1-8-1. network.minmaxファイル	44
図 4-1-9-1. os.infoファイル	45
図 4-1-10-1. os.minmaxファイル	46
図 4-1-11-1. pacct.infoファイル	47
図 4-1-12-1. pmap.infoファイル	48
図 4-1-13-1. process.minmaxファイル	50
図 4-1-14-1. sardo.logファイル	51
図 4-1-15-1. sardo.paramファイル	52
図 4-1-16-1. segment.infoファイル	52
図 4-5-1. memstat_yymmddhhmmss_infoファイル	54
図 5-2-1. 傾向ファイルのレコード形式	69
図 5-2-2. baselineレコードの形式	69
図 6-1-2-1. estimateファイルの内容(その 1)	71
図 6-1-2-2. estimateファイルの内容(その 2)	72
図 6-1-2-3. estimateファイルの内容(その 3)	73
図 6-2-1-1. 傾向データsptrendirディレクトリの構成	74
図 6-2-1-2. 傾向データとbaselineレコードの関係	75
図 6-2-2-1. sardo.evalの処理	76
図 6-2-2-2. sardo.evalが作成するsardo.baselineファイルの例	76
図 6-2-3-1. 評価の流れとbaselineファイルのカスタマイズ	77
図B-1. sardo.shによる 1 日分の振る舞い	96
図B-2. sardo.sh起動前後の振る舞い	96
図B-3. sardo.sh実行中(性能情報採取中アイドル)の振る舞い(その 1)	96
図B-4. STATSPACKとorado.editの振る舞い	97
図B-5. sardo.sh実行中(性能情報採取中アイドル)の振る舞い(その 2)	97
図B-6. sardo.sh実行中(性能情報採取中アイドル)の振る舞い(その 3)	97
図B-7. sardo.sh終了時の振る舞い	98
図B-8. sardo.sh終了後の振る舞い	98
図B-9. sardo.editの振る舞い	99
図B-10. sardo.sh性能評価時のsardo.logファイル	101
図B-11. sardo.sh性能評価時のorado.logファイル	102
図C-1. sardo.shのメモリ消費量(その 1)	103
図C-2. sardo.shのメモリ消費量(その 2)合計	104

図D-1. sar(1)-u 5 秒間隔のグラフ	105
図D-2. sar(1)-u 10 秒間隔のグラフ	105
図D-3. sar(1)-u 60 秒(1 分)間隔のグラフ	106
図D-4. sar(1)-u 600 秒(10 分)間隔のグラフ	106
図D-5. sar(1)-d 5 秒間隔のグラフ	106
図D-6. sar(1)-d 10 秒間隔のグラフ	106
図D-7. sar(1)-d 60 秒(1 分)間隔のグラフ	107
図D-8. sar(1)-d 600 秒(10 分)間隔のグラフ	107

表目次

表 2-2-1. 提供されるモジュール	11
表 2-4-1-1. sardo.sh採取開始・終了時の採取情報	13
表 2-4-2-1. sardo.sh実行中の性能情報採取コマンド	14
表 2-4-3-1. プロセスアカウント	15
表 2-4-4-1. 任意のタイミングで採取できるメモリ情報	15
表 2-5-1-1. sardoの作成するファイル	17
表 2-5-2-1. sardoの作成するディレクトリ	19
表 4-1-6-1. memory.infoファイルの項目	41
表 4-1-7-1. network.infoファイルの項目	42
表 4-1-9-1. os.infoファイルの項目	45
表 4-1-11-1. pacct.infoファイルの内容	47
表 4-1-12-1. pmap.infoファイルの内容	49
表 4-1-16-1. setment.infoファイルの内容	52
表 4-2-1. beginディレクトリのファイル	53
表 4-3-1. dataディレクトリのファイル	53
表 4-6-1. miscディレクトリのファイル	55
表 4-9-1. preserveディレクトリの内容	56
表 4-9-2. preserve/diskinfoディレクトリの内容	56
表 4-9-3. preserve/netsinfoディレクトリの内容	56
表 4-11-1. speditdirの性能要素ディレクトリ	59
表 4-12-1. splistdirディレクトリの内容	65
表 4-12-2. splistdir/networkディレクトリの内容	65
表 4-12-3. splistdir/processディレクトリの内容	66
表 4-12-4. splistdir/process/pacctinfo/d_HHMMSS_HHMMSSディレクトリの内容	66
表 4-12-5. splistdir/process/pmapinfo/pmap_[begin end]ディレクトリの内容	66
表 4-13-1. sptrendirディレクトリの内容	67
表 5-2-1. baselineレコードの計算式	69
表B-1. ハードウェア構成	95
表B-2. ソフトウェア構成	95
表B-3. 測定内容	95
表B-4. sardo.sh起動前後のCPUとディスク入出力待ちの変化	96
表B-5. アイドル時のCPUとディスク入出力待ち	97
表B-6. sardo.sh終了後アイドル時の平均値	98

はじめに

Solarisシステムにおいて、各種性能情報を採取・編集するsardoツールと、付属のグラフ描画マクロ grapher.xls^{【備考 1】}について解説します。

本書は sardo と grapher.xls の解説を主題にしております。Solaris OS(Operating System)で提供される sar(1M)や vmstat(1M)、iostat(1M)等のコマンド出力形式、内容等の詳細はそれぞれの解説書、オンラインマニュアル等をご覧ください。なお、コマンドの後ろの(1M)や(1)は UNIX におけるセクションを示しており、(1M)は運用管理コマンド、(1)は一般コマンドを表しています。

参考文献では、数々の UNIX OS が提供する各種性能情報採取コマンドについて、使用方法と要点が詳細に解説されています。問題は、各種コマンドをどのように、どのようなタイミングで発行して情報を採取してよいのか、判断が難しいことです。また、CPU やメモリ、ディスク入出力、及びネットワークが同じ時刻にどう影響し合っているのか、分析に苦慮することがあります。

本ツールは、OS の各種資源の消費、使用度合い等を測る上で必要な各種の性能情報採取コマンドを組み合わせて実行し、有効な性能値を得、ビジュアルなグラフを簡単に作成するためのツールです。定常業務における性能管理や、性能問題の追及等、また、開発過程におけるサイジング、ベンチマーク、チューニング等の場面において、定量的な性能情報を採取する時に威力を発揮します。

この結果、システム管理者や開発担当者が容易にボトルネックや問題を発見することができるようになります。

次に sardo の特徴を示します。

- (1) 操作は直感的ファンクションで簡単!! 設定も簡単です。
- (2) 性能情報採取のディスク容量を見積ることができます。
- (3) 現在採取可能な情報を漏らさず正確に採取することで、取りこぼしを最小限に抑えました。このため、採取データ量が大きくなることがあります^{【備考 2】}。
- (4) 性能情報を自動的に分析、各種性能値の傾向をつかむことで評価基準(ベースライン)を作成することができます。
- (5) 性能情報分析に有効なコマンドを取捨選択して採用し、オーバーヘッドのかからない方法を取り入れています^{【備考 3】}。
- (6) 採取データは情報採取後 tar(1)コマンドでアーカイブされ、gzip(1)コマンドで圧縮されますので、ポータビリティに富んでいます。

これらの機能でご利用の方々のシステム運用管理や性能分析などの作業負荷を大幅に軽減することができます。

【備考 1】 グラファーの名前は、grapher3.xls(Office 2000、XP、2003 用)、grapher7.xls(Office 2007、2010 用)に分かれています。総称として“grapher.xls”と記述します。個々のグラファーを示す場合は区別して記述します。

【備考 2】 唯一、マイナスの特徴です。情報を網羅的に採取する反面のトレードオフとしてご理解下さい。

【備考 3】 情報採取中のシステムオーバーヘッドを確認しました。付録 B. ツールのシステム性能に及ぼす影響 に実測データを示します。ご覧ください。

本書の構成

第 1 章 sardo の簡単セットアップと使用手順を示します。

通常、第 1 章のみで sardo を実行、操作することができます。応用操作、環境変数の設定等が必要な場合は第 2 章以降をご覧ください。

第 2 章 sardo の仕組みと考え方を解説します。

sardo がどのように性能情報を採取しているのかを解説します。時として、数ある性能情報採取ツールや、運用監視ソフトウェアは、情報採取のために高いオーバーヘッドがかかることがあります。本来の性能情報にノイズが混入し、正しい性能分析が困難になります。sardo は「静かに情報を採取し、通常処理に迷惑をかけない」よう、最大限に工夫されています。

第 3 章 sardo の詳細な操作、使用方法を解説します。

sardo の各種パラメタ、使用方法を詳解します。一歩進んだ使用法は第 6 章の応用に解説します。

第 4 章 sardo の出力するファイルの形式を解説します。

sardo は多くの性能情報を採取します。sardo によってどのような情報が採取されるのかを解説します。全ての情報は、様々な性能分析に使用します。常に全てを確認する必要はありません。

第 5 章 sardo で使用の計算式を解説します。

ディスク入出力スループット、ブロック長、メモリ量などの計算、ネットワーク関連の各種データを基にした転送レートなど、各種の性能情報から性能値を導き出す計算を解説します。

第 6 章 sardo の応用を解説します。

cron(1)コマンドで sardo を起動するようしておくと、煩わしい操作は簡略化されます。ベースライン管理は性能分析のみならず、性能監視にも役立ちます。トラップコードを組み込んで、高度な性能管理や、障害対応が可能になります。

第 1 章 sardo の簡単使用ガイド

- (1) スーパーユーザでツールのファイルを gunzip 解凍、tar -xvf 復元します。
- (2) sardo.shを実行して性能情報を採取します。ここでは例として 5 秒の採取間隔^{【備考 1】}で 3 分間 (3min)情報採取することになります^{【備考 2】}。3~4 分待ちます。実行中、端末から“ps”コマンドを入力すると多くのプロセスが実行されていることを確認することができます。約 3 分経過すると、いくつかの“Killed”のメッセージが表示され、情報採取が終了します。

```
# ./sardo.sh auto 5 now 3min <CR>
```

- (3) 編集のためにファイルを一旦解凍します。

```
# ./sardo.sh unpack <CR>
```

- (4) sardo.edit を実行して性能情報を編集します。

```
# ./sardo.edit <CR>
```

- (5) ファイルをアーカイブ・圧縮します。

```
# ./sardo.sh tar <CR>
```

- (6) /var/tmpディレクトリの“**sarstats_\$HOSTNAME_001_yymmddhhmmss.tar.gz**”のファイルを、グラファーをセットした Windows PC にコピーし、解凍・復元します^{【備考 3】}。“\$HOSTNAME”は実行システムのhostname(1)コマンドで表示される名前です。なお、(3)の sardo.sh unpackで解凍した時に作成される、編集前のtarアーカイブファイルは不要です。
- (7) **grapher.xls**^{【備考 4】}を実行します。解凍したファイルの、“**sarstats_\$HOSTNAME_001**”を選択すると全てのディレクトリのグラフが描画されます。speditdirディレクトリ下の“**os**”や“**disk**”等の個々のディレクトリを選択すると、そのディレクトリのグラフを描画することができます。

【備考1】 サンプリングインターバルです。採取する対象の動作・振る舞いや実行時間をもとに決定します。対象がバッチ処理で、実行中にリソースの消費が大きく変化し、その変化の度合いが秒単位、分単位、あるいは時間単位になるのかによって、採取間隔を決定します。リソースが短時間に变化的場合、採取間隔を短く設定します。最小単位は5秒です。

【備考2】 採取目的の処理(バッチ処理やトランザクション処理)がどのくらいで終了するのかによって、適切な終了時刻を指定します。現在時刻**now**(今から)の他、**10:30**のように開始時刻を指定することができます。終了時刻は**min**(分)の他、**hour**(時間)、または、**22:30**のように指定することができます。最大1日24時間で、日を跨って指定することもできます。

【備考3】 Microsoft OfficeがインストールされたWindows PCにコピーしてください。

【備考4】 Microsoft Officeのバージョンで使用するマクロが変わります。

Office 2000(XP)、2003では**grapher3.xls**、Office 2007、2010では**grapher7.xls**を使用します。

第 2 章 sardo の仕組みと考え方

2-1. sardo のコンセプト

システム性能管理の第一歩は通常の振る舞いを知っておくことです。通常の振る舞いとは、日次処理だけではなく、1 日の動作、週次処理における動作や月次、旬次、半期、年度末などのいつもと違う処理日の振る舞いも含めた全てです。オンライン処理のピークや平均の応答時間、バッチ処理がどれだけの時間、どの程度の負荷で動作するののかを知ることで、ある時点の振る舞いが正しい動きかどうかを判断することができるのです。

応答性能やバッチ性能の要件は通常、要求仕様書や SLA(Service Level Agreement)で明示されているはずですが、往々にしてそれらが無いまま、エンドユーザのクレームや、バッチ処理の終了が翌日のオンライン開始時刻に迫ってくる等の不具合で性能問題が顕在化することがあります。このような場合、システム運用管理者はメカジキ本(参考文献:UNIX パフォーマンスチューニングの名著です。その表紙にプリントされているのがメカジキです)を片手に各種の性能値とにらめっこして原因を探ります。

もっとも性質(たち)の悪いのは、非常に高価なシステム管理ソフトウェアを導入し、長い時間と労力をかけて数多くの性能値のスレシールド(しきい値)を設定しているシステムで、何らかのトラップが上がって来た場合です。「〇〇のトラップが上がりました!!」と管理ソフトウェアが言っています。ところが、それがどのような問題で発生したのかよく判らない場面です。これは決して稀ではなく、現実にとこにでもある風景です。

それらのシステム管理ソフトウェアはトラップを上げることはできますが、原因となる情報を正確に採取しているものは極めて少ないのです…

sardo は、システム管理ソフトウェアのようにトラップを上げる目的のツールではなく、「システムがどのように振る舞っているのかを正確に把握する」目的で作られたツールです。トラップ処理も可能ですが、本来の目的ではありません。また、一般のシステム管理ソフトウェアは、トラップのための処理に多くのシステム資源を消費していることは知られていません。

次に sardo のシステム管理者や開発担当者の作業負荷を徹底的に軽減するための優れた機能をご紹介します。

easy to use

複雑なコマンド、ファンクションを排除し、直感的なパラメタにしました。長いファイル名は入力を簡単にするため、選択を番号等で示し、判りやすくしました。また、PATH 環境変数に依存しませんので、シェルスクリプトの在処や/usr/bin等を意識する必要は全くありません。また、グラファーは、クリックのみの操作で、またたく間にグラフを描画してくれます。

正確な情報を採取

性能情報の多くは、ある時点のカーネルの持つテーブル(sysinfoと呼ばれています)の値を保存しておき、一定時間経過後の値をそれと比較、あるいは計算して求めます。この測定間隔を「サンプリングインターバル」と呼びます。「ある時点のプロセス数」や「ある時点のファイルオープン数」等はその時の値を表示するので問題はありませんが、「秒あたりのユーザ CPU 使用率」や「秒あたりの待ち率」等、計算によって求められる値は、サンプリングインターバルの長短によってなまりが生じます。このなまりの影響を少なくするために、特定の性能情報は 5 秒を基本としたサンプリングインターバルで採取し、情報を正確なものにしています。また、同じ測定区間で違ったサンプリングインターバルで情報を採取したい場合、複数の sardo を実行することができます。なお、付録 D. サンプリングタイムの違いによる測定値の変化に具体的な所見を述べていますので、ご覧下さい。

高いポータビリティ

採取された性能情報は tar(1)コマンドでアーカイブされ、gzip(1)コマンドで圧縮されます。これによりデータの移動が簡単です。

バッチモードの採用

日々の運用を助けるため、自動化のためのバッチモードを用意しました。通常はメッセージで知らされる実行結果は戻り値で返されます。この機能により、他のシェルスクリプトや cron(1)コマンドによる起動、停止処理が可能です。

容量見積りが簡単

性能情報を採取・保存するためのファイル容量を計算し、見積ることができます。この機能により、面倒な机上計算を無くしました。

しきい値管理が可能

発生確度をトレンド(傾向)ファイルに作成し、スレシヨルド(しきい値)を含むベースライン(基準値)ファイルを計算して出力します。このベースラインをカスタマイズすることにより、定常的な性能管理を簡単な操作で可能にしました。

2-2. 構成

sardo は 7 個のシェルスクリプト、2 個の実行可能バイナリモジュール、およびグラフャーで構成されます。性能情報採取は“sardo.sh”シェルスクリプト、性能情報の編集は“sardo.edit”シェルスクリプトを使用します。グラフャーは Office をインストールした Windows PC で実行します。

シェルスクリプト、及び実行可能モジュールは PATH 環境変数でパスを通した所定のディレクトリにインストールされることをお勧めします。

次に提供されるモジュールの一覧を示します。なお、グレーのモジュールは、sardo.sh、または sardo.edit から呼び出されるもので、直接使用することはありません。

表 2-2-1. 提供されるモジュール

No	名前	内容
1	sardo.divi	sardo.edit から呼び出される x86 用の高速エディタ
2	sardo.divs	sardo.edit から呼び出される SPARC 用の高速エディタ
3	sardo.edit	性能情報編集シェルスクリプト
4	sardo.eval	採取された性能情報を評価分析するシェルスクリプト
5	sardo.event	トラップや、その他の情報採取することが可能なシェルスクリプト
6	sardo.event.sample	sardo.event のサンプルシェルスクリプト
7	sardo.orainfo	Oracle や、その他の情報を採取するためのシェルスクリプト
8	sardo.orastat	Oracle や、その他の情報を採取するためのシェルスクリプト
9	sardo.sh	性能情報採取シェルスクリプト
10	grapher3.xls	VBA マクロを持つ、グラフ描画エクセルファイル(グラフャー) Office 2000、XP、および Office 2003 用
11	grapher7.xls	VBA マクロを持つ、グラフ描画エクセルファイル(グラフャー) Office 2007、および Office 2010 用

2-3. 性能情報採取と編集の流れ

次に、性能情報を採取してそのデータを編集するモデルを示しています。一つの採取単位を「sardo インスタンス」と呼びます。

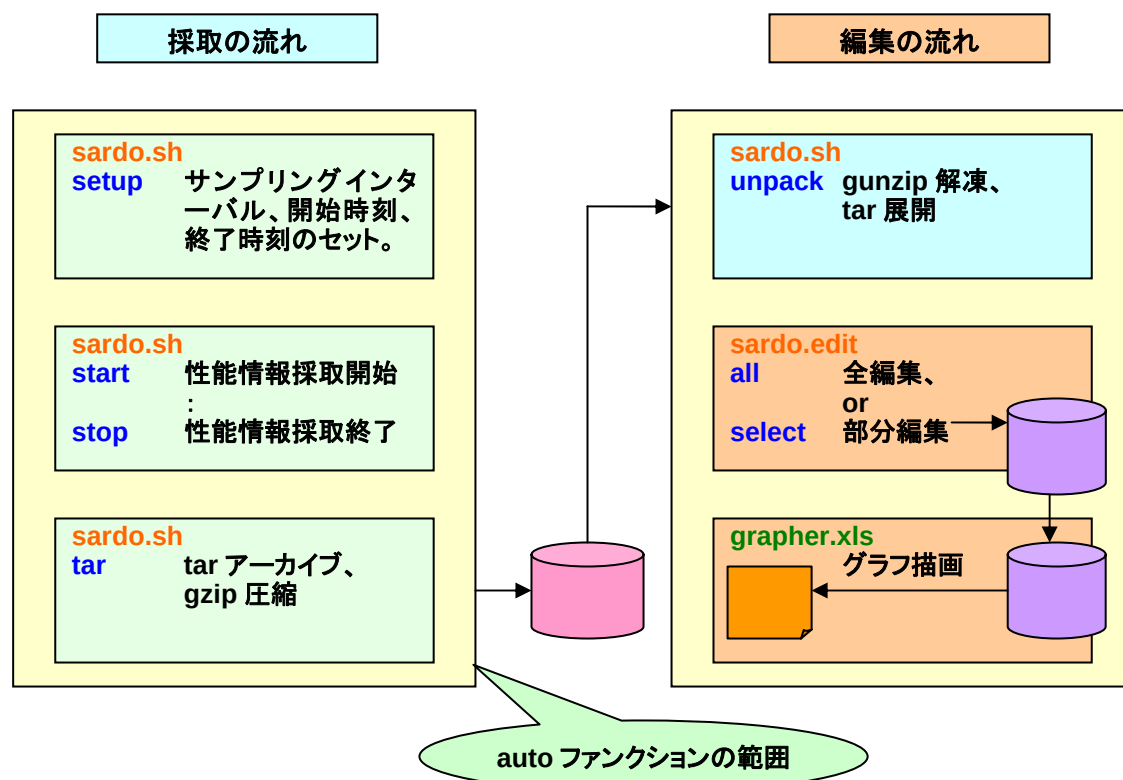


図 2-3-1. sardo の性能情報採取と編集のモデル

Solaris の OS バージョンは 2.5 以上です。一部の性能データは Solaris 8 以上で採取されますが、それ以前のバージョンでは採取されないものがあります。なお、Solaris 2.5 では、プロセスマップ形式が違います。

同じシステム内で性能情報編集中に sardo インスタンスを変えて編集することも可能です。sardo.edit の編集処理は、性能情報採取中に実行すべきではありません。CPU とディスク入出力の負荷が高いためです。

この理由は、性能情報を採取している一方で他の処理が入ると、CPU やディスクの負荷が本来の振る舞いから違ってしまい、採取目的の正しい性能値を得られないためです。

なお、sardo インスタンスを変更して性能情報を採取・編集する方法は環境変数をセットして、sardo.sh を実行します。詳細は、第 6 章 sardo の応用で解説します。

2-4. 性能情報採取コマンド

sardo.sh は、

- ① 採取開始時と終了時に採取する情報
- ② サンプリングインターバル単位、および 5 秒毎に採取する情報
- ③ プロセスアカウンティング情報
- ④ 採取中の操作によって、任意のタイミングで採取する情報

の大きく 4 種類の性能情報を採取します。本節で、各情報の内容を解説します。

2-4-1. 採取開始・終了時の採取情報

採取開始時と終了時に、同一コマンドによって情報を採取し、前後の値を比較することで変化をとらえます。次に開始・終了時に実行するコマンドを示します。

表 2-4-1-1. sardo.sh 採取開始・終了時の採取情報

No	コマンド/オプション	内容
1	crash(1M)	kmastat ファンクションによるカーネルメモリの詳細な情報とカーネルシンボル physinstalled の値(実装メモリサイズ)
2	df(1M) -k	1 ブロック 1,024 バイト単位のファイルシステムの空きと、使用ブロックの情報
3	ipcs(1) -a	UNIX プロセス間通信機能の使用状況
4	kstat(1M)	カーネルの持つシンボル、各種変数の詳細な値
5	mdb(1) -k ::memstat	OS のメモリ使用状況
6	netstat(1M) -s	TCP/IP、UDP のネットワーク統計情報
7	netstat(1M) -k	カーネルの各種変数値
8	pmap(1) -x	全プロセスのメモリ使用状況
9	swap(1M) -s	スワップ領域使用状況
10	vmstat(1M) -s	ブートからの各種システムイベント情報

2-4-2. サンプリングインターバル単位に採取する情報

サンプリングインターバル単位の情報とは、時間の流れとともに変化する値をとらえ、システム全体の振る舞いを知るために採取します。5 秒毎に採取されるデータは sar(1)コマンドの代わりに、バイナリファイルで性能情報を採取する sadc(1M)コマンドを使用しています。基本的に同じコマンドとお考え下さい。

sar(1)コマンドでは、長いサンプリングインターバルで情報を採取すると、値がなまってしまいう項目があるため、正しいシステムの振る舞いがつかめなくなります。これを防ぐため、基本の 5 秒単位で情報を採取しています。5 秒単位で sar(1)データを採取することで、なまりの無い性能情報の採取が可能になりました。一方、使用するディスク領域は多くなります。

次に、sardo.sh 実行中の性能情報採取コマンドを示します。

表 2-4-2-1. sardo.sh 実行中の性能情報採取コマンド

No	コマンドとオプション	採取間隔 【備考 1】	内容
1	iostat(1M) -xdn	n	ディスクアクティビティ
2	mpstat(1M)	n	構成されている全 CPU のアクティビティ
3	netstat(1M) -an	n	ネットワークの接続情報
4	netstat(1M) -l	n	構成されている全ての LAN インタフェースのパケット入出力情報
5	netstat(1M) -s	n	TCP/IP、UDP の統計情報
6	ps(1) ^{【備考 2】}	n	プロセス情報
7	vmstat(1M) -i	n	インタラプト情報
8	vmstat(1M)	n	性能情報全般をテキストファイルで採取
9	vmstat(1M) -p	n	メモリステータスを採取(Solaris 8 以上)
10	sadc(1M)	5	システムアクティビティレポートをバイナリで採取

【備考 1】 サンプリングインターバルです。“n”は実行時の引き数で指定された秒単位のインターバルです。
sadc(1M)コマンドのみ 5 秒の固定です。

【備考 2】 ps(1)コマンドの引き数は次の通りです。

-e -o user,pid,ppid,c,pri,nice,stime,etime,time,osz,rss,tty,args

2-4-3. プロセスアカウンティング情報

プロセスアカウンティングは、本来、監査の目的で使用する機能です。sardo では視点を変えた使用方法になっています。全てのプロセスの終了状態をはじめ、CPU 使用、メモリ使用量、システムコールの呼び出し回数など、貴重な情報を得ることが出来ます。

なお、プロセスアカウンティングデータは応用範囲が広いため、一層の活用方法を検討しています。

表 2-4-3-1. プロセスアカウント

No	コマンドとオプション	採取間隔 【備考 1】	内容
1	accton(1M) ^{【備考 2】}	-	プロセスアカウンティング情報

【備考 1】 sardo.sh 起動時にプロセスアカウンティングが実行されていないことを確認して accton(1M)コマンドを実行します。既に起動中の場合はメッセージでその旨を知らせます。この場合、sardo.edit でプロセスアカウン情報は編集されません。

【備考 2】 accton(1M)コマンドはプロセス終了時、そのプロセスのユーザ名、メモリ量、CPU 消費時間等の情報を 40 バイトのバイナリレコードにして記録します。

2-4-4. 操作によって任意のタイミングで採取する情報

プロセスが割り当てるメモリと、OS 全体のメモリ使用状況は、sardo の開始・終了時に採取しています。任意の時点で、これらの情報を入手したい場合、sardo.sh に引数を指定して実行することで、採取することができます。

sardo.sh の引数に“pmap”を指定して実行すると、全プロセスのメモリ使用状態を採取します。引数に“memstat”を指定して実行すると、OS 全体のメモリ使用状態を採取することができます。

プロセスのメモリ使用状態は pmap(1)コマンド-x オプション、OS 全体のメモリ使用状況は、mdb(1M)コマンド-k オプションのファンクション“::memstat”機能を使用しています。

pmap は実行中の全プロセスに対して“pmap -x”を実行するため、システムに負荷をあたえます。また、“mdb -k”コマンドも負荷が高いです。このため、タイミングを見計らって実行するようにしてください。

表 2-4-4-1. 任意のタイミングで採取できるメモリ情報

No	コマンドとオプション	採取タイミング	内容
1	mdb(1) -k ::memstat	任意	OS のメモリ使用状況(Solaris 10 以上)
2	pmap(1) -x	任意	全プロセスのメモリ使用状況

2-5. 性能情報採取・編集結果について

2-5-1. sardo の作成するファイル

sardo は CPU、メモリ、ディスク入出力、及びネットワークの 4 大性能要素毎に、最小、最大、平均値等をまとめたものや、性能項目に関する注目すべきポイントをファイルにして \$LOGDIR 環境変数の示す sarstats_\$(hostname)_\$(sid) ディレクトリ下に作成します。\$(hostname) 環境変数は採取されたシステムのホスト名、\$(sid) 環境変数は sardo.sh 実行時のインスタンスで、“001”がデフォルトです。このディレクトリを「sardo ディレクトリ」と呼びます。

次に sardo が作成するファイルを示します。緑色太字のファイルは sardo.sh が作成します。他は sardo.edit が編集時に作成します。ファイル口内の番号は、第 4 章の項番号を示しています。

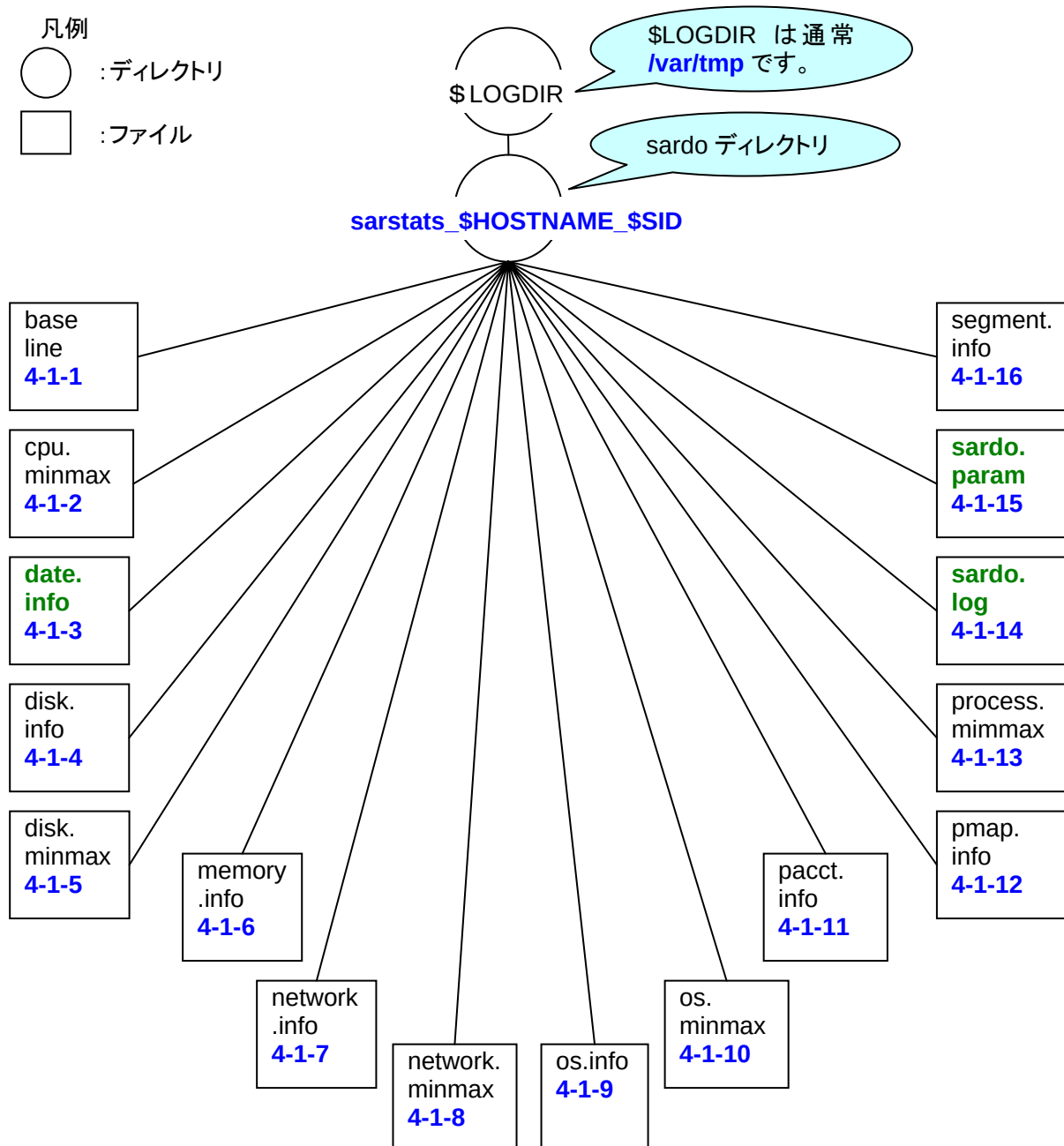


図 2-5-1-1. sardo の作成するファイル

次に、sardo の作成するファイルの概要を示します。

表 2-5-1-1. sardo の作成するファイル

参照項	ファイル名	内容
4-1-1	baseline	採取情報を基に計算したベースライン(基準値)
4-1-2	cpu.minmax	CPU に関する各種性能値の最小、最大、平均の情報
4-1-3	date.info	採取日付
4-1-4	disk.info	採取開始・終了時のディスクパーティションサイズ
4-1-5	disk.minmax	ディスク入出力に関する各種性能値の最小、最大、平均値
4-1-6	memory.info	メモリ使用に関する各種情報の計算値
4-1-7	network.info	ネットワーク使用に関する各種情報の計算値
4-1-8	network.minmax	ネットワークに関する各種性能値の最小、最大、平均値
4-1-9	os.info	OS の各種性能情報の計算値
4-1-10	os.minmax	OS に関する各種性能値の最小、最大、平均値
4-1-11	pacct.info	プロセスアカウントのサマリ情報
4-1-12	pmap.info	プロセスマップ
4-1-13	process.minmax	プロセスに関する CPU 消費時間(秒)の最小、最大、平均
4-1-14	sardo.log	sardo の実行ログ
4-1-15	sardo.param	sardo の実行パラメタ
4-1-16	segment.info	メモリセグメント情報

