sarコマンドと Glance の違い

■ 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 -u オプションを想定。）

(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。 -（★）
(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/Idel 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）

（★）部分は、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。

■ glance 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 (Mesuarement 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 プロセスに Message 機能を用いて送信する。
(3) scopeux プロセスはバイナリ値のままファイル(/var/opt/perf/datafiles/log*)に格納する。

2. GLB_＊ とPROC_＊ の計測値の違いについて

GBL_＊ メトリックで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ヘッドは少なくて済みます。
一方、PRO_＊、APP_＊ 等のメトリックは、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、個々のプロセスのテーブルを走査することは、オーバヘッドが高くなります。
（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うならば、GBL_＊ と PROC_＊（APP_＊ も含む）は計算方法が異なります。
（それぞれのメトリックで値の取り方が違うためで、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_＊ および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 の結果と類似する値を提供する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
（≒　：　近似値であり、同じを意味しているわけではありません。）