NEW FEATURES AND PERFORMANCE

A BRIEF TO PHP 7.3

Laruence
PHP DEVELOPER

SELF INTRODUCTION

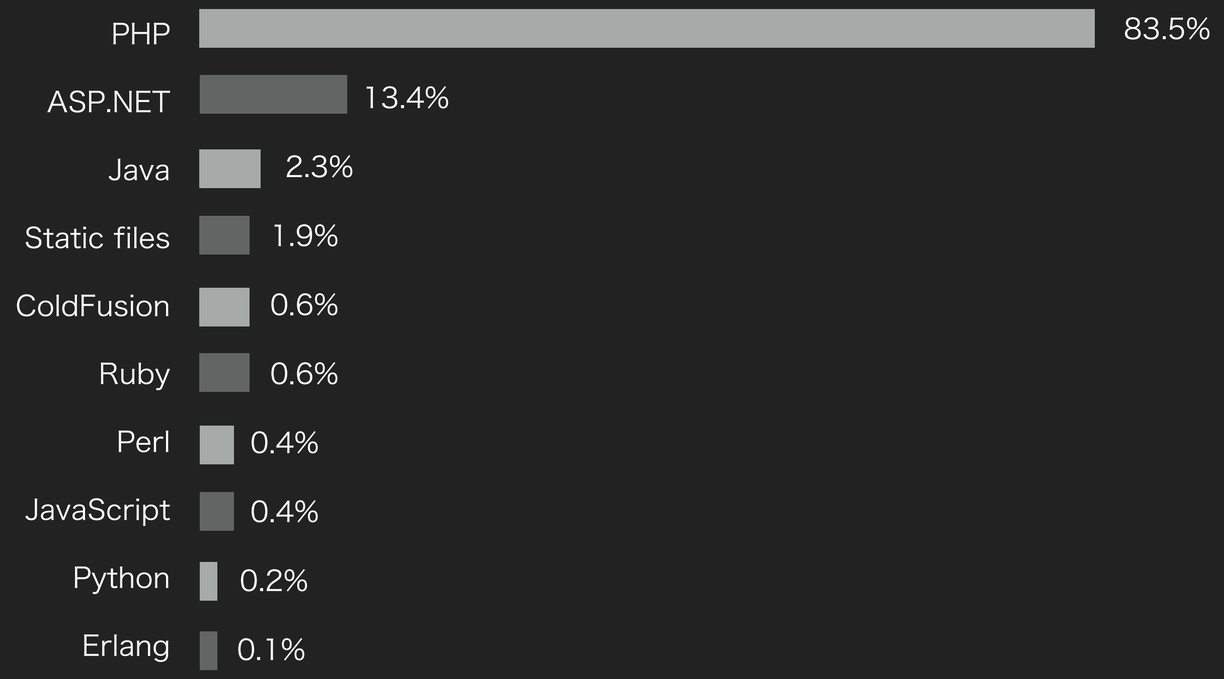
A PROGRAMER

- Author Of Yaf, Yar, Yac, Yaconf, Taint Projects
- Maintainer Of Opcache, Msgpack Projects
- PHP Core Developer Since 2011
- Zend Consultant Since 2013
- PHP7 Core Developer
- Chief Software Architect At Lianjia Since 2015

PHP HISTORY

BORN FOR WEB

- Created In 1994 By Rasmus Lerdorf
- 20+ Years Programming Language
- Most Popular Web Service Program Language
- Php7 Is Released At 3 Dec 2015
- Latest Version Is PHP7.2.5



W3Techs.com, 1 April 2018

HEREDOC AND NEWDOC

- The Ending Token No Loner Needs To Be The First Strings Of Line
- The Ending Token Can Be Indented
- The Exact Number Of Spaces/Tabs Used In The Ending Token Will Be Striped

```php
function example($arr) {
    $out = "";
    foreach($arr as $str) {
        $out .= <<<DOC
            repeat {$str}
 DOC;
    }
    return $arr;
}
```

```php
function example($arr) {
    $out = "";
    foreach($arr as $str) {
        $out .= <<<DOC
            repeat {$str}
        DOC;
    }
    return $arr;
}
```

