

Benchmarking MediaWiki with phpbench

Custom Policy

Actions ▾

Draft Post

This is a draft, and is only visible to you and other users who can edit [Doing the needful](#). Use "Publish" to publish this post.

This post gives a quick introduction to a benchmarking tool, [phpbench](#), ready for you to experiment with in core and skins/extensions.[1]

What is phpbench?

From their [documentation](#):

PHPBench is a benchmark runner for PHP analogous to PHPUnit but for performance rather than correctness.

In other words, while a PHPUnit test will tell you if your code behaves a certain way given a certain set of inputs, a PHPBench benchmark only cares how long that same piece of code takes to execute.

The tooling and boilerplate will be familiar to you if you've used PHPUnit. There's a command-line runner at `vendor/bin/phpbench`, benchmarks are discoverable by default in `tests/Benchmark`, a configuration file (`benchmark.json`) allows for setting defaults across all benchmarks, and the benchmark tests classes and tests look pretty similar to PHPUnit tests.

Here's an example test for the `Html::openElement()` function:

```
namespace MediaWiki\Tests\Benchmark;

class HtmlBench {

    /**
     * @Assert("mode(variant.time.avg) < 85 microseconds +/- 10%")
     */
    public function benchHtmlOpenElement() {
        \Html::openElement( 'a', [ 'class' => 'foo' ] );
    }
}
```

So, taking it line by line:

- `class HtmlBench` (placed in `tests/Benchmark/includes/HtmlBench.php`) – the class where you can define the benchmarks for methods in a class. It would make sense to create a single benchmark class for a single class under test, just like with PHPUnit.
- `public function benchHtmlOpenElement() {}` – method names that begin with `bench` will be executed by `phpbench`; other methods can be used for set-up / teardown work. The contents of the method are benchmarked, so any set-up / teardown work should be done elsewhere.
- `@Assert("mode(variant.time.avg) < 85 microseconds +/- 10%")` – we define a [phpbench assertion](#) that the average execution time will be less than 85 microseconds, with a tolerance of +/- 10%.

If we run the test with `composer phpbench`, we will see that the test passes. One thing to be careful with is to allow for some slack in the assertion, as otherwise you'd end up with flaky tests in CI.

Measuring performance while developing

One neat feature in PHPBench is the ability to tag current results and compare with another run. Looking at the [HTMLBench](#) benchmark test from above, for example, we can compare the work done in [rMW5deb6a2a4546: Html::openElement\(\) micro-optimisations](#) to get before and after comparisons of the performance changes.

Here's a benchmark of [e82c5e52d50a9afd67045f984dc3fb84e2daef44](#), the commit before the performance improvements added to `Html::openElement()` in [rMW5deb6a2a4546: Html::openElement\(\) micro-optimisations](#)

```
> git checkout -b html-before-optimizations e82c5e52d50a9afd67045f984dc3fb84e2daef44 # get the old
HTML::openElement code before optimizations
> git review -x 727429 # get the core patch which introduces phpbench support
> composer phpbench -- tests/Benchmark/includes/HtmlBench.php --tag=original
```

And the output [2]:

```
> phpbench run --config=tests/Benchmark/phpbench.json --report=aggregate 'tests/Benchmark/includes/HtmlBench.php'
'--tag=original'
PHPBench (1.1.2) running benchmarks...
with configuration file: /Users/kostajh/src/mediawiki/w/tests/Benchmark/phpbench.json
with PHP version 7.4.24, xdebug ✓, opcache ✘

\MediaWiki\Tests\Benchmark\HtmlBench

benchHtmlOpenElement.....R1 I1 ✓ M 18.514µs (±1.94%)

Subjects: 1, Assertions: 1, Failures: 0, Errors: 0
Storing results ... OK
Run: 1346543289c75373e513cc3b11fbf5215d8fb6d0
+-----+-----+-----+-----+-----+
| benchmark | subject | set | revs | its | mem_peak | mode | rstdev |
+-----+-----+-----+-----+-----+
| HtmlBench | benchHtmlOpenElement | 50 | 5 | 2.782mb | 18.514µs | ±1.94% |
+-----+-----+-----+-----+
```

Note that we've used `--tag=original` to store the results. Now we can check out the newer code, and use `--ref=original` to compare with the baseline:

```
> git checkout -b html-after-optimizations 5deb6a2a4546318d1fa94ad8c3fa54e9eb8fc67c # get the new
HTML::openElement code with optimizations
> git review -x 727429 # get the core patch which introduces phpbench support
> composer phpbench -- tests/Benchmark/includes/HtmlBench.php --ref=original --report=aggregate
```

