Ultimate tool for monitoring, SAR is available in sysstat package.

Some tips I use:

1. Convert from **binary** to text (to import on kSar for example)

```
/usr/bin/env LC_TIME=POSIX sar -A -f saXX > saXX.txt
```

   ! binary format is not supported anymore... you will need an old system/VM to do so.

   The POSIX time is to avoid AM/PM in the txt file and avoid kSar error:

```
 java.lang.NumberFormatException: For input string: "all"
        at
sun.misc.FloatingDecimal.readJavaFormatString(FloatingDecimal.java:1250
)
        at java.lang.Float.valueOf(Float.java:417)
        at java.lang.Float.<init>(Float.java:519)
        at net.atomique.ksar.Linux.Parser.parse(Parser.java:624)
        at net.atomique.ksar.kSar.parse(kSar.java:750)
        at net.atomique.ksar.FileRead.run(FileRead.java:62)
```

2. Time convert, if kSar complains about time stamps:

```
java.lang.NumberFormatException: For input string: "all"
        at
sun.misc.FloatingDecimal.readJavaFormatString(FloatingDecimal.java:1250
)
        at java.lang.Float.valueOf(Float.java:417)
        at java.lang.Float.<init>(Float.java:519)
        at net.atomique.ksar.Linux.Parser.parse(Parser.java:624)
        at net.atomique.ksar.kSar.parse(kSar.java:750)
        at net.atomique.ksar.FileRead.run(FileRead.java:62)
```

   You need to convert the file with POSIX time stamps using Perl:

```
#!/usr/bin/env perl

use strict;
use warnings;

use Time::Piece;

while (<>) {
    if ( my ($time_str) = m/^([\d\:]+ [AP]M)/ ) {
        my $new_time_str = Time::Piece->strptime( $time_str, "%I:%M:%S
%p" )
            ->strftime("%H:%M:%S");
        s/$time_str/$new_time_str/;
    }
    print;
```

```
}
```

Reference: http://unix.stackexchange.com/questions/227991/change-time-format-in-a-file

From:
https://wiki.fortier-family.com/ - **Warnaud's Wiki**

Permanent link:
https://wiki.fortier-family.com/software/cli/sar

Last update: **2021/12/29 21:09**