用 Async I/O 解放 PHP Web 效能

Ricky Su

PHPConf Taiwan

2015/10/09

About me

Ricky 是我 ~~哥~~

- Symfony 愛好者
- PHP也有Day 固定攝影師。
- 目前在 Niceday 玩體驗 擔任高級水電工。
- ricky@ez2.us
- http://www.facebook.com/ricky.su.35
- https://github.com/RickySu/

Agenda

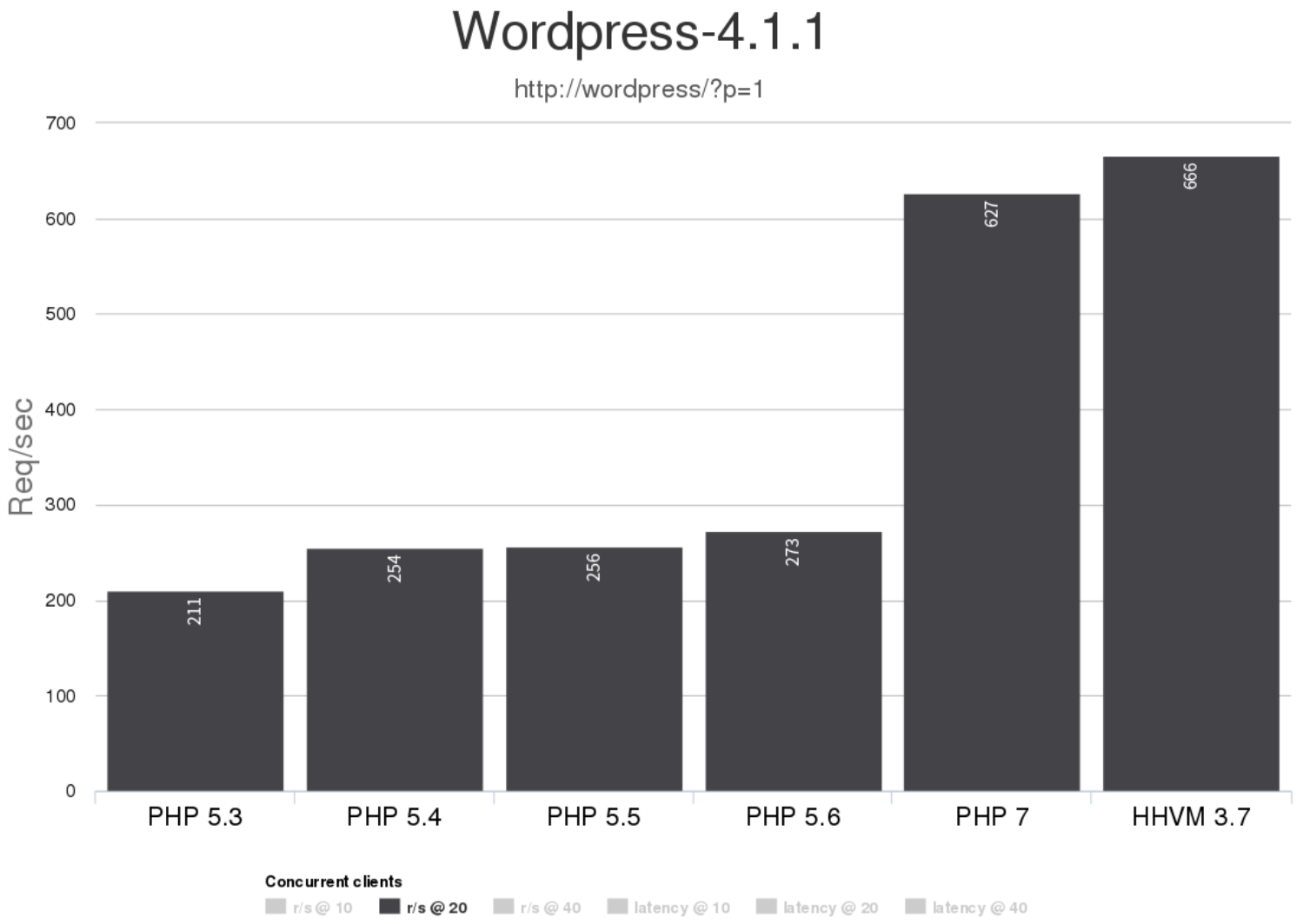
- PHP 慢在那
- 如何加速執行
- Async I/O Extensions
- Aync I/O 應用
- Performance Benchmark