And the output [3]:

```
> phpbench run --config=tests/Benchmark/phpbench.json --report=aggregate 'tests/Benchmark/includes/HtmlBench.php'
'--ref=original' '--report=aggregate'
PHPBench (1.1.2) running benchmarks...
with configuration file: /Users/kostajh/src/mediawiki/w/tests/Benchmark/phpbench.json
with PHP version 7.4.24, xdebug ✓, opcache ✘
comparing [actual vs. original]

\MediaWiki\Tests\Benchmark\HtmlBench

benchHtmlOpenElement.....R5 I4 ✓ [M 8.194µs vs. M 18.514µs] -55.74% (±0.50%)

Subjects: 1, Assertions: 1, Failures: 0, Errors: 0
+-----+-----+-----+-----+-----+
| benchmark | subject | set | revs | its | mem_peak | mode | rstdev |
+-----+-----+-----+-----+-----+
| HtmlBench | benchHtmlOpenElement | 50 | 5 | 2.782mb 0.00% | 8.194µs -55.74% | ±0.50% -74.03% |
+-----+-----+-----+-----+
```

We can see that the execution time roughly halved, from 18 microseconds to 8 microseconds. (For understanding the other columns in the report, it's best to read through the [Quick Start guide for phpbench](#).)

Testing with extensions

A slightly more complex example is available in [GrowthExperiments](#) ([patch](#)). That patch makes use of `setUp/tearDown` methods to prepopulate the database entries needed for the code being benchmarked:

```
/*
 * @BeforeMethods ("setUpLinkRecommendation")
 * @AfterMethods ("tearDownLinkRecommendation")
 * @Assert("mode(variant.time.avg) < 20000 microseconds +/- 10%")
 */
public function benchFilter() {
    $this->linkRecommendationFilter->filter( $this->tasks );
}
```

The `setUpLinkRecommendation` and `tearDownLinkRecommendation` methods have access to MediaWikiServices, and generally you can do similar things you'd do in an integration test to setup and teardown the environment. This test is towards the opposite end of the spectrum from the core test discussed above which looks at `Html::openElement()`; here, the goal is to look at a higher level function that involves database queries and interacting with MediaWiki services.

What's next

You can experiment with the tooling and see if it is useful to you. Some open questions:

- do we want to use `phpbench`? are the scripts in `maintenance/benchmarks` already sufficient for everyone's benchmarking needs?
- we already have a benchmarking tools in `maintenance/benchmarks` that extend a `Benchmark` class; would it make sense to convert these to use `phpbench`?
- what are sensible defaults for "revs" and "iterations" as well as retry thresholds?
- do we want to run `phpbench` assertions in CI?
 - if yes, do we want assertions about using absolute times (e.g. "this function should take less than 20 ms") or relative assertions ("patch code is within 10% +/- of old code")
 - if yes, do we want to aggregate reports over time, so we can see trends for the code we benchmark?
 - {your questions/comments welcome!}

Looking forward to your feedback!

[1] thank you, [@hashar](#), for working with me to include this in Quibble and roll out to CI to help with evaluation!

[2]

```
> phpbench run --config=tests/Benchmark/phpbench.json --report=aggregate 'tests/Benchmark/includes
/HtmlBench.php' '--tag=original'
PHPBench (1.1.2) running benchmarks...
with configuration file: /Users/kostajh/src/mediawiki/w/tests/Benchmark/phpbench.json
with PHP version 7.4.24, xdebug ✓, opcache ✘

\MediaWiki\Tests\Benchmark\HtmlBench

benchHtmlOpenElement.....R1 I1 ✓ M 18.514µs (±1.94%)

Subjects: 1, Assertions: 1, Failures: 0, Errors: 0
Storing results ... OK
Run: 1346543289c75373e513cc3b11fbf5215d8fb6d0
+-----+-----+-----+-----+-----+
| benchmark | subject           | set | revs | its  | mem_peak | mode   | rstdev |
+-----+-----+-----+-----+-----+-----+
| HtmlBench | benchHtmlOpenElement |    50 |    5 | 2.782mb | 18.514µs | ±1.94% |
+-----+-----+-----+-----+-----+
```

[3]

```
> phpbench run --config=tests/Benchmark/phpbench.json --report=aggregate 'tests/Benchmark/includes
/HtmlBench.php' '--ref=original' '--report=aggregate'
PHPBench (1.1.2) running benchmarks...
with configuration file: /Users/kostajh/src/mediawiki/w/tests/Benchmark/phpbench.json
```

```
with PHP version 7.4.24, xdebug ✓, opcache ✘
comparing [actual vs. original]
```

```
\MediaWiki\Tests\Benchmark\HtmlBench
```

```
benchHtmlOpenElement.....R5 I4 ✓ [Mo8.194µs vs. Mo18.514µs] -55.74% (±0.50%)
```

```
Subjects: 1, Assertions: 1, Failures: 0, Errors: 0
```

benchmark	subject	set	revs	its	mem_peak	mode	rstdev
HtmlBench	benchHtmlOpenElement		50	5	2.782mb 0.00%	8.194µs -55.74%	±0.50% -74.03%



[Previous Post](#)

How we deploy code

Unpublished draft by [kostajh](#).

</> Staff Software Engineer, Growth Team

Projects None

Subscribers [zeljkofilipin](#), [hashar](#)



B I T ☰ | ≡ ≡ </> “ ” ☁ | 😊

✖ ✎ ✎ ✎

Submit