

Siege and Beyond:

An Intro to Benchmarking and Stress Tests

WordCamp Vancouver 2015

Mike Schroder (DH-Shredder)

WordPress Platform Lead at DreamHost

@GetSource - <http://www.getsource.net>

Who Am I?



- **Mike Schroder**, a.k.a **DH-Shredder**, a.k.a. **@GetSource**
- Third Culture Kid, enjoy Coffee & Sailing
- WordPress 3.9 Co-Lead and Core Contributor
- Happy DreamHost Employee

What will happen if my site
hits the *reddit front page*?

What if reddit *doesn't matter* to me?

Prove *how many* users can access
your site at one time.

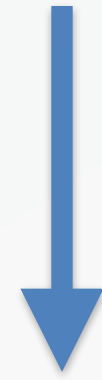
Load Testing *or* Stress Testing?

Load Tests: *Will my site run well with the expected amount of users?*

Stress Tests: *How many users before
the site/infrastructure cracks?*

Are we testing *dynamic or cached* requests?

INTERNET



FRONT-END CACHE



PHP

Varnish/Static: ~50ms

PHP: ~100ms
(without stress)

PHP: ~1500ms
(with stress)

All methods are *not* equal.

“Users/minute” *does not* necessarily mean “real users during a minute”.

There are *many* utilities for stress testing.

Let's start with *Siege*.

You can run cached *or* dynamic tests.

It's used only from the *command line*.

It only runs tests from *one* server.

Linux: Packages.
Mac: Homebrew.

<https://www.joedog.org/siege-home/>

Primary configuration (.siegerc)

To run **dynamic** tests

```
login-url = http://example.com/wp-login.php POST  
log=[user]&pwd=[password]&wp-submit=Log+In&  
redirect_to=http%3A%2F%2Fexample.com%2Fwp-admin%2F&testcookie=1
```

A *list* of URLs.

List format

```
HOST=http://example.com
```

```
${HOST}/
```

```
${HOST}/?feed=rss2
```

```
${HOST}/?p=41
```

```
${HOST}/?feed=rss2
```

```
${HOST}/?p=44
```

```
${HOST}/
```

```
...
```

Results

```
Transactions:           712 hits
Availability:          100.00 %
Elapsed time:           59.12 secs
Data transferred:      3.70 MB
Response time:         1.23 secs
Transaction rate:      12.04 trans/sec
Throughput:            0.06 MB/sec
Concurrency:           14.75
Successful transactions: 702
Failed transactions:    0
Longest transaction:   8.49
Shortest transaction:  0.20
```

Where can I find *test content*?

HHVM's *oss-performance* is one option.

<https://github.com/hhvm/oss-performance/tree/master/targets/wordpress>

To seed from your own content, take a
look at your *logs*, or try:

<http://example.com/>

<http://example.com/?feed=rss2>

http://example.com/?p=%{*:1-52}

You can also use *sproxy*,
not to be confused with stunnel.

Loader.io is a *service* to run stress tests.

However, it has a friendly *GUI* to run tests and see the results.

Response Times

Average 24 ms
Min/Max 7 / 1073 ms

Response Counts

Success 14874 Timeout 0
400/500 124 / 0 Network 0

Bandwidth

Sent 1.73 MB
Received 441.32 MB

Redirects

Valid 358
Invalid 0

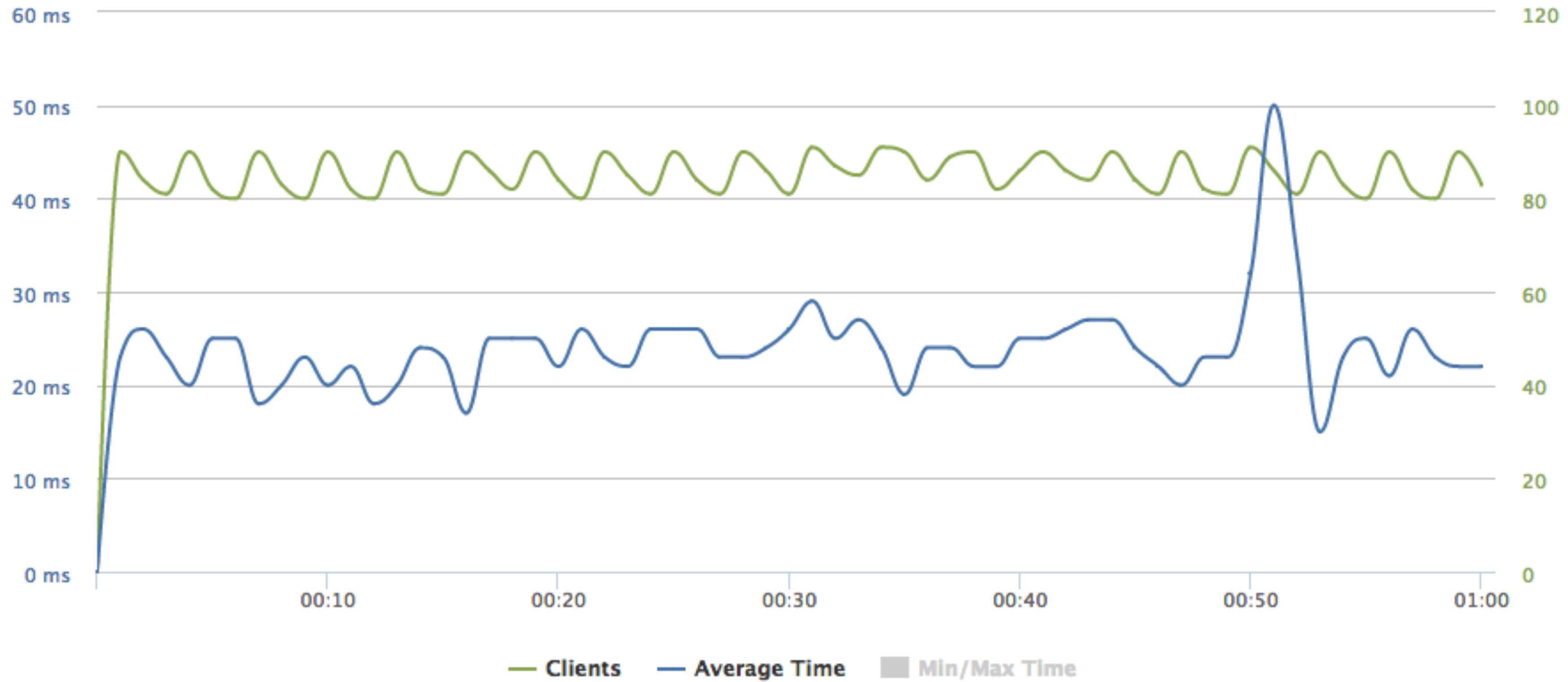
Times

Details

Bandwidth

Distribution

▶ Watch simulation



Single data center, but from a
cluster of AWS servers.

More complex user behavior?

Try a service like *LoadStorm*.

What kinds of *issues* will I find?

How do I *debug* problems once they're found?

Bonus: Your WordPress site runs slowly,
and even *more slowly* as soon as you
enable a persistent object-cache.

Why?

alloptions.

Memcached *slab size*.

wp_options cleanup.

Questions?

- getsource.net/tag/siege/
- joedog.org/siege-manual/
- joedog.org/sproxy-home/
- github.com/hhvm/oss-performance

Mike Schroder (DH-Shredder)

WordPress Platform Lead at DreamHost

@GetSource - <http://www.getsource.net>