TRAILING COMMA IN FUNC CALLS

- Allow Trailing Comma In Function And Method Calls
- Only One Extra Comma Are Allowed
- You Can Not Use This In Function/Method Declarations

```php
function func_call($num1, $num2, $num3, $num4) {
  return $num1 + $num2 + $num3 + $mum4;
}
```

```php
func_call(
  $argument1,
  $argument2,
  $argument3,
  $argument4,
);
```

REFERENCE IN LIST

- Items In List Can Be Assigned By Reference

```php
$animal = ['dog', 'cat'];

list($a, &$b) = $animal;

$b = "monkey";

var_dump($animal);

//output:
//array(2) {
//  [0]=>
//  string(3) "dog"
//  [1]=>
//  &string(5) "monkey"
//}
```

```php
$animal = ['dog', 'cat'];

[$a, &$b] = $animal;

$b = "monkey";

var_dump($animal);

//output:
//array(2) {
//  [0]=>
//  string(3) "dog"
//  [1]=>
//  &string(5) "monkey"
//}
```

HRTIME

- Hrtime Is A High-Resolution Monotonic Timer
- It Doesn’t Depends On The System Clock

```php
$start = hrtime(true);
//..........
$end = hrtime(true);

$elapsed = $end - $start;
```

HRTIME

- Hrtime Is A High-Resolution Monotonic Timer
- It Doesn’t Depends On The System Clock

```php
$start = hrtime(true);
//..........
$end = hrtime(true);

$elapsed = $end - $start;
```

MULTIPLY NEW FEATURES & BUGFIX

- Add Net_Get_Interfaces()
- Instanceof Now Allows Literals As The First Operand
- New Flag Json_Throw_On_Error, Json_Encode/Decode Could Throw Exception Instead Warning Now
- Added Is_Countable() Function, To Check Whether A Value May Be Passed Tocount()
- Multiply New Methods Added In Gmp
- Https://Github.Com/Php/Php-Src/Blob/Master/Upgrading

FASTER NEW GC

- Composer Gc Performance Problem Resolved
- Multiple Bugs Fixed In The Mean Time

Disable GC when computing deps, refs #3482

master

Browse code

Seldaek authored Dec 2, 2014 — 1 parent 91dd999 commit ac676f47f7bbc619678a29deae097b6b0710b799

Showing 1 changed file with **2 additions** and **0 deletions**.

Unified | Split

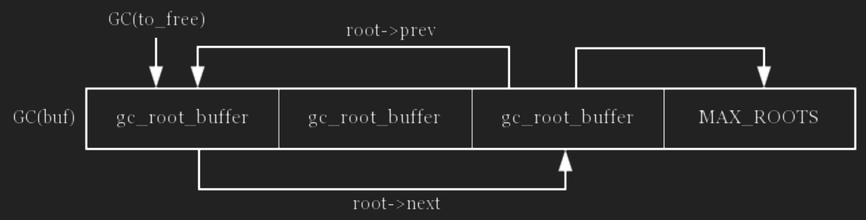
2 src/Composer/Installer.php — View

```
@@ -160,6 +160,8 @@ public function __construct(IOInterface $io, Config $config, RootPackageInterfac
160 160      */
161 161     public function run()
162 162     {
    163 +         gc_disable();
    164 +
163 165         if ($this->dryRun) {
164 166             $this->verbose = true;
165 167             $this->runScripts = false;
```

