



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

\$ man perf-bench.1

PERF-BENCH(1) perf Manual PERF-BENCH(1)

NAME

perf-bench - General framework for benchmark suites

SYNOPSIS

perf bench [<common options>] <subsystem> <suite> [<options>]

DESCRIPTION

This perf bench command is a general framework for benchmark suites.

COMMON OPTIONS

-r, --repeat=

Specify number of times to repeat the run (default 10).

-f, --format=

Specify format style. Current available format styles are:

default

Default style. This is mainly for human reading.

.ft C

% perf bench sched pipe # with no style specified

(executing 1000000 pipe operations between two tasks)

Total time:5.855 sec

5.855061 usecs/op

170792 ops/sec

.ft

simple

This simple style is friendly for automated processing by scripts.

.ft C

% perf bench --format=simple sched pipe # specified simple

5.988

.ft

SUBSYSTEM

sched

Scheduler and IPC mechanisms.

syscall

System call performance (throughput).

mem

Memory access performance.

numa

NUMA scheduling and MM benchmarks.

futex

Futex stressing benchmarks.

epoll

Eventpoll (epoll) stressing benchmarks.

internals

Benchmark internal perf functionality.

uprobe

Benchmark overhead of uprobe + BPF.

all

All benchmark subsystems.

SUITES FOR sched

messaging

Suite for evaluating performance of scheduler and IPC mechanisms. Based on hackbench by Rusty Russell.

Options of messaging

-p, --pipe

Use pipe() instead of socketpair()

-t, --thread

Be multi thread instead of multi process

-g, --group=

Specify number of groups

-l, --nr_loops=

Specify number of loops

Example of messaging

.ft C

% perf bench sched messaging # run with default

options (20 sender and receiver processes per group)

(10 groups == 400 processes run)

Total time:0.308 sec

% perf bench sched messaging -t -g 20 # be multi-thread, with 20 groups

(20 sender and receiver threads per group)

(20 groups == 800 threads run)

Total time:0.582 sec

.ft

pipe

Suite for pipe() system call. Based on pipe-test-1m.c by Ingo Molnar.

Options of pipe

-l, --loop=

Specify number of loops.

-G, --cgroups=

Names of cgroups for sender and receiver, separated by a comma. This is useful to check cgroup context switching overhead. Note that perf doesn't create nor delete the cgroups, so users should make sure that the cgroups exist and are accessible before use.

Example of pipe

.ft C

% perf bench sched pipe

(executing 1000000 pipe operations between two tasks)

Total time:8.091 sec

8.091833 usecs/op

123581 ops/sec

% perf bench sched pipe -l 1000 # loop 1000

(executing 1000 pipe operations between two tasks)

Total time:0.016 sec

16.948000 usecs/op

59004 ops/sec

% perf bench sched pipe -G AAA,BBB

(executing 1000000 pipe operations between cgroups)

Running 'sched/pipe' benchmark:

Executed 1000000 pipe operations between two processes

Total time: 6.886 [sec]

6.886208 usecs/op

145217 ops/sec

.ft

SUITES FOR syscall

basic

Suite for evaluating performance of core system call throughput (both usecs/op and ops/sec metrics). This uses a single thread simply doing getpid(2), which is a simple syscall where the result is not cached by glibc.

SUITES FOR mem

memcpy

Suite for evaluating performance of simple memory copy in various ways.

Options of memcpy

-s, --size

Specify size of memory to copy (default: 1MB). Available units are B, KB, MB, GB and TB (case insensitive).

-p, --page

Specify page-size for mapping memory buffers (default: 4KB). Available values are 4KB, 2MB, 1GB (case insensitive).

-k, --chunk

Specify the chunk-size for each invocation. (default: 0, or full-extent) Available units are B, KB, MB, GB and TB (case insensitive).

-f, --function

Specify function to copy (default: default). Available functions are depend on the architecture. On x86-64, x86-64-unrolled, x86-64-movsq and x86-64-movsb are supported.

-l, --nr_loops

Repeat memcpy invocation this number of times.

-c, --cycles

Use perf's cpu-cycles event instead of gettimeofday syscall.

memset

Suite for evaluating performance of simple memory set in various ways.

Options of memset

-s, --size

Specify size of memory to set (default: 1MB). Available units are B, KB, MB, GB and TB (case insensitive).

-p, --page

Specify page-size for mapping memory buffers (default: 4KB). Available values are 4KB, 2MB, 1GB (case insensitive).

-k, --chunk

Specify the chunk-size for each invocation. (default: 0, or full-extent) Available units are B, KB, MB, GB and TB (case insensitive).

-f, --function

Specify function to set (default: default). Available functions are depend on the architecture. On x86-64, x86-64-unrolled, x86-64-stosq and x86-64-stosb are supported.

-l, --nr_loops

Repeat memset invocation this number of times.

-c, --cycles

Use perf's cpu-cycles event instead of gettimeofday syscall.

mmap

Suite for evaluating memory subsystem performance for mmap()'d memory.

Options of mmap

-s, --size

Specify size of memory to set (default: 1MB). Available units are B, KB, MB, GB and TB (case insensitive).

-p, --page

Specify page-size for mapping memory buffers (default: 4KB). Available values are 4KB, 2MB, 1GB (case insensitive).

-r, --randomize

Specify seed to randomize page access offset (default: 0, or not randomized).

-f, --function

Specify function to set (default: all). Available functions are demand and populate, with the first demand faulting pages in the region and the second using an eager mapping.

-l, --nr_loops

Repeat mmap() invocation this number of times.

-c, --cycles

Use perf's cpu-cycles event instead of gettimeofday syscall.

SUITES FOR numa

mem

Suite for evaluating NUMA workloads.

SUITES FOR futex

hash

Suite for evaluating hash tables.

wake

Suite for evaluating wake calls.

wake-parallel

Suite for evaluating parallel wake calls.

requeue

Suite for evaluating requeue calls.

lock-pi

Suite for evaluating futex lock_pi calls.

SUITES FOR epoll

wait

Suite for evaluating concurrent epoll_wait calls.

ctl

Suite for evaluating multiple epoll_ctl calls.

SUITES FOR internals

synthesize

Suite for evaluating perf's event synthesis performance.

perf(1)

perf

05/25/2026

PERF-BENCH(1)