



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

\$ man perf-inject.1

PERF-INJECT(1) perf Manual PERF-INJECT(1)

NAME

perf-inject - Filter to augment the events stream with additional information

SYNOPSIS

perf inject <options>

DESCRIPTION

perf-inject reads a perf-record event stream and repipes it to stdout. At any point the processing code can inject other events into the event stream - in this case build-ids (-b option) are read and injected as needed into the event stream.

Build-ids are just the first user of perf-inject - potentially anything that needs userspace processing to augment the events stream with additional information could make use of this facility.

OPTIONS

-b, --build-ids

Inject build-ids of DSOs hit by samples into the output stream. This means it needs to process all SAMPLE records to find the DSOs.

--buildid-all

Inject build-ids of all DSOs into the output stream regardless of hits and skip SAMPLE processing.

--known-build-ids=

Override build-ids to inject using these comma-separated pairs of build-id and path.

Understands file://filename to read these pairs from a file, which can be generated with

perf buildid-list.

-v, --verbose

Be more verbose.

-i, --input=

Input file name. (default: stdin)

-o, --output=

Output file name. (default: stdout)

-s, --sched-stat

Merge sched_stat and sched_switch for getting events where and how long tasks slept.

sched_switch contains a callchain where a task slept and sched_stat contains a timeslice

how long a task slept.

-k, --vmlinux=<file>

vmlinux pathname

--ignore-vmlinux

Ignore vmlinux files.

--kallsyms=<file>

kallsyms pathname

--itrace

Decode Instruction Tracing data, replacing it with synthesized events. Options are:

- i synthesize instructions events
- y synthesize cycles events
- b synthesize branches events
- c synthesize branches events (calls only)
- r synthesize branches events (returns only)
- x synthesize transactions events
- w synthesize ptwrite events
- p synthesize power events (incl. PSB events for Intel PT)
- o synthesize other events recorded due to the use
 of aux-output (refer to perf record)
- I synthesize interrupt or similar (asynchronous) events
 (e.g. Intel PT Event Trace)
- e synthesize error events
- d create a debug log
- f synthesize first level cache events

- m synthesize last level cache events
- M synthesize memory events
- t synthesize TLB events
- a synthesize remote access events
- g synthesize a call chain (use with i or x)
- G synthesize a call chain on existing event records
- l synthesize last branch entries (use with i or x)
- L synthesize last branch entries on existing event records
- s skip initial number of events
- q quicker (less detailed) decoding
- A approximate IPC
- Z prefer to ignore timestamps (so-called "timeless" decoding)
- T use the timestamp trace as kernel time

The default is all events i.e. the same as `--itrace=iybxwpe`,
except for perf script where it is `--itrace=ce`

In addition, the period (default 100000, except for perf script where it is 1)
for instructions events can be specified in units of:

- i instructions
- t ticks
- ms milliseconds
- us microseconds
- ns nanoseconds (default)

Also the call chain size (default 16, max. 1024) for instructions or
transactions events can be specified.

Also the number of last branch entries (default 64, max. 1024) for
instructions or transactions events can be specified.

Similar to options g and l, size may also be specified for options G and L.

On x86, note that G and L work poorly when data has been recorded with
large PEBS. Refer `linkperf:perf-intel-pt[1]` man page for details.

It is also possible to skip events generated (instructions, branches, transactions,
ptwrite, power) at the beginning. This is useful to ignore initialization code.

`--itrace=i0nss1000000`

skips the first million instructions.

The 'e' option may be followed by flags which affect what errors will or will not be reported. Each flag must be preceded by either '+' or '-'.

The flags are:

- o overflow
- l trace data lost

If supported, the 'd' option may be followed by flags which affect what debug messages will or will not be logged. Each flag must be preceded by either '+' or '-'. The flags are:

- a all perf events
- e output only on errors (size configurable - see linkperf:perf-config[1])
- o output to stdout

If supported, the 'q' option may be repeated to increase the effect.

--strip

Use with --itrace to strip out non-synthesized events.

-j, --jit

Process jitdump files by injecting the mmap records corresponding to jitted functions. This option also generates the ELF images for each jitted function found in the jitdumps files captured in the input perf.data file. Use this option if you are monitoring environment using JIT runtimes, such as Java, DART or V8.

-f, --force

Don't complain, do it.

--vm-time-correlation[=OPTIONS]

Some architectures may capture AUX area data which contains timestamps affected by virtualization. This option will update those timestamps in place, to correlate with host timestamps. The in-place update means that an output file is not specified, and instead the input file is modified. The options are architecture specific, except that they may start with "dry-run" which will cause the file to be processed but without updating it. Currently this option is supported only by Intel PT, refer perf-intel-pt(1)

--guest-data=<path>,<pid>[,<time offset>[,<time scale>]]

Insert events from a perf.data file recorded in a virtual machine at the same time as the input perf.data file was recorded on the host. The Process ID (PID) of the QEMU hypervisor process must be provided, and the time offset and time scale (multiplier) will likely be needed to convert guest time stamps into host time stamps. For example,

for x86 the TSC Offset and Multiplier could be provided for a virtual machine using Linux command line option no-kvmclock. Currently only mmap, mmap2, comm, task, context_switch, ksymbol, and text_poke events are inserted, as well as build ID information. The QEMU option -name debug-threads=on is needed so that thread names can be used to determine which thread is running which VCPU. Note libvirt seems to use this by default. When using perf record in the guest, option --sample-identifier should be used, and also --buildid-all and --switch-events may be useful.

--convert-callchain

Parse DWARF callchains and convert them to usual callchains. This also discards stack and register data from the samples. This will lose inlined callchain entries.

--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 inject --guestmount=~/.guestmount
```

SEE ALSO

perf-record(1), perf-report(1), perf-archive(1), perf-intel-pt(1)

perf

05/25/2026

PERF-INJECT(1)