

【 sar コマンドと GlancePlus の違いについて 】

以下、HP-UX の sar コマンドと GlancePlus の違いを説明します。

●sar (1M) 情報取得方法について

実行時のオプションで kmem 内のどこを見るかが決定されます。

以下は、“-u”オプションを想定した説明になります。

- (1) /dev/kmem を開く
- (2) CPU 関連のテーブルを探す
- (3) テーブルの先頭と（もしくはそこからそう遠くない）各 CPU 動作情報を示すポインタ先を記録 *1
- (4) Kernel は、10 ミリ秒毎にポインタ先を更新（注意：更新時点での User/Sys/WIO/IdeI を加算）
- (5) sar (1M) は、（デフォルト）5 秒毎にポインタ先を自分の領域にコピー
- (6) 全体から割合を演算
- (7) 表示（sar -u ... を叩いた後に表示される値に相当）

*1：個々の CPU 毎に存在し、10 ミリ秒毎の更新時点での状態を保存しているに過ぎません。
すなわち、10 ミリ秒以内での変化は記録されません。

●GlancePlus 情報取得方法について

1. 性能情報を取得する仕組み

- (1) MI (Mesurement Interface) が実行毎に Kernel 内の管理テーブルを走査
（Kernel 内の管理テーブル情報を単純に自分の領域にコピーします。）
その結果を midaemon プロセスが読み取り可能な共有メモリ領域に書き込み
- (2) midaemon は、何を記録するか の定義ファイル（/var/opt/perf/parm）に基づき、読み出し値もしくは
直前の値との XOR 値を scopeux/oacore プロセスに Message 機能を用いて送信する
- (3) scopeux/oacore プロセスは、バイナリ値のままファイル(*2)に格納する

*2:

scopeux : /var/opt/perf/datafiles/log* ファイル
oacore : /var/opt/OV/databases/oa/*.db ファイル

2. GLB_*と PROC_*の計測値の違いについて

GLB_*メトリックは、予め取得元の場所が決められているため走査時のオーバーヘッドは少なく済みます。
一方、PRO_*やAPP_*などのメトリックは、個々のプロセス毎のテーブルを全て走査して該当するものを
取得するため、オーバーヘッドが高くなります。

（サーバ負荷状況により取得に漏れが発生する場合も考えられます。）

また、計算方法の面から言いますと、GLB_*と PROC_*（APP_* も含む）は計算方法が異なります。
これは、それぞれのメトリックの値の取り方が違うためであり、製品の仕様です。

GBL_*メトリックは、カーネルから直接（上記項番“1.”の仕組みで）データを取得します。
この場合のデータ更新周期は1／100秒（10ミリ秒）になります。

PROC_*やAPP_*メトリックは、そのアプリケーションに属するプロセスの情報を集計した値を取りますが、
プロセスの情報は、pstat()経由で作成されます。
この場合、データの更新周期は1／10秒（0.1秒）となります。

sar -u コマンドの結果と類似する値を提供する GlancePlus のメトリックは以下の GBL_*になります。

sar : "%usr" ≡ GBL_CPU_USER_MODE_UTIL

sar : "%system" ≡ GBL_CPU_SYS_MODE_UTIL

sar : "%total" ≡ GBL_CPU_TOTAL_UTIL

（≡は、近似値であり、同じ値を示すことを意味しているわけではありません。）

以上