kffl / **speedbump** Public

Code · Issues 2 · Pull requests 4 · Discussions · Actions · Projects · Security · In

master · **7** Branches · **8** Tags · Go to file · Add file · Code

kffl M 04e6c7 · 7 months ago · 53 Commits

.github	build(deps): bump …	10 months ago
assets	docs: add combine…	2 years ago
lib	feat: add `host` option	2 years ago
.gitignore	Initial commit	2 years ago
CONTRIBU…	docs: add contribut…	2 years ago
Dockerfile	build: add Dockerfile	2 years ago
LICENSE	Initial commit	2 years ago
README.md	docs: add `host` fla…	2 years ago
args.go	chore: bump versio…	2 years ago
args_test.go	feat: add `host` option	2 years ago
go.mod	feat: logging impro…	2 years ago
go.sum	feat: logging impro…	2 years ago
main.go	feat!: `lib` API impr…	2 years ago

About

TCP proxy for simulating variable, yet predictable network latency 🌐 ⏳

#go #golang #tcp #load-testing #tcp-proxy #observability #network-latency

Readme

Apache-2.0 license

Activity

942 stars

8 watching

26 forks

Report repository

Releases 8

v1.1.0 Latest
on Nov 20, 2022

+ 7 releases

Packages

No packages published

Contributors 4

kffl Paweł Kuffel

roertbb Robert Banaszak

dependabot[bot]

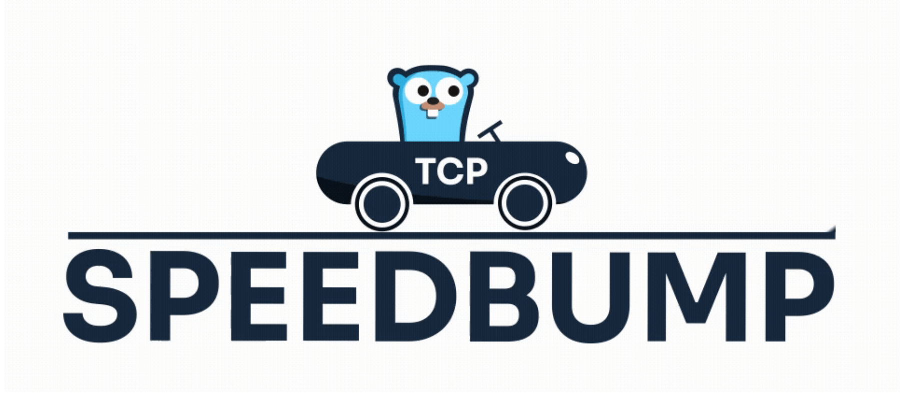
szkf Simon

Languages

● **Go** 99.0% ● **Dockerfile** 1.0%

README Apache-2.0 license

speedbump - TCP proxy with variable latency



Speedbump is a TCP proxy written in Go which allows for simulating variable network latency.

CI passing | go report A+ | docker pulls 13k | version v1.1.0 | GO reference

Usage

Installation

The easiest way to install speedbump is to download pre-built binaries for your platform that are automatically attached to each release under *Assets*. If you wish to build speedbump from source, clone this repository and run `go build`. Alternatively, you can run speedbump as a container using the kffl/speedbump image.

Basic usage examples

Spawn a new instance listening on port 2000 that proxies TCP traffic to localhost:80 with a base latency of 100ms and sine wave amplitude of 100ms (resulting in maximum added latency being 200ms and minimum being 0), period of which is 1 minute:

