Happyr

# Hack in 15 minutes

Tobias Nyholm

Happyr

# About me

- Tobias Nyholm, @tobiasnyholm
- Happyr
- Certified Symfony developer
- Co-host of the Sound of Symfony podcast
- Organizer of Symfony Sweden

Happyr

# Facebook had a problem

Happyr

# HHVM

- Use as PHP-FPM
- Linux packages
- http://hhvm.com/blog/1817/fastercgi-with-hhvm

Happyr

# HHVM

## PHP

```
Requests per second:    23.17 [#/sec] (mean)
Time per request:       2157.579 [ms] (mean)
Time per request:       43.152 [ms] (mean, across all concurrent requests)
Transfer rate:          275.42 [Kbytes/sec] received
```

## HHVM

```
Requests per second:    184.71 [#/sec] (mean)
Time per request:       270.689 [ms] (mean)
Time per request:       5.414 [ms] (mean, across all concurrent requests)
Transfer rate:          2194.38 [Kbytes/sec] received
```

Happyr

# HHVM

## PHP

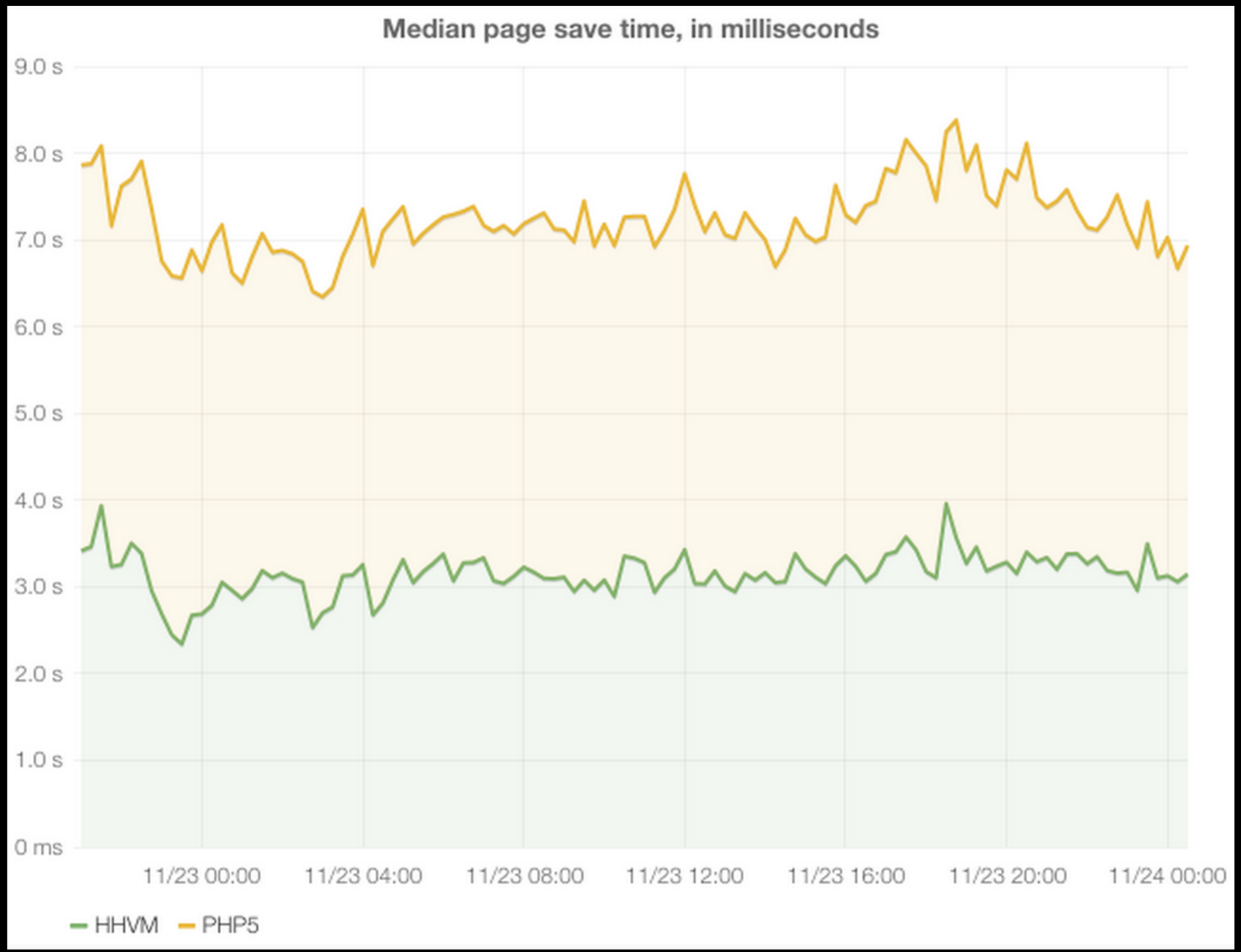
```
Requests per second: 13789.24 [#/sec] (mean) Fib(5)
Requests per second: 3202.31 [#/sec] (mean)  Fib(15)
Requests per second: 118.94 [#/sec] (mean)   Fib(25)*
Requests per second: 8.40 [#/sec] (mean)     Fib(30)**
```