2-5-2. sardo の作成するディレクトリ

sardo の作成するディレクトリを次に示します。ディレクトリは、sardo.sh の実行時(性能情報採取時)、および sardo.edit の実行(性能情報の編集: 緑太字)によって作成され、各種性能情報のファイルを保存します。memstat ディレクトリは、“sardo.sh memstat”、pmap ディレクトリは、“sardo.sh pmap”ファンクションによって作成されます。なお、ディレクトリ内の番号は、第 4 章の節番号を示しています。

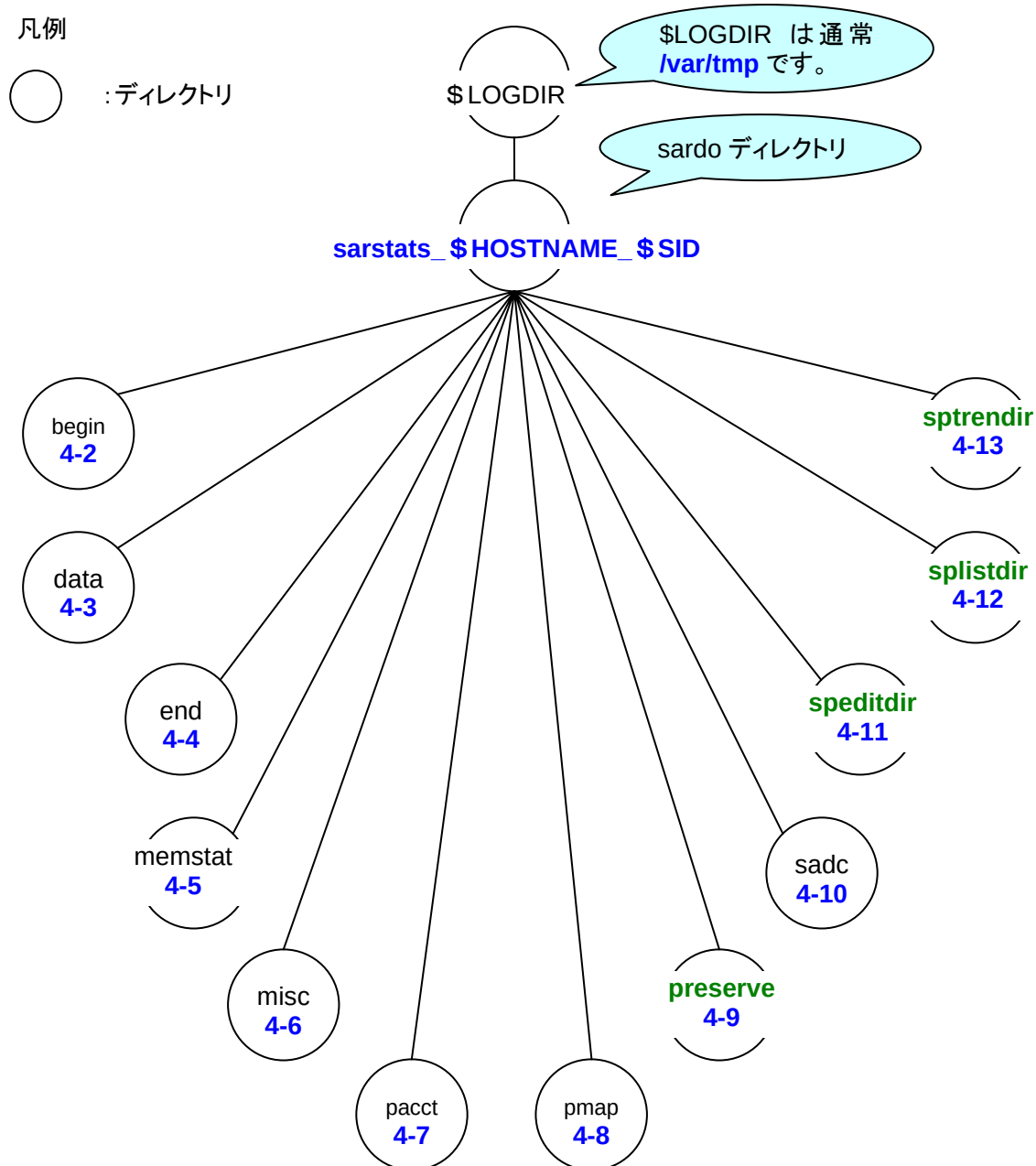


図 2-5-2-1. sardo の作成するディレクトリ

次に sardo の作成するディレクトリの内容を示します。

表 2-5-2-1. sardo の作成するディレクトリ

参照説	ディレクトリ名	内容
4-2	begin	sardo.sh の起動時に採取する情報
4-3	data	sardo.sh の実行中に採取する情報
4-4	end	sardo.sh の終了時に採取する情報
4-5	memstat	sardo.sh memstat によって採取される memstat 情報
4-6	misc	OS の各種情報
4-7	pacct	プロセスアカウンティングで採取されるバイナリファイル
4-8	pmap	sardo.sh pmap によって採取される全プロセスの pmap(1) -x 情報
4-9	preserve	sardo.edit によって作成される編集情報を保存するディレクトリ
4-10	sadc	sardo.sh の実行中に採取される sadc(1M)コマンドのバイナリファイル
4-11	speditdir	sardo.edit で作成される項目別に分割された編集結果
4-12	splistdir	sardo.edit で作成される編集出力
4-13	sptrendir	sardo.edit で作成されるネットワークと OS の傾向ファイル

※作成されるファイルの内容は、第 4 章 sardo のファイル形式 に解説します。ご覧ください。

第 3 章 sardo の操作

sardo をご使用になる前に

情報を採取・編集するためのディスク領域が必要です。sardo.sh、sardo.eval、及び sardo.edit シェルスクリプト実行時、LOGDIR 環境変数を空き領域のあるファイルシステムに設定して下さい。省略時は"/var/tmp"になります。

sh(1)の場合の LOGDIR 環境変数設定例

```
# LOGDIR=/opt/work <CR>
# export LOGDIR <CR>
```

csh(1)の場合の LOGDIR 環境変数設定例

```
% setenv LOGDIR /opt/work <CR>
```

sardo.sh は crash(1M)や adb(1)コマンドを実行するため、root(スーパーユーザ)の権限が必要です。sardo.edit と sardo.eval は一般ユーザでも実行可能ですが、ファイルパーミッションや、ulimit(1)のプロセス資源制限値等の関係上、root(スーパーユーザ)権限で実行することをお勧めします。

sardo にはこの他にいくつかの環境変数がありますが、通常の操作で設定することはないため、第 6 章 sardo の応用 で解説します。

なお、操作解説では各ファンクションの実行形式を解説します。“[]”(カギ括弧)のついていない引き数は必須のパラメタです。引き数が“[]”(カギ括弧)でくられているものはオプションです。

3-1. sardo.sh の操作

3-1-1. 自動実行 auto

自動的に性能情報を採取し、採取が終了するとファイルを tar(1)アーカイブ、gzip(1)圧縮するファンクションです。このファンクションは後述の setup、start、stop、及び tar ファンクションを組み合わせています。このため、起動後にその起動端末を電源オフや、ログアウトすると、採取プロセスをはじめ、終了処理プロセス等が終了してしまいます。

このような状況が考えられる場合は nohup(1)コマンドによって sardo.sh を起動するようにして下さい。

なお、auto ファンクションのメッセージは何も表示されず、ログファイル sardo.log にのみ記録されます。コマンドの終了状態は戻り値で判断することが出来ます。

実行形式

```
# sardo.sh auto interval starttime endtime <CR>
```

引き数の解説

第 1 引き数: “auto”です。

第 2 引き数: interval

サンプリングインターバルを指定します。秒単位で指定します。値の後ろに“sec”を付けると明示的に秒であることを示します。値の後ろに“min”を付けると分単位です。秒単位で指定可能な値は 5 から 3600 の間、分単位で指定可能な値は 1 から 60 の間です(最大 1 時間です)。

第 3 引き数: starttime

開始時刻を指定します。基本形式は HH:MM[:SS]の時刻です。

“now”を指定するとただちに採取を開始します。

第 4 引き数: endtime

終了時刻を指定します。基本形式は HH:MM[:SS]の時刻です。指定の簡略方式として値に“min”、または“hour”を付け、相対的に終了時間を指定することが出来ます。“min”は開始時刻から何分間、“hour”は開始時刻から何時間を示します。分単位で指定可能な値は 1 から 1440 の間、時間単位で指定可能な値は 1 から 24 の間です(最大 1 日、24 時間です)。

戻り値

0: 正常に起動されました。

1: 起動が誤りになりました。sardo.log ファイルをご確認下さい。

実行例

(1) 600 秒のサンプリングインターバルで 18:00:00 から 20:00:00 まで採取する例。

```
# sardo.sh auto 600 18:00:00 20:00:00 <CR>
#
```

18:00:00 まで時間が
あるとプロンプトは返
らずにその時間まで
待ちます。

(2) 5 秒のサンプリングインターバルで直ちに採取を開始し 10 分間採取する例。

```
# sardo.sh auto 5sec now 10min <CR>
#
```

(3) 1 分のサンプリングインターバルで 17:30:00 から 1 時間採取する例。

```
# sardo.sh auto 1min 17:30:00 1hour <CR>
#
```

現在時刻が 17:30:00
を過ぎているとただち
に開始します。

(4) 900 秒のサンプリングインターバルで直ちに採取を開始し 24 時間採取する例。

```
# sardo.sh auto 900sec now 24hour <CR>
#
```


3-1-2. 情報採取ディスク容量の見積り estimate

採取済みの圧縮 tar アーカイブファイルを基に、性能情報データのファイルサイズを見積るためのファンクションです。見積りデータとして、sardo ディレクトリに“estimate”の名前のファイルを作成します。具体的なファイルの解説は第 4 章 sardo のファイル形式 をご覧下さい。

実行形式

```
# sardo.sh estimate [ hostname sid directoryname ] <CR>
```

引き数の解説

第 1 引き数: “estimate”です。

第 2 引き数: **hostname**

目的の圧縮 tar アーカイブファイルのホスト名を指定します。無指定の場合、実行システムのホスト名が仮定されます。

第 3 引き数: **sid**

目的の圧縮 tar アーカイブファイルの SID を指定します。無指定の場合、実行環境の SARDO 環境変数の値が仮定されます。

SARDO 環境変数は第 6 章 sardo の応用 に解説します。

第 4 引き数: **directoryname**

目的の圧縮 tar アーカイブファイルが置かれているディレクトリ名を指定します。無指定の場合、sardo.sh の LOGDIR 環境変数で指定されたディレクトリ名が使用されます。

実行例

```
# sardo.sh estimate act061 001 /var/tmp <CR>
#
```

戻り値

0: 正常に終了しました。

1: 誤りが発生しました。

3-1-3. メモリ使用状況の出力 memstat

任意の時点で、メモリ全体の使用状況を mdb(1)コマンド-k オプション、::memstat ファンクションを使用して memstat ディレクトリに採取するファンクションです(Solaris 10 以上)。

実行形式

```
# sardo.sh memstat <CR>
```

引き数の解説

第 1 引き数: “memstat”です。

実行例

```
# sardo.sh memstat <CR>
INFO: an intermediate mdb ::memstat command was completed.
#
```

戻り値

0: 正常に終了しました。
1: 誤りが発生しました。

3-1-4. プロセスマップの出力 pmap

任意の時点で、実行されている全プロセスを pmap(1)コマンド-x オプションを使用して pmap ディレクトリに採取するファンクションです。

実行形式

```
# sardo.sh pmap <CR>
```

引き数の解説

第 1 引き数: “pmap”です。

実行例

```
# sardo.sh pmap <CR>
INFO: an intermediate pmap -x command was completed.
#
```

戻り値

0: 正常に終了しました。
1: 誤りが発生しました。

3-1-5. 手動実行のためのセットアップ setup

サンプリングインターバル、開始時刻、終了時刻を sardo.param ファイルに記録し、採取準備するためのファンクションです。

setup が終了すると次項に示す start ファンクションによって採取を開始します。

実行形式

```
# sardo.sh setup interval starttime endtime <CR>
```

引き数の解説

第 1 引き数: “**setup**”です。

第 2～4 引き数: auto ファンクションと同じ形式です。サンプリングインターバル、開始時刻、終了時刻を指定します。

実行例

(1) 300 秒のサンプリングインターバルで 19:00:00 から 20:00:00 迄を設定する例。

```
# sardo.sh setup 300 19:00:00 20:00:00 <CR>
INFO: spsetup was successfully completed.
#
```

(2) 10 秒のサンプリングインターバルで現在時刻から 20 分後を設定する例。

```
# sardo.sh setup 10sec now 20min <CR>
INFO: spsetup was successfully completed.
#
```

(3) 20 分のサンプリングインターバルで 12:30:00 から 8 時間後を設定する例。

```
# sardo.sh setup 20min 12:30:00 8hour <CR>
INFO: spsetup was successfully completed.
#
```

(4) 600 秒のサンプリングインターバルで現在時刻から 20 時間後を設定する例。

```
# sardo.sh setup 600sec now 20hour <CR>
INFO: spsetup was successfully completed.
#
```

戻り値

0: 正常に終了しました。

1: 誤りが発生しました。

3-1-6. 手動による採取開始 start

setup ファンクションの設定に基づき採取を開始するファンクションです。auto ファンクションと同様に、起動後にその起動端末を電源オフや、ログアウトすると、採取プロセスをはじめ、終了処理プロセスが終了してしまいます。このような状況が考えられる場合は、nohup(1)コマンドによって sardo.sh を起動するようにして下さい。

実行形式

```
# sardo.sh start <CR>
```

引き数の解説

第 1 引き数: “**start**”です。

実行例

```
# sardo.sh setup 600sec now 20hour <CR>
INFO: spsetup was successfully completed.
# sardo.sh start <CR>
INFO: sampling interval : 600
INFO: sampling count : 118
INFO: current time : 63274 17:34:34
INFO: start and end time : 63152 17:32:32 133952 13:32:31
INFO: sampling mode : OVERTHEDAY
INFO: sardo will be started with 600 sec interval.
INFO: process accounting has been started.
INFO: crash - was successfully completed.
INFO: df -k was successfully completed.
INFO: ipcs -a was successfully completed.
INFO: netstat -s was successfully completed.
INFO: netstat -k was successfully completed.
INFO: pmap -x was successfully completed.
INFO: swap -s was successfully completed.
INFO: netstat - was successfully scheduled.
INFO: netstat - was successfully scheduled.
INFO: vmstat -p was successfully scheduled.
INFO: iostat -xdn was successfully scheduled.
INFO: vmstat - was successfully scheduled.
INFO: mpstat - was successfully scheduled.
INFO: ps - was successfully scheduled.
INFO: vmstat -i was successfully scheduled.
INFO: netstat -s was successfully scheduled.
INFO: netstat -an was successfully scheduled.
INFO: sadc command was successfully scheduled.
INFO: wait for the end time [ 13:32:31 ] to finish.
INFO: termproc pid [ 22896 ] was started successfully.
#
```

戻り値

- 0: 正常に起動されました。
- 1: 起動が誤りで終了しました。sardo.log ファイルをご確認下さい。

3-1-7. 強制停止 stop force

sardo.sh 実行中に強制的に停止するファンクションです。

実行形式

```
# sardo.sh stop force <CR>
```

引き数の解説

第 1 引き数: “stop”です。

第 2 引き数: “force”です。必ず指定してください。

実行例

```
# sardo.sh stop force <CR>
.....Killed
..Killed
..Killed
..Killed
....
INFO: crash - was successfully completed.
INFO: df -k was successfully completed.
INFO: ipcs -a was successfully completed.
INFO: netstat -s was successfully completed.
INFO: netstat -k was successfully completed.
INFO: pmap -x was successfully completed.
INFO: swap -s was successfully completed.
INFO: process accounting was stopped.
INFO: list devinfo was successfully completed.
#
```

戻り値

0: 正常に終了しました。

1: 誤りが発生しました。

3-1-8. アーカイブの作成 tar

採取終了した sardo ディレクトリを tar アーカイブし、gzip(1)コマンドで圧縮するためのファンクションです。このファンクション実行時、sardo ディレクトリの下のディレクトリで起動すると sardo ディレクトリの削除が行われませんのでご注意ください。

実行形式

```
# sardo.sh tar <CR>
```

引き数の解説

第 1 引き数: “tar”です。

実行例

```
# sardo.sh tar <CR>
a sarstats_act061_001/ 0K
a sarstats_act061_001/sardo.param 1K
a sarstats_act061_001/sardo.log 3K
a sarstats_act061_001/date.info 1K
a sarstats_act061_001/pacct/ 0K
a sarstats_act061_001/pacct/pacct_info 36K
a sarstats_act061_001/misc/ 0K
      :(途中省略)
a sarstats_act061_001/end/crash_info 16K
a sarstats_act061_001/end/crash_info 16K
a sarstats_act061_001/end/df_k_info 1K
a sarstats_act061_001/end/ipcs_a_info 2K
a sarstats_act061_001/end/netstat_s_info 6K
a sarstats_act061_001/end/netstat_k_info 79K
a sarstats_act061_001/end/pmap_x_info 209K
a sarstats_act061_001/end/swap_s_info 1K
a sarstats_act061_001/sadc/ 0K
a sarstats_act061_001/sadc/sadc_info 207K
INFO: tar archive and compress were successfully completed.
#
```

戻り値

0: 正常に終了しました。
1: 誤りが発生しました。

3-1-9. 展開 unpack

tarファンクションで作成された圧縮tarアーカイブファイルを解凍、復元するファンクションです。複数のファイルが\$LOGDIR ディレクトリ下に存在した場合はファイルを選択する旨のメッセージが表示されます。この場合、番号を応答することでファイルを選択することが出来ます。

実行形式

```
# sardo.sh unpack <CR>
```

引き数の解説

第 1 引き数: “unpack”です。

実行例

```
# sardo.sh unpack <CR>
1. sarstats_act011_001_030802145148.tar.gz Aug 2 14:51 500073
2. sarstats_act061_001_030802131815.tar.gz Aug 2 13:18 316517
Several unpack files found.

Select file number ? : 2 <CR>
INFO: the unpack file sarstats_act061_001_030802131815.tar.gz was chosen.
x sarstats_act061_001, 0 bytes, 0 tape blocks
x sarstats_act061_001/sardo.param, 21 bytes, 1 tape blocks
x sarstats_act061_001/sardo.log, 2600 bytes, 6 tape blocks
x sarstats_act061_001/date.info, 11 bytes, 1 tape blocks
x sarstats_act061_001/pacct, 0 bytes, 0 tape blocks
x sarstats_act061_001/pacct/pacct_info, 44040 bytes, 87 tape blocks
: (途中省略)
x sarstats_act061_001/end/crash_info, 15569 bytes, 31 tape blocks
x sarstats_act061_001/end/df_k_info, 514 bytes, 2 tape blocks
x sarstats_act061_001/end/ipcs_a_info, 1248 bytes, 3 tape blocks
x sarstats_act061_001/end/netstat_s_info, 6064 bytes, 12 tape blocks
x sarstats_act061_001/end/netstat_k_info, 79732 bytes, 156 tape blocks
x sarstats_act061_001/end/pmap_x_info, 213266 bytes, 417 tape blocks
x sarstats_act061_001/end/swap_s_info, 99 bytes, 1 tape blocks
INFO: spunpack was successfully completed.
#
```

戻り値

0: 正常に終了しました。
1: 誤りが発生しました。

3-2. sardo.edit の操作

3-2-1. 全データの編集 all

sardo ディレクトリ下の採取データを編集するファンクションです。sardo.edit は高速のファイル編集プログラムを持っており、出来るだけ短時間で編集出来るよう工夫されていますが、大量の採取データの場合、編集時間がかかることがあります。このような場合、いくつかの編集をスキップする方法もあります。編集をスキップする方法は第 6 章 sardo の応用 に解説します。

実行形式

```
# sardo.edit [all] <CR>
```

引き数の解説

第 1 引き数: “all”です。省略時、all 指定と見なされます。

実行例

```
# sardo.edit all <CR>
.....
INFO: making os info sar_x files has been done.
INFO: making vmstat_info has been done.
INFO: making vmstat_p_info has been done.
.....
INFO: making sar -d information has been done.
.....
INFO: making iostat_xdn_* files has been done.
.
INFO: making mpstat_cpu_info files has been done.
.....
INFO: making vmstat_i_* files has been done.
...
INFO: making netstat_IF_file files has been done.
INFO: making ps info files was started.
      : (途中省略)
INFO: making minmax files has been done.
INFO: making memory.info file has been done.
...
INFO: making network.info file has been done.
INFO: making os.info file has been done.
.....
INFO: making trend files has been done.
#
```

戻り値

0: 正常に終了しました。
1: 誤りが発生しました。

3-2-2. 部分データの抽出編集 select

ある時間帯のデータを抽出して編集するファンクションです。一旦、sardo.edit で全編集すると preserve ディレクトリに編集情報が作成されます。select はその preserve ディレクトリの編集情報から指定した部分を抽出するファンクションです。

実行形式

```
# sardo.edit select starttime endtime <CR>
```

引き数の解説

第 1 引き数: “select”です。

第 2 引き数: starttime

抽出の開始時刻を HH:MM[:SS]形式で指定します。

第 3 引き数: endtime

抽出の終了時刻を HH:MM[:SS]形式で指定します。

実行例

```
# sardo.edit select 16:43:52 18:42:50 <CR>
```

```
      :  
基本的に“all”と同じメッセージです。このため省略させていただきます。  
      :
```

```
#
```

戻り値

0: 正常に終了しました。

1: 誤りが発生しました。

3-3. sardo.eval の操作

sardo.edit の編集した性能データを基にして性能情報の値を評価するシェルスクリプトです。baseline ファイルは項目の取捨選択や、しきい値、評価時間の変更等のカスタマイズが可能です。sardo.eval は baseline ファイルを基に、現在の性能データを評価・分析します。

baseline ファイルの内容、項目等の解説は第 4 章 sardo のファイル形式、sardo.eval の解説は第 6 章 sardo の応用 をご覧下さい。

実行形式

sardo.eval <CR>

引き数の解説

引き数はありません。

実行例

●全て OK の例

```
# sardo.eval <CR>
%usr          OK
%sys          OK
%wio          OK
runq-sz       OK
freemem       OK
freeswap      OK
pgscan/s      OK
inpackets     OK
outpackets    OK
#
```

●freemem がベースラインを下回った場合の例

```
# sardo.eval <CR>
%usr          OK
%sys          OK
runq-sz       OK
freemem       NG UNDER 23:25:15 98499 5 81 89
freeswap      OK
pgscan/s      OK
inpackets     OK
outpackets    OK
#
```

戻り値

- 0: 正常に終了しました。
- 1: 条件を超えた(又は下回った)データを発見しました。

3-4. グラフの描画 grapher.xls

sardo.edit の結果、speditdir ディレクトリ下に作成された cpu、disk、network、os、及び process の性能要素単位に分かれているディレクトリを選択してグラフを描画します。Microsoft Office がインストールされている Windows PC で実行します。

- (1) grapher n .xlsを選択して「マクロを有効にする(E)」をクリックします。セキュリティは「低」にしてください。グラフャーの“ n ”はOfficeのレベルです【備考 1】。
- (2) 「Graphs will be～Please click here.」のテキストボックスをクリックします。

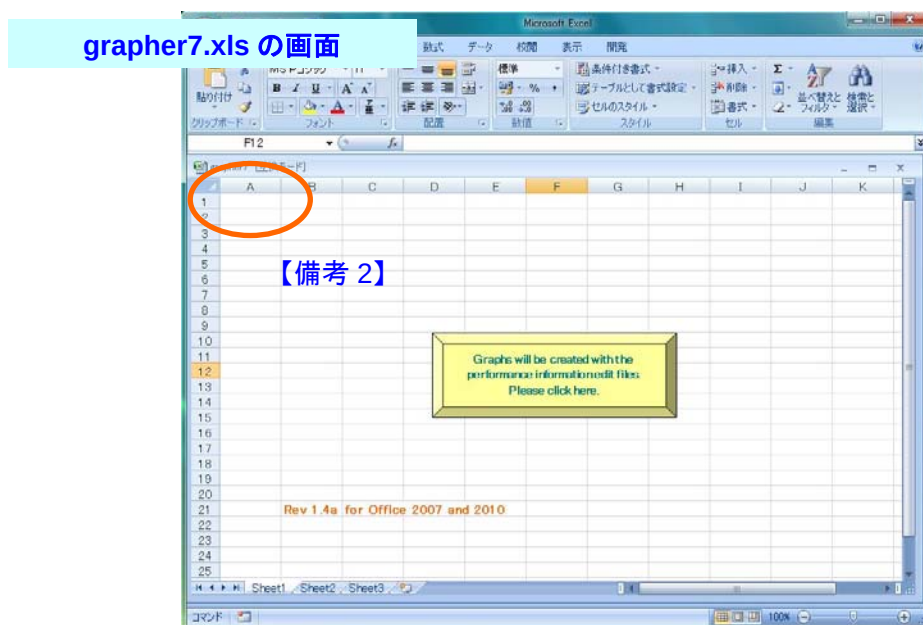
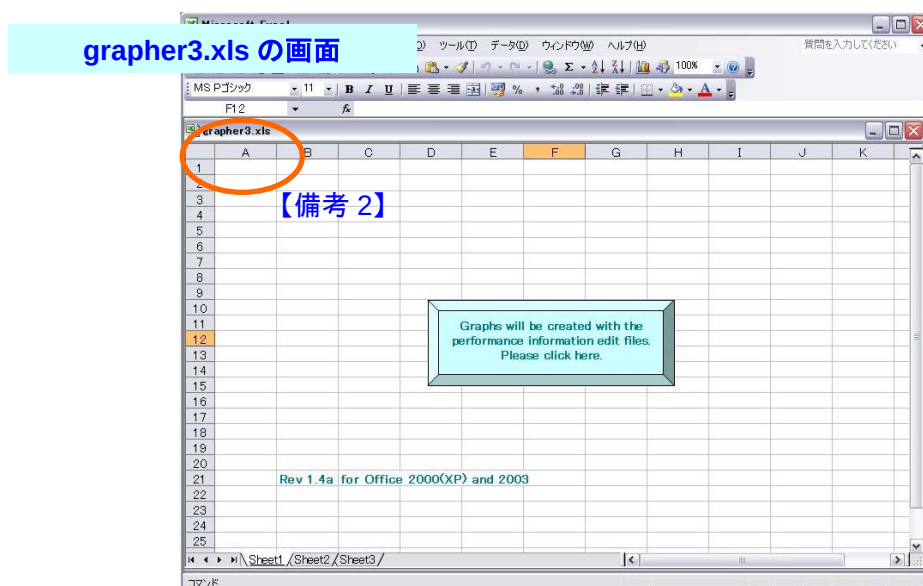


図 3-4-1. グラフ描画エクセル

【備考 1】 Office 2000(XP)、2003 用のマクロは“3”、Office 2007、2010 用のマクロは“7”です。

【備考 2】 グラフャーの初期画面ではカーソル位置は F12 になっています。A1 カラムをクリックして値を入れると、実印刷のコピー数として扱われます。Windows PC にプリンターが接続されている場合、グラフ描画された各性能要素が印刷されます。

- (3) フォルダ選択画面が表示されます。図 3-4-2 で、一番先頭のフォルダ“**sarstats_x2270_001**”を選択した場合、全部のグラフを描画することができます。



図 3-4-2. フォルダ選択画面

- (4) 単一の性能要素をグラフ描画する場合、speditdir フォルダ下の性能要素単位に分かれている“**os**”などのフォルダを選択してグラフを描画します。次のように、目的のフォルダを選択して **OK** をクリックします。



図 3-4-3. 性能要素の選択画面

(5) フォルダ名を確認の上、**はい(Y)**をクリックします。

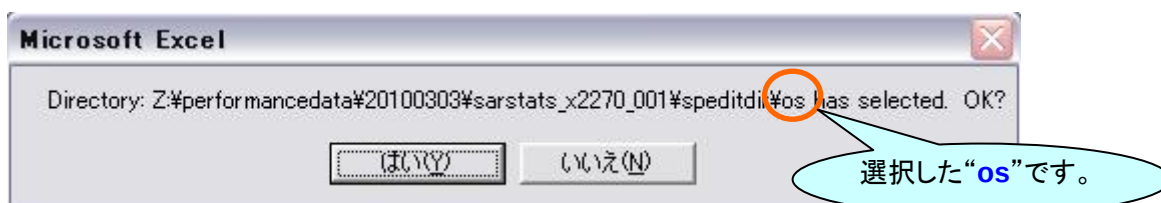


図 3-4-4. フォルダ確認画面

(6) 確認のメッセージボックスが表示されますのでOKでしたら**はい(Y)**をクリックします。

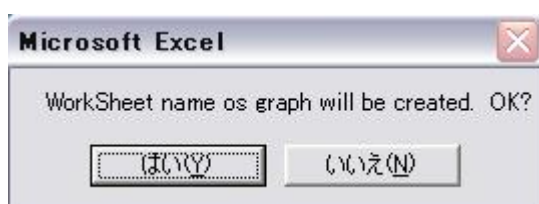


図 3-4-5. グラフ名の確認メッセージボックス

(7) 続いて日付確認のためのメッセージボックスが表示されます。この日付はdate.infoファイル内の日付で、作成されるグラフにスタンプされます。次の図は、確認のためのメッセージボックスです。OKであれば**はい(Y)**をクリックします。他の日付に変更したい場合、**いいえ(N)**をクリックすると図 3-4-7 の入力ボックスが表示されます。任意の日付を入力して下さい。図 3-4-8 は確認のためのメッセージボックスです。

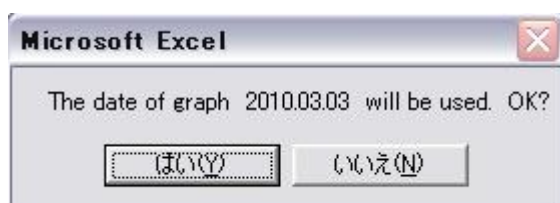


図 3-4-6. グラフ日付確認のメッセージボックス

(8) グラフ日付確認のメッセージボックスに**はい(Y)**を応答するとグラフの描画が始まります。**いいえ(N)**を応答すると日付入力のための入力ボックスが表示されます。任意の日付を入力し、**OK**をクリックすると再度確認のメッセージボックスが表示されます。

