



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

\$ man perf-diff.1

PERF-DIFF(1) perf Manual PERF-DIFF(1)

NAME

perf-diff - Read perf.data files and display the differential profile

SYNOPSIS

perf diff [baseline file] [data file1] [[data file2] ...]

DESCRIPTION

This command displays the performance difference amongst two or more perf.data files captured via perf record.

If no parameters are passed it will assume perf.data.old and perf.data.

The differential profile is displayed only for events matching both specified perf.data files.

If no parameters are passed the samples will be sorted by dso and symbol. As the perf.data files could come from different binaries, the symbols addresses could vary. So perf diff is based on the comparison of the files and symbols name.

OPTIONS

-D, --dump-raw-trace

Dump raw trace in ASCII.

--kallsyms=<file>

kallsyms pathname

-m, --modules

Load module symbols. WARNING: use only with -k and LIVE kernel

-d, --dsos=

Only consider symbols in these dsos. CSV that understands file://filename entries. This

option will affect the percentage of the Baseline/Delta column. See --percentage for more info.

-C, --comms=

Only consider symbols in these comms. CSV that understands file://filename entries. This option will affect the percentage of the Baseline/Delta column. See --percentage for more info.

-S, --symbols=

Only consider these symbols. CSV that understands file://filename entries. This option will affect the percentage of the Baseline/Delta column. See --percentage for more info.

-s, --sort=

Sort by key(s): pid, comm, dso, symbol, cpu, parent, srcline. Please see description of --sort in the perf-report man page.

-t, --field-separator=

Use a special separator character and don't pad with spaces, replacing all occurrences of this separator in symbol names (and other output) with a . character, that thus it's the only non valid separator.

-v, --verbose

Be verbose, for instance, show the raw counts in addition to the diff.

-q, --quiet

Do not show any warnings or messages. (Suppress -v)

-f, --force

Don't do ownership validation.

--symfs=<directory>

Look for files with symbols relative to this directory.

-b, --baseline-only

Show only items with match in baseline.

-c, --compute

Differential computation selection - delta, ratio, wdiff, cycles, delta-abs (default is delta-abs). Default can be changed using diff.compute config option. See COMPARISON METHODS section for more info.

--cycles-hist

Report a histogram and the standard deviation for cycles data. It can help us to judge if the reported cycles data is noisy or not. This option should be used with -c cycles.

-p, --period

Show period values for both compared hist entries.

-F, --formula

Show formula for given computation.

-o, --order

Specify compute sorting column number. 0 means sorting by baseline overhead and 1 (default) means sorting by computed value of column 1 (data from the first file other base baseline). Values more than 1 can be used only if enough data files are provided.

The default value can be set using the diff.order config option.

--percentage

Determine how to display the overhead percentage of filtered entries. Filters can be applied by --comms, --dsos and/or --symbols options.

"relative" means it's relative to filtered entries only so that the sum of shown entries will be always 100%. "absolute" means it retains the original value before and after the filter is applied.

--time

Analyze samples within given time window. It supports time percent with multiple time ranges. Time string is a%/n,b%/m,... or a%-b%,c%-%d,....

For example:

Select the second 10% time slice to diff:

```
perf diff --time 10%/2
```

Select from 0% to 10% time slice to diff:

```
perf diff --time 0%-10%
```

Select the first and the second 10% time slices to diff:

```
perf diff --time 10%/1,10%/2
```

Select from 0% to 10% and 30% to 40% slices to diff:

```
perf diff --time 0%-10%,30%-40%
```

It also supports analyzing samples within a given time window

<start>,<stop>. Times have the format seconds.nanoseconds. If 'start'

is not given (i.e. time string is 'x.y') then analysis starts at

the beginning of the file. If stop time is not given (i.e. time

string is 'x.y,') then analysis goes to the end of the file.