## HHVM

```
Requests per second: 8842.70 [#/sec] (mean) Fib(5)
Requests per second: 8892.66 [#/sec] (mean) Fib(15)
Requests per second: 5581.37 [#/sec] (mean) Fib(25)
Requests per second: 737.56 [#/sec] (mean)  Fib(30)
```

Happyr

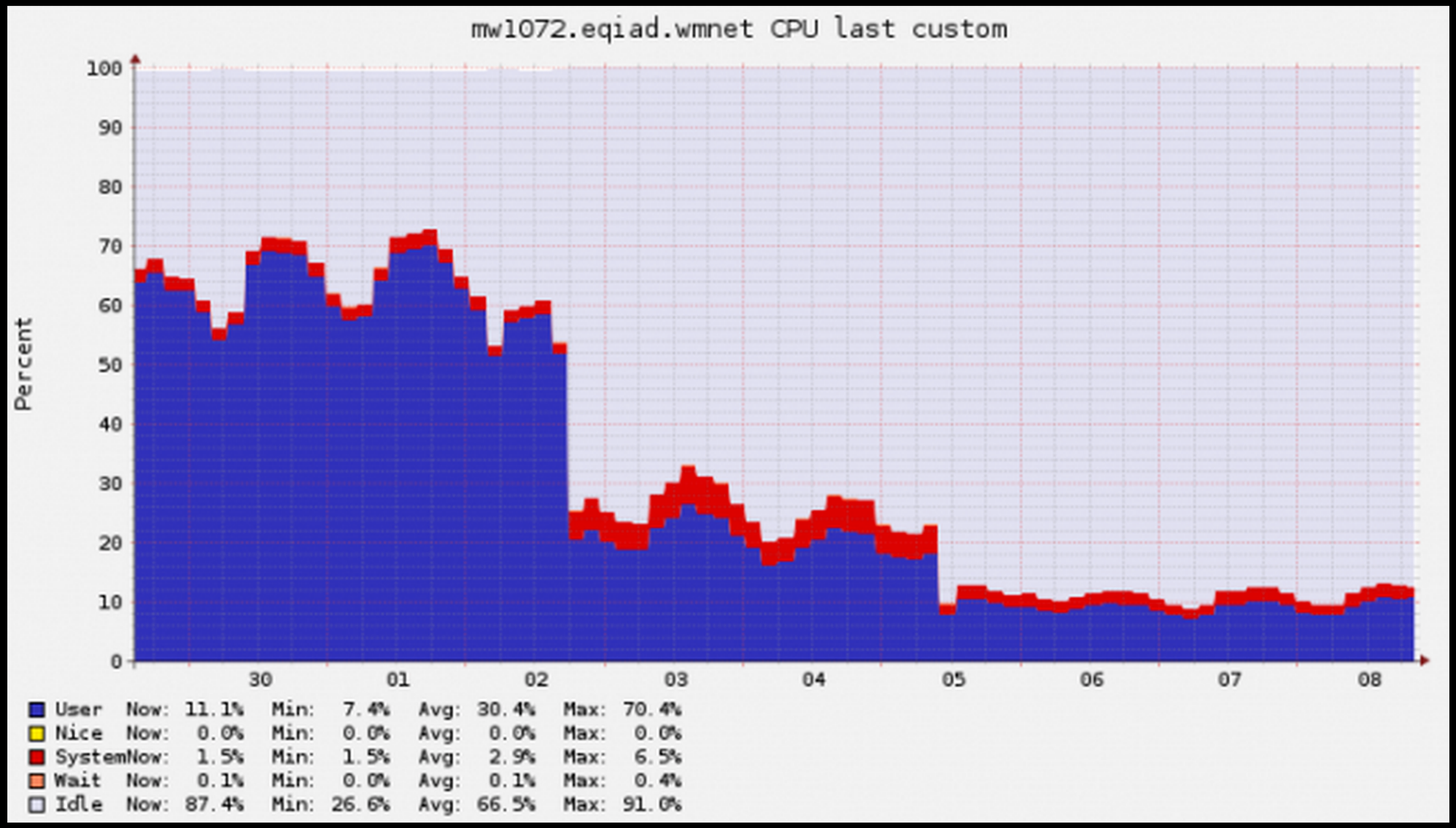
# Wikipedia




http://hhvm.com/blog/7205/wikipedia-on-hhvm

Happyr

# Wikipedia




http://hhvm.com/blog/7205/wikipedia-on-hhvm

Happyr

# What about Hack?

Happyr

**Happyr**

123 == “123foo”
“123” != “123foo”

"6" == " 6"

"133" == "0133"

NULL < -1
NULL == 0

“025” == 031

“foo” == TRUE
“foo” == 0
TRUE != 0

133 ! = 0133

Happyr

# Hack

- Backwards compatible with PHP
- Slightly faster on HHVM
- Syntactic sugar
- *Gradually* typed

Happyr

# Static typed languages

- C, Go, Java, Haskell

```
public class Foobar{
  String foo;
  int bar;

  String baz(int i) {
    return “Biz”;
  }
}
```

```
func main() {
  var i, j int = 1, 2
  k := 3
  c, python, java := true, false, "no!"

  fmt.Println(i, j, k, c, python, java)
  //output: 1 2 3 true false no!
}
```

Happyr

# Dynamic typed language

- PHP, Ruby, Python, Javascript

```
class Storage {
  private $foo;

  String set($i) {
    $this->foo=$i
  }
}
```

```
$storage = new Storage();
$storage->set(4711);
$storage->set(‘Foobar’);
$storage->set(array(‘foo’));
$storage->set(new User());
$storage->set(null);
```

Happyr

# But what if…

```
$storage = new Storage();
$storage->set(4711);

// …
$array = $storage->get();