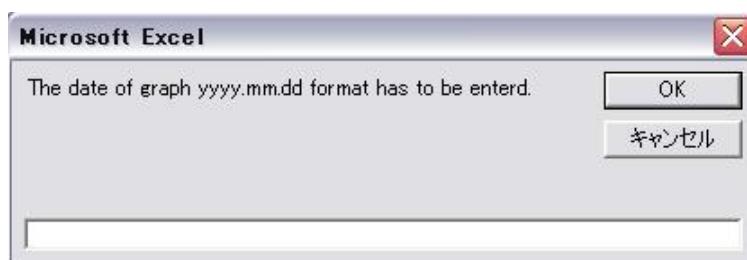


図 3-4-7. 日付入力の入力ボックス

- (9) グラフになってからヘッダーやフッターを操作するのは手間がかかります。くどい操作のようですが手間を掛けないため、また、手作業を出来るだけ少なくするために、正しい日付を入力するようにしましょう。をクリックするとグラフの描画が始まります。をクリックするとステップ(8)に戻ります。日付を変更する場合はキャンセルし、再度ステップ(2)から操作してください。



図 3-4-8. 再度の日付確認メッセージボックス

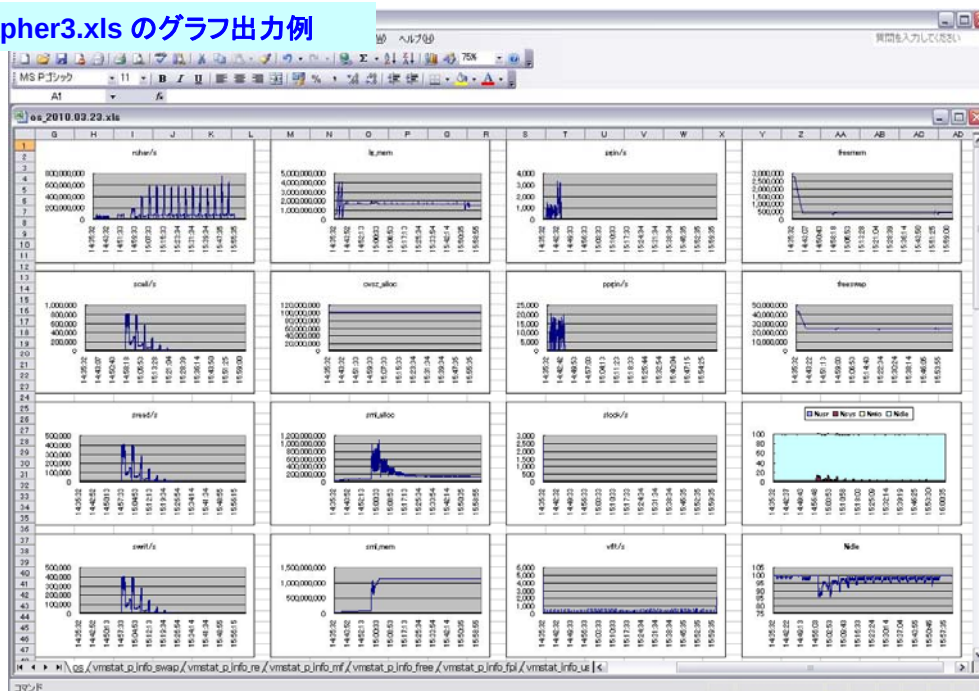
- (10) しばらくの間グラフが描画されます。シートは A4 サイズ横です。grapher3.xls では 1 ページに 30 個のグラフ、grapher7.xls では 1 ページに 16 個のグラフが収まります。なお、Windows PC にプリンターが構成されている場合、図 3-4-1 でボタンをクリックする前に、A1 カラムに、“1”を入れると 1 部、“2”を入れると 2 部がプリンターに印書されます。
- (11) グラフ描画が終了すると、前述の図 3-4-5 で示されたグラフ名(すなわち選択した性能要素名)に日付を付けたブック名でグラフファイルが作成されます。ファイルの作成場所はマクロの実行フォルダです。次ページの図 3-4-9 に作成されたグラフを示します。
- (12) (1)～(11)を繰り返して目的のグラフを描画・印書して下さい。

grapher3.xlsはOffice 2000、XP、または 2003 をインストールしたWindows PCで実行できます。
grapher7.xlsはOffice 2007、または 2010 をインストールしたWindows PCで実行できます【備考】。

※Solaris の StarOffice や、Linux の OpenOffice では実行できません。

【備考】 Windows Vista や Windows 7 + Office 2007、2010 で grapher7.xls を実行すると非常に遅いことがわかりました。互換モードのためか、現在原因を調査中です。類似のスペックを持つ PC で Windows XP + Office 2003 + grapher3.xls の組み合わせは高速に描画できました。このため、グラフの描画は、Windows XP + Office 2000 または 2003 + grapher3.xls をお勧めします。なお、ウィルスソフトが入っていると一層遅くなります… ;-(

grapher3.xls のグラフ出力例



grapher7.xls のグラフ出力例

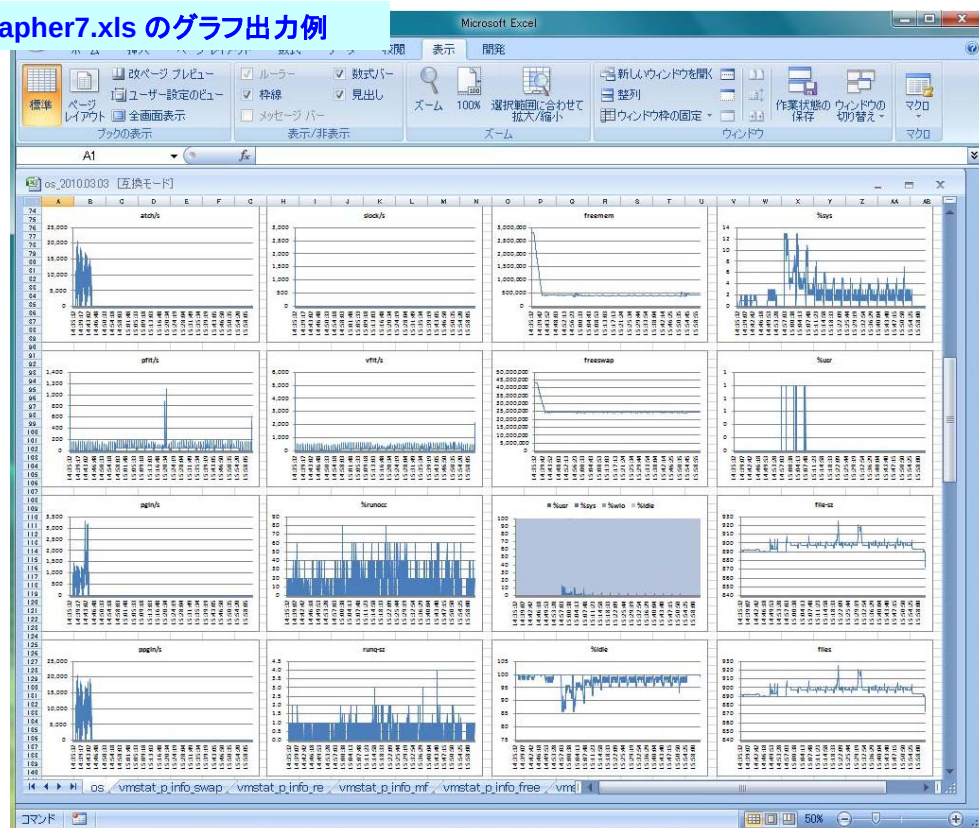


図 3-4-9. グラフの出力例

第 4 章 sardo のファイル形式

sardo ディレクトリ下に作成される各種ファイルの形式を解説します。

4-1. sardo ディレクトリ下のファイル形式

4-1-1. baseline ベースライン

baseline は主要な性能値の基準となるベースを示すファイルです。sardo.edit の編集によって作成されます。sardo.edit は、特定の性能値が示すピーク値がどれだけ持続したかを確認して baseline を作成します。持続期間はサンプリングインターバルの単位です。sardo.eval コマンドは、baseline ファイルをもとに、それより負荷が高いか、または低いかを検査します。詳細は第 6 章 sardo の応用をご覧ください。

次に baseline ファイルの例を示します。

# cat baseline				
%usr	86	1min	6	over
%sys	77	130	24	over
runq-sz	41	65	13	over
freemem	99386	1min	5	under
freeswap	5912721	1min	5	under
pgscan/s	200	300	2	over
inpackets	621	5min	4	over
outpackets	1072	5min	4	over
#				

図 4-1-1-1. baseline ファイル

4-1-2. cpu.minmax CPU 性能値の最小・平均・最大

CPU 性能値の最小・最大値とその発生時刻、および平均・合計値を示すファイルです。

次に cpu.minmax ファイルの例を示します。

root@act011 # cat cpu.minmax	cpu_0_csw	87 22:48:45	3318 22:53:45	979	29383
mpstat_0_info_csw	cpu_0_icsw	0 22:48:45	327 22:57:45	112	3348
mpstat_0_info_idl	cpu_0_idl	0 22:53:45	100 22:48:45	36	1070
mpstat_0_info_intr	cpu_0_intr	263 22:48:45	1208 23:02:45	833	25000
mpstat_0_info_ithr	cpu_0_ithr	49 22:47:45	476 22:51:45	282	8445
mpstat_0_info_migr	cpu_0_migr	3 22:47:45	331 23:15:45	172	5149
mpstat_0_info_minf	cpu_0_minf	26 22:55:45	32975 23:16:45	18416	552434
mpstat_0_info_smtx	cpu_0_smtx	3 22:48:45	3043 23:16:45	1460	43805
mpstat_0_info_srw	cpu_0_srw	0 22:47:45	4 22:53:45	1	20
mpstat_0_info_sys	cpu_0_sys	0 22:48:45			1265
mpstat_0_info_syscl	cpu_0_syscl	0 22:48:45			345620
mpstat_0_info_usr	cpu_0_usr	0 22:48:45			663
mpstat_0_info_xcal	cpu_0_xcal	0 22:48:45	569 22:57:45	17	3886
mpstat_1_info_csw	cpu_1_csw	0 22:48:45	3162 22:54:45	971	28166
mpstat_1_info_icsw	cpu_1_icsw	0 22:48:45	238 22:58:45	110	3185
mpstat_1_info_idl	cpu_1_idl	0 22:54:45	93 22:49:45	37	1069
mpstat_1_info_intr	cpu_1_intr	155 22:50:45	1055 22:58:45	623	18075
mpstat_1_info_ithr	cpu_1_ithr	81 22:48:45	219 22:54:45	173	5008
mpstat_1_info_migr	cpu_1_migr	3 22:48:45	336 23:07:45	169	4894
mpstat_1_info_minf	cpu_1_minf	43 22:49:45	32789 23:13:45	17719	513854
mpstat_1_info_smtx	cpu_1_smtx	3 22:49:45	3002 23:13:45	1402	40659
mpstat_1_info_srw	cpu_1_srw	0 22:48:45	6 22:53:45	1	21
mpstat_1_info_sys	cpu_1_sys	1 22:48:45		45	1197
mpstat_1_info_syscl	cpu_1_syscl	152 22:50:45		6	330278
mpstat_1_info_usr	cpu_1_usr	0 22:48:45		22	635
mpstat_1_info_xcal	cpu_1_xcal	0 22:49:45	238 22:58:45	81	2349
vmstat_1_Total_info_rate	Total_rate	100 22:46:49	100 22:46:49	100	3000
vmstat_1_Total_info_total	Total_total	360475 22:46:49	534884 23:15:53	447759	13432772
vmstat_1_clock_info_rate	clock_rate	100 22:46:49	100 22:46:49	100	3000
vmstat_1_clock_info_total	clock_total	360468 22:46:49	534877 23:15:53	447752	13432562
vmstat_1_fdc0_info_total	fdc0_total	7 22:46:49	7 22:46:49	7	210
root@act011 #					

図 4-1-2-1. cpu.minmax ファイル

【備考】 合計する意味の無い項目もあります。ご了承ください。

4-1-3. date.info 採取日付

sardo.sh 実行時の日付がセットされるファイルです。

```
# cat date.info
2010.02.17
#
```

図 4-1-3-1. date.info ファイル

4-1-4. disk.info ディスク容量の採取前と後

sardo.sh 実行時、情報採取前後のディスク容量を確認するためのファイルです。df(1)コマンド-k オプションの出力を編集したもので、単位はキロバイトで表示されます。

```
# cat disk.info
filesystem      kbytes      used      used%
begin
/                239468544   6205415   2.88
/lib/libc.so.1   221728295   6205415   3.11
/export          239468544   23        0.00
/export/home     239468544   26        0.00
/export/home/cxadmin 239468544   328       0.00
/export/home/dbadmin 239468544   253215    0.12
/export/home/katsumi 239468544   345       0.00
/export/home/oracle 239468544   5332085   2.47
/oracledb        239468544   6507508   3.02
/oracleinventory 239468544   2725      0.00
/rpool           239468544   34        0.00
/www             239468544   396039    0.18
total            2855882279  24903158  0.97
/tmp             6505820     92        0.00
end
/                239468544   6239353   2.90
/lib/libc.so.1   221706766   6239353   3.13
/export          239468544   23        0.00
/export/home     239468544   26        0.00
/export/home/cxadmin 239468544   328       0.00
/export/home/dbadmin 239468544   253215    0.12
/export/home/katsumi 239468544   345       0.00
/export/home/oracle 239468544   5332085   2.47
/oracledb        239468544   6528963   3.03
/oracleinventory 239468544   2725      0.00
/rpool           239468544   34        0.00
/www             239468544   396039    0.18
total            2855860750  24992489  0.97
/tmp             3236012     68        0.00
#
```

図 4-1-4-1. disk.info ファイル

【備考】 上記例は ZFS ファイルシステムの例です。UFS の場合、各々のファイルシステム単位の容量と割当量が見易くなっています。

4-1-5. disk.minmax ディスク性能値の最小・平均・最大

ディスク性能値の最小・最大値とその発生時刻、および平均・合計値を示すファイルです。

次に disk.minmax ファイルの例を示します。

```

root@act011 # cat disk.minmax
iostat_xdn_cld0_%b      1 22:48:45      66 22:51:45      11      341
iostat_xdn_cld0_%w      0 22:48:45      39 22:51:45       4      123
iostat_xdn_cld0_actv     0 22:48:45       1 22:51:45       0       5
iostat_xdn_cld0_asvc_t   3 22:48:45      13 22:51:45       5      143
iostat_xdn_cld0_kr/s     0 23:07:45     8941 22:51:45     813    24404
iostat_xdn_cld0_kw/s     13 22:50:45     3031 22:58:45     599    17959
iostat_xdn_cld0_r/s      0 22:48:45       72 22:51:45       7      198
iostat_xdn_cld0_w/s      2 22:49:45      76 22:52:45      20     598
iostat_xdn_cld0_wait     0 22:48:45       5 22:51:45       1       19
iostat_xdn_cld0_wsvc_t   0 22:49:45      57 22:50:45      14     423
sar_d_cmdk0_%busy       0 22:48:54      98 22:50:40     12    4170
sar_d_cmdk0_avque       0 22:48:54      17 22:50:14       1     290
sar_d_cmdk0_avserv      0 22:48:54      17 22:50:30       5    1625
sar_d_cmdk0_avwait      0 22:48:54     130 22:50:14       7    2682
sar_d_cmdk0_bks/s       0 22:48:54    43554 22:50:40    2830   1021576
sar_d_cmdk0_r+w/s       0 22:48:54      217 23:02:46      26     9547
root@act011 #

```

図 4-1-5-1. disk.minmax ファイル

※各列の意味は図 4-1-2-1 と同様です。

4-1-6. memory.info メモリ消費ページとバイト

メモリの消費量を計算した結果のファイルです。

次に memory.info ファイルの例を示します。

# cat memory.info			ページ数【備考】
PhysInstalled	1031722	4225933312	
Phymem	1029673	4217540608	
UnixSize	2049	8392704	
KernelMemory(avg)	501815	2055434240	
PageTableArraySize	16120	66027520	
FreeMemory(avg)	202547	829632512	
UserMemory(end)	188609	772542464	
FreeSwap(avg)	915786	3751059456	
DiskBufferCache(end)1/100	62258	255008317	
DiskBufferCache(end)2/100	124516	510016635	バイト数
#			

図 4-1-6-1. memory.info ファイル

次に、memory.info ファイルの項目を示します。

表 4-1-6-1. memory.info ファイルの項目

No	項目名	内容
1	PhysInstalled	実装メモリサイズ
2	Phymem	物理メモリサイズ(PhysInstalled - Phymem=UnixSize)
3	UnixSize	カーネルサイズ
4	KernelMemory(avg)	カーネルメモリ平均値
5	PageTableArraySize	ページテーブル構造体配列
6	FreeMemory(avg)	平均空きメモリ(ページ単位)【備考】
7	UserMemory(xxx)	ユーザプロセスの合計サイズ。sardo.sh 開始時と終了時のサイズを比較し、大きい方の値を表示。“()”(括弧)内の xxx は開始時の場合“bgn”、終了時の場合“end”。
8	FreeSwap(avg)	平均空きスワップ
9	DiskBufferCache(xxx)1/100	使用中の全ファイルの df(1)コマンド-k オプションの“used”ブロックの 1 パーセント値(90:10 の理論と呼ばれる、アプリケーションがアクセスするディスクデータはディスク全域の 10/100 に集中し、さらにその 10 パーセントがフラクタルに使用されると言われています。この 1 パーセントの領域がディスクバッファキャッシュに存在すると仮定した場合の推定値。“()”(括弧)内の xxx は開始時の場合“bgn”、終了時の場合“end”。
10	DiskBufferCache(xxx)2/100	No.9 の 2 倍値。“()”(括弧)内の xxx は開始時の場合“bgn”、終了時の場合“end”。

【備考】 ページサイズは、i386 の場合 4,096 バイト、SPARC の場合 8,192 バイトです。

4-1-7. network.info ネットワーク性能値のまとめ

ネットワーク関連性能値をまとめたファイルです。一部、計算して値を出しています。

次に network.info ファイルを示します。

# cat network.info		
udpInDatagrams	0	項目
udpOutDatagrams	0	
InPackets	4707	バイトや個数などを示す値。
InPacketPerSec	3	
OutPackets	6251	
OutPacketsPerSec	3	
tcpInInorderBytesBegin	8614	
tcpInInorderBytesEnd	201485	
tcpInInorderBytes	192871	
tcpOutDataBytesBegin	11995	
tcpOutDataBytesEnd	553988	
tcpOutDataBytes	541993	
TCPAverageInPacketSize	36	
TCPAverageOutPacketSize	100	
AverageInBps	2214	
AverageOutBps	4148	
AverageTotalThroughPut	6362	
MaximumInputBps	6170	
MaximumOutputBps	25101	
MaximumTotalThroughPut	31271	
#		

図 4-1-7-1. network.info ファイル

次に、network.info ファイルの項目を示します。

表 4-1-7-1. network.info ファイルの項目

No	項目名	内容
1	AverageInBps	入力の平均秒あたりビット数
2	AverageOutBps	出力の平均秒あたりビット数
3	AverageTotalThroughPut	入出力の合計秒あたりビット数 (AverageInBps + AverageOutBps)
4	InPacketPerSec	秒あたりの平均入力パケット数
5	InPackets	LAN インタフェース全部の合計入力パケット数
6	LANcollision	LAN インタフェース単位の netstat(1M)コマンド-I オプションの出力情報中、“Collis”の項目に値が記録されていた場合、“FOUND”と表示されます。これらのファイル名一覧は sardo ディレクトリの“lan_colls”ファイルに記録されます。
7	LANinterfaceerrors	LAN インタフェース単位の netstat(1M)コマンド-I オプションの出力情報中、“lerrs”または“Oerrs”の項目に値が記録されていた場合、“FOUND”と表示されます。これらのエラーファイル名一覧は sardo ディレクトリの“lan_errors”ファイルに記録されます。

8	MaximumInputBps	ピーク時の秒あたり入力ビット数。平均を見ると低い値でも、瞬間的に高いビットレートを示すことがあります。このピーク時の入力秒あたりビット数。
9	MaximumOutputBps	ピーク時の秒あたり出力ビット数。平均を見ると低い値でも、瞬間的に高いビットレートを示すことがあります。このピーク時の出力秒あたりビット数。
10	MaximumTotalThroughPut	最大の入出力合計秒あたりビット数。MaximumInputBps と MaximumOutputBps は必ずしも同じタイミングかどうかは値だけでは判断できないため、minmax ファイルやグラフでタイミングを確認する必要があります。
11	OutPackets	LAN インタフェース全部の合計出力パケット数
12	OutPacketsPerSec	秒あたり平均出力パケット数
13	TCPaverageInPacketSize	TCP レベル入力平均パケット長
14	TCPaverageOutPacketSize	TCP レベル出力平均パケット長
15	networkerror	netstat(1M)コマンド-s オプションの出力情報の中で、チェックサムエラーやその他のエラーがカウントされていた場合、“FOUND”と表示されます。splistdir の netsinfo ディレクトリ下に“*Err*”の文字列を持つファイルが存在すると表示されます。これらのエラーファイル名一覧は sardo ディレクトリの“network_errors”ファイルに記録されます。
16	tcpInInorderBytes	TCP レベルの採取期間中入力バイト数 (tcpInInorderBytesEnd - tcpInInorderBytesBegin)
17	tcpInInorderBytesBegin	情報採取開始時の TCP 入力バイト数
18	tcpInInorderBytesEnd	情報採取終了時の TCP 入力バイト数
19	tcpOutDataBytes	TCP レベルの採取期間中の出力バイト数 (tcpOutDataBytesEnd - tcpOutDataBytesBegin)
20	tcpOutDataBytesBegin	情報採取開始時の TCP 出力バイト数
21	tcpOutDataBytesEnd	情報採取終了時の TCP 出力バイト数
22	udpInDatagrams	UDPの入力データグラム数【備考】
23	udpOutDatagrams	UDPの出力データグラム数【備考】

【備考】 NFS 環境で表示されることがあります。エラーではありません。

4-1-8. network.minmax ネットワーク性能値の最小・平均・最大

ネットワーク関連性能値の最小・最大値とその発生時刻、および平均・合計値を示すファイルです。

次に network.minmax ファイルの例を示します。

```

root@act011 # cat network.minmax
inbps_info          inBitPerSecond      555 22:47:45      6170 22:57:45      2293      64208
ipInDelivers         ipInDelivers         0 22:47:49      525 23:03:55      142      4124
ipInReceives         ipInReceives         0 22:47:49      501 23:03:55      118      3414
ipOutRequests        ipOutRequests        0 22:47:49      891 23:03:55      172      4997
netstat_e1000g0_info_inp000005 e1000g0_inpackets  10 22:55:45      500 23:03:45      119      3580
netstat_e1000g0_info_out000006 e1000g0_outpackets  4 22:55:45      887 23:03:45      171      5124
netstat_lo0_info_inpackets lo0_inpackets       0 22:47:45      86 22:48:45      38      1127
netstat_lo0_info_outpackets lo0_outpackets      0 22:47:45      86 22:48:45      38      1127
netstat_total_info_inpackets total_inpackets     39 23:12:45      532 23:03:45      157      4707
netstat_total_info_outpackets total_outpackets    34 22:55:45      919 23:03:45      208      6251
outbps_info          outBitPerSecond     511 22:47:45      25101 23:03:45      4296     120285
tcpActiveOpens       tcpActiveOpens      0 22:47:49      73 22:48:49      25      724
tcpAttemptFails      tcpAttemptFails     0 22:47:49      37 22:48:49      12      362
tcpCurrEstab         tcpCurrEstab        0 22:47:49      2 22:50:49      0      2
tcpInAckBytes        tcpInAckBytes       0 22:47:49      123984 23:03:55      12355     358294
tcpInAckSegs         tcpInAckSegs        0 22:47:49      467 23:03:55      105     3057
tcpInDupAck          tcpInDupAck         0 22:47:49      2 22:48:49      0      2
tcpInInorderBytes    tcpInInorderBytes   0 22:47:49      19020 22:57:52      6404     185725
tcpInInorderSegs     tcpInInorderSegs    0 22:47:49      108 22:58:53      30      882
tcpInSegs            tcpInSegs           0 22:47:49      521 23:03:55      142     4113
tcpOutAck            tcpOutAck           0 22:47:49      6 22:48:49      3      91
tcpOutAckDelayed     tcpOutAckDelayed    0 22:47:49      4 22:53:51      3      87
tcpOutControl        tcpOutControl       0 22:47:49      78 22:48:49      25      730
tcpOutDataBytes      tcpOutDataBytes     0 22:47:49      127600 23:03:55      17826     516965
tcpOutDataSegs       tcpOutDataSegs      0 22:47:49      896 23:03:55      182     5267
tcpOutRsts           tcpOutRsts          0 22:47:49      37 22:48:49      12      362
tcpOutSegs           tcpOutSegs          0 22:47:49      899 23:03:55      185     5358
tcpPassiveOpens      tcpPassiveOpens     0 22:47:49      1 22:48:49      0      2
tcpRttUpdate         tcpRttUpdate        0 22:47:49      467 23:03:55      105     3055
tcp_ipv4_info_BOUND  tcp_ipv4_info_BOUND 1 22:47:49      1 22:47:49      1      29
tcp_ipv4_info_ESTABLISHED ESTABLISHED        2 22:47:49      6 22:51:49      6     164
tcp_ipv4_info_IDLE   IDLE               2 22:47:49      29 23:00:54      8     228
tcp_ipv4_info_LISTEN LISTEN             36 22:47:49      39 22:51:49      39    1123
tcp_ipv4_info_TIME_WAIT TIME_WAIT          0 22:47:49      1 22:48:49      0      1
udpInDatagrams       udpInDatagrams      0 22:47:49      4 22:50:49      0     10
udpNoPorts           udpNoPorts          0 22:47:49      4 22:53:51      1     28
udpOutDatagrams      udpOutDatagrams     0 22:47:49      4 22:50:49      0     10
root@act011 #

```

図 4-1-8-1. network.minmax ファイル

※各列の意味は図 4-1-2-1 と同様です。

4-1-9. os.info OS 各種性能値のまとめ

OS 関連性能値をまとめたファイルです。一部、計算して値を出しています。

次に os.info ファイルの例を示します。

# cat os.info	
SystemCallPerSec	23036
ReadSystemCallPerSec	1174
WriteSystemCallPerSec	629
ReadCharacterPerSec	6179216
WriteCharacterPerSec	333425
ppgins(bytes)	0
ppgouts(bytes)	0
ReadBytePerCall	5263
WriteBytePerCall	530
ForkCountPerSec	183
ExecCountPerSec	183
PagingRateInReadWrite	0.00
SemaphoreCountPerSec	30
ReadWriteCallRatio	7.83
ForkExecRatio	0.79
IPCScallRatio	0.13
#	

バイトや回数等を示す値。

項目

図 4-1-9-1. os.info ファイル

次に、os.info ファイルの項目を示します。

表 4-1-9-1. os.info ファイルの項目

No	項目名	内容
1	ExecCountPerSec	平均の秒あたりシステムコール実行数
2	ForkCountPerSec	平均の秒あたりプロセスフォーク数
3	ForkExecRatio	SystemCallPerSec に占める Fork/ExecCountPerSec の割合
4	IPCScallRatio	SystemCallPerSec に占める Semaphore/MsgCountPerSec の割合
5	PagingRateInReadWrite	ppgins(bytes)と ppgouts(bytes)、が ReadCharacterPerSec と WriteCharacterPerSec に占める割合
6	ReadBytePerCall	1 回の読み取りシステムコールの平均バイト数(リードのブロック化係数)
7	ReadCharacterPerSec	平均の秒あたり読み取りバイト数
8	ReadSystemCallPerSec	平均の秒あたり読み取りシステムコール数
9	ReadWriteCallRatio	SystemCallPerSec に占める ReadSystemCallPerSec と WriteSystemCallPerSec の割合
10	SemaphoreCountPerSec	平均の秒あたりセマフォシステムコール数
11	SystemCallPerSec	平均の秒あたりシステムコール数
12	WriteBytePerCall	1 回の書き込みシステムコールの平均バイト数(ライトのブロック化係数)
13	WriteCharacterPerSec	平均の秒あたり書き込みバイト数
14	WriteSystemCallPerSec	平均の秒あたり書き込みシステムコール数
15	ppgins(bytes)	平均の秒あたりページインのバイト数
16	ppgouts(bytes)	平均の秒あたりページアウトのバイト数

4-1-10. os.minmax OS 性能値の最小・平均・最大

OS 関連性能値の最小・最大値とその発生時刻、および平均・合計値を示すファイルです。

次に os.minmax ファイルの例を示します。

```

root@act011 # cat os.minmax
sar_a_namei/s      1 22:47:04      9821 22:59:20      3850      1378242
sar_b_rcache      100 22:46:54      100 22:46:54      100      35800
sar_b_wcache      100 22:46:54      100 22:46:54      100      35800
sar_c_exec/s       0 22:46:59      424 23:04:31      183      65637
sar_c_fork/s       0 22:46:59      424 23:04:31      183      65597
sar_c_rchar/s     120 22:46:59     35987536 22:56:20     6179216     2212159424
sar_c_scall/s     291 22:47:44     59002 22:53:25     23036      8246917
sar_c_sread/s      4 22:46:59      7812 23:01:46      1174      420441
sar_c_swrit/s      8 22:46:59     5594 23:02:31      629      225313
sar_c_wchar/s     1626 22:47:44     3696488 22:50:40     333425     119366309
sar_k_lg_alloc     509745232 22:46:54 2187063952 23:16:31 1867504408 668566577936
sar_k_lg_mem      534261760 22:46:54 2233270272 23:16:41 1907696817 682955460608
sar_k_ovsz_alloc   45658112 22:46:54 67294848 22:56:45 51388519 18397089792
sar_k_sml_alloc   27314876 22:47:04 90021918 22:53:25 62435702 22351981423
sar_k_sml_mem     30068992 22:46:54 111882560 23:16:31 96349326 34493058752
sar_m_sema/s       0 22:46:54      1061 22:53:20       30      10882
sar_p_atrch/s      0 22:46:59      8298 23:05:01     4467     1599132
sar_p_pflt/s       0 22:46:59     28013 23:12:21     13390     4793639
sar_p_slock/s      0 22:46:59     2773 22:58:15       12      4300
sar_p_vflt/s       1 22:47:24     67528 23:12:21     36508     13063976
sar_q_runocc       0 22:46:59      100 22:52:30       48     17053
sar_q_runq-sz      0 22:46:59       40 22:59:20        6      2176
sar_r_freemem     100787 22:59:56     702530 22:46:54     202547     72511976
sar_r_freeswap    5996674 22:59:56    12986195 22:46:59    7326293     2622812718
sar_u_usr          0 22:46:59       85 22:50:40       22      7894
sar_u_sys          0 22:46:59       72 22:57:15       42     15035
sar_u_wio          0 22:46:54        0        0        0
sar_u_idle         0 22:52:35      100 22:46:59       36     12876
sar_u_sys          0 22:52:35      100 22:46:59       36     12876
sar_u_usr          0 22:46:59       72 22:57:15       42     15035
sar_u_usr          0 22:46:59       85 22:50:40       22      7894
sar_v_file-sz     499 22:46:54     1521 22:58:45      901     322630
sar_v_files        499 22:46:54     1521 22:58:45      901     322630
sar_v_inod-sz     70712 22:46:54     70712 22:46:54     70712     25314896
sar_v_proc-sz     16400 22:46:54     16400 22:46:54     16400     5871200
sar_v_procs        83 22:46:54       209 22:56:45      131     46851
sar_w_pswch/s     132 22:49:19    20862 22:53:25     1951     698348
vmstat_info_cd     0 22:46:50       220 23:02:40       26      9473
vmstat_info_cs     131 22:49:15     21737 22:53:20     1952     702554
vmstat_info_free   405892 22:59:50    2810116 22:46:50    808650     291114016
vmstat_info_id     0 22:52:30      100 22:46:55       36     12918
vmstat_info_in     455 22:49:15     4276 23:02:25     1456     524130
vmstat_info_mf     1 22:46:55     67884 23:04:25     36648     13192640
vmstat_info_r      0 22:46:50       40 22:59:15        5      1857
vmstat_info_re     0 22:46:55     8290 23:04:45     4485     1614486
vmstat_info_swap   3002816 23:00:00    6493096 22:46:55    3661163    1318018552
vmstat_info_sy     0 22:46:55       72 22:57:15       42     15156
vmstat_info_us     0 22:46:55       85 22:50:35       22     7935
vmstat_p_info_free 431588 23:00:45    2807828 22:47:45    808717     24261520
vmstat_p_info_mf   95 22:48:45     65780 23:16:45     36638     1099147
vmstat_p_info_re   11 22:48:45     7789 23:11:45     4484     134530
vmstat_p_info_swap 3065448 23:00:45    6490092 22:47:45    3661177    109835324
root@act011 #
  
