



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

\$ man perf-mem.1

PERF-MEM(1) perf Manual PERF-MEM(1)

NAME

perf-mem - Profile memory accesses

SYNOPSIS

perf mem [<options>] (record [<command>] | report)

DESCRIPTION

"perf mem record" runs a command and gathers memory operation data from it, into perf.data.

Perf record options are accepted and are passed through.

"perf mem report" displays the result. It invokes perf report with the right set of options

to display a memory access profile. By default, loads and stores are sampled. Use the -t

option to limit to loads or stores.

Note that on Intel systems the memory latency reported is the use-latency, not the pure load

(or store latency). Use latency includes any pipeline queuing delays in addition to the

memory subsystem latency.

On Arm64 this uses SPE to sample load and store operations, therefore hardware and kernel

support is required. See perf-arm-spe(1) for a setup guide. Due to the statistical nature of

SPE sampling, not every memory operation will be sampled.

On AMD this use IBS Op PMU to sample load-store operations.

COMMON OPTIONS

-f, --force

Don't do ownership validation

-t, --type=<type>

Select the memory operation type: load or store (default: load,store)

-v, --verbose

Be more verbose (show counter open errors, etc)

-p, --phys-data

Record/Report sample physical addresses

--data-page-size

Record/Report sample data address page size

RECORD OPTIONS

<command>...

Any command you can specify in a shell.

-e, --event <event>

Event selector. Use perf mem record -e list to list available events.

-K, --all-kernel

Configure all used events to run in kernel space.

-U, --all-user

Configure all used events to run in user space.

--ldlat <n>

Specify desired latency for loads event. Supported on Intel, Arm64 and some AMD processors. Ignored on other archs.

On supported AMD processors:

- /sys/bus/event_source/devices/ibs_op/caps/ldlat file contains '1'.
- Supported latency values are 128 to 2048 (both inclusive).
- Latency value which is a multiple of 128 incurs a little less profiling overhead compared to other values.
- Load latency filtering is disabled by default.

REPORT OPTIONS

-i, --input=<file>

Input file name.

-C, --cpu=<cpu>

Monitor only on the list of CPUs provided. Multiple CPUs can be provided as a comma-separated list with no space: 0,1. Ranges of CPUs are specified with - like 0-2. Default is to monitor all CPUS.

-D, --dump-raw-samples

Dump the raw decoded samples on the screen in a format that is easy to parse with one

sample per line.

`-s, --sort=<key>`

Group result by given key(s) - multiple keys can be specified in CSV format. The keys are specific to memory samples are: `symbol_daddr`, `symbol_iaddr`, `dso_daddr`, `locked`, `tlb`, `mem`, `snoop`, `dcacheline`, `phys_daddr`, `data_page_size`, `blocked`.

- ? `symbol_daddr`: name of data symbol being executed on at the time of sample
- ? `symbol_iaddr`: name of code symbol being executed on at the time of sample
- ? `dso_daddr`: name of library or module containing the data being executed on at the time of the sample
- ? `locked`: whether the bus was locked at the time of the sample
- ? `tlb`: type of tlb access for the data at the time of the sample
- ? `mem`: type of memory access for the data at the time of the sample
- ? `snoop`: type of snoop (if any) for the data at the time of the sample
- ? `dcacheline`: the cacheline the data address is on at the time of the sample
- ? `phys_daddr`: physical address of data being executed on at the time of sample
- ? `data_page_size`: the data page size of data being executed on at the time of sample
- ? `blocked`: reason of blocked load access for the data at the time of the sample

And the default sort keys are changed to `local_weight`, `mem`, `sym`, `dso`, `symbol_daddr`, `dso_daddr`, `snoop`, `tlb`, `locked`, `blocked`, `local_ins_lat`.

`-F, --fields=`

Specify output field - multiple keys can be specified in CSV format. Please see `perf-report(1)` for details.

In addition to the default fields, 'perf mem report' will provide the following fields to break down sample periods.

- ? `op`: operation in the sample instruction (load, store, prefetch, ...)
- ? `cache`: location in CPU cache (L1, L2, ...) where the sample hit
- ? `mem`: location in memory or other places the sample hit
- ? `dtlb`: location in Data TLB (L1, L2) where the sample hit
- ? `snoop`: snoop result for the sampled data access

Please take a look at the OUTPUT FIELD SELECTION section for caveats.

`-T, --type-profile`

Show data-type profile result instead of code symbols. This requires the debug information and it will change the default sort keys to: `mem`, `snoop`, `tlb`, `type`.

-U, --hide-unresolved

Only display entries resolved to a symbol.

-x, --field-separator=<separator>

Specify the field separator used when dump raw samples (-D option). By default, The separator is the space character.

In addition, for report all perf report options are valid, and for record all perf record options.

OVERHEAD CALCULATION

Unlike perf-report(1), which calculates overhead from the actual sample period, perf-mem overhead is calculated using sample weight. E.g. there are two samples in perf.data file, both with the same sample period, but one sample with weight 180 and the other with weight 20:

```
$ perf script -F period,data_src,weight,ip,sym
100000 629080842 |OP LOAD|LVL L3 hit|... 20 7e69b93ca524 strcmp
100000 1a29081042 |OP LOAD|LVL RAM hit|... 180 ffffffff82429168 memcpy

$ perf report -F overhead,symbol
50%  [.] strcmp
50%  [k] memcpy

$ perf mem report -F overhead,symbol
90%  [k] memcpy
10%  [.] strcmp
```

OUTPUT FIELD SELECTION

"perf mem report" adds a number of new output fields specific to data source information in the sample. Some of them have the same name with the existing sort keys ("mem" and "snoop"). So unlike other fields and sort keys, they'll behave differently when it's used by -F/--fields or -s/--sort.

Using those two as output fields will aggregate samples altogether and show breakdown.

```
$ perf mem report -F mem,snoop
...
# ----- Memory ----- --- Snoop ----
#   RAM Uncach Other   HitM Other
# .....
#
```

3.5% 0.0% 96.5% 25.1% 74.9%

But using the same name for sort keys will aggregate samples for each type separately.

```
$ perf mem report -s mem,snoop
```

#	Overhead	Samples	Memory access	Snoop
#				
#				
	47.99%	1509	L2 hit	N/A
	25.08%	338	core, same node Any cache hit	HitM
	10.24%	54374	N/A	N/A
	6.77%	35938	L1 hit	N/A
	6.39%	101	core, same node Any cache hit	N/A
	3.50%	69	RAM hit	N/A
	0.03%	158	LFB/MAB hit	N/A
	0.00%	2	Uncached hit	N/A

SEE ALSO

[perf-record\(1\)](#), [perf-report\(1\)](#), [perf-arm-spe\(1\)](#)

perf 05/25/2026 PERF-MEM(1)