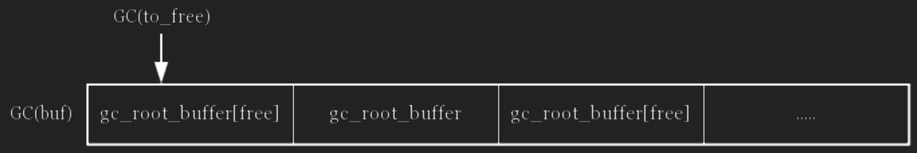
FASTER NEW GC

- Removed Double Link List, Plain Array Instead
- Dynamically Resize-Able Root Buffers
- Dynamically Gc Threshold Adjustment
- Reduced Gc_Buffer_Root Size From 32 To 8 (X86_64)
- 5X Boost Performance In Tests

```c
typedef struct _gc_root_buffer {
    zend_refcounted          *ref;
    struct _gc_root_buffer   *next;
    struct _gc_root_buffer   *prev;
    uint32_t                 refcount;
} gc_root_buffer;
```




```c
typedef struct _gc_root_buffer {
    zend_refcounted *ref;
} gc_root_buffer;
```



SIMD OPTIMIZATION FRAMEWORK

- Runtime Cpu Feature Detection(Sse, Avx)
  - Zend_Cpu_Support_Sse42
- Cpu Specific Version Runtime Selection (Ifunc/Function Pointer)
- Addslashes
  - 2X+ Performance Improved In Test

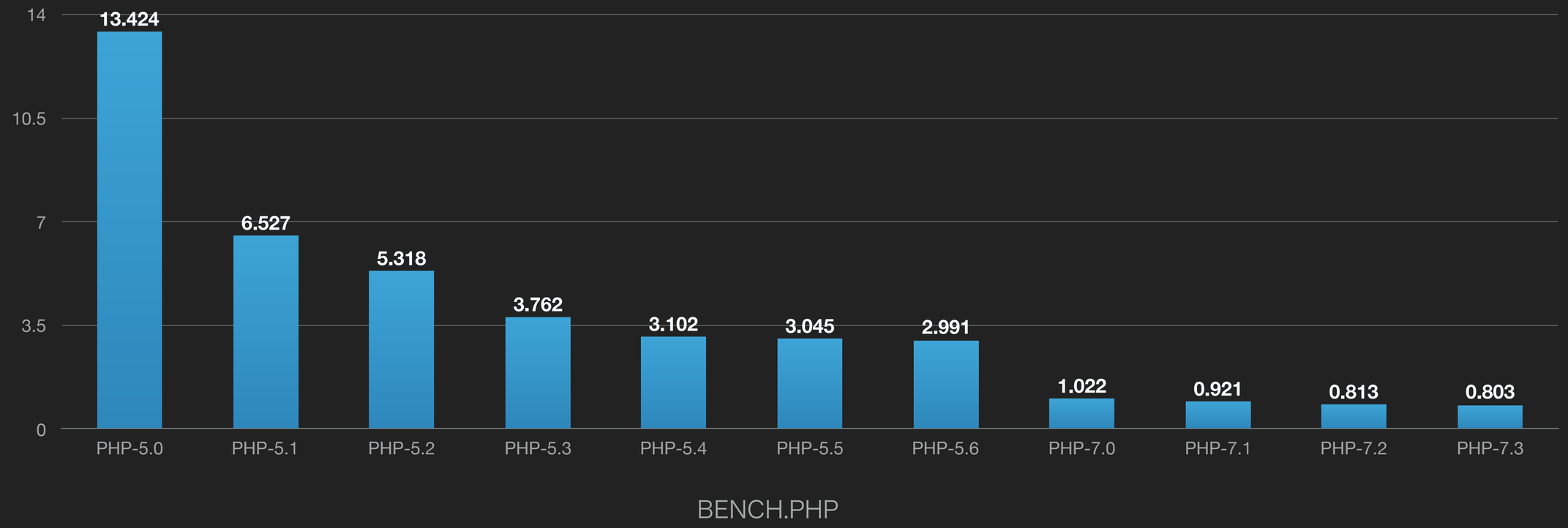
0x00	n	o	s	l	a	h	s	s	.	p	l	a	i	n	t	e
0x10	a	d	d	s	l	a	s	h	e	\	s	.	\r	\n	\n	o
0x20	p	l	a	i	n	t	e	x	t	s	t	r	i	n	g	.