```

図 4-1-10-1. os.minmax ファイル

※各列の意味は図 4-1-2-1 と同様です。

4-1-11. pacct.info プロセスアカウント情報のまとめ

プロセスアカウントティング情報をまとめたファイルです。

次に、pacct.info ファイルの例を示します(部分)。

command	userid	elaps	calcount	totalcpu	rate%	syscpu	sysmsec	usercpu	usermsec
oracle	oracle	10638.86	2841	557.42	30.97	237.23	83.5023	320.19	112.7033
sqlplus	oracle	7461.81	2576	133.42	7.41	79.03	30.6793	54.39	21.1141
mdb	root	5.89	6	5.52	0.31	1.31	218.3333	4.21	701.6667
oradoc.sh	oracle	10772.70	103	8.30	0.46	5.79	56.2136	2.51	24.3689
awk	oracle	887.38	66905	1.47	0.08	0.00	0.0000	1.47	0.0220
kstat	root	1.18	2	0.71	0.04	0.10	50.0000	0.61	305.0000
exp	oracle	13.49	1	0.74	0.04	0.25	250.0000	0.49	490.0000
gzip	oracle	0.38	1	0.38	0.02	0.01	10.0000	0.37	370.0000
vmstat	root	3601.21	35	0.44	0.02	0.20	5.7143	0.24	6.8571
sadc	root	1801.60	1	1.15	0.06	0.93	930.0000	0.22	220.0000
#pmap	root	41.61	271	38.79	2.16	38.66	142.6568	0.13	0.4797
gunzip	oracle	0.06	1	0.06	0.00	0.01	10.0000	0.05	50.0000
tar	oracle	0.48	2	0.43	0.02	0.33	195.0000	0.04	20.0000
#sardo.sh	root	5.80	1	0.13	0.01	0.10	100.0000	0.03	30.0000
sardo.sh	root	7306.36	10	0.23	0.01	0.21	21.0000	0.02	2.0000
iostat	root	1800.32	1	0.04	0.00	0.02	20.0000	0.02	20.0000
#netstat	root	3601.53	66	0.12	0.01	0.10	1.5152	0.02	0.3030
wc	oracle	0.43	41	0.01	0.00	0.00	0.0000	0.01	0.2439
mpstat	root	1800.32	1	0.02	0.00	0.01	10.0000	0.01	10.0000
ed	oracle	1.10	32	0.60	0.03	0.59	18.4375	0.01	0.3125
#lsnrctl	oracle	3.32	1	0.08	0.00	0.07	70.0000	0.01	10.0000
wc	root	0.04	4	0.00	0.00	0.00	0.0000	0.00	0.0000
uname	root	0.12	12	0.00	0.00	0.00	0.0000	0.00	0.0000
uname	oracle	8.85	311	0.00	0.00	0.00	0.0000	0.00	0.0000
swap	root	0.02	2	0.00	0.00	0.00	0.0000	0.00	0.0000
sort	root	0.04	1	0.00	0.00	0.00	0.0000	0.00	0.0000
sort	oracle	0.20	18	0.00	0.00	0.00	0.0000	0.00	0.0000
sleep	root	9028.40	131	0.00	0.00	0.00	0.0000	0.00	0.0000
sleep	oracle	600.00	1	0.00	0.00	0.00	0.0000	0.00	0.0000
sh	root	0.04	4	0.00	0.00	0.00	0.0000	0.00	0.0000
sh	oracle	15.73	10	0.00	0.00	0.00	0.0000	0.00	0.0000
sed	root	0.31	31	0.00	0.00	0.00	0.0000	0.00	0.0000
sed	oracle	1431.44	144499	0.00	0.00	0.00	0.0000	0.00	0.0000

図 4-1-11-1. pacct.info ファイル

pacct.info ファイルは、usercpu(ユーザ CPU)の時間でソートされます。

次に、各列の内容を示します。

表 4-1-11-1. pacct.info ファイルの内容

No	項目	内容
1	command	コマンド名
2	userid	コマンドを実行したユーザ ID
3	elaps	コマンドの壁時間: elapsed wall clocktime(秒)の合計
4	calcount	コマンド実行回数
5	totalcpu	コマンドの CPU 時間(秒)合計
6	rate%	測定区間内のコマンドの CPU 消費割合
7	syscpu	システム CPU 時間(秒)の合計
8	sysmsec	コマンドあたりシステム CPU 時間(ミリ秒)
9	usercpu	ユーザ CPU 時間(秒)の合計
10	usermsec	コマンドあたりユーザ CPU 時間(ミリ秒)

4-1-12. pmap.info プロセスのメモリ使用状況

pmap(1)コマンド-x オプションで出力した全プロセスのメモリ使用状況をまとめたファイルです。

次に、pmap.info ファイルの例を示します。

Microsoft Excel

ファイル(F) 編集(E) 表示(V) 挿入(I) 書式(O) ツール(T) データ(D) ウィンドウ(W) ヘルプ(H)

質問を入力してください

MS Pゴシック 11 B I U 背景色 塗りつぶし 罫線 枠線 文字色 背景色 塗りつぶし 罫線 枠線 文字色

A1 processname

pmaminfo.xls

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	processname	txt	rtxt	shs	rshs	shm	rshm	dsz	rdsz	hsz	rhsz	rsh	rshs	ssz	rssz	pid
2	/bin/sh	76	76	1264	968	0	0	44	32	132	128	0	0	8	8	6569
3	/bin/sh	76	76	1264	968	0	0	44	32	136	132	0	0	8	8	1510
4	/bin/sh	76	76	1264	968	0	0	44	32	136	132	0	0	8	8	9298
5	/bin/sh	76	76	1428	1056	0	0	60	56	316	316	0	0	8	8	26029
6	/lib/svc/bin/svc.config	344	344	0	0	0	0	0	0	7924	7884	0	0	16	16	9
7	/lib/svc/bin/svc.startd	196	196	0	0	0	0	256	256	8908	7948	0	0	8	8	7
8	/lib/svc/method/iscsid	8	8	0	0	0	0	68	68	8	8	0	0	4	4	233
9	/oracle/db/product/11.2.0/dbhome_1/bin/tnslsnr	768	624	0	0	0	0	0	0	1920	1872	0	0	4	4	1575
10	/sbin/init	36	36	2204	1736	0	0	60	48	24	24	0	0	8	8	1
11	/usr/X11/bin/Xorg	1768	1568	22284	17420	132	132	11956	11884	8504	8468	0	0	100	100	675
12	/usr/apache2/bin/httpd	280	276	13792	7420	0	0	1620	1348	1992	1548	0	0	20	20	718
13	/usr/apache2/bin/httpd	280	276	13792	7420	0	0	1620	1348	1992	1548	0	0	20	20	719
14	/usr/apache2/bin/httpd	280	276	13792	7420	0	0	1620	1348	1992	1548	0	0	20	20	720
15	/usr/apache2/bin/httpd	280	276	13792	7420	0	0	1620	1348	1992	1548	0	0	20	20	721
16	/usr/apache2/bin/httpd	280	276	13792	7420	0	0	1620	1348	1992	1548	0	0	20	20	722
17	/usr/apache2/bin/httpd	280	276	13792	7420	0	0	1624	1348	1988	1532	0	0	20	20	715
18	/usr/bin/postmaster	2276	1300	4036	3116	0	0	556	232	1172	124	576	492	24	24	724
19	/usr/bin/postmaster	2276	1300	4036	3116	0	0	556	232	444	260	576	492	24	24	725
20	/usr/bin/postmaster	2276	1300	4036	3116	408400	408400	596	236	140	116	576	492	24	24	682
21	/usr/bin/postmaster	2276	1300	4036	3116	408400	408400	596	236	332	304	576	492	24	24	723
22	/usr/dt/bin/dtlogin	112	112	5048	3272	0	0	336	236	160	48	20	20	20	20	670
23	/usr/dt/bin/dtlogin	112	112	5520	3676	0	0	424	280	296	128	328	280	28	28	717
24	/usr/java/bin/java	56	56	4	4	0	0	76	64	67832	67412	0	0	24	24	776
25	/usr/lib/autofs/automountd	124	120	0	0	0	0	4	4	220	136	0	0	4	4	502
26	/usr/lib/autofs/automountd	124	120	0	0	0	0	64	64	12	12	0	0	4	4	500
27	/usr/lib/crypto/kcfd	12	12	0	0	0	0	0	0	408	408	0	0	4	4	165
28	/usr/lib/dmi/dmispd	52	52	2352	1824	0	0	372	324	424	208	0	0	12	12	630
29	/usr/lib/dmi/snmpd	120	120	0	0	0	0	64	64	672	388	0	0	4	4	631
30	/usr/lib/fm/fmd/fmd	192	192	0	0	0	0	0	0	5016	4752	0	0	4	4	580
31	/usr/lib/inet/inetd	60	60	8	8	0	0	0	0	1556	1336	0	0	4	4	388
32	/usr/lib/nfs/lockd	24	24	1616	1256	0	0	280	228	208	92	0	0	4	4	351
33	/usr/lib/nfs/nfs4cbd	20	20	8	8	0	0	132	40	184	68	0	0	4	4	342
34	/usr/lib/nfs/nfsmapid	16	16	0	0	0	0	0	0	1032	24	0	0	4	4	341
35	/usr/lib/nfs/statd	36	36	1832	1468	0	0	380	292	416	220	144	116	8	8	340
36	/usr/lib/picl/picld	16	16	0	0	0	0	0	0	736	704	0	0	4	4	164
37	/usr/lib/power/powerd	20	20	0	0	0	0	0	0	244	244	0	0	4	4	134
38	/usr/lib/saf/sac	20	20	1792	1432	0	0	48	36	92	28	0	0	8	8	377
39	/usr/lib/saf/tytmon	56	56	1984	1616	0	0	180	112	84	84	0	0	16	16	383
40	/usr/lib/saf/tytmon	56	56	1984	1616	0	0	180	76	56	40	0	0	8	8	407
41	/usr/lib/snmp/snmpd	76	76	2108	1648	0	0	128	68	272	148	0	0	8	8	599
42	/usr/lib/ssh/sshd	308	308	3340	2752	0	0	80	68	60	60	0	0	12	12	539
43	/usr/lib/ssh/sshd	308	308	3920	3284	0	0	284	180	108	108	308	260	20	20	1531

コマンド

【備考】 空白で区切られたファイルのため、テキストで表示すると桁がずれて見づらいです。このため、エクセルファイルに取り込んで例示しています。

図 4-1-12-1. pmap.info ファイル

次に、pmap.info ファイルの内容を示します。

表 4-1-12-1. pmap.info ファイルの内容

No	項目	内容
1	processname	フルパスのプロセス名
2	txt	テキストセグメントサイズ
3	rtxt	常駐しているテキストセグメントサイズ
4	shs	共用ライブラリサイズ
5	rshs	常駐している共用ライブラリのサイズ
6	shm	ISM 型共有メモリサイズ
7	rshm	常駐している ISM 型共有メモリのサイズ
8	dsz	データセグメントサイズ
9	rdsz	常駐しているデータセグメントのサイズ
10	hsz	ヒープセグメントサイズ
11	rhsz	常駐しているヒープセグメントのサイズ
12	rsh	プライベート型共有メモリサイズ
13	rrsh	常駐しているプライベート型共有メモリのサイズ
14	ssz	スタックセグメントサイズ
15	rssz	常駐しているスタックセグメントのサイズ
16	pid	プロセス ID

【備考】 値は 1,024 バイト 1 単位で表示されます。

4-1-13. process.minmax プロセス CPU 消費の最小・平均・最大

プロセス CPU の最小・最大値とその発生時刻、および平均・合計値を示すファイルです。

ExecutedProcess は、サンプリングインターバル(測定間隔)内で起動されたプロセス数です。

次に、process.minmax ファイルの例を示します。

ExecutedProcess	ExecutedProcess	17 22:48:49	24420 23:03:55	10771	333906
Xorg_675_670_4	Xorg_675_670_4_cpu	3 22:48:49	4 23:05:55	3	105
dtgreet_789_717_1	dtgreet_789_717_1_cpu	0 22:48:49	1 22:57:52	1	20
fmd_580_1_1	fmd_580_1_1_cpu	1 22:48:49	1 22:48:49	1	31
fsflush_3_0_4	fsflush_3_0_4_cpu	2 22:48:49	4 23:11:55	3	85
inetd_388_1_1	inetd_388_1_1_cpu	1 22:48:49	1 22:48:49	1	31
init_1_0_47	init_1_0_47_cpu	0 22:48:49	47 23:16:55	12	358
java_776_1_25	java_776_1_25_cpu	23 22:48:49	25 23:07:55	24	752
nscd_157_1_3	nscd_157_1_3_cpu	0 22:48:49	3 23:00:54	2	62
ora_13586_1_2	ora_13586_1_2_cpu	0 22:48:45	2 23:00:54	0	2
ora_1614_1_1	ora_1614_1_1_cpu	0 22:48:45	1 23:05:55	0	12
ora_1616_1_3	ora_1616_1_3_cpu	0 22:48:45	3 23:10:55	1	45
ora_1628_1_1	ora_1628_1_1_cpu	0 22:48:45	1 23:05:55	0	12
ora_1632_1_2	ora_1632_1_2_cpu	0 22:48:45	2 23:07:55	1	31
ora_1634_1_4	ora_1634_1_4_cpu	0 22:48:45	4 23:02:55	3	85
ora_1636_1_1	ora_1636_1_1_cpu	0 22:48:45	1 23:04:55	0	13
ora_1638_1_1	ora_1638_1_1_cpu	0 22:48:45	1 22:51:49	1	26
ora_1642_1_1	ora_1642_1_1_cpu	0 22:48:45	1 22:50:49	1	27
ora_1644_1_1	ora_1644_1_1_cpu	0 22:48:45	1 23:08:55	0	9
ora_1672_1_1	ora_1672_1_1_cpu	0 22:48:45	1 22:54:52	1	23
ora_1674_1_3	ora_1674_1_3_cpu	0 22:48:45	3 23:10:55	1	45
ora_1676_1_2	ora_1676_1_2_cpu	0 22:48:45	2 22:50:49	0	2
ora_1690_1_98	ora_1690_1_98_cpu	0 22:48:45	98 22:53:50	9	268
ora_1694_1_4	ora_1694_1_4_cpu	0 22:48:45	4 22:50:49	0	4
ora_1696_1_8	ora_1696_1_8_cpu	0 22:48:45	8 22:50:49	0	8
ora_2057_1_6	ora_2057_1_6_cpu	0 22:48:45	6 22:55:52	1	17
ora_27047_1_1	ora_27047_1_1_cpu	0 22:48:45	1 23:12:55	0	1
ora_3126_1_1	ora_3126_1_1_cpu	0 22:48:45	1 23:10:55	0	1
ora_7325_1_2	ora_7325_1_2_cpu	0 22:48:45	2 22:57:52	0	3
oracleora11g_25889_25882_5	oracleora11g_25889_25882_5_cpu	0 22:48:45	5 23:02:55	0	5
postmaster_723_682_1	postmaster_723_682_1_cpu	1 22:48:49	1 22:46:49	1	31
sado_1471_1_1	sado_1471_1_1_cpu	0 22:48:45	1 23:00:54	1	16
sched_0_0_40	sched_0_0_40_cpu	40 22:48:49	40 22:46:49	40	1240
sh_26029_811_52	sh_26029_811_52_cpu	0 22:48:45	52 23:16:55	11	364
sshd_805_804_1	sshd_805_804_1_cpu	0 22:48:49	1 23:14:55	0	3
svc.configd_9_1_8	svc.configd_9_1_8_cpu	8 22:48:49	8 22:46:49	8	248
svc.startd_7_1_3	svc.startd_7_1_3_cpu	3 22:48:49	3 22:46:49	3	93

図 4-1-13-1. process.minmax ファイル

※各列の意味は図 4-1-2-1 と同様です。

4-1-14. sardo.log sardo 実行ログ

sardo.log は、sardo.sh、および sardo.edit の実行状況が記録されます。

次に、sardo.log ファイルを示します(部分)。ログの内容は付録 A に解説しています。ご覧ください。

```

root@act011 # cat sardo.log
100217224645 SETUP :INFO: spsetup was successfully completed.
100217224645 START :INFO: sampling interval : 60
100217224645 START :INFO: sampling count : 30
100217224645 START :INFO: current time : 82005 22:46:45
100217224645 START :INFO: start and end time : 82005 22:46:45 83805 23:16:45
100217224645 START :INFO: sampling mode : INTHE DAY
100217224645 START :INFO: sardo will be started with 60 sec interval.
100217224645 START :INFO: process accounting has been started.
100217224646 START :INFO: crash[adb] - was successfully completed.
100217224646 START :INFO: df -k was successfully completed.
100217224647 START :INFO: ipcs -a was successfully completed.
100217224647 START :INFO: kstat - was successfully completed.
100217224648 START :INFO: mdb -k was successfully completed.
100217224648 START :INFO: netstat -s was successfully completed.
100217224648 START :INFO: netstat -k was skipped.
100217224648 START :INFO: pmap -x was successfully completed.
100217224649 START :INFO: swap -s was successfully completed.
100217224649 START :INFO: vmstat -s was successfully completed.
100217224649 START :INFO: netstat - was successfully scheduled.
100217224649 START :INFO: netstat -an was successfully scheduled.
100217224649 START :INFO: vmstat -p was successfully scheduled.
100217224649 START :INFO: iostat -x -n was successfully scheduled.
100217224649 START :INFO: mpstat - was successfully scheduled.
100217224649 START :INFO: ps - was successfully scheduled.
100217224649 START :INFO: vmstat -i was successfully scheduled.
100217224649 START :INFO: netstat -s was successfully scheduled.
100217224649 START :INFO: netstat -an was successfully scheduled.
100217224649 START :INFO: sado command was successfully scheduled.
100217224649 START :INFO: vmstat command was successfully scheduled.
100217224650 START :INFO: wait for the end time [ 23:16:45 ] to finish.
100217224650 START :INFO: termproc pid [ 1510 ] was started successfully.
100217224650 START :INFO: remaining time is 1800 seconds.
100217225806 MEMSTA :INFO: an intermediate mdb ::memstat command was completed.
100217225831 PMAP :INFO: an intermediate pmap -x command was completed.
100217231651 START :INFO: wait for the end time [ 23:16:45 ] to finish.
100217231651 START :INFO: remaining time is 5 seconds.
100217231702 STOP :INFO: crash[adb] - was successfully completed.
100217231702 STOP :INFO: df -k was successfully completed.
100217231703 STOP :INFO: ipcs -a was successfully completed.
100217231703 STOP :INFO: kstat - was successfully completed.
100217231704 STOP :INFO: mdb -k was successfully completed.
100217231704 STOP :INFO: netstat -s was successfully completed.
100217231704 STOP :INFO: netstat -k was skipped.
100217231725 STOP :INFO: pmap -x was successfully completed.
100217231725 STOP :INFO: swap -s was successfully completed.
100217231725 STOP :INFO: vmstat -s was successfully completed.
100217231725 STOP :INFO: process accounting was stopped.
100217231726 STOP :INFO: list devinfo was successfully completed.
100217231726 STOP :INFO: pagesize - was successfully completed.
100217231726 STOP :INFO: du -k was successfully completed.
100217231727 STOP :INFO: uname -p was successfully completed.
100217231727 STOP :INFO: uname -r was successfully completed.
100217231727 STOP :INFO: hostid - was successfully completed.
100217231727 STOP :INFO: sysdef - was successfully completed.
100217231727 STOP :INFO: prtdiag -v was successfully completed.
100217231728 STOP :INFO: uptime - was successfully completed.
100217233121 EDIT :INFO: started as for act011 /var/tmp.
100217233122 EDIT :INFO: making os info sar_x files has been done.
100217233122 EDIT :INFO: making vmstat_info has been done.
100217233122 EDIT :INFO: making vmstat_p_info has been done.
100217233123 EDIT :INFO: making sar -d information has been done.
100217233123 EDIT :INFO: making iostat_xdn_* files has been done.

```

図 4-1-14-1. sardo.log ファイル

4-1-15. sardo.param sardo パラメタファイル

sardo.param ファイルは、サンプリングインターバル、採取開始時刻、および採取終了時刻がセットされます。

次に、sardo.param ファイルの例を示します。

```
# cat sardo.param
60 22:46:45 23:16:45
#
```

図 4-1-15-1. sardo.param ファイル

4-1-16. segment.info メモリセグメントの割当状況

メモリセグメントの割当状況がファイルに作成されます。

次に、segment.info ファイルの例を示します。

項目	常駐サイズ
# cat segment.info	
DataSize 76472 49504 65	
StackSize 1552 1552 100	
ISMsharedMemSize 246296 246296 100	
PrivateSharedMemSize 8 8 100	
SharedLibrarySize 24088 18392 76	
TextSize 416320 217952 52	
Total 764736 533704 70	
#	

図 4-1-16-1. segment.info ファイル

次に、segment.info ファイルの内容を示します。

表 4-1-16-1. setment.info ファイルの内容

No	項目	内容
1	DataSize	データセグメントの合計サイズ
2	StackSize	スタックセグメントの合計サイズ
3	ISMsharedMemSize	ISM 型共有メモリの合計サイズ
4	PrivateSharedMemSize	プライベート型共有メモリの合計サイズ
5	SharedLibrarySize	共用ライブラリの合計サイズ
6	TextSize	テキストセグメントの合計サイズ
7	Total	No1～6 までの合計、すなわちユーザプロセスの合計サイズ

4-2. begin 開始時の情報

次に、sardo 開始時に採取される、begin ディレクトリ下のファイルを示します。ファイル形式の解説は省略させていただきます。

表 4-2-1. begin ディレクトリのファイル

No	ファイル名	採取コマンドとオプション	内容
1	crash_info	mdb(1) -k ::kmastat、 physinstalled、physmem	Solaris 10 以上は mdb(1)を使用 Solaris 9 以下は adb(1)を使用
2	df_k_info	df(1M) -k	1ブロック1,024バイト単位のファイルシステム の空きと使用ブロック数
3	ipcs_a_info	ipcs(1) -a	プロセス間通信機能の使用状況
4	kstat_info	kstat(1M) netstat(1M) -k	Solaris 8 以上は kstat(1M) Solaris 8 以下は netstat(1M) -k
5	mdb_k_info	mdb(1) -k	カーネル情報
6	mdb_memstat_info	mdb(1) -k ::memstat	メモリ使用状況のサマリ
7	netstat_k_info	netstat(1M) -k	Solaris 8 のみ
8	netstat_s_info	netstat(1M) -s	TCP/IP、UDP のネットワーク統計情報
9	pmap_x_info	pmap(1) -x	全プロセスのプロセスマップ(メモリ)
10	swap_s_info	swap(1M) -s	スワップ情報
11	vmstat_s_info	vmstat(1M) -s	累積型のカーネル変数

4-3. data 採取中の情報

data ディレクトリは、採取中の性能情報が保存されます。

次に、data ディレクトリ下のファイルを示します。ファイル形式の解説は省略させていただきます。

表 4-3-1. data ディレクトリのファイル

No	ファイル名	採取コマンドとオプション	内容
1	iostat_xdn_info	iostat(1M) -xdn	入出力情報
2	mpstat_info	mpstat(1M)	CPU 情報
3	netstat_an_info	netstat(1M) -an	IP アドレスの名前変換はしません
4	netstat_e1000g0_info	netstat(1M) NIC	NIC を指定して入出力パケット数
5	netstat_lo0_info	netstat(1M) lo0	ループバックの入出力パケット数
6	netstat_s_info	netstat(1M) -s	詳細なネットワーク統計情報
7	netstat_total_info	-	全 NIC の合計入出力パケット数
8	ps_info	ps(1) -e -o user,pid,ppid, c,pri,nice,stime,etime, time,osz,rss,ty,ty,args	プロセス情報を左記のオプション で採取
9	vmstat_i_info	vmstat(1M) -i	インタラプトの合計とレート
10	vmstat_info	vmstat(1M) TI	TI は採取間隔(秒)
11	vmstat_p_info	vmstat(1M) -p	メモリ情報

4-4. end 終了時の情報

終了時の情報は、開始時の値と比較して変化を見るために採取しています。内容は begin ディレクトリ下のファイルと同じ情報が採取されます。4-2 節をご覧ください。

4-5. memstat ファンクション memstat の採取情報

sardo 性能情報採取の任意のタイミングで、“**sardo.sh memstat**”を実行することで作成されます。mdb(1) -k の ::memstat ファンクションの出力が、日付時刻スタンプ付きで保存されます。ファイル名は次のようになります。

memstat_100323122101_info

→ yymmddhhmmss の日付・時刻スタンプ。

【memstat_yymmddhhmmss_info ファイルの例】

Page Summary	Pages	MB	%Tot
-----	-----	-----	-----
Kernel	128051	500	12%
ZFS File Data	411641	1607	40%
Anon	301607	1178	29%
Exec and libs	27466	107	3%
Page cache	6857	26	1%
Free (cachelist)	25677	100	2%
Free (freelist)	128374	501	12%
Total	1029673	4022	

図 4-5-1. memstat_yymmddhhmmss_info ファイル

なお、begin と end ディレクトリ下の memstat 情報のファイル名は、“**mdb_memstat_info**”のファイル名で採取されます。

4-6. misc システム情報

misc ディレクトリには、各種のシステム情報やファイルが作成されます。各種計算や、sardo.edit 編集時に条件として入力する値を含んでいます。ファイル形式の解説は省略させていただきます。

表 4-6-1. misc ディレクトリのファイル

No	ファイル名	ファイル・コマンド	内容
1	device_info	/etc/path_to_inst	path_to_inst から作成したデバイス名一覧
2	du_k_info	du(1B) -k	sardo ディレクトリ下のサイズ
3	fsize_info	wc(1) -c	sardo ディレクトリ下の各ファイルのサイズ
4	hostid_info	hosted(1)	ホスト ID
5	hosts	/etc/hosts	hosts ファイル
6	pagesize_info	pagesize(1)	バイト単位のページサイズ
7	prtdiag_v_info	prtdiag(1M)	prtdiag コマンドの出力
8	sar	/usr/sbin/sar	編集に使用する sar コマンド(実行可能バイナリ)
9	sysdef_info	sysdef(1M)	システム定義情報
10	system	/etc/system	system ファイル
11	uname_p_info	uname(1) -p	システムのアーキテクチャ
12	uname_r_info	uname(1) -r	Solaris 10 のバージョン
13	uptime_info	uptime(1)	uptime 情報
14	vfstab	/etc/vfstab	vfstab ファイル

4-7. pacct プロセスアカウンティング情報(バイナリ)

pacct_info のファイル名で、accton(1M)コマンドによるプロセスアカウンティング情報がバイナリで採取されます。

この情報によって、性能情報採取中に実行→終了した全プロセスの情報が採取されます。

pacct_info のバイナリファイルは、sardo.edit で使用され、各種のプロセス情報が編集されます。

4-8. pmap ファンクション pmap の採取情報

全プロセスの pmap(1)コマンド、-x オプションによるセグメント情報が採取されます。ファイル名は次のようになります。

`pmap_x_100217225811_info`

→ yymmddhhmmss の日付・時刻スタンプ。

4-9. preserve 編集情報の保存ディレクトリ

sardo.edit の部分編集機能で使用する編集済みファイルが保存されています。

表 4-9-1. preserve ディレクトリの内容

No	ファイル・ディレクトリ	内容
1	diskinfo	ディスクの編集済みファイルが入れられるディレクトリ
2	netsinfo	ネットワークの編集済みファイルが入れられるディレクトリ
3	pacct_d.save	編集済みのプロセスアカウント情報のファイル
4	psinfo	ps(1)コマンド出力を編集したファイルが入れられるディレクトリ

表 4-9-2. preserve/diskinfo ディレクトリの内容

No	ファイル	内容
1	iostat_xdn_cxtxdx	iostat(1M)のディスク単位の編集情報
2	sar_d_cmdkx	sar(1)のディスク単位の編集情報

表 4-9-3. preserve/netsinfo ディレクトリの内容

No	ファイル ^{【備考 1】}	内容
1	inbps_info ^{【備考 2】}	入力ネットワークスループット(bps:ビット/秒)
2	ipDefaultTTL	IP デフォルト TTL
3	ipForwDatagrams	フォワードされたデータグラム数
4	ipForwProhibits	フォワードしなかったデータグラム数
5	ipForwarding	このエンティティがゲートウェイかどうかのフラグ
6	ipFragCreates	フラグメント作成数
7	ipFragFails	フラグメントされずに破棄されたデータグラム数
8	ipFragOKs	正常にフラグメントされたデータグラム数
9	ipInAddrErrors	IP アドレスが無効のために破棄したデータグラム数
10	ipInCksumErrs	チェックサムエラーの数
11	ipInDelivers	受信したデータグラム数
12	ipInDiscards	破棄したデータグラム数
13	ipInHdrErrors	ヘッダーにエラーがあったデータグラム数
14	ipInIPv6	IPv6 形式の入力データグラム数
15	ipInReceives	受信データグラムの総数

16	ipInUnknownProtos	不明なプロトコルの受信数
17	ipOutDiscards	送信で破棄されたデータグラム数
18	ipOutIPv6	IPv6 形式の出力データグラム数
19	ipOutNoRoutes	宛先のないルーティングデータグラム数
20	ipOutRequests	送信データグラムの総数
21	ipOutSwitchIPv6	IPv6 に移された送信データグラム数
22	ipReasmDuplicates	パケット再構成の重複数
23	ipReasmFails	パケット再構成のエラー数
24	ipReasmOKs	パケット再構成 OK の数
25	ipReasmPartDups	パケット再構成で、部分的に重複した数
26	ipReasmReqds	再構成要求数
27	ipReasmTimeout	パケット再構成タイムアウト数
28	ipRoutingDiscards	パケット再構成で破棄されたデータグラム数
29	ipsecInFailed	IPsec の入力エラー数
30	ipsecInSucceeded	IPsec の入力成功数
31	outbps_info ^{【備考 2】}	出力ネットワークスループット(bps:ビット/秒)
32	rawipInOverflows ^{【備考 3】}	ROWIP の入力オーバーフロー数
33	tcpActiveOpens	アクティブにオープン(こちらから)した数
34	tcpAttemptFails	コネクション確立中にリセット(RST)パケットを受信した数
35	tcpCurrEstab	現在の接続数
36	tcpEstabResets	接続確立がリセットされた数
37	tcpHalfOpenDrop	接続が何らかの原因で中断し破棄された数
38	tcpInAckBytes	ACK の受信バイト数
39	tcpInAckSegs	ACK の受信セグメント数
40	tcpInAckUnsent	-
41	tcpInClosed	-
42	tcpInDupAck	重複した ACK の受信数(セグメント順序が違う、または消失)
43	tcpInDupBytes	重複バイト受信数
44	tcpInDupSegs	重複セグメント受信数
45	tcpInErrs	TCP の入力エラー数
46	tcpHalfOpenDrop	接続が何らかの原因で中断し切断された数
47	tcpInInorderBytes	正しい順序の受信バイト数
48	tcpInInorderSegs	正しい順序の受信セグメント数
49	tcpInPartDupBytes	部分重複バイト受信数
50	tcpInPartDupSegs	部分重複セグメント受信数
51	tcpInPastWinBytes	-
52	tcpInPastWinSegs	-
53	tcpInSegs	受信したセグメント数
54	tcpInUnorderBytes	正しくない順序の受信バイト数
55	tcpInUnorderSegs	正しくない順序の受信セグメント数
56	tcpInWinProbe	ウィンドウ検査セグメント受信数
57	tcpInWinUpdate	ウィンドウ更新受信数
58	tcpListenDrop	接続確立時に接続完了キューから破棄された数
59	tcpListenDropQ0	接続確立時に接続中のキューから破棄された数

