

The WebSocket god of thunder

MIT license

1.3k stars 165 forks

Star

Notifications

Code

Issues 33

Pull requests 9

Actions

Projects

Wiki

Security

Insights

master

Go to file



3rd-Eden Merge pull request #44 from evaisse/master ...

on Feb 5, 2017 31

[View code](#)

README.md

# Thor

Thor is WebSocket benchmarking/load generator. There are a lot of benchmarking tools for HTTP servers. You've got ab, siege, wrk and more. But all these tools only work with plain ol HTTP and have no support for WebSockets - even if they did they wouldn't be suitable, as they would be testing short running HTTP requests instead of long running HTTP requests with a lot of messaging traffic. Thor fixes all of this.

## Dependencies

Thor requires Node.js to be installed on your system. If you don't have Node.js installed you can download it from <http://nodejs.org> or build it from the github source repository: <http://github.com/joyent/node>.

Once you have Node.js installed, you can use the bundled package manager `npm` to install this module:

```
npm install -g thor
```

The `-g` command flag tells `npm` to install the module globally on your system.

## Usage

```
thor [options] <urls>
```

Thor can hit multiple URL's at once; this is useful if you are testing your reverse proxies, load balancers or just simply multiple applications. The url that you supply to `thor` should be written in a WebSocket compatible format using the `ws` or `wss` protocols:

```
thor --amount 5000 ws://localhost:8080 wss://localhost:8081
```

The snippet above will open up `5000` connections against the regular `ws://localhost:8080` and also `5000` connections against the *secured* `wss://localhost:8081` server, so a total of `10000` connections will be made.

One thing to keep in mind is you probably need to bump the amount of file descriptors on your local machine if you start testing WebSockets. Set the `ulimit -n` on machine as high as possible. If you do not know how to do this, Google it.

## Options

```
Usage: thor [options] ws://localhost
```

Options:

<code>-h, --help</code>	output usage information
<code>-A, --amount &lt;connections&gt;</code>	the amount of persistent connections to generate
<code>-C, --concurrent &lt;connections&gt;</code>	how many concurrent-connections per second
<code>-M, --messages &lt;messages&gt;</code>	messages to be send per connection
<code>-P, --protocol &lt;protocol&gt;</code>	WebSocket protocol version
<code>-B, --buffer &lt;size&gt;</code>	size of the messages that are send
<code>-W, --workers &lt;cpus&gt;</code>	workers to be spawned
<code>-G, --generator &lt;file&gt;</code>	custom message generators
<code>-M, --masked</code>	send the messaged with a mask
<code>-b, --binary</code>	send binary messages instead of utf-8
<code>-V, --version</code>	output the version number

Some small notes about the options:

- `--protocol` is the protocol version number. If you want to use the *HyBi drafts 07-12* use `8` as argument or if you want to use the *HyBi drafts 13-17* drafts which are the default version use `13`.
- `--buffer` should be size of the message in bytes.
- `--workers` as Node.js is single threaded this sets the amount of sub processes to handle all the heavy lifting.

## Custom messages

Some WebSocket servers have their own custom messaging protocol. In order to work with those servers we introduced a concept called `generators` a generator is a small JavaScript file that can output `utf8` and `binary` messages. It uses a really simple generator by default.

Checkout <https://github.com/observing/thor/blob/master/generator.js> for an example of a generator.

```
thor --amount 1000 --generator <file.js> ws://localhost:8080
```

## Example

```
thor --amount 1000 --messages 100 ws://localhost:8080
```

This will hit the WebSocket server that runs on localhost:8080 with 1000 connections and sends 100 messages over each established connection. Once `thor` is done with smashing your connections it will generate a detailed report:

```
Thor:                                     version: 1.0.0
```

```
God of Thunder, son of Odin and smasher of WebSockets!
```

```
Thou shall:
```

- Spawn 4 workers.
- Create all the concurrent/parallel connections.
- Smash 1000 connections with the mighty Mjölnir.

```
The answers you seek shall be yours, once I claim what is mine.
```

```
Connecting to ws://localhost:8080
```

```
Opened 100 connections
Opened 200 connections
Opened 300 connections
Opened 400 connections
Opened 500 connections
Opened 600 connections
Opened 700 connections
Opened 800 connections
Opened 900 connections
Opened 1000 connections
```

```
Online           15000 milliseconds
Time taken       31775 milliseconds
Connected        1000
Disconnected     0
Failed           0
Total transferred 120.46MB
```

Total received 120.43MB

Durations (ms):

	min	mean	stddev	median	max
Handshaking	217	5036	4094	3902	14451
Latency	0	215	104	205	701

Percentile (ms):

	50%	66%	75%	80%	90%	95%	98%	98%
100%								
Handshaking	3902	6425	8273	9141	11409	12904	13382	13945
14451								
Latency	205	246	266	288	371	413	437	443
701								

## License

MIT

## Releases

🔖 1 tags

## Packages

No packages published

## Used by 7



## Contributors 4



# Languages

● JavaScript 100.0%