echo $array[0];

// …
$user = $storage->get();
echo $user->getUsername();
```

Happyr

# Hack modes

- Strict - Type check everything
- Partial - Check whatever is annotated
- Decl - Check nothing

"Show some code"

*–The audience*

Happyr

```php
<?php

class MessageProvider {
  private $counter=0;
  private $message="";

  public function set($text) {
    $this->message = $text;
    $this->counter++;
    return $this->counter;
  }
}
```

Happyr

```
<?hh

class MessageProvider {
  private $counter=0;
  private $message="";

  public function set($text) {
    $this->message = $text;
    $this->counter++;
    return $this->counter;
  }
}
```

Happyr

```
<?hh

class MessageProvider {
  private int $counter=0;
  private string $message="";

  public function set($text) {
    $this->message = $text;
    $this->counter++;
    return $this->counter;
  }
}
```

Happyr

```
<?hh

class MessageProvider {
  private int $counter=0;
  private string $message="";

  public function set(string $text) {
    $this->message = $text;
    $this->counter++;
    return $this->counter;
  }
}
```

Happyr

```
<?hh

class MessageProvider {
  private int $counter=0;
  private string $message="";

  public function set(string $text): int {
    $this->message = $text;
    $this->counter++;
    return $this->counter;
  }
}
```

```
<?hh

function main() {
  $mp= new MessageProvider();
  $var = $mp->set(“hello”);
  $mp->set($var);
}

main();
```