60	tcpMaxConn	最大の TCP 接続数
61	tcpOutAck	ACK の送信数
62	tcpOutAckDelayed	遅延 ACK の送信数
63	tcpOutControl	制御セグメントの送信数
64	tcpOutDataBytes	データの送信バイト数
65	tcpOutDataSegs	データの送信セグメント数
66	tcpOutFastRetrans	-
67	tcpOutRsts	リセット(RST)の送信数
68	tcpOutSackRetrans	-
69	tcpOutSegs	送信セグメントの数総数
70	tcpOutUrg	緊急送信数
71	tcpOutWinProbe	ウィンドウ検査セグメント送信数
72	tcpOutWinUpdate	ウィンドウ更新送信数
73	tcpPassiveOpens	パッシブにオープン(相手から)された数
74	tcpRetransBytes	再送バイト数
75	tcpRetransSegs	再送セグメント数
76	tcpRtoAlgorithm ^{【備考 3】}	再送パケットタイムアウトのアルゴリズム。1:その他、2:定数 RTO、3:MIL-ST D-1778、4:VanJacobson アルゴリズム
77	tcpRtoMax	ミリ秒の再送タイムアウト最大値
78	tcpRtoMin	ミリ秒の再送タイムアウト最小値
79	tcpRttNoUpdate	往復時間非更新数
80	tcpRttUpdate	往復時間更新数
81	tcpTimKeepalive	キープアライブ中の数
82	tcpTimKeepaliveDrop	キープアライブ中、破棄された数
83	tcpTimKeepaliveProbe	キープアライブ検査セグメント数
84	tcpTimRetrans	再送数
85	tcpTimRetransDrop	再送破棄数
86	udpInCksumErrs	入力チェックサムエラー数
87	udpInDatagrams	入力データグラム数
88	udpInOverflows	入力オーバーフロー数
89	udpNoPorts	待ち受けていなかったため破棄したデータグラム数
90	udpOutDatagrams	送信したデータグラム数

【備考 1】 netsinfo ディレクトリ下は、netstat(1M)コマンド-s オプションの出力項目をファイル名にしたものです。値が“0”、または無しの場合は作成されません。

【備考 2】 No.1 の inbps_info と、No.31 の outbps_info は計算して求めた値です。

【備考 3】 データ採取されていますが、編集しません。また、値が“0”(ゼロ)や、無しの項目も編集しません。

4-10. sadc sadc(1M)出力ファイル(バイナリ)

sar(1)情報をテキストで採取すると、ディスク容量がかさむとともに、データ採取のためのディスクオーバーヘッドがかかります。これを防ぐため、sar(1)情報は sadc(1M)コマンドによってバイナリファイルとして採取します。この出力ファイルが保存されます。ファイル名は“sadc_info”です。

4-11. speditdir 性能要素単位の編集結果

sardo.edit が編集した結果のファイルが入られるディレクトリです。グラファーはこれらのディレクトリを入力にしてグラフを描画します。

表 4-11-1. speditdir の性能要素ディレクトリ

No	ディレクトリ	内容
1	cpu	CPU に関連する各種性能情報
2	disk	ディスクに関連する各種性能情報
3	network	ネットワークに関連する各種性能情報
4	os	OS 関連の各種性能情報
5	process	プロセスアカウンティング、ps(1)の編集情報

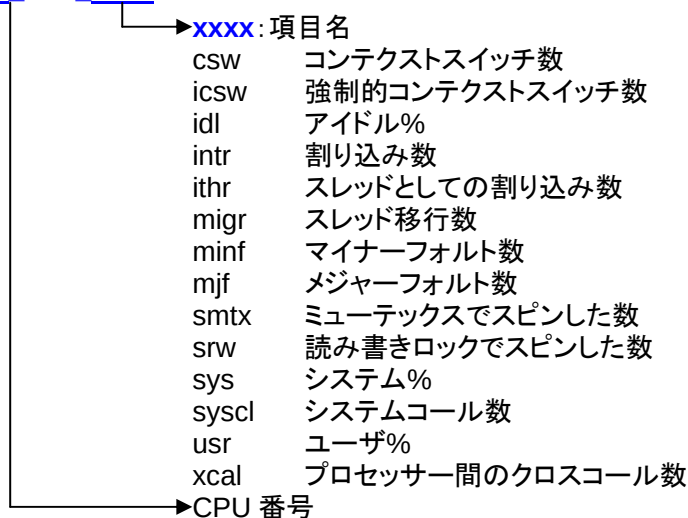
4-11-1. cpu 性能要素ディレクトリ

mpstat(1M)コマンドの出力を CPU 単位に分割し、項目名を持つファイルにして作成します。また、vmstat(1M)コマンド-i オプションの出力を、デバイス毎の割り込みカウントとして作成します。

ファイル名は次のようになります。

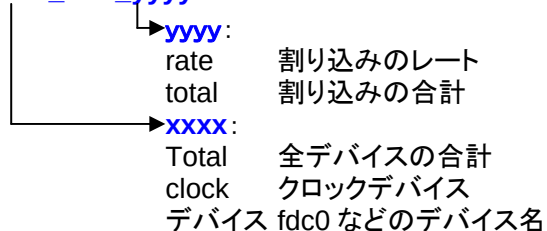
mpstat(1M)の出力

mpstat_n_info_xxxx



vmstat(1M) -i オプションの出力

vmstat_i_xxxx_info_yyyy



4-11-2. disk 性能要素ディレクトリ

iostat(1M)コマンド-xdn オプションの出力をディスク単位に分割し、項目名を持つファイルにして作成します。また、sar(1)コマンド-d オプションの出力を、デバイス単位に分割し、sar コマンドが表示する項目名を持つファイルにして作成します。

ファイル名は次のようになります。

iostat(1M)-xdn の出力

iostat_xdn_xxxx_yyyy

→ **yyyy**: 項目名

%b ディスクビジー%

%w ディスク待ち%

actv トランザクションアクティビティの平均値

asvc_t トランザクションの平均サービス時間(ミリ秒)

kr_s kr/s: 読み取りのキロバイト/秒

kw_s kw/s: 書き込みのキロバイト/秒

r_s r/s: 読み取り回数/秒

svc_t トランザクションの平均応答時間(ミリ秒)

w_s w/s: 書き込み回数/秒

wait トランザクション待ちの平均値(キューレングス)

wsvc_t 待ちキュー中の平均サービス時間(ミリ秒)

→ **xxxx**:

デバイス名

sar(1)-d の出力

sar_d_xxxx_yyyy

→ **yyyy**: 項目名

%busy ディスクビジー%

avque 平均キュー

avserv 平均サービス時間(ミリ秒)

await 平均待ち時間(ミリ秒)

blks_s blks/s: 秒あたりの転送ブロック数

read_s read/s: 秒あたりの読み取り回数

r+w_s r+w/s: 秒あたりの読み取り・書き込み回数

write_s write/s: 秒あたりの書き込み回数

→ **xxxx**:

デバイス名

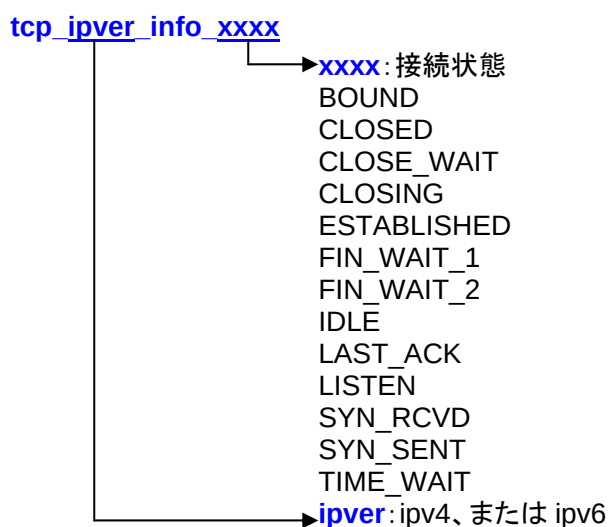
4-11-3. network 性能要素ディレクトリ

netstat(1M)コマンド-s オプション、netstat(1M)コマンド-an オプション、および netstat(1M)コマンドのネットワークインタフェース別のパケット入出力数を編集した情報が作成されます。また、計算で求めた入出力スループット(ビット/秒)のファイルが作成されます。

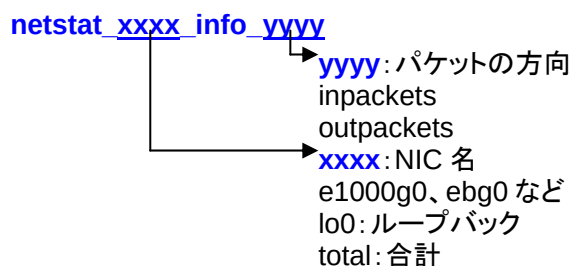
netstat(1M)-s の出力

前述、表 4-9-3 に示すファイル名で作成されます。入力スループットは“**inbps_info**”、出力スループットは“**outbps_info**”のファイル名です。

netstat(1M)-an の出力



netstat(1M) NIC 名の出力



4-11-4. os 性能要素ディレクトリ

sar(1)コマンドの-d オプションを除くオプション、vmstat(1M)コマンドの時間間隔指定、およびvmstat(1M)コマンド-p オプションの出力を編集した情報が作成されます。項目名の“/”(スラッシュ)は“_”(下線)に変換されます。

sar(1)の出力

sar x yyyy

→ **yyyy**: 項目

→ **x**: オプション

x **yyyy**

a	iget_s:iget/s namei_s:namei/s dirbk_s:dirbk/s
b	bread_s:bread/s lread_s:lread/s %rcache bwrit_s:bwrit/s lwrit_s:lwrit/s %wcache pread_s:pread/s pwrit_s:pwrit/s
c	scall_s:scall/s sread_s:sread/s swrit_s:swrit/s fork_s:fork/s exec_s:exec/s rchar_s:rchar/s wchar_s:wchar/s
k	sml_mem_alloc:sml_mem の alloc sml_mem_fail:sml_mem の fail lg_mem_alloc:lg_mem の alloc lg_mem_fail:lg_mem の fail ovsz_alloc ovsz_fail:ovsz_alloc の fail
m	msg_s:msg/s sema_s:sema/s
p	atch_s:atch/s pgin_s:pgin/s ppgin_s:ppgin/s pflt_s:pflt/s vflt_s:vflt/s slock_s:slock/s
q	runq-sz %runocc swpq-sz %swpocc
r	freemem freeswap
u	u:%usr、%sys、%wio、%idle の 4 項目全部 %usr

```
%sys
%wio
%idle
v   proc-sz
    procv: proc-sz の ov
    inod-sz
    inodov: inod-sz の ov
    file-sz
    fileov: file-sz の ov
    lock-sz
w   swpin_s: swpin/s
    bswin_s: bswin/s
    swpot_s: swpot/s
    bswot_s: bswot/s
    pswch_s: pswch/s
```

vmstat(1M)時間間隔指定の出力

[vmstat_info_xxxx](#)

→ **xxxx**: 項目

```
r
b
w
swap
free
re
mf
pi
po
fr
de
sr
cd
f0
s0
s1
in
sy
cs
us
sy
id
```

vmstat(1M)-p の出力

[vmstat_p_info_xxxx](#)

→ **xxxx**: 項目

swap
free
re
mf
fr
de
sr
epi
epo
epf
api
apo
apf
fpi
fpo
fpf

4-11-5. process 性能要素ディレクトリ

プロセスアカウント、および ps(1)コマンドの出力情報を基に、性能情報採取期間中に起動・終了したプロセス個々の情報を作成します。

また、採取のサンプリングインターバル内に起動されたプロセス数を ExecutedProcess ファイルに作成します。

コマンド情報

[xxxxx](#) [PID](#) [PPID](#) [sec](#)

→ **sec**: プロセスの実行時間(秒)

→ **PPID**: 親プロセスの PID

→ **PID**: 自プロセスの PID

→ **xxxxx**: コマンド名(ps 情報の CMD で表示される名前)

実行プロセス数

[ExecutedProcess](#)

4-12. splistdir 性能要素単位の各コマンド出力

splistdir 直下のディレクトリ内容は次のとおりです。speditdir ディレクトリと同様の構造になっています。各ディレクトリ下のファイルは、各種コマンドの出力と同じ形式です。

また、netstat(1M)や vmstat(1M)のような時間の表示されないコマンド出力に時間スタンプが挿入されます。

表 4-12-1. splistdir ディレクトリの内容

No	ディレクトリ名	内容
1	cpu	mpstat(1M)と vmstat(1M)コマンド-i オプションの出力に時間スタンプを付けたファイル。
2	disk	sar(1M)コマンド-d オプションの出力、及び iostat(1M)コマンド-xdn オプションの出力に時間スタンプを付けたファイル。
3	network	netstat(1M)コマンド-s オプションの出力を分解し時間スタンプを付けた項目(タグ)名のファイルをこの下の netsinfo ディレクトリに作成。 netstat(1M)コマンド-an オプションの出力からコネクション確立やリッスン数等の状態を示す情報を抜き出し、それに時間スタンプを付けたファイルです。現在のところ TCP/IP は Version4 の情報のみ対象。 netstat(1M)コマンド-l オプションの出力を編集した LAN インタフェース毎の時間スタンプ付きファイル。
4	os	sar(1M)コマンド-d オプションを除く他のオプション出力、及び、vmstat(1M)コマンド(オプション無し)、vmstat(1M)コマンド-p オプションの出力に時間スタンプを付けたファイル。
5	process	pacctinfo、pmapinfo、及び psinfo ディレクトリが作られます。

splistdir/network ディレクトリは、netstat(1M)コマンド-s オプションの出力のみ“netsinfo”ディレクトリの下に作成されます。また、ソケット情報のファイル“socket”が作成されます。次に、splistdir/network ディレクトリの内容を示します。

表 4-12-2. splistdir/network ディレクトリの内容

No	ディレクトリ・ファイル名	内容
1	netsinfo	netstat(1M)コマンド-s オプションの出力を分解し時間スタンプを付けた項目(タグ)名そのもののファイル。
2	netstat_NIC_info	netstat(1M)コマンド-l オプションの出力を total と qfe0 や eri0、lo0 等の LAN インタフェース毎に分解し、時間スタンプを付けたファイル。
3	socket	netstat(1M)コマンド-an オプションで表示される“Active UNIX domain sockets”のタイプ数を採取時間毎に示すファイル。
4	tcp_ipv4	No.5 の tcp_ipv4_info の元になるファイルで、netstat(1M)コマンド-an オプションの“TCP: v4”に現れる接続状態毎の数を採取時間毎に示すファイル。
5	tcp_ipv4_info	No.4 の tcp_ipv4 ファイルの“LISTEN”、“ESTABLISHED”等の項目毎に値を一覧にしたファイル。
6	tcp_ipv6	No.4 と類似の“TCP: v6”に区分される接続状態毎の値を採取時間毎に示すファイル。
7	udp_ipv4	No.4と類似の“UDP: v4”に区分されるUDPの状態毎の値を採取時間毎に示すファイル。
8	udp_ipv6	No.7と同様の“UDP: v6”に区分されるUDPの状態毎の値を採取時間毎に示すファイル。

splistdir/process ディレクトリは、プロセスアカウンティング、プロセスマップ、および ps(1)の情報が作成されます。次に splistdir/process ディレクトリの内容を示します。

表 4-12-3. splistdir/process ディレクトリの内容

No	ディレクトリ名	内容
1	pacctinfo	ps(1)コマンドの時間スタンプをベースに acctcom(1)コマンドを用いて HHMMSS の時間情報を得、プロセスアカウント情報をサンプリングインターバル毎のファイルにして“a_HHMMSS_HHMMSS”ディレクトリに作成。“a_HHMMSS_HHMMSS”のプロセスアカウントデータから、エラプス時間、メモリ量、システムとユーザ CPU 使用率別の順位一覧を“d_HHMMSS_HHMMSS”ディレクトリに作成。ディレクトリ名の最初の HHMMSS はプロセスの開始時刻、後ろの HHMMSS はプロセスの終了時刻で、この間に起動・終了したプロセスの情報を作成(トップ 24)。
2	pmapinfo	全プロセスのメモリマップ。sardo.sh の開始時に採取されたプロセスメモリマップは“pmap_begin”ディレクトリに、終了時に採取されたプロセスメモリマップは“pmap_end”ディレクトリに作成。sardo.sh の pmap ファンクションで採取した任意の時点のプロセスメモリマップは日付時刻を持つ、“pmap_x_yymmddHHMMSS”のディレクトリに作成。
3	psinfo	ps(1)データより抽出した情報を“p_ProgramName_PID_PPID”のファイル名で作成。ProgramName は ps(1)出力情報の“CMD”で表示されるコマンドのベース名(ディレクトリパスを除いたプログラム名)。PID はそのプロセス ID、PPID は親プロセスの ID。

次に、splistdir/process/pacctinfo/d_HHMMSS_HHMMSS ディレクトリ下のファイルを示します。

表 4-12-4. splistdir/process/pacctinfo/d_HHMMSS_HHMMSS ディレクトリの内容

No	ディレクトリ名	内容
1	elaps	その区間に終了した上位 24 プロセスのエラプス時間順のリスト
2	memsize	その区間に終了した上位 24 プロセスのメモリサイズ順のリスト
3	syscpu	その区間に終了した上位 24 プロセスのシステム CPU 消費時間順リスト
4	usercpu	その区間の終了した上位 24 プロセスのユーザ CPU 消費時間順リスト

次に、splistdir/process/pmapinfo/pmap_[begin|end]ディレクトリの内容を示します。

表 4-12-5. splistdir/process/pmapinfo/pmap_[begin|end]ディレクトリの内容

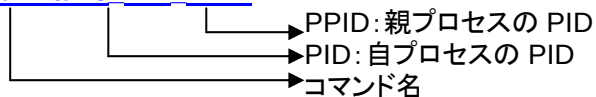
No	ディレクトリ名	内容
1	pmap_info	sardo ディレクトリ下の pmap_info ファイルと同じ内容。begin と end のどちらかメモリ消費の大きい方が sardo ディレクトリ下にコピーされます。
2	segment_info	プロセスのデータ、共有メモリ、共用ライブラリ、テキスト、スタックのセグメント情報。sardo ディレクトリ下の segment.info ファイルと同じもの。begin か end のどちらかメモリ消費の大きい方が sardo ディレクトリにコピーされます。

【備考】 このファイルで表示される値の単位は、1,024 バイト単位です。

次に、splistdir/process/psinfo ディレクトリ下に作成されるプロセス情報のファイル名を示します。

p_ProgramName_PID_PPID

p_ProgramName_PID_PPID



4-13. sptrendir 傾向データ

sptrendir は次のディレクトリが作成されます。sardo.edit が作成するファイルで、性能値のトレンド(傾向)をヒストグラムで読み取ることができます。ファイルの詳細は第 6 章 sardo の応用 性能値の評価をご覧ください。

表 4-13-1. sptrendir ディレクトリの内容

No	ディレクトリ名	内容
1	network	ネットワークパケットのトレンド(傾向)ファイルがヒストグラムで作られます。 ファイルは次の 2 個 netstat_total_info_inpackets netstat_total_info_outpackets
2	os	OS 関連性能情報のトレンド(傾向)ファイルがヒストグラムで作られます。 ファイル名の形式 sar x TAG TAG: %usr や scall_s など x: a、b、c、g、k、m、p、q、r、u、v、及び w ※ファイル名として使用出来ない"/"(スラッシュ)や":"(コロン)は"_"(アンダーバー)に変換されます。

第 5 章 sardo の計算式

5-1. ネットワークスループットの計算

ネットワークスループットの計算は以下のように行っています。

- ① 秒単位のパケット数を求めます。

データは netstat(1M)コマンド-I オプションの inpackets 及び outpackets を用います。

$$\text{パケット数} \div \text{サンプリングインターバル} = \text{①}$$

- ② バイト数を秒単位にします。

データは netstat(1M)コマンド-s オプションの tcpInInorderBytes と tcpOutDataBytes を用います。

$$\text{バイト数} \div \text{サンプリングインターバル} = \text{②}$$

- ③ 平均パケット長を求めます。

$$\text{②} \div \text{①} = \text{③}$$

- ④ ③の平均パケット長に 66 バイト(イーサネットヘッダーの 64 バイト、及びトレイラの 2 バイト)を加えます。

$$\text{③} + 66 = \text{④}$$

- ⑤ 平均パケット数を掛けて秒あたりのバイト数を求めます。

$$\text{④} \times \text{①} = \text{⑤}$$

- ⑥ ⑤のバイト数を 8 ビットに変換します。

$$\text{⑤} \times 8 \text{ ビット} = \text{スループット(bps)の値}$$

5-2. baseline レコードの計算

次に、sardo.edit の作成する傾向ファイルのレコード形式を示します。

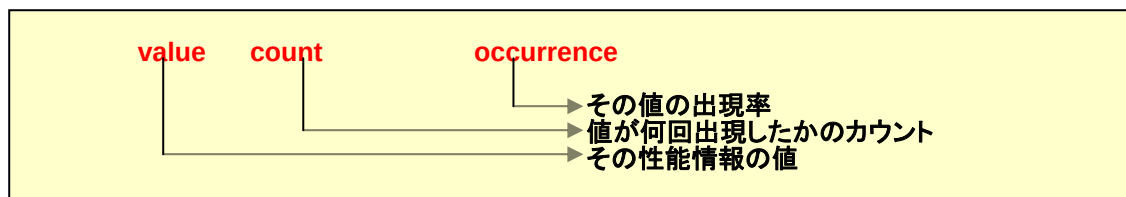


図 5-2-1. 傾向ファイルのレコード形式

次に、baseline レコードの形式を示します。

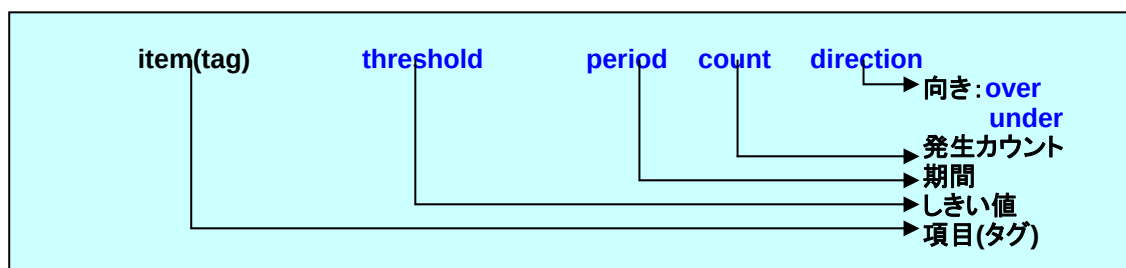


図 5-2-2. baseline レコードの形式

sardo.edit は、次表に示す計算式で baseline レコードを生成しています。傾向ファイル中のサンプルの値は、over Direction の場合上位 5 レコード、under Direction の場合下位 5 レコードを用います。なお、具体例は第 6 章 sardo の応用 で解説します。

表 5-2-1. baseline レコードの計算式

baseline レコードの項目	over Direction(～を超える)	under Direction(～を下回る)
threshold(しきい値)	$\text{value} * (1 + \text{occurrence} / 100)$ %の場合、100 を上回らない。	$\text{value} * (1 - \text{occurrence} / 100)$ 現在のところ下限の検査は行っていない。
period(評価期間)	$\text{count} * \text{interval}$ 但し、interval を下回らない。また採取期間を上回らない。	
count(発生回数)	$\text{count} * (1 - \text{occurrence} / 100)$ 但し、1 を下回らない。	

第 6 章 sardo の応用

6-1. 性能情報のファイルサイズを見積る

sardo 採取情報のファイルサイズは対象のシステム構成やラッシュ度合いに大きく左右されますので次の手順でドライラン(試運転)を行い、実測データを元にサイズをお見積り下さい。

なお、この見積り情報は sardo.sh の採取情報を対象にしています。sardo.edit はこれより多くの情報を編集・出力します。

6-1-1. 見積り用ファイル estimate の作成

- (1) sardo ディレクトリを任意のファイルシステムに設定する場合、LOGDIR 環境変数にそのファイルシステムを設定します。デフォルトは“/var/tmp”です。
- (2) 次のタイプインで 5 分間のデータを採取します。

```
# ./sardo.sh auto 5 now 5min <CR>
```

- (3) 数分後、LOGDIR 環境変数で指定したディレクトリに次の圧縮 tar アーカイブファイルが出来上がっています。hostname はそのシステムのホスト名です。yymmddHHMMSS は日付時刻です。

```
# cd /var/tmp <CR>
# ls <CR>
sarstats_act011_001_100328005130.tar.gz
#
```

- (4) アーカイブファイルが確認し、次のタイプインで見積り情報を作成します。

```
# ./sardo.sh estimate <CR>
```

- (5) この結果、アーカイブファイルが解凍・tar 展開され、LOGDIR 環境変数で示されるディレクトリ下に sarstats_hostname_001 のディレクトリが作成されます。これが採取情報のファイルサイズを見積るための基礎データになります。この基礎データを基に、**estimate** ファイルが作成されます。目的のサンプリングインターバル、開始時刻、終了時刻や、採取数に応じたファイルサイズをお見積りください。なお、そのファイルシステムが小さい場合は LOGDIR 環境変数で他のファイルシステムに移して再度、確認してください。

6-1-2. estimate ファイルの見方

estimate ファイルは大きく 3 つの部分に分かれています。図 6-1-2-1 に最初の部分(その 1)の例を示します。ここには環境的な情報が表示されます。値はブロック単位で、1 ブロックは 1,024 バイトです。試験的に採取した sardo データの各種ブロック数が表示されます。

注目するポイントは、

- ① packedarchivesize で示される圧縮 tar アーカイブファイルのブロック数
- ② sardodirectorysize で示される sardo の情報採取終了時のブロック数
- ③ tarfilesize で示される tar アーカイブファイルのブロック数です。

sardo.sh は性能情報を採取し、終了時に tar(1)コマンドでアーカイブしたファイルを gzip(1)コマンドで圧縮します。この時、元の採取データ(②)、tar(1)でアーカイブしたファイル(③)、この tar アーカイブを圧縮したファイル(①)の 3 つを合計したサイズが、瞬間的に最も大きな領域サイズになります。

図 6-1-2-1 の例では 5 秒のサンプリングインターバルで 5 分間採取した場合のブロック数を示しています。

```
hostname          : act061
hostid            : 8310060f
date              : 2003.07.28

unitvalue         : 1024 bytes (kilo-bytes) 1 block unit

archivefilecount  : 1
availblock        : 3004466
datadirectorysize : 1011
logdirectory      : /var/tmp
onetimetotalsize   : 687
paccountsize      : 74
packedarchivesize : 496 ←①
pmapdirsize       : 449
pmapfilecount     : 2
sadcdirectorysize : 257
sampling interval : 5
samplingtime      : 300
samplingcount     : 60
sardodirectorysize : 2483 ←②
tarfilesize       : 2408 ←③
time info         : 80983 22:29:43 81283 22:34:43

dataunit          : 17 X 60
onetimeunit       : 687 X 1
pacctunit         : 74 X n
pmapunit          : 224 X n
pacctunit         : 74 X n
pmapunit          : 224 X n
sadcunit          : 4 X 60
```

図 6-1-2-1. estimate ファイルの内容(その 1)

2 つ目の部分は横軸にサンプリングインターバル、縦軸に採取時間を示すもので、具体的な推定ブロックサイズを表にしています。図 6-1-2-2 に(その 2)の例を示します。

例えば 60 秒(1 分)のサンプリングインターバルで 12 時間採取する場合の推定ブロックサイズは横軸 60sec、縦軸 43200sec 720min 12hour の交わる点で 270574 ブロック=約 300 メガバイトであることがわかります。実際の情報採取の場合は、サンプリングインターバルと採取時間を元にこの交点のブロック数を求めることで簡単に見積ることが出来ます。

なお、estimateファイルは試験的に採取した実ファイルブロックから計算して作成されます。システムのラッシュ度合いによって増減の可能性がありますので、交点で求められた値にラッシュの度合いを加味することをお勧めいたします【備考】。

sampling time			sampling interval (sec)								
sec	min	hour	3600	1800	900	600	60	30	10	5	
300	5	0	3196	3196	3196	3196	3384	3571	4322	5447	
600	10	0	4877	4877	4915	4915	5252	5627	7128	9378	
1800	30	0	11601	11639	11676	11714	12726	13851	18352	25102	
3600	60	1	21724	21762	21837	21912	23937	26187	35188	48688	
7200	120	2	41933	42008	42158	42308	46359	50859	68860	95861	
10800	180	3	62142	62255	62480	62705	68780	75531	102532	143034	
14400	240	4	82351	82501	82801	83101	91202	100202	136204	190206	
18000	300	5	102560	102748	103123	103498	113623	124874	169876	237379	
21600	360	6	122769	122994	123444	123894	136045	149546	203548	284552	
25200	420	7	142978	143241	143766	144291	158467	174217	237220	331724	
28800	480	8	163187	163487	164087	164687	180888	198889	270892	378897	
32400	540	9	183396	183734	184409	185084	203310	223561	304564	426070	
36000	600	10	203605	203980	204730	205480	225731	248232	338236	473242	
39600	660	11	223814	224227	225052	225877	248153	272904	371908	520415	
43200	720	12	244023	244473	245373	246273	270574	297576	405580	567588	
46800	780	13	264232	264720	265695	266670	292996	322247	439253	614760	
50400	840	14	284441	284966	286016	287066	315418	346919	472925	661933	
54000	900	15	304650	305213	306338	307463	337839	371591	506597	709106	
57600	960	16	324859	325459	326659	327859	360261	396262	540269	756278	
61200	1020	17	345068	345706	346981	348256	382682	420934	573941	803451	
64800	1080	18	365277	365952	367302	368652	405104	445606	607613	850624	
68400	1140	19	385486	386199	387624	389049	427526	470277	641285	897796	
72000	1200	20	405695	406445	407945	409445	449947	494949	674957	944969	
75600	1260	21	425904	426692	428267	429842	472369	519621	708629	992142	
79200	1320	22	446113	446938	448588	450238	494790	544292	742301	1039314	
82800	1380	23	466322	467185	468910	470635	517212	568964	775973	1086487	
86400	1440	24	486531	487431	489231	491031	539633	593636	809645	1133660	

図 6-1-2-2. estimate ファイルの内容(その 2)

【備考】 サイズに強い影響を及ぼす要素は、プロセスの発生度合いによるプロセスアカウンティングファイル、ps データファイル、及びプロセスマップファイルと、ネットワーク接続数や転送量によるネットワーク関連データファイル、及びディスク構成による sadc バイナリファイルです。

3 つ目の部分は 5 秒のサンプリングインターバルで 24 時間採取した場合、sardo の採取データがそのファイルシステムにいくつ採取・保存できるのかを示す情報です。言い換えれば最大の性能情報を採取する条件で、採取可能なファイル数と、保存可能なファイル数を知ることができます。

図 6-1-2-3 に(その 3)の例を示します。この最大ブロック数の合計は図 6-1-2-2(その 2)の右下に表示されるブロック数(1133660)に近似していることがわかります。

$$\begin{aligned}\text{推定最大ブロック数} &\doteq \text{④のブロック数} \times 2 + \text{⑤の圧縮 tar アーカイブファイルブロック数} \\ &= 513903 \times 2 + 105853 = 1133659 \text{ ブロック} \doteq 1.16 \text{ ギガバイト}\end{aligned}$$

