hyperfine Command-line benchmarking

CONCEPTS

Commands run directly without a shell. Quote each command, and add -S when you need pipes, && or redirects.

setup once
↓ then
prepare before each run
↓ then
command timed
↓ then
conclude after each run
↓ finally
cleanup once
One benchmark, per command

The first command is the reference; the others are reported relative to it. Default is at least 10 runs, aiming for about 3 seconds.

RUNNING

hyperfine '<cmd>'
Benchmark a command

hyperfine '<cmd1>' '<cmd2>'
Compare commands side by side

hyperfine -r <n> '<cmd>'
Run exactly n times

hyperfine -m <n> '<cmd>'
Run at least n times

hyperfine -M <n> '<cmd>'
Run at most n times

hyperfine -w <n> '<cmd>'
Do n warmup runs first

hyperfine -n <name> '<cmd>'
Give a command a display name

hyperfine -i '<cmd>'
Ignore a non-zero exit code

hyperfine --show-output '<cmd>'
Show the command's output

PREPARING RUNS

-p '<cmd>' / --prepare
Run before every timing run

-C '<cmd>' / --conclude
Run after every timing run

-s '<cmd>' / --setup
Run once before a benchmark

-c '<cmd>' / --cleanup
Run once after a benchmark

--prepare 'make clean' 'make'
Rebuild from scratch each run

PARAMETERS

-P <var> <min> <max> '<cmd> {var}'
Scan a numeric range

-P <var> <min> <max> -D <step> ...
Set the scan step size

-L <var> a,b,c '<cmd> {var}'
Try a list of values

-L <v1> a,b -L <v2> x,y '<cmd> {v1} {v2}'
Benchmark all combinations

SHELL

-N / --shell=none
Run without a shell (default)

-S
Run through the default shell

--shell <name>
Use a specific shell, e.g. bash or zsh

OUTPUT AND EXPORT

--export-json <file>
Save stats and every sample as JSON

--export-markdown <file>
Save a Markdown table

-h / --help
Show help

-V / --version
Show version