NAME

cfilt - A filter for Collect

SYNOPSIS

usr/sbin/cfilt [-aN] [-f [*input-file*] [*expression*] [*expression* ...]] [-p] [-u]

OPTIONS

`-a [`*`number`*`]`

Average values for *N* (number of samples).

`-f`

Selects the input data file and the expressions used to operate on that data. If this option is not used, cfilt gets input from stdin.

`-p`

Selects only the samples that contain process data. This is useful when a separate process interval was given to Collect, such `-i1,4`, but you want to graph process data against some non-process data, such as cpu idle.

`-u`

Unbuffered mode as `-u` for the `cat` command: forces a flush for each line. Used by the `-live` option of `collgui`.

DESCRIPTION

The `cfilt` utility allows the arbitrary selection of values from the output of Collect. It condenses the output of Collect into one line per sample, or per *N* samples, if using the `-a` option to average *N* samples. The data in this form can be graphed using `gnuplot` or Excel.

`cfilt` can also be used *live*, that is, as a filter to Collect while it's collecting and writing to standard output. This only works if no normalization is being

done, as that requires that all samples be seen so that `cfilt` can determine the highest value, which is then used to normalize.

The first two columns in cfilt's output are always the *epoch-second* and sample-number. The epoch-second is the internal UNIX time format, the number of seconds since the beginning of the *epoch*, January 1st, 1970. This is extracted directly from the Collect output: at the beginning of each record there is a line similar to the following:

```
### RECORD 1 (873230968:160) (Tue Dec 2 22:09:28 2000) ###
```

In this example, epoch-seconds is 873230968. The sample-number is also extracted from this line. In this example it is 1.

EXPRESSIONS

An expression has the following syntax:

```
subsystem:selection-criterion:tag-expr1:tag-expr2:...:tag-exprN
```

Subsystem

The following comprise the subsystems `cfilt recognizes:` *proc*, *disk*, *mem*, *net*, *cpu*, *sin*, *file*, *tty*, and *lsm* (first 3 chars are significant) A subsystem can be one of: *proc*, *disk*, *mem*, *net*, *cpu*, *sin*, *file*, *tty*, and *lsm* (first 3 characters are significant).

If a plus-sign (+) is on the end, or no selection criterion has been given, then numerical values are summed for all lines of a subsystem. If a selection criterion has been provided, and there is no plus sign on the end of the subsystem name, then for each value in the selection criterion, the corresponding values for each <tag-expr> will be printed. For example, given the following output from Collect:

DISK Statistics

#DSK	NAME	B/T/L	R/S	RKB/S	W/S	AVS	QLEN	%BUSY
0	rz1	0/1/0	5	300	10	10	0	70
1	rz2	0/2/0	7	400	11	10	0	80
2	rz3	0/3/0	9	500	12	10	0	90

Assuming that cfilt is called with the single following expression,

disk:r/s

cfilt would sum reads/second for all disks. That is, 5+7+9=21. The output of cfilt would be:

<epoch-seconds> <sample#> 21

The expression disk+:name=rz1,rz2:r/s would sum reads/second for disks rz1 and rz2, 5+7=12. (name=rz1,rz2 is a selection-criterion, which is discussed below.) The output of cfilt would be:

<epoch-seconds> <sample#> 12

The expression disk+:name=rz1,rz2:rkb/s+wkb/s would sum KiloBytes read and written for disks rz1 and rz2, 300+400+1000+2000=3700, as follows (rkb/s+wkb/s is a tag-expression, which is discussed below.):

<epoch-seconds> <sample#> 3700

The expression disk:name=rz1,rz2:r/s would print reads/second for rz1 and reads/second for rz2, as follows:

<epoch-seconds> <sample#> 5 7

RESTRICTIONS

The following restriction applies when using `cfilt` in this release:

When normalizing and averaging are used, the highest normalized value will not necessarily be as high as the value used for normalizing.

EXAMPLES

1. This command provides a CPU summary:

   # **cfilt -fdata.in cpu:user+sys:intr#:sysc#:cs#**

   The command generates the following output:

   `[time] [sample#] [user+sys] [interrupts] [syscalls] [conswitch]` (where interrupts, syscalls, and conswitch are normalized).

2. This command provides a system overview:

   # **cfilt -fdata.in cpu:idle net:inpck+outpck# mem:free#**

   The command generates the following output:

   ```
   [time] [sample#] [cpu:idle] [net:inpck=outpck (normalized)] [mem:free (normalized)]
   ```

3. This command provides user process information:

   # **cfilt -fdata.in proc+:user=smith:rss**

   The command generates the following output:

   ```
   [time] [sample#] [rss (resident set size) for all pro-
   cesses owned by smith]
   ```

4. This command provides user process information:

   # **cfilt -fdata.in pro:pid=1234,8888:rss:vsz**

   The command generates the following output:

   ```
   [time] [sample#] [rss(pid=1234)] [vsz (pid=1234)]
   [rss(pid=8888)] [vsz(pid=8888)]
   ```

5. The command provides process information:

   # **cfilt -fdata.in pro+:pid=1234,8888:rss:vsz**

   ```
   [time] [sample#] [rss(sum for pid 1234, 8888)] [vsz
   (sum)]
   ```

6. The command provides process information:

   # **cfilt -fdata.in pro+:rss:vsz**

   ```
   [time] [sample#] [rss(sum all procs)] [vsz (sum for all
   procs)]
   ```

FILES

`/usr/sbin/cfilt`
The executable image.

SEE ALSO

Commands: `collect`(8), `collgui`(8)

Manuals: *System Configuration and Tuning Guide*, *System Administration Guide*