PHP

Apache

https://www.pinterest.com/pin/441634307179323188/

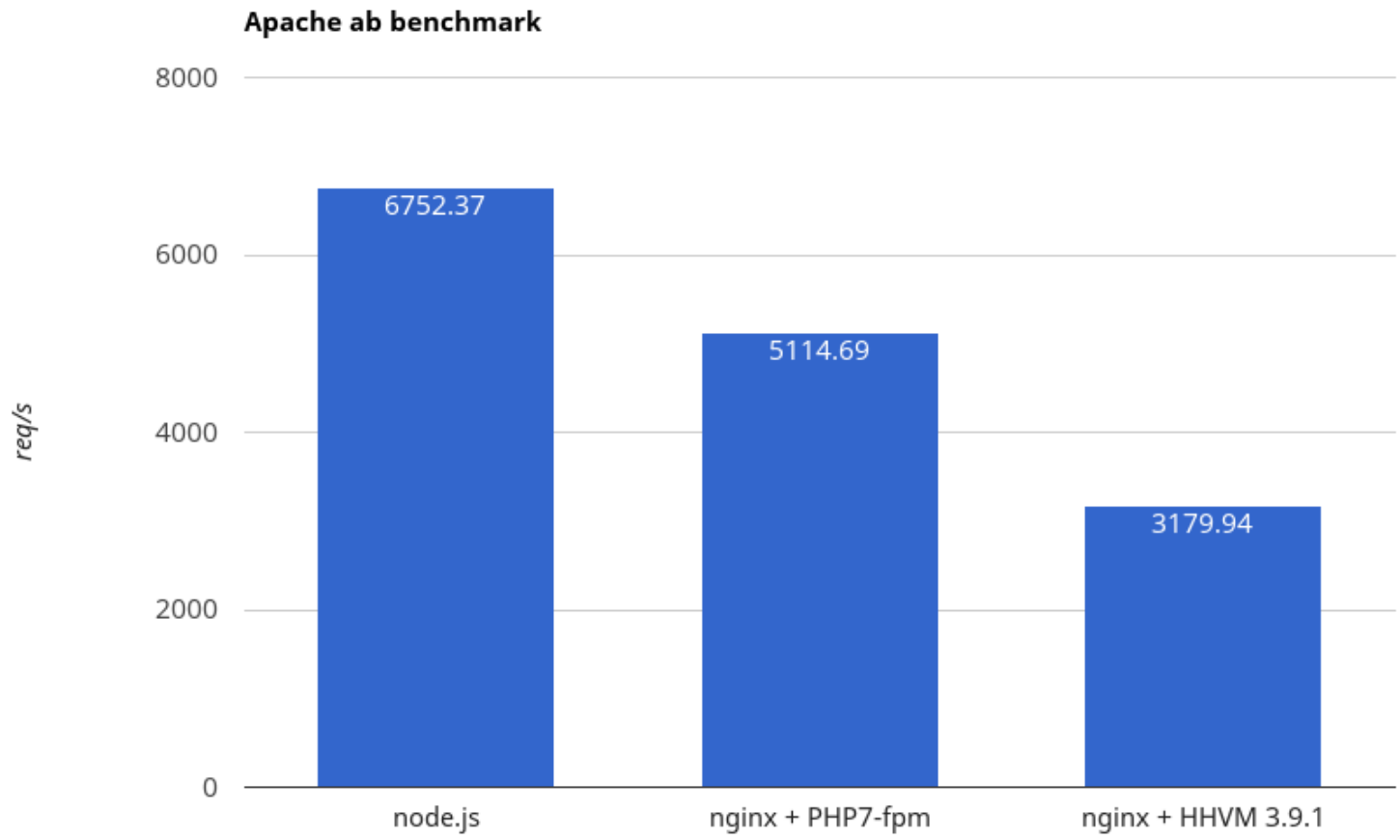
PHP 慢？