Multiple ranges can be separated by spaces, which requires the argument

to be quoted e.g. --time "1234.567,1234.789 1235,"

Time string is 'a1.b1,c1.d1:a2.b2,c2.d2'. Use ':' to separate timestamps for different perf.data files.

For example, we get the timestamp information from 'perf script'.

```
perf script -i perf.data.old
```

```
mgen 13940 [000] 3946.361400: ...
```

```
perf script -i perf.data
```

```
mgen 13940 [000] 3971.150589 ...
```

```
perf diff --time 3946.361400,:3971.150589,
```

It analyzes the perf.data.old from the timestamp 3946.361400 to the end of perf.data.old and analyzes the perf.data from the timestamp 3971.150589 to the end of perf.data.

--cpu

Only diff samples for 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 -: 0-2.

Default is to report samples on all CPUs.

--pid=

Only diff samples for given process ID (comma separated list).

--tid=

Only diff samples for given thread ID (comma separated list).

--stream

Enable hot streams comparison. Stream can be a callchain which is aggregated by the branch records from samples.

COMPARISON

The comparison is governed by the baseline file. The baseline perf.data file is iterated for samples. All other perf.data files specified on the command line are searched for the baseline sample pair. If the pair is found, specified computation is made and result is displayed.

All samples from non-baseline perf.data files, that do not match any baseline entry, are displayed with empty space within baseline column and possible computation results (delta) in their related column.

Example files samples: - file A with samples f1, f2, f3, f4, f6 - file B with samples f2, f4, f5 - file C with samples f1, f2, f5

Example output: x - computation takes place for pair b - baseline sample percentage

? perf diff A B C

baseline/A compute/B compute/C samples

b		x	f1
b	x	x	f2
b			f3
b	x		f4
b			f6
	x	x	f5

? perf diff B A C

baseline/B compute/A compute/C samples

b	x	x	f2
b	x		f4
b		x	f5
	x	x	f1
	x		f3
	x		f6

? perf diff C B A

baseline/C compute/B compute/A samples

b		x	f1
b	x	x	f2
b	x		f5
		x	f3
	x	x	f4
		x	f6

COMPARISON METHODS

delta

If specified the Delta column is displayed with value d computed as:

$$d = A \rightarrow \text{period_percent} - B \rightarrow \text{period_percent}$$

with: - A/B being matching hist entry from data/baseline file specified (or

perf.data/perf.data.old) respectively.

- ? period_percent being the % of the hist entry period value within single data file
- ? with filtering by -C, -d and/or -S, period_percent might be changed relative to how entries are filtered. Use --percentage=absolute to prevent such fluctuation.

delta-abs

Same as 'delta` method, but sort the result with the absolute values.

ratio

If specified the Ratio column is displayed with value r computed as:

$$r = A \rightarrow \text{period} / B \rightarrow \text{period}$$

with: - A/B being matching hist entry from data/baseline file specified (or perf.data/perf.data.old) respectively.

- ? period being the hist entry period value

wdiff:WEIGHT-B,WEIGHT-A

If specified the Weighted diff column is displayed with value d computed as:

$$d = B \rightarrow \text{period} * \text{WEIGHT-A} - A \rightarrow \text{period} * \text{WEIGHT-B}$$

- ? A/B being matching hist entry from data/baseline file specified (or perf.data/perf.data.old) respectively.
- ? period being the hist entry period value
- ? WEIGHT-A/WEIGHT-B being user supplied weights in the -c option behind : separator like -c wdiff:1,2.
- ? WEIGHT-A being the weight of the data file
- ? WEIGHT-B being the weight of the baseline data file

cycles

If specified the [Program Block Range] Cycles Diff column is displayed. It displays the cycles difference of same program basic block amongst two perf.data. The program basic block is the code between two branches.

[Program Block Range] indicates the range of a program basic block. Source line is reported if it can be found otherwise uses symbol+offset instead.

SEE ALSO

perf-record(1), perf-report(1)