この例の場合、そのファイルシステムに保存可能な圧縮 tar アーカイブファイルは 17 個、情報採取は 2 日分可能であることを示しています。

```
the largest blocks :  
datadirectory      : 293760  
onetimedirectories : 687  
pacctdirectory     : 21312  
pmapdirectory      : 129024  
sadcdirectory      : 69120  
sardodirectory    : 513903 ←④  
  
packedarchivefile : 105853 ←⑤  
  
INFO: /var/tmp can hold [ 17 ] packed tar archive files,  
INFO: during the sardo is gathering the latest one of data.  
INFO: sardo can gather maximum for [ 2 ] days data in the directory.  
  
END REPORT
```

図 6-1-2-3. estimate ファイルの内容(その 3)

6-2. 性能値の評価

各種の性能データからシステムの振る舞いを評価する時、どの程度の負荷が正常な動きで、それをどの程度超過するか、又は下回ると通常動作ではないかを判断します。この超過したか、又は下回ったかの基準となる値がスレシヨルドバリュウ(しきい値)です。

次に、どの区間で何回超過、又は下回った事象が発生したのかを測ることが重要です。システム性能管理の難しさはここにあります。

突発的な値の超過や下回った事象、重たいバッチ処理中の明らかに起こるべくして発生する事象等は問題ではなく、通常の振る舞いに見做して良いでしょう。

sardo はこの点を踏まえ、baseline レコードの計算によってしきい値、評価期間、及び発生回数を作成します。

本節ではこの baseline レコードの応用について解説します。

6-2-1. baseline レコード作成の仕組み

sardo.edit は編集の際に性能情報の傾向(トレンド)データを作成します。この傾向データを元に基準値(ベースライン)ファイル baseline が作成されます。

sardo.edit の作成する傾向データはヒストグラムになっています。編集対象のデータから、sar(1)コマンドの出力と、LAN の入出力パケット情報を示す netstat(1M)コマンド-I オプションの出力を元に、項目毎のヒストグラムが作成されます。

なお、sar(1)コマンド-d オプションのデータはデバイス名を持っています。性能情報の評価にあたってデバイス名を特定して評価することは難しいため除外しています。

次に、sptrendir ディレクトリの構成を示します。

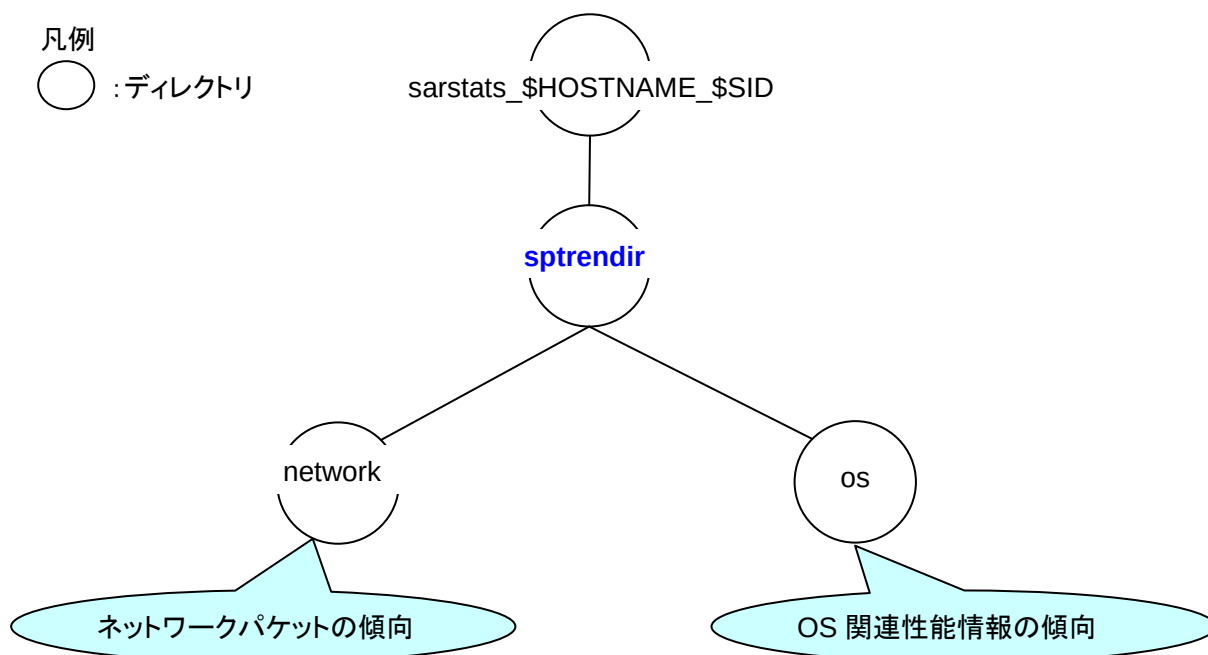


図 6-2-1-1. 傾向データ sptrendir ディレクトリの構成

sardo.edit はこれらの傾向データを基に、sardo ディレクトリに baseline ファイルを作成します。次は、傾向データと baseline ファイルのレコードがどのような関係になっているかを示しています。

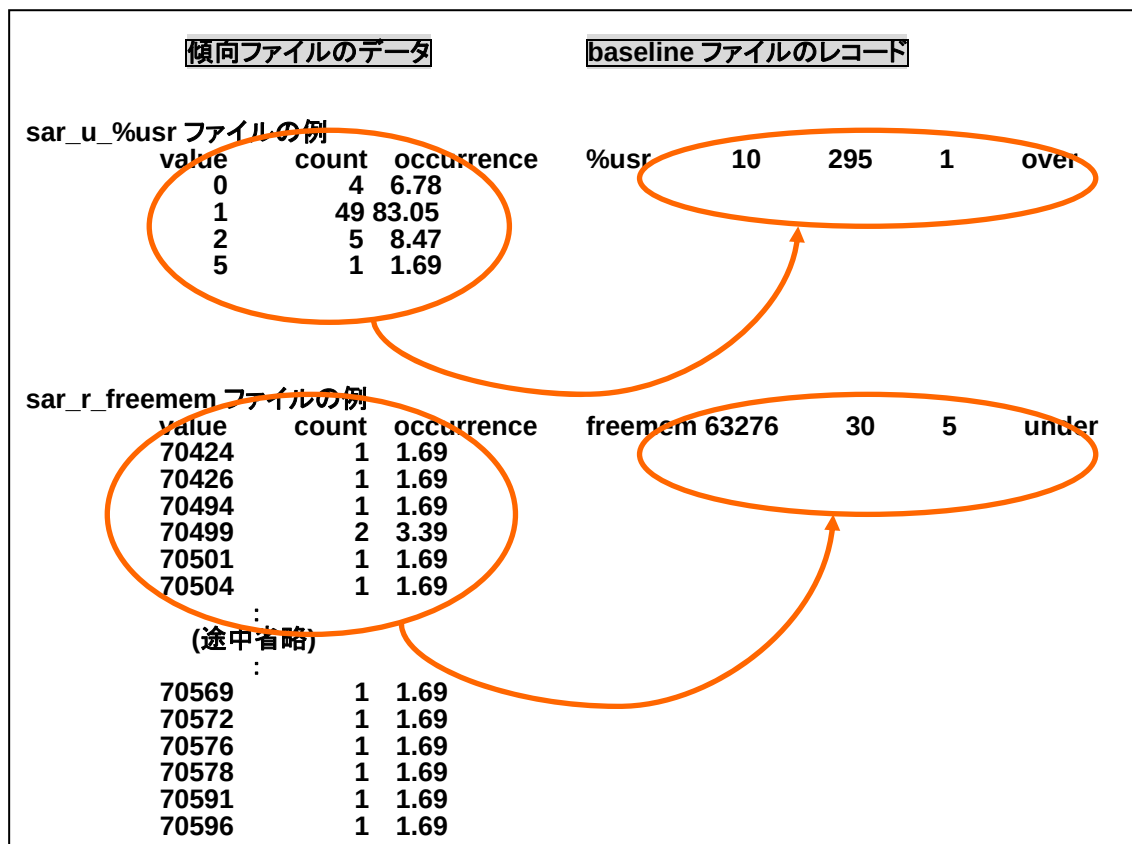


図 6-2-1-2. 傾向データと baseline レコードの関係

6-2-2. baseline を用いた性能データの評価

sardo.eval シェルスクリプトは、基準の baseline ファイルから、新たに採取した性能情報を比較して評価します。次に sardo.eval の標準出力メッセージと、評価結果の sardo.baseline ファイルの関係を示します。

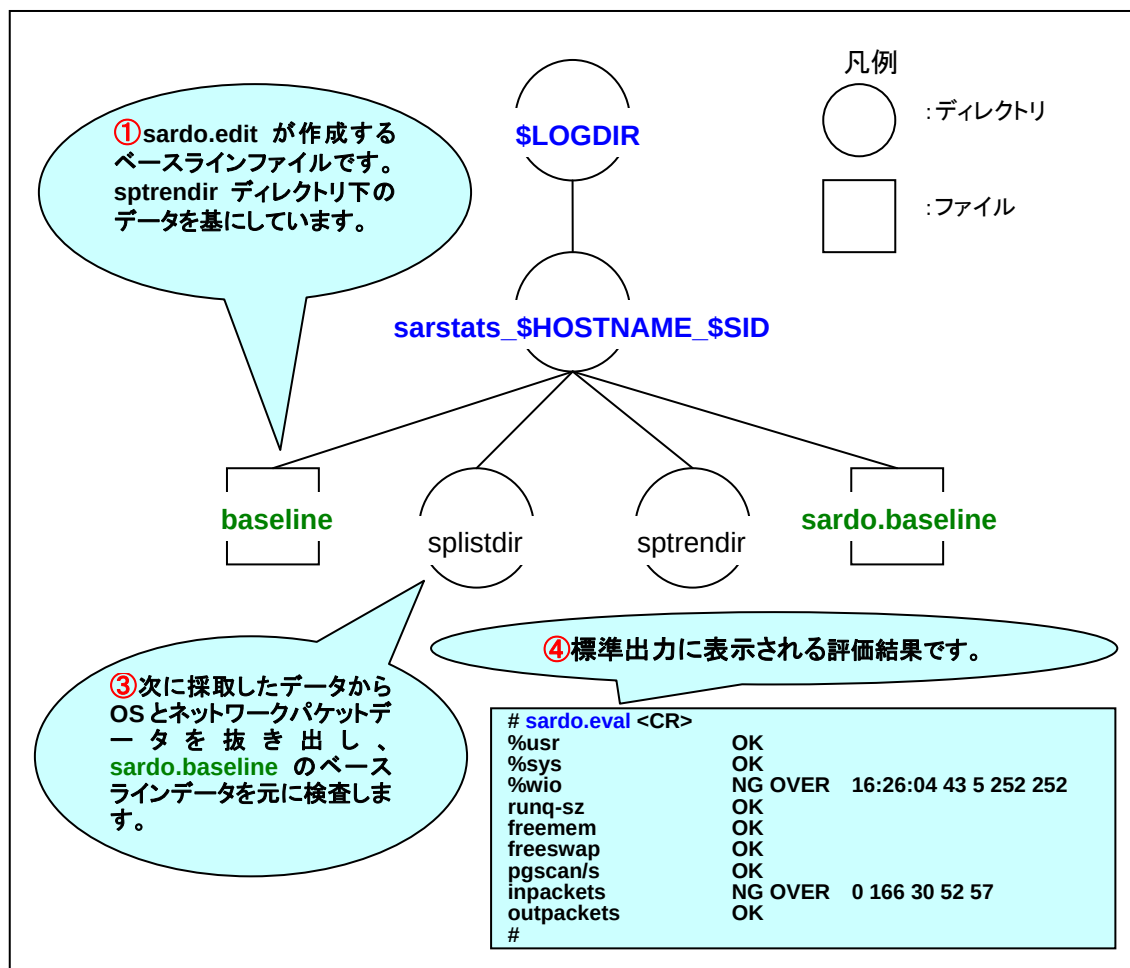


図 6-2-2-1. sardo.eval の処理

次は、sardo.eval が作成する sardo.baseline ファイルの例です。

%usr	77	30	4	over
%sys	73	75	14	over
%wio	35	30	6	over
runq-sz	6	35	6	over
freeswap	1985569	30	3	under
pgscan/s	200	150	2	over
inpackets	150	210	3	over
outpackets	163	150	4	over

図 6-2-2-2. sardo.eval が作成する sardo.baseline ファイルの例

6-2-3. baseline ファイルのカスタマイズ

sardo.edit の作成する baseline ファイルを基にしきい値や回数をカスタマイズし、sardo.eval で評価することができます。

次にカスタマイズと評価の流れを示します。

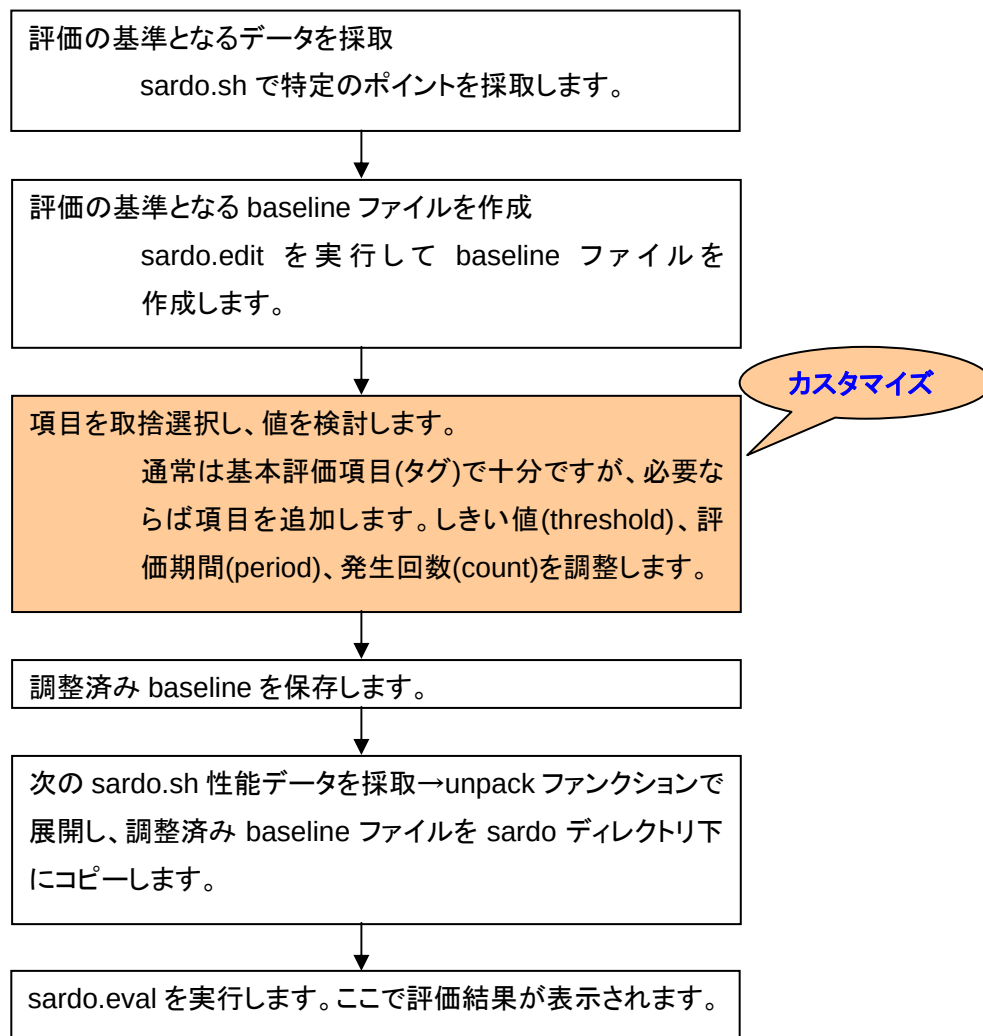


図 6-2-3-1. 評価の流れと baseline ファイルのカスタマイズ

6-3. 環境変数

6-3-1. 共通の環境変数

sardo.sh、sardo.edit、及び sardo.eval 共通の環境変数は次の通りです。

- BATCHMODE** **NO** が省略時値です。メッセージは端末画面に出力されます。また、sardo.log にも記録されます(一部記録されないメッセージもあります)。
YES の場合、バッチモードになります。メッセージは端末画面に出力されず、sardo.log ファイルにのみ記録されます。実行結果は戻り値で判断します。
- LOGDIR** **/var/tmp** が省略時値です。sardo ディレクトリの決定に使用されます。採取情報を記録するのに十分な空き領域のあるファイルシステム(ディレクトリ)を指定して下さい。
- HOSTNAME** ホスト名を表示する `uname -n` の実行システムの出力が省略時に用いられます。hostname(1)コマンドの出力と同じものです。sardo ディレクトリの決定に使用されます。
- SARDO** **1** が省略時値です。複数の sardo インスタンスを使用・編集したい場合は 2~999 までの値を指定することが出来ます。値が 1~2 桁の数字の場合は先頭に 0(ゼロ)が付加され 3 桁の数字になり、sardo ディレクトリの決定に使用されます。

【主な使用目的】

sardo.sh で性能情報を採取する場合、LOGDIR と SARDO 環境変数を組み合わせて、複数の sardo.sh を実行することができます。

sardo.edit を他システムで実行する場合、採取した tar アーカイブ、gzip 圧縮された sarstats ファイルを移動し、HOSTNAME、LOGDIR、および SARDO 環境変数を採取時のものにセットして編集することができます。

cron(1M)コマンドで、sardo.sh の各種ファンクションを組み込んだシェルスクリプトを実行する場合、BATCHMODE 環境変数を使用すると、戻り値によるシェルスクリプト内の制御が可能です。

6-3-2. sardo.edit 固有の環境変数

sardo.edit 固有の環境変数は次の通りです。

DISKEDIT	ALL が省略時値です。ディスク装置が多数構成されている場合、ディスク性能情報の編集に時間がかかることがあります。この際、編集をスキップする場合に指定します。sar(1M)コマンド-d オプションのディスク情報、及び iostat(1M)コマンド-xdn オプションのディスク情報が編集されます。 IOSTAT を指定すると iostat(1M)コマンド-xdn オプションのディスク情報のみ編集されます。 SAR を指定すると sar(1M)コマンド-d オプションのディスク情報が編集されます。 NO を指定すると両方とも編集されません。
PACCTEDIT	YES が省略時値です。プロセスアカウントの出力情報が編集されます。採取期間が長時間にわたり、多数のプロセスが起動・終了するシステムで、プロセスアカウントリング情報が大きくなると編集に時間がかかることがあります。このような場合、 NO を指定すると編集されません。
PSEDIT	YES が省略時値です。ps(1)コマンドの出力情報が編集されます。 NO を指定すると編集されません。

【主な使用目的】

編集時間が長くなった場合に、ディスク情報やプロセスアカウント、ps(1)情報のいずれかを確認することで、編集を制限することができます。また、grapher3.xls で Excel ファイルの制限値を超える編集結果になるような場合、ファイル出力を少なくすることができます。

6-3-3. sardo.eval 固有の環境変数

IMMEDIATE	NO が省略時値で、評価する項目が複数あった場合、全ての評価項目を確認して結果を表示します。 YES を指定するとしきい値を超えたもの(over Direction の場合)、または下回ったもの(under Direction の場合)を発見するとその項目を表示してただちに終了します。
-----------	---

6-4. sardo.event イベントチェックルーチン

sardo.sh ではサンプリングインターバル毎に制御を得て、独自の処理を行うイベントルーチンを組み込むことができます。

イベントルーチンの名前は、**sardo.event** の固定の名前です。これをパスの通ったディレクトリに置いておきます。

実行パーミッションを **sardo.event** に与えて sardo.sh を実行すると、サンプリングインターバル毎に制御をもらうことができます。実行パーミッションが OFF の場合は実行されません。

例として、/usr/local/bin ディレクトリに次のシェルスクリプトを作成します。

```
[root@act191 bin]# echo $PATH <CR>                ←パスを確認します。
/usr/lib64/qt-3.3/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin
[root@act191 bin]# cat sardo.event <CR>
#!/bin/sh

tail -1 $SARDODIR/data/vmstat_info |
awk '{
if ( $4 < 360000 )
    printf("WARN: free memory lower than 360000 pages. [ %d ]¥n",$4)
}'
[root@act191 bin]# chmod 700 sardo.event <CR>        ←実行パーミッションをセットします。
[root@act191 bin]# ls -l sardo.event<CR>            ←実行パーミッションを確認します。
-rwx-----. 1 root root 143 12 月   9 17:50 2010 sardo.event
[root@act191 bin]#
```

sardo.sh を実行してメモリを消費していくと次のようにメッセージが表示されます。

```
[root@act191 sardo]# ./sardo.sh auto 5 now 5min <CR>
[root@act191 sardo]# WARN: free memory lower than 360000 pages. [ 327380 ]
WARN: free memory lower than 360000 pages. [ 170724 ]
WARN: free memory lower than 360000 pages. [ 66700 ]
WARN: free memory lower than 360000 pages. [ 61624 ]
WARN: free memory lower than 360000 pages. [ 71792 ]
:
※条件が満たされなくなるとメッセージは表示されなくなります。
:
[root@act191 sardo]#
```

sardo.event 内では環境変数 **SARDODIR** を利用して sardo.sh の採取している各種情報を参照することができます(上記サンプルシェルの上から3行目です。この例では vmstat_info ファイルを参照しています)。

もちろん、イベントルーチン内で sar(1)や iostat(8)コマンドを実行しても良いですが、せっかく sardo.sh が採取している色々な性能情報があるので、できるだけそれらを利用してシステムの負荷を低減すべきと考えます…

このようにして、sardo.event に色々な機能を組み込むことで、トラップやメール送信などが可能です。

6-5. cron(8)デーモンによる sardo.sh の自動起動

sardo.sh を cron(8)デーモンで自動的に起動する方法を解説します。

- (1) cron(8)で起動されるシェルスクリプトを作成します。次は、sardo.sh を、5 秒のサンプリングインターバルで cron(8)によって起動された時刻“now”から5 分間採取する例です。起動の際、インスタンス ID を sardo.sid から得てプラス 1 して使用し、識別に利用します。

```
[root@act192 sardo]# cat exsardo.sh <CR>
#!/bin/sh

BATCHMODE=YES
export BATCHMODE

SARDO=`/bin/awk '{ print $1 + 1 }' /home/katsumi/sardo/sardo.sid`
if [ $SARDO -gt 999 ]; then
    SARDO=0
fi
export SARDO
/bin/echo $SARDO > /home/katsumi/sardo/sardo.sid
/home/katsumi/sardo/sardo.sh auto 5 now 5min
[root@act192 sardo]#
```

- (2) (1)の **exsardo.sh** で参照される SARDO のインスタンス ID をファイルに作成します。次は、初期値を“7”にセットする例です。

```
[root@act192 ~]# echo 7 > /home/katsumi/sardo/sardo.sid <CR>
[root@act192 ~]# cat /home/katsumi/sardo/sardo.sid <CR>
7
[root@act192 ~]#
```

- (3) crontab(1)コマンドの“-e”オプションで起動設定します。

```
[root@act192 ~]# crontab -e <CR>
vi(1)コマンドの画面が表示されます。
次は、15 時 10 分から 15 時 50 分まで自動採取する例です。
0 15 * * 1-5 /home/katsumi/sardo/exsardo.sh
10 15 * * 1-5 /home/katsumi/sardo/exsardo.sh
20 15 * * 1-5 /home/katsumi/sardo/exsardo.sh
30 15 * * 1-5 /home/katsumi/sardo/exsardo.sh
40 15 * * 1-5 /home/katsumi/sardo/exsardo.sh
50 15 * * 1-5 /home/katsumi/sardo/exsardo.sh
“ZZ”または“w!”で書き込んで vi(1)コマンドを終了します。
"/tmp/crontab.XXXEnXKK0" 6L, 270C written
crontab: installing new crontab ←このメッセージが表示されると設定は OK です。
[root@act192 ~]#
```

【参考】 crontab のフィールド指定

1	分	0-59
2	時	0-23
3	月内日	1-31
4	月	1-12
5	曜日	0-7(日曜日が 0 です。)
6	実行プログラム名	

- (4) 全部の情報採取が終了すると次のように/var/tmp に情報が採取されます。

```
[root@act192 ~]# cd /var/tmp <CR>
[root@act192 tmp]# ls -l <CR>
total 2464
-rw-r--r--. 1 root root 223039 Dec 10 15:05 sarstats_act192_008_101210150527.tar.gz
-rw-r--r--. 1 root root 225483 Dec 10 15:15 sarstats_act192_009_101210151527.tar.gz
-rw-r--r--. 1 root root 514860 Dec 10 15:25 sarstats_act192_010_101210152528.tar.gz
-rw-r--r--. 1 root root 514182 Dec 10 15:35 sarstats_act192_011_101210153526.tar.gz
-rw-r--r--. 1 root root 513402 Dec 10 15:45 sarstats_act192_012_101210154528.tar.gz
-rw-r--r--. 1 root root 516610 Dec 10 15:55 sarstats_act192_013_101210155527.tar.gz
[root@act192 tmp]#
```

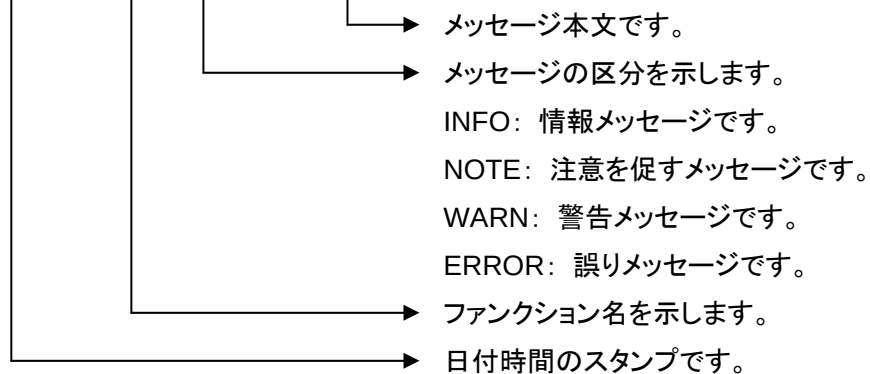
- (5) 後ほど自身のシステムで編集することができます。編集時、アーカイブの解凍・復元は環境変数 SARDO にインスタンス ID をセットして“sardo.sh unpack”し、“sardo.edit”を実行して編集します。sftp(1)コマンドなどで他システムに転送して編集する場合は、環境変数 HOSTNAME、および環境変数 LOGDIR でディレクトリを指定してから“sardo.sh unpack”し、“sardo.edit”を実行してください。

付録 A. メッセージ

\$LOGDIR/sarstats_\$(HOSTNAME)_\$(SID) ディレクトリのログファイル sardo.log に記録されるメッセージの形式は次のようになっています。画面には先頭の日付時間とファンクション名は表示されません。

なお、sardo.log に採取されないメッセージについてはメッセージ区分とメッセージ本文の間に"*"(アスタリスク)を付けて区別して記述します。

030729110726 SELECT :INFO: making os.info file has been done.



付録 A-1. sardo.sh のメッセージ

区分	メッセージ	対処等
	ERROR: \$BASENAME -\${SUFFICS} return value wasn't 0	\$BASENAME、オプション \$SUFFICSで示されるコマンドの実行が誤り終了しました。お手数ですが、sardo.shのtarファンクションでアーカイブしたファイルを弊社までご送付下さい。
	ERROR: \$CMD isn't executable.	\$CMDで示されるコマンドが実行モジュールではありません。該当のコマンドのパーミッションを確認して下さい。
	ERROR: \$DIRNAME/\$BASENAME isn't executable.	\$DIRNAME/\$BASENAMEで示されるコマンドが実行モジュールではありません。該当のコマンドのパーミッションを確認して下さい。
	ERROR: argument missing. [\$FUNCTION]	引き数が誤っています。
	ERROR: can't make \$SARDODIR/\${SID} direcotry.	開始時、または終了時の情報収集コマンドの実行にて、\$SARDODIR/\${SID}で示されるディレクトリの作成に失敗しました。ディスク空き領域が不足している可能性があります。
	ERROR: can't make sardo directory.	sardoディレクトリの作成に失敗しました。ディスク空き領域が不足している可能性があります。
	ERROR: file \$i copy was failed.	\$iで示されるファイルをmiscディレクトリにコピーしましたが誤り終了しました。ディスク空き領域が不足している可能性があります。または\$iで示されるファイルが有りません。
	ERROR: gzip command was failed.	圧縮コマンドが誤り終了しました。ディスク空き領域が不足している可能性があります。
	ERROR: instance \$INSTANCE has already been started.	\$INSTANCEで示されるsardoインスタンスが既に実行中です。
	ERROR: making device info file was failed.	device_infoファイルの作成が誤り終了しました。ディスクの空き領域が不足している可能性があります。
	ERROR: making misc directory was failed.	miscディレクトリの作成に失敗しました。ディスク空き領域が不足している可能性があります。
	ERROR: making pmap directory was failed.	pmapディレクトリの作成に失敗しました。ディスク空き領域が不足している可能性があります。
	ERROR: moving fsizer_info file was failed.	一時領域からfsizer_infoファイルをmiscディレクトリに移動しようとしたが誤りになりました。ディスク空き領域が不足している可能性があります。
	ERROR: process accounting directory couldn't be made.	プロセスアカウント情報を入れるディレクトリの作成に失敗しました。ディスク空き領域が不足している可能性があります。

ERROR: remove \$SARDODIR was failed.

ERROR: sadc command return value wasn't 0

ERROR: sadc process pid [\$SADCPID] wasn't running.

ERROR: sardo instance \$INSTANCE is still running.

ERROR: sardosize_info file exists.

ERROR: spautoexec setup function terminated.

ERROR: spautoexec start function terminated.

ERROR: spsetup parameter setting was failed.

ERROR: spstart instance \${SID} sardo.param wasn't found.

tarファンクションの後処理でsardo採取ディレクトリを削除する際に誤りが発生しました。圧縮tarアーカイブファイルが正しく作成されていた場合は手動にてsardoディレクトリを削除して下さい。圧縮tarアーカイブファイルが作成されていなかった場合は誤り原因を取り除き再度tarファンクションを実行して下さい。

sadcコマンドが誤り終了しました。ディスク空き領域が不足している可能性があります。

採取終了時点で実行されているはずのsadcプロセスが走っていませんでした。sardo.shを起動した後、その端末を電源オフした場合はプロセスが消失します。起動端末を落とすことが考えられる場合はnohup(1)コマンドによってsardo.shを起動する必要があります。この原因では無い場合、採取情報は完結していませんので次の手順で情報採取の上、弊社までご連絡願います。

(1) sardo.sh stop force <CR>
で強制停止。

(2) もしsardoディレクトリ下にpidlistファイルあればそれを削除します。

(3) sardo.shのtarファンクションでアーカイブに保存し弊社までご送付願います。

同じSARDO IDのインスタンスでsardo.shが実行中です。又は、sardo.editが実行中です。状態をご確認下さい。

estimateファンクション実行時にsardoディレクトリの下にsardosize_infoファイルが存在しました。空き領域計算に使用されるファイルです。ご確認の上ファイルを消去後再実行願います。通常発生することはありません。

自動実行時のsetupファンクションが異常終了しました。sardo.logを確認して下さい。

自動実行時のstartファンクションが異常終了しました。sardo.logを確認して下さい。

setupファンクションの引き数が誤っている可能性があります。ご確認下さい。またはディスク空き領域が不足している可能性があります。

startファンクションでパラメタファイルが見つかりませんでした。SARDO環境変数を確認して下さい。また、sardo.paramファイルが正しく作成されているかを確認して下さい。

ERROR: starting process account was failed.

ERROR: stop process accounting was failed.

ERROR: tar command was failed.

ERROR: the current time greater than the end time.

ERROR: the end parameter is invalid.

ERROR: the sampling interval invalid.

