



Linux Ubuntu 26.04 Manual Pages on command 'perf-kvm.1'

\$ man perf-kvm.1

PERF-KVM(1) perf Manual PERF-KVM(1)

NAME

perf-kvm - Tool to trace/measure kvm guest os

SYNOPSIS

```
perf kvm [--host] [--guest] [--guestmount=<path>
          [--guestkallsyms=<path> --guestmodules=<path> | --guestvmlinux=<path>]]
          {top|record|report|diff|buildid-list} [<options>]
perf kvm [--host] [--guest] [--guestkallsyms=<path> --guestmodules=<path>
          | --guestvmlinux=<path>] {top|record|report|diff|buildid-list|stat} [<options>]
'perf kvm stat [record|report|live] [<options>]
```

DESCRIPTION

There are a couple of variants of perf kvm:

'perf kvm [options] top <command>' to generates and displays
a performance counter profile of guest os in realtime
of an arbitrary workload.

'perf kvm record <command>' to record the performance counter profile
of an arbitrary workload and save it into a perf data file. We set the
default behavior of perf kvm as --guest, so if neither --host nor --guest
is input, the perf data file name is perf.data.guest. If --host is input,
the perf data file name is perf.data.kvm. If you want to record data into
perf.data.host, please input --host --no-guest. The behaviors are shown as
following:

Default("") -> perf.data.guest

--host -> perf.data.kvm
--guest -> perf.data.guest
--host --guest -> perf.data.kvm
--host --no-guest -> perf.data.host

'perf kvm report' to display the performance counter profile information recorded via perf kvm record.

'perf kvm diff' to displays the performance difference amongst two perf.data files captured via perf record.

'perf kvm buildid-list' to display the buildids found in a perf data file, so that other tools can be used to fetch packages with matching symbol tables for use by perf report. As buildid is read from /sys/kernel/notes in os, then if you want to list the buildid for guest, please make sure your perf data file was captured with --guestmount in perf kvm record.

'perf kvm stat <command>' to run a command and gather performance counter statistics.

Especially, perf 'kvm stat record/report' generates a statistical analysis of KVM events. Currently, vmexit, mmio (x86 only) and ioport (x86 only) events are supported. 'perf kvm stat record <command>' records kvm events and the events between start and end <command>.

And this command produces a file which contains tracing results of kvm events.

'perf kvm stat report' reports statistical data which includes events handled sample, percent_sample, time, percent_time, max_t, min_t, mean_t.

'perf kvm stat live' reports statistical data in a live mode (similar to record + report but with statistical data updated live at a given display rate).

OPTIONS

-i, --input=<path>

Input file name, for the report, diff and buildid-list subcommands.

-o, --output=<path>

Output file name, for the record subcommand. Doesn't work with report, just redirect the output to a file when using report.

--host

Collect host side performance profile.

--guest

Collect guest side performance profile.

--guestmount=<path>

Guest OS root file system mount directory. Users mount guest OS root directories under <path> by a specific filesystem access method, typically, sshfs. For example, start 2 guest OS, one's pid is 8888 and the other's is 9999:

```
$ mkdir ~/guestmount
```

```
$ cd ~/guestmount
```

```
$ sshfs -o allow_other,direct_io -p 5551 localhost:/ 8888/
```

```
$ sshfs -o allow_other,direct_io -p 5552 localhost:/ 9999/
```

```
$ perf kvm --host --guest --guestmount=~ /guestmount top
```

--guestkallsyms=<path>

Guest OS /proc/kallsyms file copy. perf reads it to get guest kernel symbols. Users copy it out from guest OS.

--guestmodules=<path>

Guest OS /proc/modules file copy. perf reads it to get guest kernel module information.

Users copy it out from guest OS.

--guestvmlinux=<path>

Guest OS kernel vmlinux.

--guest-code

Indicate that guest code can be found in the hypervisor process, which is a common case for KVM test programs.

--stdio

Use the stdio interface.

-v, --verbose

Be more verbose (show counter open errors, etc).

STAT REPORT OPTIONS

--vcpu=<value>

analyze events which occur on this vcpu. (default: all vcpus)

--event=<value>

event to be analyzed. Possible values: vmexit, mmio (x86 only), ioport (x86 only).

(default: vmexit)

-k, --key=<value>

Sorting key. Possible values: sample (default, sort by samples number), percent_sample (sort by sample percentage), time (sort by average time), percent_time (sort by time percentage), max_t (sort by maximum time), min_t (sort by minimum time), mean_t (sort by mean time).

-p, --pid=

Analyze events only for given process ID(s) (comma separated list).

STAT LIVE OPTIONS

-d, --display

Time in seconds between display updates

-m, --mmap-pages=

Number of mmap data pages (must be a power of two) or size specification in bytes with appended unit character - B/K/M/G. The size is rounded up to the nearest power-of-two page value.

-a, --all-cpus

System-wide collection from all CPUs.

-p, --pid=

Analyze events only for given process ID(s) (comma separated list).

--vcpu=<value>

analyze events which occur on this vcpu. (default: all vcpus)

--event=<value>

event to be analyzed. Possible values: vmexit, mmio (x86 only), ioport (x86 only).
(default: vmexit)

-k, --key=<value>

Sorting key. Possible values: sample (default, sort by samples number), time (sort by average time).

--duration=<value>

Show events other than HLT (x86 only) or Wait state (s390 only) that take longer than duration usecs.

--proc-map-timeout

When processing pre-existing threads /proc/XXX/mmap, it may take a long time, because the file may be huge. A time out is needed in such cases. This option sets the time out limit. The default value is 500 ms.

SEE ALSO

perf-top(1), perf-record(1), perf-report(1), perf-diff(1), perf-buildid-list(1), perf-stat(1)

perf

05/25/2026

PERF-KVM(1)