perf Linux profiler

CONCEPTS

perf stat counts events, perf record samples them: count first to see what is off, then sample to find where.

perf record samples events
↓ writes
perf.data sample file
↓ read by
report / script annotate, diff
↓ shows
Hot spots functions, stacks

Call graphs need frame pointers by default (-g). If stacks look bogus, use --call-graph dwarf.

On permission errors or missing kernel data, check kernel.perf_event_paranoid and try sudo.

Without a command perf stat and record measure all CPUs; end with Ctrl+C or append -- sleep <secs>.

COUNTING

perf stat *<cmd>*
Count default events for a command

perf stat -r *<n>* *<cmd>*
Repeat, show mean and stddev

perf stat -e *<ev>*,*<ev>* *<cmd>*
Count chosen events

perf stat -p *<pid>* **-- sleep** *<secs>*
Count a running process for a while

perf stat -a -- sleep *<secs>*
Count system-wide

perf stat -d *<cmd>*
Add cache and more detailed events

perf stat -I *<ms>* **-a**
Print counts every interval

perf stat -C *<cpus>* **-a -- sleep** *<secs>*
Count only chosen CPUs

perf stat -x, -o *<file>* *<cmd>*
CSV output to a file

COUNTING (cont.)

perf stat -e cycles:u *<cmd>*
Event modifier: user space only

perf stat -M *<metric>* *<cmd>*
Show a named metric

SAMPLING

perf record -- *<cmd>*
Sample a command into perf.data

perf record -g -- *<cmd>*
Sample with call stacks

perf record -F *<hz>* **--** *<cmd>*
Set the sampling frequency

perf record -a -g -- sleep *<secs>*
Sample all CPUs for a while

perf record -g -p *<pid>*
Attach to a running process (Ctrl+C ends)

perf record --call-graph dwarf -- *<cmd>*
Stacks without frame pointers

perf record -e *<ev>* **--** *<cmd>*
Sample on another event

perf record -o *<file>* **--** *<cmd>*
Write to a named file

perf record -C *<cpus>* **-- sleep** *<secs>*
Sample only some CPUs

perf record -D *<ms>* **--** *<cmd>*
Delay start to skip warm-up

REPORTING

perf report
Browse samples (TUI)

perf report --stdio
Plain text output

perf report --no-children
Self cost only, not inclusive

perf report -n
Show sample counts

perf report -i *<file>*
Read another data file

perf report -s *<keys>*
Group by comm, dso, symbol, cpu

perf report --percent-limit *<pct>*
Hide small entries

REPORTING (cont.)

perf report -G
Inverted call graph

perf report --comms *<name>* **--dsos** *<lib>*
Filter by process or library

perf report --symbols *<sym>*
Filter by function

perf report -g caller
Call tree from callers downward

perf annotate *<symbol>*
Per-instruction samples

perf script
Dump raw samples as text

LIVE VIEW

perf top
Live hottest functions

perf top -p *<pid>*
Only one process

perf top -e *<event>*
Another event

perf top -g
Collect call graphs

perf top -K
Hide kernel symbols

s / **S**
In top: annotate / stop annotating

q In top: quit

EVENTS

perf list
List available events

perf list sw
Software events only

perf list hw cache
Hardware and cache events

perf list tracepoint
Kernel tracepoints

perf list '*<glob>***'**
Filter events by pattern

sudo sysctl kernel.perf_event_paranoid=-1
Allow unprivileged events

OTHER SUBCOMMANDS

perf trace *<cmd>*
strace-like syscall trace