ERROR: the start and the end time are identical.

ERROR: the termination process starting was failed.

ERROR: the time format invalid.

ERROR: the time parameters are inconsistent.

ERROR: wc command was failed.

ERROR: whichcmd argument missing.

ERROR:*argument missing.

ERROR:*directory \$DIRECTORYNAME already exists.

ERROR:*directory \${ESDIR} was not found.

ERROR:*du command was failed.

ERROR:*making sarstats_\${HOST}_\${SID} directory was failed.

ERROR:*no any sardo archive files were found.

ERROR:*no any unpack files were found.

プロセスアカウンティングの起動に失敗しました。ディスク空き領域が不足している可能性があります。

プロセスアカウンティングの停止に失敗しました。システム運用管理者にお問い合わせ下さい。

tarコマンドが失敗しました。ディスク空き領域が不足している可能性があります。

現在時刻が終了時刻を超えています。時間を確認して下さい。

autoまたはsetup関数関数の最後の引き数が誤っています。時間の関係を確認して下さい。

また、採取が1日の範囲かどうかを確認して下さい。

サンプリングインターバルの指定が誤りです。引き数を確認して下さい。

開始時刻と終了時刻が同じです。

終了処理プロセスの起動に失敗しました。システムのプロセス数制限に到達した可能性があります。想定外の事象ですので、システム運用管理者にご確認下さい。

サンプリングインターバル、開始時刻、終了時刻の何れかに矛盾があります。

同上。

ファイルサイズ情報を採取する際に誤りとなりました。ディスク空き領域が不足している可能性があります。通常発生することはありません。sardo.shシェルスクリプトが壊れている可能性があります。

引き数誤りです。

unpack関数関数実行時、既に同じ名前のディレクトリが存在しました。SARDO環境変数と当該ディレクトリ関係を確認後、再実行して下さい。

ESDIRで示される親ディレクトリがありません。LOGDIR環境変数を確認して下さい。

estimate関数関数中のdu(1)コマンド-kオプションが失敗しました。ディスク空き領域が不足している可能性があります。

estimate関数関数時にsardoディレクトリ作成に失敗しました。ディスク空き領域が不足している可能性があります。

estimate関数関数実行時、目的の圧縮tarアーカイブファイルが見つかりませんでした。

unpack関数関数実行時、解凍目的の圧縮tarアーカイブファイルが見つかりませんでした。

ERROR:*now sardo_\${ESHOST}_\${ESID} directory is active.	estimateファンクション実行時、解凍した結果のsardoディレクトリが既に存在します。
ERROR:*several sardo archive files were found.	estimateファンクション実行時、いくつかの圧縮tarアーカイブファイルが発見されました。SARDO環境変数と目的のファイルの関係を確認してください。estimateファンクションではホスト名とSARDO環境変数の組み合わせは唯一無二でなくてはなりません。
ERROR:*spestimate invalid param.	estimateファンクションの引き数が誤っています。
ERROR:*tar -xvf command was failed.	unpackファンクション内のtarコマンドが失敗しました。ディスクの空き領域が不足している可能性があります。
ERROR:*tar command was failed.	estimateファンクション内のtarコマンドが失敗しました。ディスクの空き領域が不足している可能性があります。
ERROR:*uid is not root.	sardo.shはスーパーユーザで実行してください。
ERROR:*gunzip command was failed.	unpack、又はestimateファンクションにてgunzipコマンドが失敗しました。ディスク空き領域が不足している可能性があります。
INFO: \$BASENAME -\${SUFFICS} was successfully completed.	開始時と終了時の情報採取コマンドが正常に終了しました。
INFO: \$BASENAME -\${SUFFICS} was successfully scheduled.	自走型(sarなどの)コマンドが正常に起動されました。
INFO: an intermediate pmap -x command was completed.	sardo.shのpmapファンクションでプロセスメモリマップの出力が終了しました。
INFO: awaiting start \$waitsec sec.	開始時刻まで\$waitsec秒の間待ちます。
INFO: current time : \$ctime \$CTIME	現在時刻のsardo.logファイルスタンプです。
INFO: list devinfo was successfully completed.	devinfo_infoファイルのリストが正しく終了しました。
INFO: process accounting has already been started.	プロセスアカウンティングが既に起動済みです。先に実行されたsardo.shでプロセスアカウンティングが起動された場合はそちらを参照してください。もし先に実行されていない場合はそのシステムが元々プロセスアカウンティングを起動しています。
INFO: process accounting has been started.	可能であれば、次回から、システムの設定を解除し、sardo.shでプロセスアカウンティングを採取するよう、システム運用管理者にご相談下さい。
INFO: process accounting was stopped.	プロセスアカウンティングが起動されました。
	プロセスアカウンティングが停止されました。

INFO: process accounting wasn't stopped.

INFO: remaining time is \$REMAIN seconds.

INFO: sadc command was successfully scheduled.

INFO: sampling count : \$COUNT

INFO: sampling interval : \$INTERVAL

INFO: sampling mode : \$SAMPLING

INFO: sardo will be started with \$INTERVAL sec interval.

INFO: spsetup was successfully completed.

INFO: start and end time : \$stime \$STIME \$etime \$ETIME

INFO: termproc pid [\$PID] was started successfully.

INFO: wait for the end time [\$ETIME] to finish.

INFO: *spunpack was successfully completed.

INFO: *tar archive and compress were successfully completed.

INFO: *the unpack file \$FILENAME was chosen.

NOTE: spstop invalid mode.

NOTE: *\$DIR directory won't be cleared.

NOTE: *Although tar archive and compress
(続き)were successfully completed,

NOTE: *any unpack files list couldn't be displayed.

NOTE: *the unpack file \$FILENAME was assumed.

NOTE: *the working directory \$EXECDIRECTORY is below \$LOGDIR. (前述の続き)実行ディレク

後に実行されたsardo.shの停止時に先に実行中のsardo.shのプロセスアカウンティングを停止することはありません。この場合の停止しなかった旨を示すメッセージです。情報採取時間を示すメッセージです。

sadcコマンドが正常に起動されました。

サンプリングカウントを示すメッセージです。

サンプリングインターバルを示すメッセージです。

サンプリングの区間を示すモードです。その日中の情報採取の場合は、

“INTHEDAY”

日またがりの(すなわち午前0時を超える)場合は、

“OVERTHEDAY”

と表示されます。

サンプリングインターバルを示すメッセージです。

setupファンクションが正常に終了しました。

開始時刻と終了時刻を示すメッセージです。

終了処理プロセスが正常に起動されました。

終了時刻までの時間を示します。

unpackファンクションが正常に終了しました。

tarアーカイブと圧縮が正常に終了しました。

unpackファンクション時、引き数にファイルが指定されませんでしたので、\$FILENAMEで示されるファイルを選択しました。

stopファンクションの副引き数

は“force”でなくてはなりません。終了処理が不完全な可能性があります。

tar、またはautoファンクション実行の際、実行ディレクトリがsardoディレクトリの下でした。このため、当該のディレクトリは削除されませんでした。

しかしながら、tarアーカイブと

圧縮は正常に終了しました。

unpackファンクションで入力ファイルが指定されませんでしたので

\$FILENAMEを仮定しました。

トリがsardoディレクトリの下でした。

付録 A-2. sardo.edit のメッセージ

区分	メッセージ	対処等
ERROR:	copying \${PREFIX}_\${i} file was failed.	iostat_xdn編集済みファイルのコピーに失敗しました。ディスク空き領域が不足している可能性があります。高速ファイルデバイダの誤りです。
ERROR:	MESSAGE. argument error error in open the file %s %d file %s open fail. errno=%d file %s write fail. errno=%d fstat errno=%d getrlimit errno=%d %d iostat_xdn record length error malloc errno=%d open file %s errno=%d pacct_d record length error ps_d record length error read %s errno=%d read %s fail. errno=%d read fail errno=%d sar_d record length error setrlimit errno=%d %d ulink error %s errno=%d write fail. errno=%d	左記のメッセージがMESSAGEに表示されます。%sにはファイル名、%dには誤り番号(errno)等が表示されます。
ERROR:	arg1 \$INDIR wasn't directory.	通常表示されることはありません。引き数の誤りです。
ERROR:	argument missing. [\$FUNCTION]	通常表示されることはありません。
ERROR:	can't find sardo directory.	selectファンクション時、ネットワーク編集情報をコピーする際にエラーが発生しました。ディスク空き領域が不足している可能性があります。
ERROR:	copying \${i} file was failed.	selectファンクション時、ps(1)コマンド編集情報をコピーする際にエラーが発生しました。ディスク空き領域が不足している可能性があります。
ERROR:	copying \${i} was failed.	時間情報を付加したiostat -xdnファイルをspeditdir下にコピーする時に誤りが発生しました。ディスク空き領域が不足している可能性があります。
ERROR:	copying iostat -xdn files was failed.	selectファンクションにおいて、iostat -xdnファイルに時間情報を付加しsplisdirにコピーする時に誤りが発生しました。ディスク空き領域が不足している可能性があります。
ERROR:	copying iostat_xdn_* file was failed.	preserveディレクトリからiostat_xdn_*ファイルをコピーした時誤りが発生しました。ディスク空き領域が不足している可能性があります。
ERROR:	copying iostat_xdn_* files from preserve was failed.	上記の逆で、splisdirディレクトリからpreserveにiostat_xdn_*ファイルをコピーした時、誤りが発生しました。ディスク空き領域が不足している可能性があります。
ERROR:	copying iostat_xdn_* files to preserve was failed.	

ERROR: copying FILES was failed.

mpstat info
netsinfo (netstat -s情報です)
netstat -l
sar_* files
sar_d files
sar_d_* files
tcp_ipv4_info file
vmstat info
vmstat_info file
vmstat_p_info file

ERROR: copying netsinfo from preserve was failed.

ERROR: copying psinfo files from preserve directory was failed.

ERROR: copying psinfo files to preserve directory was failed.

ERROR: copying sar_d_* files to preserve was failed.

ERROR: file \$INFILE was not found.

ERROR: fork argument missing.

ERROR: making \$DIR was failed.

ERROR: making \${TRENDIR} direcotry was failed.

ERROR: making \${TRENDIR}/\${i} directory was failed.

ERROR: making ExecutedProcess file was failed.

ERROR: making pacctinfo directory was failed.

FILESで示されるsplistdirディレクトリ下の編集済み情報をspeditdirディレクトリにコピーする時に誤りが発生しました。ディスク空き領域が不足している可能性があります。左記のメッセージがFILESに表示されます。

netsinfoファイルをpreserveディレクトリにコピーした時、誤りが発生しました。ディスク空き領域が不足している可能性があります。

splistdir/psinfoディレクトリ下にpreserveのファイルをコピーした時に誤りが発生しました。ディスク空き領域が不足している可能性があります。

preserveディレクトリにpsinfoディレクトリ下のファイルをコピーする時に誤りが発生しました。ディスク空き領域が不足している可能性があります。

sar_d_*ファイルをpreserveにコピーする時に誤りが発生しました。ディスク空き領域が不足している可能性があります。

minmaxファイルの作成時、所定のファイルがありませんでした。ディスク空き領域が不足している可能性があります。ディスク空き領域が十分で、再実行しても同様の問題が発生する場合はtarファンクションでアーカイブし、弊社までご連絡下さい。

forkファンクションで引き数に誤りがあります。

プロセスアカウントの上位24位ファイル(syscpu、usercpu、memory、又はelaps)を入れるディレクトリの作成に失敗しました。ディスク空き領域が不足している可能性があります。sptrendirディレクトリの作成に失敗しました。ディスク空き領域が不足している可能性があります。

sptrendirディレクトリ下のos、又はnetworkディレクトリの作成に失敗しました。ディスク空き領域が不足している可能性があります。

プロセスアカウント情報から抽出した終了プロセス情報のファイル"ExecutedProcess"の作成に失敗しました。ディスク空き領域が不足している可能性があります。

pacctinfoディレクトリの作成に失敗しました。ディスク空き領域が不足し

ERROR: making preserve directory was failed.

ERROR: making preserve/diskinfo directory was failed.

ERROR: making preserve/netsinfo directory was failed.

ERROR: making preserve/psinfo directory was failed.

ERROR: making psinfo directory was failed.

ERROR: mkdir \$i was failed.

ERROR: mkdir \$i/\$j was failed.

ERROR: mkdir netsinfo was failed.

ERROR: mkdir pmap_\${DIRNAME} was failed.

ERROR: mkdir pmap_\${i} was failed.

ERROR: moving \$k was failed.

ERROR: pmap file \$INFILE was not found.

ERROR: pmap function needs file name.

ERROR: pmapinfo directory making was failed.

ERROR: ps_info file wasn't found.

ERROR: return value wasn't zero at \$0.

ERROR: sadc_info file was not found.

ERROR: sadc_info file wasn't found.

ERROR: sardo instance \$INSTANCE is still running.

ERROR: the file had two fields which couldn't be forked.

ている可能性があります。
preserveディレクトリの作成に失敗しました。ディスク空き領域が不足している可能性があります。

preserve/diskinfoディレクトリの作成に失敗しました。ディスク空き領域が不足している可能性があります。
preserve/netsinfoディレクトリの作成に失敗しました。ディスク空き領域が不足している可能性があります。
preserve/psinfoディレクトリの作成に失敗しました。ディスク空き領域が不足している可能性があります。
psinfoディレクトリの作成に失敗しました。ディスク空き領域が不足している可能性があります。

\$iで示されるディレクトリの作成に失敗しました。ディスク空き領域が不足している可能性があります。
\$i/\$jで示されるディレクトリの作成に失敗しました。ディスク空き領域が不足している可能性があります。
netsinfoディレクトリの作成に失敗しました。ディスク空き領域が不足している可能性があります。

pamp_\${DIRNAME}で示されるディレクトリの作成に失敗しました。ディスク空き領域が不足している可能性があります。

pmap_\${i}で示されるディレクトリの作成に失敗しました。ディスク空き領域が不足している可能性があります。

\$kで示されるファイルをsplstdirディレクトリ下からspeditdirにコピーする時に誤りが発生しました。ディスク空き領域が不足している可能性があります。

pmap関数で\$INFILEに示されるファイルがありませんでした。
pmap関数で引き数にファイル名が指定されませんでした。
pmapinfoディレクトリの作成に失敗しました。ディスク空き領域が不足している可能性があります。

ps情報の編集でps_infoファイルがありませんでした。

現在未使用のメッセージです。
sadc_infoファイルが見つかりませんでした。

DISKEDIT環境変数にSARが明示的に指定されているにもかかわらずsadc_infoファイルが見つかりませんでした。

\$INSTANCEのsardo.sh、又はsardo.edit実行中です。
fork関数の対象のファイル

ERROR: the position no \$POSITION over \$NF fields.

ERROR: the time format invalid.

ERROR: uid is not root.

ERROR: whichcmd argument missing.

ERROR:*argument missing.

INFO: copying \$TAG info to speditdir has been done.

INFO: files vmstat_i_* haven't been made.

INFO: making \${i} has been done.

INFO: making iostat -xdn information has been skipped.

INFO: making iostat_xdn_* files has been done.

INFO: making memory.info file has been done.

INFO: making minmax files has been done.

INFO: making mpstat_cpu_info files has been done.

INFO: making netstat -an info files has been done.

INFO: making netstat -s info has been done.

INFO: making netstat -s info was started.

INFO: making netstat_IF_file files has been done.

INFO: making network.info file has been done.

INFO: making os info sar_x files has been done.

INFO: making os.info file has been done.

INFO: making pacct info files has been done.

INFO: making pacct info files has been skipped.

INFO: making pacct info files was started.

INFO: making ps info files has been done.

INFO: making ps info files has been skipped.

INFO: making ps info files was started.

INFO: making sar -d information has been done.

は2つのフィールドでした(即ち、fork
済みファイル)。

minmaxファンクションで指定したフィールド位置がファイルのフィールド数を超えています。

時間パラメタ(サンプリングインターバル、開始時刻、終了時刻)の形式に誤りがあります。

実行がスーパーユーザではありません。

通常発生することはありません。

sardo.editシェルスクリプトが壊れている可能性があります。

引き数に誤りがあります。

\$TAGで示されるファイルのコピーが終了しました。

vmstat_i_*ファイルが作られませんでした。

\${i}で示されるファイルが作成されました。

DISKEDIT環境変数の指定により、iostat -xdn情報の編集は行われませんでした。

iostat -xdn情報が作成されました。

memory.infoファイルが作成されました。

minmaxファイルが作成されました。

mpstat_cpu_infoファイルが作成されました。

netstat-an関連のファイルが作成されました。

netstat -s関連のファイルが作成されました。

netstat-sの編集が開始されました(編集に時間がかかるために表示しています)。

netstat_IF_fileファイルが作成されました。

network_infoファイルが作成されました。

sar関連ファイルが作成されました。

os.infoファイルが作成されました。

プロセスアカウントの情報ファイルが作成されました。

PACCTEDIT環境変数の指定によってプロセスアカウントの編集は行われませんでした。

プロセスアカウントの編集が開始されました(編集に時間がかかるために表示しています)。

ps情報が作成されました。

PSEDIT環境変数の指定によってps情報の編集は行われませんでした。

ps情報の編集が開始されました(編集に時間がかかるために表示しています)。

sar-d(ディスク情報)の作成が終了し

INFO: making sar -d information has been skipped.

INFO: making trend files has been done.

INFO: making vmstat_i_* files has been done.

INFO: mapping pmap_*_info files has been done.

INFO: mapping pmap_x_*_info files has been done.

INFO: pagesize was assumed 8192 bytes.

INFO: timelist will be made by parameter file.

NOTE: file \${i} hasn't been made.

NOTE: pagesize_info file wasn't found.

NOTE: ps_info file wasn't found.

WARN: any network files couldn't be found.

WARN: files mpstat_cpu_info haven't been made.

WARN: files netstat_IF_file haven't been made.

WARN: iostat_xdn_info file wasn't found.

WARN: netstat_an_info file wasn't found.

WARN: netstat_s_info file wasn't found.

WARN: network.info file hasn't been made.

WARN: no any sar files haven't been made.

WARN: os info sar_x files haven't been made.

WARN: os.info file hasn't been made.

ました。

DISKEDIT環境変数の指定により sar-d情報の編集は行われませんでした。

trend(傾向)ファイルが作成されました。

vmstat-iの作成が終了しました。

beginとendに関するpmap-x情報の編集が終了しました。

任意の時点で採取されたpmap-xの編集が終了しました。

miscディレクトリにpagesize_infoファイルが無かったため、ページサイズを8192バイトに仮定しました。

プロセスアカウント情報の編集に使用する時間のリストをps_infoファイルから作成しようとしたが、

ps_infoファイルが有りませんでした。

このため、sardo.paramパラメタファイルのサンプリングインターバル、開始時刻、終了時刻を用いて時間リストを作成します。

\${i}で示されるファイルが作成されませんでした。

miscディレクトリにpagesize_infoファイルが見つかりませんでした。

ps_infoファイルが見つかりませんでした。

いくつかのネットワーク関連性能情報ファイルがありませんでした。完全なnetwork.infoファイルが作成されない可能性があります。

mpstat_cpu_infoファイルが作成されませんでした。

LANインタフェース毎のnetstat -Iファイルが作成されませんでした。

iostat -xdn情報が採取されていませんでした。

netstat-an情報が採取されていませんでした。

netstat-s情報が採取されていませんでした。

network.infoファイルが作成されませんでした。

sadc_infoファイルが無かったため sar関連ファイルが作成されませんでした。

同上

os.infoファイルが作成されませんでした。

付録 A-3. sardo.eval のメッセージ

区分	メッセージ	対処等
	*any BaseLine parameter errors were found.	ベースラインパラメタにエラーが発見されました。
	ERROR:*any baseline parameter errors were found.	ベースラインパラメタのエラー、又は他の理由によりsardo.baselineファイルが作成されませんでした。 sardoディレクトリ下にbaselineファイルが見つかりませんでした。 splistdirディレクトリの作成に失敗しました。ディスク空き領域が不足している可能性があります。
	ERROR:*baseline file wasn't found.	splistdir/osディレクトリの作成に失敗しました。ディスク空き領域が不足している可能性があります。
	ERROR:*making splistdir directory was failed.	netstat_IF_infoのファイルがありません。
	ERROR:*making splistdir/os directory was failed.	sardo.shか、又はsardo.editが実行中です。
	ERROR:*no any netstat_IF_info files were found.	sadc_infoファイルが見つかりませんでした。
	ERROR:*now sardo.sh or sardo.edit is running.	sardoディレクトリが見つかりませんでした。
	ERROR:*sadc/sadc_info file wasn't found.	\$iで示されるベースラインファイルのtagが見つかりませんでした。
	ERROR:*sardo directory was not found.	
	ERROR:*tag \$i wasn't found.	

付録 B. ツールのシステム性能に及ぼす影響

sardo.sh は 5 秒毎に sadc(1M)コマンドによる性能情報を採取し、またサンプリングインターバル毎に ps(1)、vmstat(1M)、netstat(1M)、mpstat(1M)、iostat(1M)等の多くのコマンドを実行し情報を採取します。これらの多くの情報を採取することから、ツールの負荷がシステム全体にどのように影響しているのかを測定しました。

測定環境

次にハードウェア構成を示します。

表 B-1. ハードウェア構成

No	区分	内容
1	CPU	UltraSPARC-III 900MHz × 1
2	メモリ	1GB
3	ディスク	FC-AL 73GB 内蔵ディスク × 1
4	ネットワーク	ERI0 100Mbps イーサネットアダプタ

次にソフトウェア構成を示します。

表 B-2. ソフトウェア構成

No	区分	内容
1	OS	Solaris 8 02/02
2	データベース	Oracle 9.0.1.0.0

測定内容

次に測定内容を示します。

表 B-3. 測定内容

図番号	コマンド	内容
B-1	sardo.sh	30 秒の間隔で、15 時 30 分から翌日の 15 時 29 分までの 24 時間を採取
B-2	orado.sh	23 時 55 分～0 時 15 分にかけてのorado.sh ^{【備考 1】} とorado.editの動作
B-3	sar(1)	No.1 sardo.sh開始前 3 分から 5 分間、1 秒の間隔で採取 ^{【備考 2】}
B-4	sar(1)	16 時 8 分から 5 分間、1 秒の間隔で採取 ^{【備考 2】}
B-5	sar(1)	翌日 15 時 19 分から 5 分間、1 秒の間隔で採取 ^{【備考 2】}
B-6	sar(1)	翌日 15 時 24 分から 5 分間、1 秒の間隔で採取 ^{【備考 2】}
B-7	sar(1)	sardo.shの終了前後を 1 秒の間隔で採取 ^{【備考 2】}
B-8	sar(1)	sardo.shが終了した後のアイドル時状態を 1 秒の間隔で 5 分間採取 ^{【備考 2】}
B-9	sardo.sh	sardo.edit の“all”と、“select 23:55:00 00:14:59”の動き

【備考 1】 STATSPACK を使用しています。

【備考 2】 sar(1)コマンドの最小測定間隔は 5 秒です。本測定の場合、粒度が必要でした。このため、1 秒にしています。

次に、sardo.sh の採取した 1 日分の振る舞いを示します。STATSPACK のスナップ採取と、その編集によって負荷の高い CPU 使用とディスク使用が認められます。

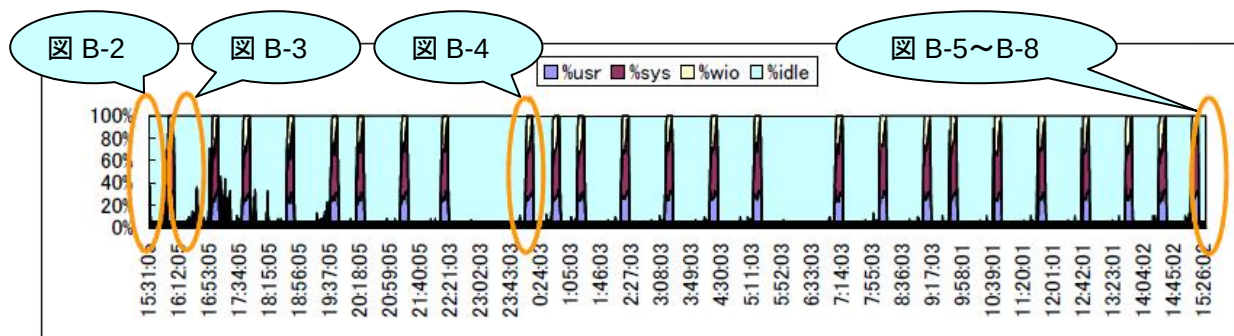


図 B-1. sardo.sh による 1 日分の振る舞い

次表に、sardo.sh 起動前後の変化を平均値で示します。

表 B-4. sardo.sh 起動前後の CPU とディスク入出力待ちの変化

タイミング	区間		%usr	%sys	%wio	%idle	CPU 合計
sardo.sh 起動前	15:27:07	15:30:00	0.20	0.17	0.16	99.47	0.37
sardo.sh 起動時	15:30:01	15:30:07	19.00	35.86	8.57	36.57	54.86
sardo.sh 起動後	15:30:08	15:32:06	0.42	0.79	0.27	98.52	1.21

次図は、sardo.sh の起動前と後を示したグラフです。15 時 30 分に起動がかかり、約 85 パーセントの CPU と、約 15 パーセントのディスク入出力待ちが記録されています。

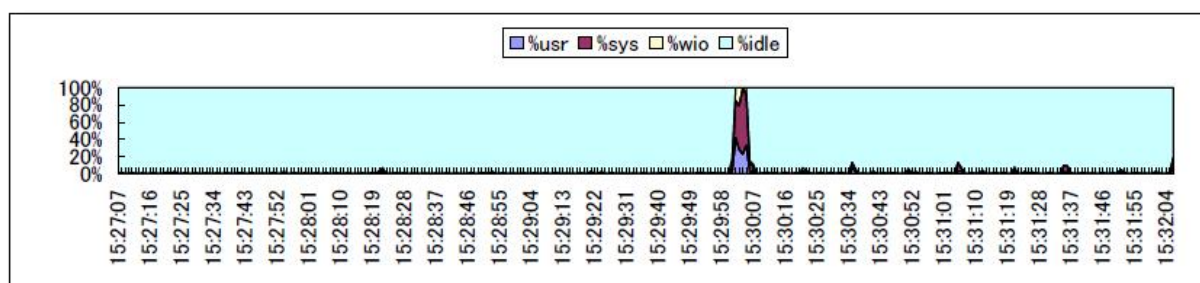


図 B-2. sardo.sh 起動前後の振る舞い

次図は、sardo.sh 実行中(性能情報採取中のアイドル)の動きです。30 秒のサンプリングインターバル毎に約 2~5 パーセントの CPU 使用が認められます。

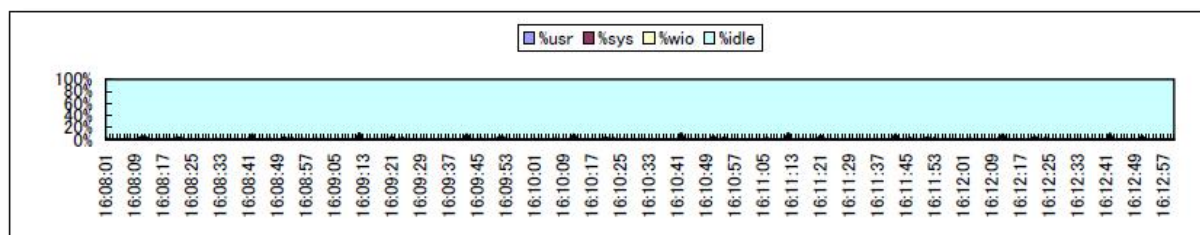


図 B-3. sardo.sh 実行中(性能情報採取中アイドル)の振る舞い(その 1)

【参考】

次は、23 時 55 分から翌日 0 時 15 分にかけて、STATSPACK と編集の動きを図にしたものです。STATSPACK によって、数パーセントの CPU と 5～40 パーセントのディスク入出力待ちが記録されています。編集(orado.edit)はバッチ処理のため、多くのリソースが消費されます。

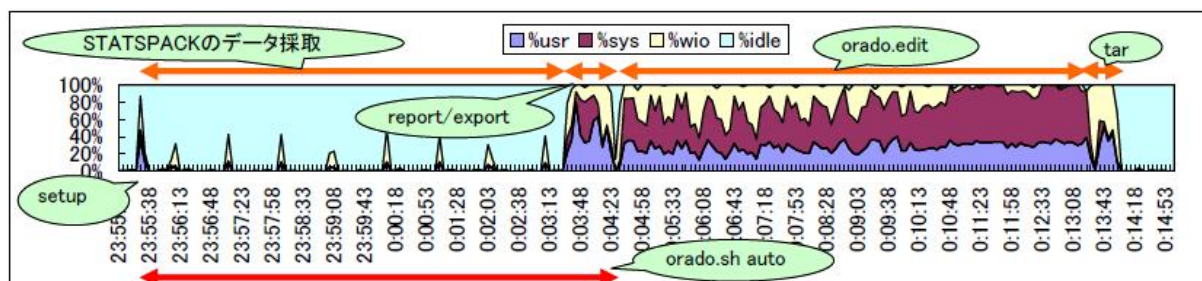


図 B-4. STATSPACK と orado.edit の振る舞い

次は、sardo.sh の終了直前のグラフです。図 B-3 とほぼ同じ動きになっています。

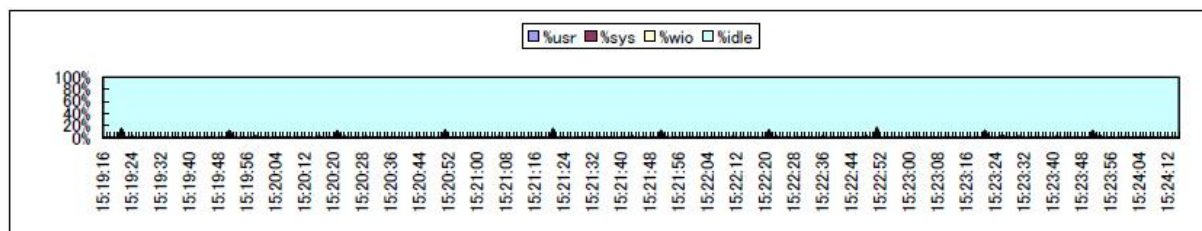


図 B-5. sardo.sh 実行中(性能情報採取中アイドル)の振る舞い(その 2)

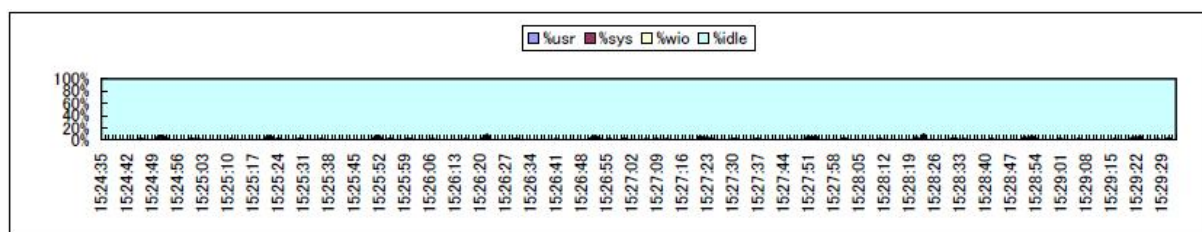


図 B-6. sardo.sh 実行中(性能情報採取中アイドル)の振る舞い(その 3)

次は、図 B-3、B-5、および B-6 の CPU 平均とディスク入出力待ちを示しています。1 パーセント以下の CPU 使用です。ディスク入出力待ちは平均 0.166 パーセントです。

表 B-5. アイドル時の CPU とディスク入出力待ち

タイミング	区間		%usr	%sys	%wio	%idle	CPU 合計
図 B-3(その 1)	16:08:01	16:13:00	0.37	0.47	0.18	98.98	0.84
図 B-5(その 2)	15:19:16	15:24:15	0.34	0.59	0.14	98.93	0.93
図 B-6(その 3)	15:24:35	15:29:33	0.34	0.55	0.15	98.96	0.89