BASE64 SIMD OPTIMIZATION

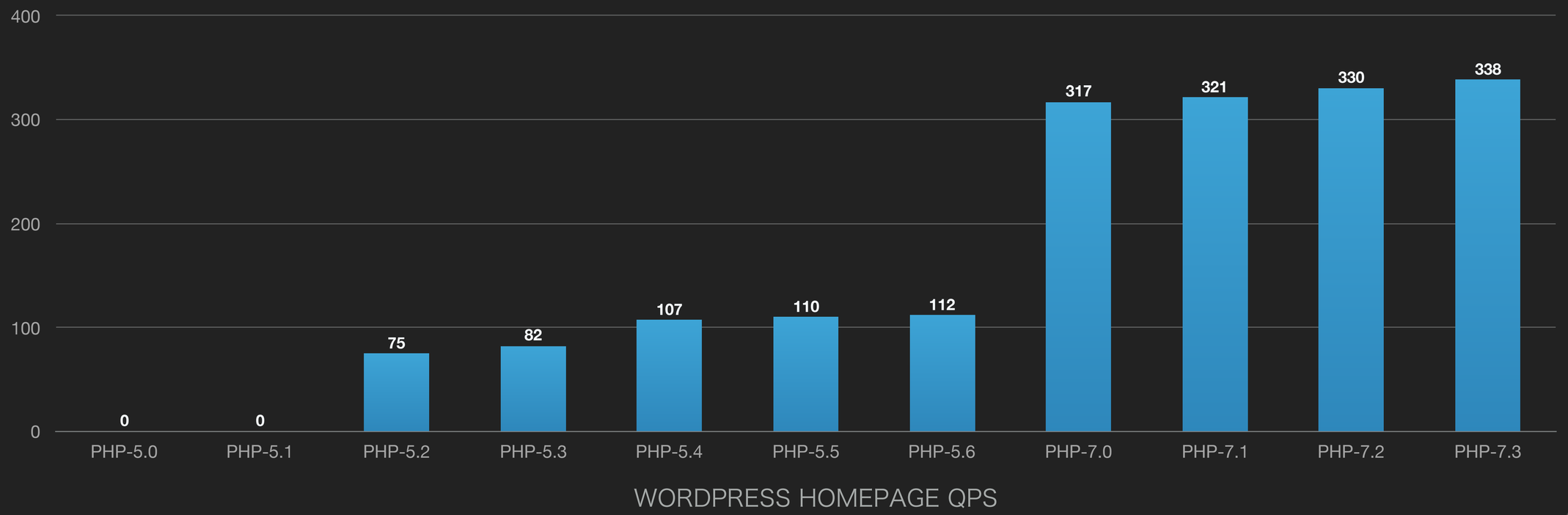
- Base64_Encode/Decode Optimized
- 3X Performance Improved In Test (Sse42)
- Http://Www.Alfredklomp.Com/Programming/Sse-Base64/

```php
$retry = 100;
$start = microtime(true);
while (--$retry) {
    $length = (int)(1000/$retry);
    $str = file_get_contents("/dev/urandom", false, NULL, 0, $length);
    //echo "String lenght: ", $length, "\n";
    $time = microtime(true);
    for ($i = 0; $i < 1000000; $i++) {
        base64_encode($str);
    }
    $time = microtime(true) - $time;
    //echo "Used time: ", $time , "s", "\n";
}
echo "Total used time: ", microtime(true) - $start, "s\n";
```

PERFORMANCE EVOLUTION



PERFORMANCE EVOLUTION



PHP 7.3 ROADMAP

- Jun 2018, Alpha1
- Jul 2018, Beta
- Aug 2018, Rc
- Nov 2018, Ga

THANKS

PHP: KEEP GETTING FASTER