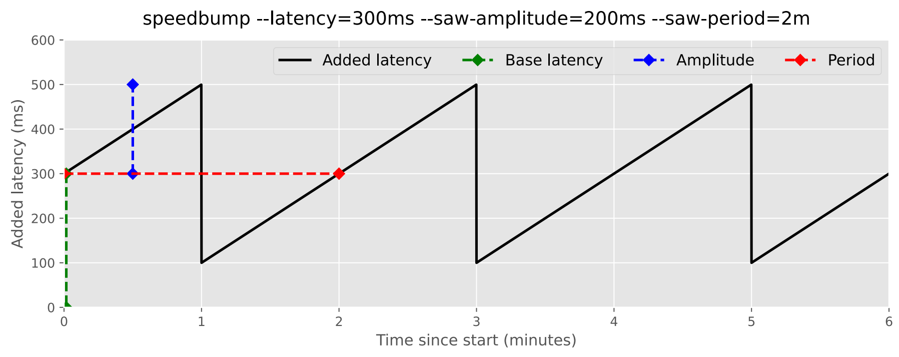
```
speedbump --latency=100ms --sine-amplitude=100ms --sine-period=1m --port=2000 localhost:80
```

or when running speedbump using the kffl/speedbump container image:

```
docker run --net=host kffl/speedbump:latest --latency=100ms --sine-amplitude=100ms \
           --sine-period=1m --port=2000 localhost:80
```

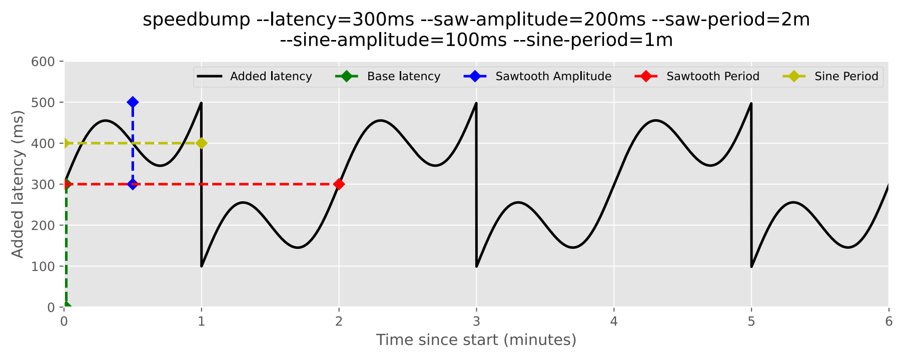
Spawn a new instance with a base latency of 300ms and a sawtooth wave latency summand with amplitude of 200ms and period of 2 minutes (visualized by the graph below):

```
speedbump --latency=300ms --saw-amplitude=200ms --
saw-period=2m --port=2000 localhost:80
```



Combining latency summands

It is possible to run speedbump with multiple latency summands at once:



CLI Arguments Reference:

Output of `speedbump --help` :

```
usage: speedbump [<flags>] <destination>

TCP proxy for simulating variable network latency.

Flags:
  --help                    Show context-sensitive
help (also try --help-long and
                          --help-man).
  --host=""                 IP or hostname to listen
on. Speedbump will bind to
                          all available network
interfaces if unspecified.
  --port=8000               Port number to listen
on.
  --buffer=64KB             Size of the buffer used
for TCP reads.
  --queue-size=1024         Size of the delay queue
storing read buffers.
  --latency=5ms             Base latency added to
```

```
proxied traffic.
  --log-level=INFO        Log level. Possible
values: DEBUG, TRACE, INFO, WARN,
                          ERROR.
  --sine-amplitude=0      Amplitude of the latency
sine wave.
  --sine-period=0         Period of the latency
sine wave.
  --saw-amplitude=0       Amplitude of the latency
sawtooth wave.
  --saw-period=0          Period of the latency
sawtooth wave.
  --square-amplitude=0    Amplitude of the latency
square wave.
  --square-period=0       Period of the latency
square wave.
  --triangle-amplitude=0  Amplitude of the latency
triangle wave.
  --triangle-period=0     Period of the latency
triangle wave.
  --version               Show application
version.

Args:
  <destination>  TCP proxy destination in
host:post format.
```

Using speedbump as a library

Speedbump can be used as a Go library via its `lib` package. Check `lib` README for additional information.