次に sardo.sh が終了する時のグラフを示します。sardo.sh は、自身が起動した各種の性能採取コマンドを終了させ、sardo ディレクトリを tar(1)コマンドでアーカイブし、その後、アーカイブを gzip(1)コマンドで圧縮します。

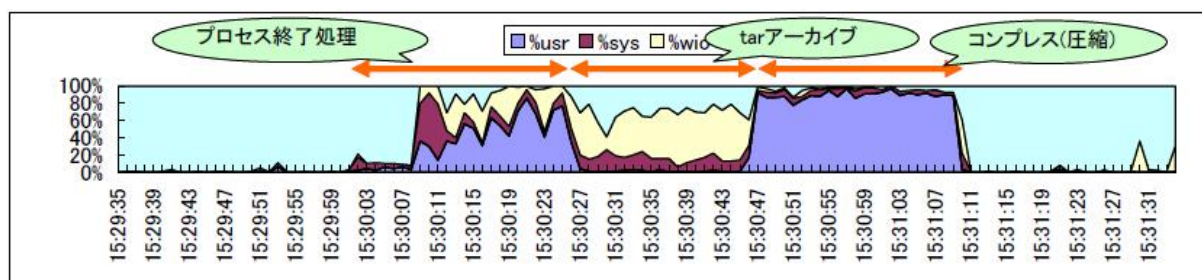


図 B-7. sardo.sh 終了時の振る舞い

次は、sardo.sh が終了した後のアイドル状態を示すグラフです。

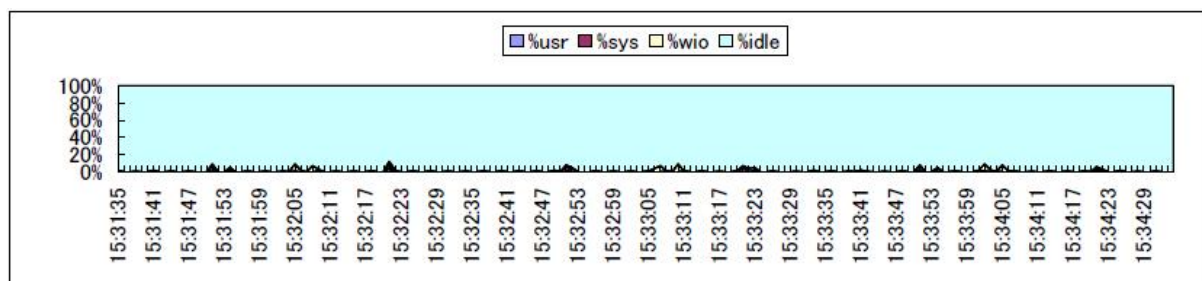


図 B-8. sardo.sh 終了後の振る舞い

sardo.sh 終了後のアイドル時は次のような CPU 使用と、ディスク入出力待ちになります。上段に sardo.sh 起動前の値を示します。

使用率が起動前に比べてやや高くなっています。これは、測定期間中にキャッシュされたディスクバッファキャッシュ量の増加により、fsflush コマンドの動作が活発になった影響と考えられます。

表 B-6. sardo.sh 終了後アイドル時の平均値

タイミング	区間		%usr	%sys	%wio	%idle	CPU 合計
sardo.sh 起動前	15:27:07	15:30:00	0.20	0.17	0.16	99.47	0.37
sardo.sh 終了後	15:31:35	15:34:34	0.27	0.30	0.52	98.91	0.57

次に、本測定で採取された 1 日分のデータを他の機会に sardo.edit で編集した時の動きです。13 時 31 分から 14 時 46 分まで、“all”による編集、14 時 50 分から 15 時 5 分まで、“select 23:55:00 00:14:59”によって部分編集しています。プロセスアカウント情報の編集時間が大部分を占めます。



図 B-9. sardo.edit の振る舞い

これらの評価で採取された sardo.log ファイルの内容を示します。

```
030815152301 SETUP :INFO: spsetup was successfully completed.
030815152301 START :INFO: sampling interval : 30
030815152301 START :INFO: sampling count : 2879
030815152301 START :INFO: current time : 55381 15:23:01
030815152301 START :INFO: start and end time : 55800 15:30:00 142199 15:29:59
030815152301 START :INFO: sampling mode : OVERTHEDAY
030815152301 START :INFO: awaiting start 419 sec.
030815153000 START :INFO: sardo will be started with 30 sec interval.
030815153001 START :INFO: process accounting has been started.
030815153001 START :INFO: crash - was successfully completed.
030815153001 START :INFO: df -k was successfully completed.
030815153002 START :INFO: ipcs -a was successfully completed.
030815153002 START :INFO: netstat -s was successfully completed.
030815153002 START :INFO: netstat -k was successfully completed.
030815153003 START :INFO: pmap -x was successfully completed.
030815153004 START :INFO: swap -s was successfully completed.
030815153004 START :INFO: netstat - was successfully scheduled.
030815153004 START :INFO: netstat - was successfully scheduled.
030815153004 START :INFO: vmstat -p was successfully scheduled.
030815153004 START :INFO: iostat -xd was successfully scheduled.
030815153004 START :INFO: vmstat - was successfully scheduled.
030815153004 START :INFO: mpstat - was successfully scheduled.
030815153004 START :INFO: ps - was successfully scheduled.
030815153004 START :INFO: vmstat -i was successfully scheduled.
030815153004 START :INFO: netstat -s was successfully scheduled.
030815153005 START :INFO: netstat -an was successfully scheduled.
030815153005 START :INFO: sadc command was successfully scheduled.
030815153006 START :INFO: wait for the end time [ 15:29:59 ] to finish.
030815153006 START :INFO: termproc pid [ 1172 ] was started successfully.
030815153006 START :INFO: remaining time is 86399 seconds.
030816153008 STOP :INFO: crash - was successfully completed.
030816153008 STOP :INFO: df -k was successfully completed.
030816153008 STOP :INFO: ipcs -a was successfully completed.
030816153008 STOP :INFO: netstat -s was successfully completed.
030816153008 STOP :INFO: netstat -k was successfully completed.
030816153010 STOP :INFO: pmap -x was successfully completed.
030816153010 STOP :INFO: swap -s was successfully completed.
030816153010 STOP :INFO: process accounting was stopped
030816153011 STOP :INFO: list devinfo was successfully completed.
030816154050 ALL :INFO: making os info sar_x files has been done.
030816154051 ALL :INFO: making vmstat_info has been done.
030816154051 ALL :INFO: making vmstat_p_info has been done.
030816154125 ALL :INFO: making sar -d information has been done.
030816154128 ALL :INFO: making iostat_xd_* files has been done.
030816154128 ALL :INFO: making mpstat_cpu_info files has been done.
030816154131 ALL :INFO: making vmstat_i_* files has been done.
030816154146 ALL :INFO: making netstat_IF_file files has been done.
030816154146 ALL :INFO: making ps info files was started.
030816155058 ALL :INFO: making ps info files has been done.
030816155058 ALL :INFO: making pacct info files was started.
030816164554 ALL :INFO: making pacct info files has been done.
030816164555 ALL :INFO: mapping pmap_*_info files has been done.
```


030816164602 ALL	:INFO: making netstat -an info files has been done.
030816164602 ALL	:INFO: making netstat -s info was started.
030816165035 ALL	:INFO: making netstat -s info has been done.
030816165043 ALL	:INFO: copying cpu info to speditdir has been done.
030816165055 ALL	:INFO: copying disk info to speditdir has been done.
030816165101 ALL	:INFO: copying network info to speditdir has been done.
030816165202 ALL	:INFO: copying os info to speditdir has been done.
030816165257 ALL	:INFO: copying process info to speditdir has been done.
030816165317 ALL	:INFO: making minmax files has been done.
030816165318 ALL	:INFO: making memory.info file has been done.
030816165318 ALL	:INFO: making network.info file has been done.
030816165319 ALL	:INFO: making os.info file has been done.
030816165333 ALL	:INFO: making trend files has been done.

図 B-10. sardo.sh 性能評価時の sardo.log ファイル

次は同じく評価で採取された orado.log ファイルの内容です。

```
030815221803 AUTO :INFO: sampling interval : 1
030815221803 AUTO :INFO: sampling count : 8
030815221803 AUTO :INFO: current time : 80283 22:18:03
030815221803 AUTO :INFO: start and end time : 86105 23:55:05 86585 00:03:05
030815221803 AUTO :INFO: sampling mode : OVERTHEDAY
030815221804 AUTO :INFO: statspack can be setup and started.
030815221823 SETUP :INFO: perfstat user account was successfully created.
030815221823 AUTO :INFO: awaiting start 1 [ min ] in 5822 sec.
030815235524 AUTO :INFO: snap job will be started with 1 minutes interval.
030815235528 START :INFO: statspack spauto job was successfully scheduled.
030815235529 AUTO :INFO: wait for the end time [ 00:03:05 ] to finish.
030815235529 AUTO :INFO: remaining time is 480 seconds.
030816000330 AUTO :INFO: NEXT_SEC will be compared to the end time.
030816000333 STOP :INFO: statspack spauto job was stopped.
030816000346 REPORT :INFO: checking result sp_15Aug03_0001_0002 was passed.
030816000347 REPORT :INFO: checking result sp_15Aug03_0002_0003 was passed.
030816000349 REPORT :INFO: checking result sp_15Aug03_0003_0004 was passed.
030816000350 REPORT :INFO: checking result sp_15Aug03_0004_0005 was passed.
030816000352 REPORT :INFO: checking result sp_15Aug03_0005_0006 was passed.
030816000354 REPORT :INFO: checking result sp_16Aug03_0006_0007 was passed.
030816000356 REPORT :INFO: checking result sp_16Aug03_0007_0008 was passed.
030816000358 REPORT :INFO: checking result sp_16Aug03_0008_0009 was passed.
030816000359 REPORT :INFO: checking result sp_16Aug03_0009_0010 was passed.
030816000405 EXPORT :INFO: export perfstat was successfully completed.
030816000418 CLEAN :INFO: perfstat user was successfully cleaned up.
030816000418 AUTO :INFO: auto function was successfully completed.
030816000419 TAR :NOTE: perfstat user account hasn't been created.
030816000432 EDIT :INFO: the hostname act061 was assumed.
030816000432 EDIT :INFO: the SID ora900 was assumed.
030816000433 EDIT :INFO: spedit start.
030816001314 EDIT :INFO: PhysicalReadsT files concatenated.
030816001314 EDIT :INFO: PhysicalReadsG files concatenated.
030816001323 EDIT :INFO: spedit was successfully completed.
030816001324 TAR :NOTE: perfstat user account hasn't been created.
030816001324 TAR :INFO: the target archive SID ora900 was assumed.
```

図 B-11. sardo.sh 性能評価時の orado.log ファイル

以上のことから、sardo.sh ツールのシステムに与える影響は次のように考察することができます。

【考察】

CPU 使用は平均で 1 パーセント未満の消費のため、影響は僅少である。

ディスク入出力は、わずかな影響と判断できる。しかしながら、本測定環境では、ハードウェア構成が小さいため、別途大規模構成での負荷状況を測定する必要がある。

ネットワークは、sardo.sh 自身はネットワークを使用していないため、性能に及ぼす影響はない。

ツールのメモリ使用量につきましては付録 C に解説します。ご覧ください。

付録 C. ツールのメモリ使用量

sardo ツールのメモリ使用量を計測しました。次に、メモリ使用量を示します。本ステータスは“sardo.sh pmap”ファンクションで採取されたものです。pmap ファンクションで出力されたデータを Excel ファイルに取り込んで作図したものです。

図中、ピンクでマークされているプロセスが、sardo.sh で実行されているプロセスです。

processname	txt	rtxt	shs	rshs	shm	rshm	dsz	rdsz	hsz	rhsz	rsh	rrsh	ssz	rssz	pid
/bin/sh	88	88	872	856	0	0	64	64	96	96	0	0	16	16	27089
/bin/sh	88	88	872	856	0	0	64	64	96	96	0	0	16	16	27112
/bin/sh	88	88	872	856	0	0	64	64	96	96	0	0	16	16	27135
/bin/sh	88	88	872	856	0	0	64	64	96	96	0	0	16	16	27158
/bin/sh	88	88	872	856	0	0	64	64	96	96	0	0	16	16	27190
/bin/sh	88	88	872	856	0	0	64	64	96	96	0	0	16	16	27227
/bin/sh	88	88	872	856	0	0	64	64	96	96	0	0	16	16	27239
/etc/init	504	264	160	160	0	0	40	40	104	104	0	0	8	8	1
/opt/SUNWspci2/bin/sunpcid	32	32	1528	1352	0	0	176	168	184	184	0	0	16	16	388
/opt/caldera/bin/calserver	88	88	1480	1304	0	0	152	152	248	160	0	0	8	8	305
/opt/caldera/bin/keyserver	72	72	1480	1304	0	0	160	160	248	136	0	0	8	8	307
/sbin/dhcpagent	480	224	160	160	0	0	40	40	32	32	0	0	16	16	32
/usr/bin/fgd	56	56	1576	880	0	0	96	96	88	72	0	0	8	8	291
/usr/bin/iostat	32	32	976	952	0	0	104	104	96	96	0	0	8	8	27025
/usr/bin/mpstat	8	8	856	840	0	0	64	64	96	96	0	0	8	8	27065
/usr/bin/netstat	48	48	1504	1328	0	0	176	168	96	96	0	0	8	8	26965
/usr/bin/netstat	48	48	1504	1328	0	0	176	168	96	96	0	0	8	8	26985
/usr/bin/sleep	8	8	848	832	0	0	56	56	8	8	0	0	8	8	27210
/usr/bin/sleep	8	8	848	832	0	0	56	56	8	8	0	0	8	8	27214
/usr/bin/sleep	8	8	848	832	0	0	56	56	8	8	0	0	8	8	27218
/usr/bin/sleep	8	8	848	832	0	0	56	56	8	8	0	0	8	8	27222
/usr/bin/sleep	8	8	848	832	0	0	56	56	8	8	0	0	8	8	27226
/usr/bin/vmstat	16	16	856	840	0	0	72	72	96	96	0	0	8	8	27005
/usr/bin/vmstat	16	16	856	840	0	0	72	72	96	96	0	0	8	8	27045
/usr/dt/bin/dtlogin	120	120	4392	3152	0	0	664	632	64	64	16	16	24	24	376
/usr/dt/bin/dtlogin	120	120	4504	3264	0	0	696	656	176	168	24	24	24	24	398
/usr/lib/ab2/dweb/sunos5/bin/dwh	1048	984	7104	6656	0	0	1032	936	296	296	0	0	16	16	348
/usr/lib/ab2/dweb/sunos5/bin/dwh	1048	984	7112	6664	0	0	1048	960	960	960	0	0	40	40	349
/usr/lib/autofs/automountd	128	128	2352	2136	0	0	512	496	248	192	0	0	8	8	221
/usr/lib/dmi/dmispd	64	64	2160	1880	0	0	520	464	904	768	0	0	8	8	381
/usr/lib/dmi/snmpXdmid	144	144	2392	2112	0	0	600	536	688	568	0	0	16	16	382
/usr/lib/im/htt	8	8	1016	992	0	0	80	80	8	8	0	0	8	8	333
/usr/lib/locale/ja/atokserver/atokm	40	40	848	832	0	0	64	64	16	16	0	0	16	16	354
/usr/lib/locale/ja/wnn/dpkeyserv	32	24	1488	1312	0	0	152	144	24	24	0	0	16	16	323
/usr/lib/locale/ja/wnn/jserver	8	8	1464	1288	0	0	136	128	0	0	0	0	8	8	329
/usr/lib/locale/ja/wnn/jserver.m	384	384	1488	1312	0	0	168	160	232	216	8	8	16	16	330
/usr/lib/lpsched	184	184	1784	1560	0	0	232	216	1064	48	0	0	16	16	253
/usr/lib/nfs/lockd	24	24	1456	1280	0	0	152	152	256	160	0	0	16	16	217
/usr/lib/nfs/statd	40	40	1640	1456	0	0	344	336	472	344	0	0	16	16	218
/usr/lib/picl/picld	16	16	1512	1472	0	0	456	448	592	592	0	0	16	16	72
/usr/lib/power/powerd	24	24	1040	1016	0	0	256	240	104	104	8	8	8	8	266
/usr/lib/sa/sadc	8	8	856	840	0	0	72	72	264	264	0	0	8	8	27175
/usr/lib/saf/sac	24	24	1488	1312	0	0	152	144	104	40	0	0	16	16	391
/usr/lib/saf/ttymon	64	64	1488	1312	0	0	152	144	64	64	0	0	16	16	392

図 C-1. sardo.sh のメモリ消費量(その 1)

processname	txt	rtxt	shs	rshs	shm	rshm	dsz	rdsz	hsz	rhsz	rsh	rrsh	ssz	rssz	pid
/usr/lib/saf/ttymon	64	64	1488	1312	0	0	152	144	64	64	0	0	16	16	396
/usr/lib/sendmail	712	384	2104	1904	0	0	304	296	192	184	0	0	64	64	285
/usr/lib/snmp/snmpdx	88	88	1824	1640	0	0	272	256	144	144	0	0	16	16	373
/usr/lib/sysevent/syseventconfd	16	16	1176	1152	0	0	288	272	8	8	0	0	8	8	63
/usr/lib/sysevent/syseventd	24	24	1216	1192	0	0	440	432	32	32	104	96	16	16	61
/usr/lib/utmpd	16	16	864	848	0	0	64	64	80	56	0	0	8	8	275
/usr/openwin/bin/Xsun	1384	760	6104	4824	0	0	17152	680	3624	3488	0	0	96	96	397
/usr/openwin/bin/fbconsole	8	8	2184	1736	0	0	184	176	56	56	0	0	8	8	400
/usr/sadm/lib/smc/bin/smcboot	24	24	1488	1312	0	0	152	144	24	24	0	0	16	16	298
/usr/sadm/lib/smc/bin/smcboot	24	24	1488	1312	0	0	152	144	24	24	0	0	16	16	299
/usr/sbin/ccv	16	16	4384	3696	0	0	560	480	32	24	0	0	8	8	282
/usr/sbin/cron	40	40	1792	1608	0	0	232	224	40	40	0	0	16	16	236
/usr/sbin/cs00	120	120	1648	1448	0	0	200	200	872	336	0	0	8	8	284
/usr/sbin/cssd	16	16	848	832	0	0	56	56	16	16	0	0	16	16	281
/usr/sbin/inetd	32	32	1568	1392	0	0	176	176	648	632	0	0	16	16	211
/usr/sbin/kkcv	96	96	4384	3696	0	0	576	496	312	296	0	0	8	8	283
/usr/sbin/nscd	56	56	1864	1672	0	0	744	736	368	368	0	0	8	8	247
/usr/sbin/rpcbind	40	40	1472	1296	0	0	152	152	536	312	0	0	16	16	153
/usr/sbin/syslogd	64	64	2056	1856	0	0	1584	592	496	488	0	0	32	32	229
/usr/sbin/vold	144	144	1968	1696	0	0	400	392	168	96	800	520	16	16	317
0csh	144	144	1192	1128	0	0	144	136	104	88	0	0	24	24	418
0csh	144	144	1192	1128	0	0	144	136	104	88	0	0	24	24	476
0sh	272	272	0	0	0	0	16	16	24	24	0	0	8	8	496
0sh	272	272	0	0	0	0	16	16	24	24	0	0	8	8	500
devfsadmd	64	64	2024	1784	0	0	480	456	528	528	96	72	16	16	73
dtgreet	64	64	6592	4472	0	0	832	800	680	680	240	208	144	144	415
htt_server	168	168	2640	2176	0	0	456	424	16	16	0	0	8	8	335
in.telnetd	24	24	1576	1400	0	0	176	168	16	16	0	0	16	16	416
in.telnetd	24	24	1576	1400	0	0	176	168	16	16	0	0	16	16	444
in.telnetd	24	24	1576	1400	0	0	176	168	16	16	0	0	16	16	474
in.telnetd	24	24	1576	1400	0	0	176	168	16	16	0	0	16	16	481
mibiisa	80	80	1784	1592	0	0	496	480	192	192	0	0	16	16	414
ora_cjq0_ora900	37176	19064	5848	4528	246296	246296	680	664	360	240	0	0	40	40	464
ora_ckpt_ora900	37176	19064	5848	4528	246296	246296	888	872	1552	1440	0	0	40	40	458
ora_d000_ora900	37176	19064	5848	4528	246296	246296	680	664	704	584	0	0	24	24	468
ora_d001_ora900	37176	19064	5848	4528	246296	246296	680	664	704	584	0	0	24	24	470
ora_d002_ora900	37176	19064	5848	4528	246296	246296	680	664	704	584	0	0	24	24	472
ora_dbw0_ora900	37176	19064	5848	4528	246296	246296	928	912	2856	2744	0	0	24	24	454
ora_lgwr_ora900	37176	19064	5848	4528	246296	246296	888	872	5792	1648	0	0	24	24	456
ora_pmon_ora900	37176	19064	5848	4528	246296	246296	680	664	944	848	0	0	24	24	452
ora_reco_ora900	37176	19064	5848	4528	246296	246296	680	664	360	240	0	0	40	40	462
ora_s000_ora900	37176	19064	5848	4528	246296	246296	680	664	848	728	0	0	24	24	466
ora_smon_ora900	37176	19064	5848	4528	246296	246296	680	664	464	344	0	0	64	64	460
		104						1320		1360				176	2960

図 C-2. sardo.sh のメモリ消費量(その 2)合計

以上のことから、次のように考察することができます。

【考察】

メモリは約 3 メガバイトと計算されました。測定対象が 512MB の機器でも 1 パーセントを切るサイズのため、影響はないと考えます。

次にサンプリングインターバルの決定に関する情報を解説します。

付録 D. サンプリングタイムの違いによる測定値の変化

sar(1)やvmstat(1M)コマンドによって採取された性能情報は、採取間隔(サンプリングインターバル)によって表示される値が変化する項目があります。sar(1)コマンドの-v オプションで表示されるプロセス数を表す proc-sz や、ファイル数を表す file-sz 等は、サンプリング時点の値を表示しており、サンプリングインターバルの違いによる表示の差はありません。

一方、sar(1)コマンドの-u オプションで表示されるシステムの CPU 消費割合%sys やユーザ CPU の消費割合%user、sar(1)コマンドの-d オプションで表示されるディスク装置の待ち%busy など、秒あたりの値を表示する項目では、違いが現れます。

sar(1)や vmstat(1M)等のコマンドは、カーネルの提供する sysinfo 構造体からある時点のデータを得ます。次のサンプリング時点に更新されたデータを得て差分を計算します。

本章では、実際の計測で得られたデータを基に、秒あたりで表示される値がどのように変化するのかを考察します。

【実測データ】

sar(1)コマンド-u オプションの実測データ

図 D-1～D-4 に sar(1)コマンド-u オプションの 5 秒、10 秒、60 秒、および 600 秒(10 分)で採取したデータを示します。

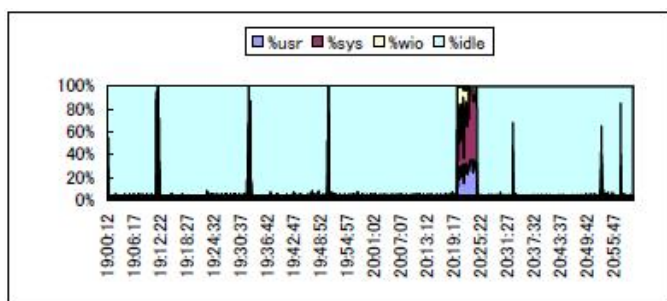


図 D-1. sar(1)-u 5 秒間隔のグラフ

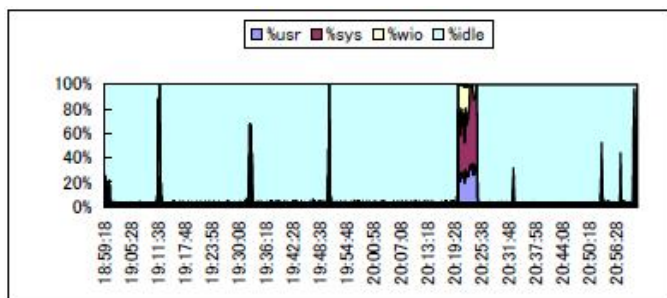


図 D-2. sar(1)-u 10 秒間隔のグラフ

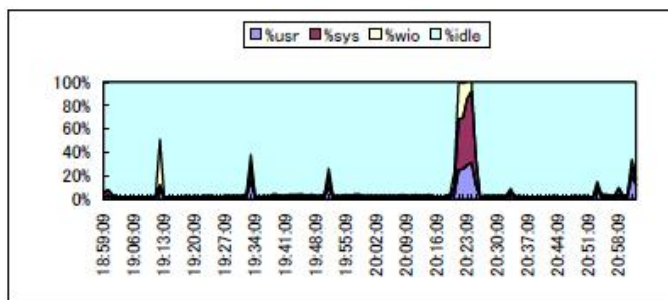


図 D-3. sar(1)-u 60 秒(1 分)間隔のグラフ

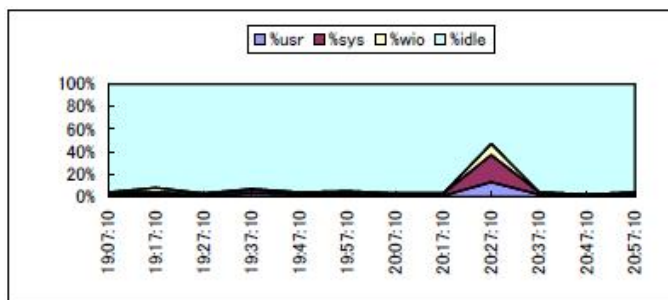


図 D-4. sar(1)-u 600 秒(10 分)間隔のグラフ

sar(1)コマンド-d オプションの実測データ

図 D-5～D-8 に sar(1)コマンド-d オプションの 5 秒、10 秒、60 秒、および 600 秒(10 分)で採取したデータを示します。

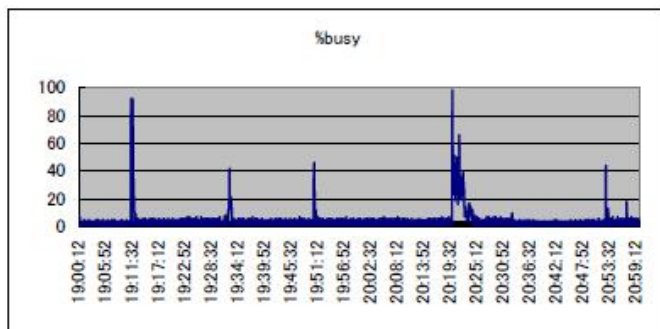


図 D-5. sar(1)-d 5 秒間隔のグラフ

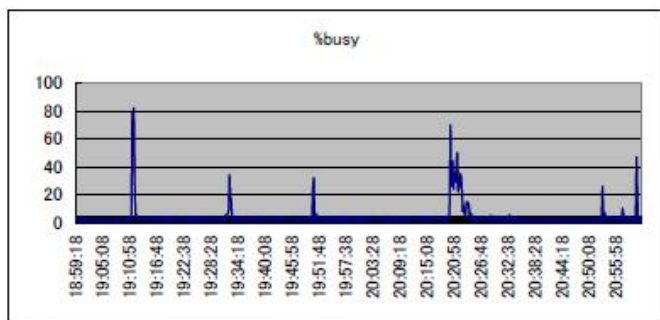


図 D-6. sar(1)-d 10 秒間隔のグラフ

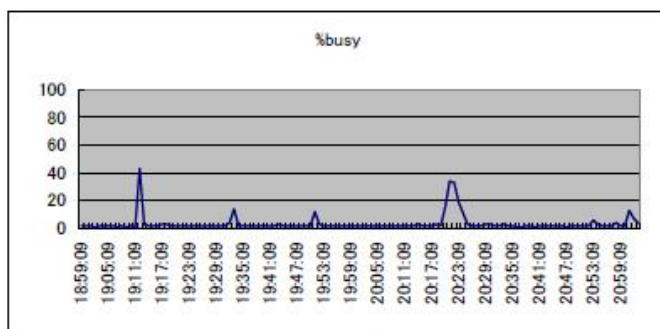


図 D-7. sar(1)-d 60 秒(1 分)間隔のグラフ

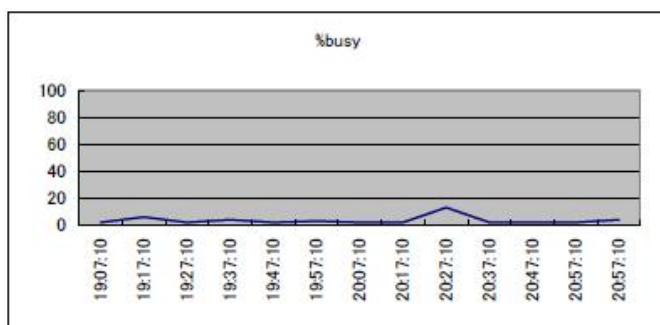


図 D-8. sar(1)-d 600 秒(10 分)間隔のグラフ

これら実測データで、サンプリングインターバルが長くなるにつれて値が小さくなっていることがあります。平均値を計算しているためにそのようになります。

以上のことから、次のように考察することができます。

【考察】

サンプリングインターバルは、測定目的のシステム処理に応じて適切に決定する必要がある。

sardoを使用すると、sadc(1M)は 5 秒のサンプリングインターバルを使用しているので正確。

性能監視ソフトウェアの**しきい値管理**はこの点に十分注意すべきである【備考】。

【備考】 数多くある性能監視ソフトウェアは、システム負荷を避けるためにサンプリングインターバルを長くしてトラップを生成していることがほとんどです。ここに落とし穴があり、実態を表していないデータを使用していることがあります。また、様々な sysinfo 上の値を得る時、通常のコマンドを使用するために余分なオーバーヘッドを消費しています(例えば CPU ロードアベレージを得る際、uptime(1)や w(1)コマンドを使用しています。これらは、C ライブラリを用いた実行プログラムを作成することで、大幅に負荷を低減することが可能です)。

参考文献

Unix システムパフォーマンスチューニング第 2 版 Gian-Paolo D.Musumeci、Mike Loukides 著 砂原秀樹監訳 岡島順治郎、高橋敏明訳 株式会社オライリー・ジャパン(原本)System Performance Tuning 2nd Edition Help for Unix System Administrators O'REILLY Gian-Paolo D.Musumeci & Mike Loukides February 2002.

ドキュメント更新履歴

日付	レビジョン	内容
2003 年 10 月 28 日	Rev.1.0	初版
2003 年 12 月 29 日	Rev.1.0a	エディトリアルバグの改訂。
2004 年 02 月 04 日	Rev.1.0b	pacct.info、disk.info 編集機能の追加。
2004 年 03 月 18 日	Rev.1.0c	pacct のフラグ追記。
2010 年 04 月 30 日	Rev.1.2b	グラファーの入れ替え、および全面改訂。
2010 年 12 月 10 日	Rev.1.2d	6-4 & 6-5 節の追加、および部分改訂。
2014 年 01 月 28 日	Rev.1.2f	P15 表 2-4-4-1 No.2 pmap(1) -k → pmap(1)-x P53 表4-2-1 No.4 netstat(1M) →netstat(1M) -k P101 図B-10 sardo.log を図11 orado.logに分割 事務所住所変更。

連絡先

株式会社 エイ・シー・テイ

〒170-0013 東京都豊島区東池袋 3-15-14 クレール東池袋 308

TEL:03-5953-4815 FAX:03-5953-4751

E-Mail: n-jun@actcrp.com

The Art of Computer Technologies, Corporation.

Claire Higashi-ikebukuro 308, 3-15-14

Higashi-Ikebukuro Toshima-ku, Tokyo

170-0013 Japan