Happyr

```
<?hh

function main() {
  $mp= new MessageProvider();
  $var = $mp->set(“hello”);
  $mp->set($var);
}

 main();
```

```
/index.php:6:14,17: Invalid argument
  /Mp.php:6:25,30: This is a string
  /Mp.php:6:40,42: It is incompatible with an int
```

Happyr

# Syntactic sugar

```
<?hh

class MessageProvider {
  private int $counter=0;
  private string $message="";

  public function __construct(
     int $cnt, string $msg
  ) {
  $this->counter = $cnt;
  $this->message = $msg;
  }
}
```

Happyr

# Syntactic sugar

```
<?hh

class MessageProvider {
    public function __construct(
        private int $counter,
        private string $message
    ) { }
}
```

Happyr

# Syntactic sugar

```
<?php

$arr = array('hello', 'bazbar', 'foo');
usort($arr, function($a, $b) {
  return strlen($a)-strlen($b);
});

//$arr = array('foo', 'hello', 'bazbar');
```

Happyr

# Syntactic sugar

```
<?php

$arr = array('hello', 'bazbar', 'foo');
usort($arr, function($a, $b) {
  return strlen($a)-strlen($b);
});
```

```
<?hh

$arr = array('hello', 'bazbar', 'foo');
usort($arr, ($a, $b) ==> strlen($a)-strlen($b));
```

Happyr

# Syntactic sugar

```php
<?php

$input = array(2, 3, 5, 6, 7);
$factor = 3;

$result = array_filter(
  array_map(
    function($a) use ($factor) {
      return $a * $factor;
    }, $input
  ), function($a) {
    return $a % 2 == 1;
  }
);

//$result = array(9, 15, 21);
```

Happyr

# Syntactic sugar

```
<?hh

$input = array(2, 3, 5, 6, 7);
$factor = 3;

$result = array_filter(
  array_map($a ==> $a * $factor, $input),
  $a ==> $a % 2 == 1
);

//$result = array(9, 15, 21);
```

Happyr

# Traits

```
trait Foo {
    public function bar() {
        return $this->biz();
    }
}

class Baz {
    use Foo;
    public function biz() {}
}
```

Happyr

# Traits with requirements

```
trait Foo {
    require extends A;
    require implements B;

    public function bar() {
        return $this->biz();
    }
}
```

Happyr

# Arrays are great!

```
$vector = [4,7,1,1];
$map = array(‘foo’=>47, ‘bar’=>11);
$set = array_unique(array(4,7,1,1));
$object = array(
    ‘name’=>’Tobias’,
    ‘age’=>25,
    ‘twitter’=>’@tobiasnyholm’,
);

$gender = $object[‘gender’];
```

"Arrays cannot be type checked"

*–Tobias Nyholm, (just now)*

Happyr

# Shape

```
type Link = shape(
  'url' => string,
  'title' => string,
  'number' => int,
);

// …
$var = shape(“url”=>’..’, “title”=>””, “number”=> 2);
```

Happyr

# Generics

```
class Storage<T> {
    protected T $var;

  public function __construct(T $var) {
        $this->var = $var;
    }

  public function get() {
        return $this->var;
  }

  public function set(T $var) {
        $this->var = $var;
    }
}
```

```
function getIntStorage(): Storage<int> {
    return new Storage(4711);
}
function getStrStorage(): Storage<string> {
    return new Storage("Hello");
}

function main() {
  $int = getIntStorage();
  $str = getStrStorage();

  $str->set("Foobar"); // No error
  $int->set("World"); // Error
}

main();
```

Happyr

# How to get started?

- Online tutorial
  (http://hacklang.org/tutorial)

- Start coding on you machine
  (http://developer.happyr.com/start-coding-with-hack)

- Miles Johnson has created Titon, “a fullstack Hack framework”.
  (https://github.com/titon/framework)

- Study the Happyr ExcerptBundle
  (https://github.com/Happyr/ExcerptBundle)