http://talks.php.net/velocity15#/wpbench

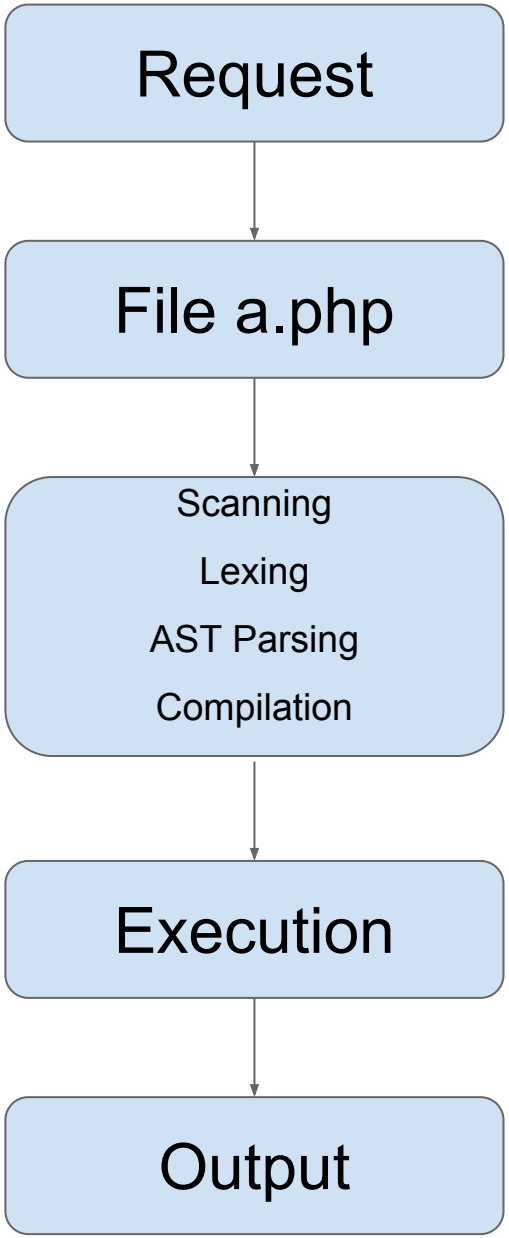
PHP7 夠快了？

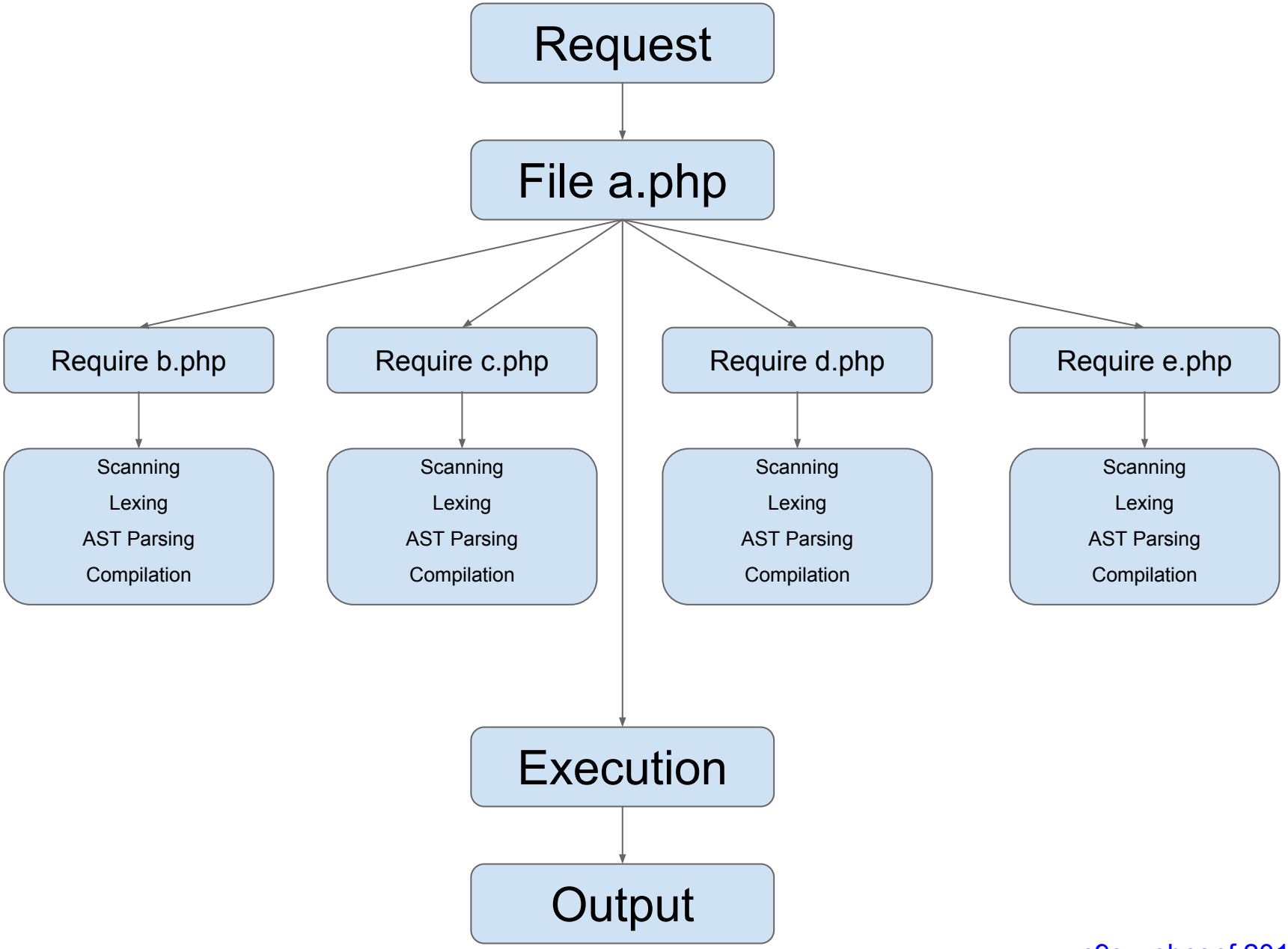
die ‘hello world’;

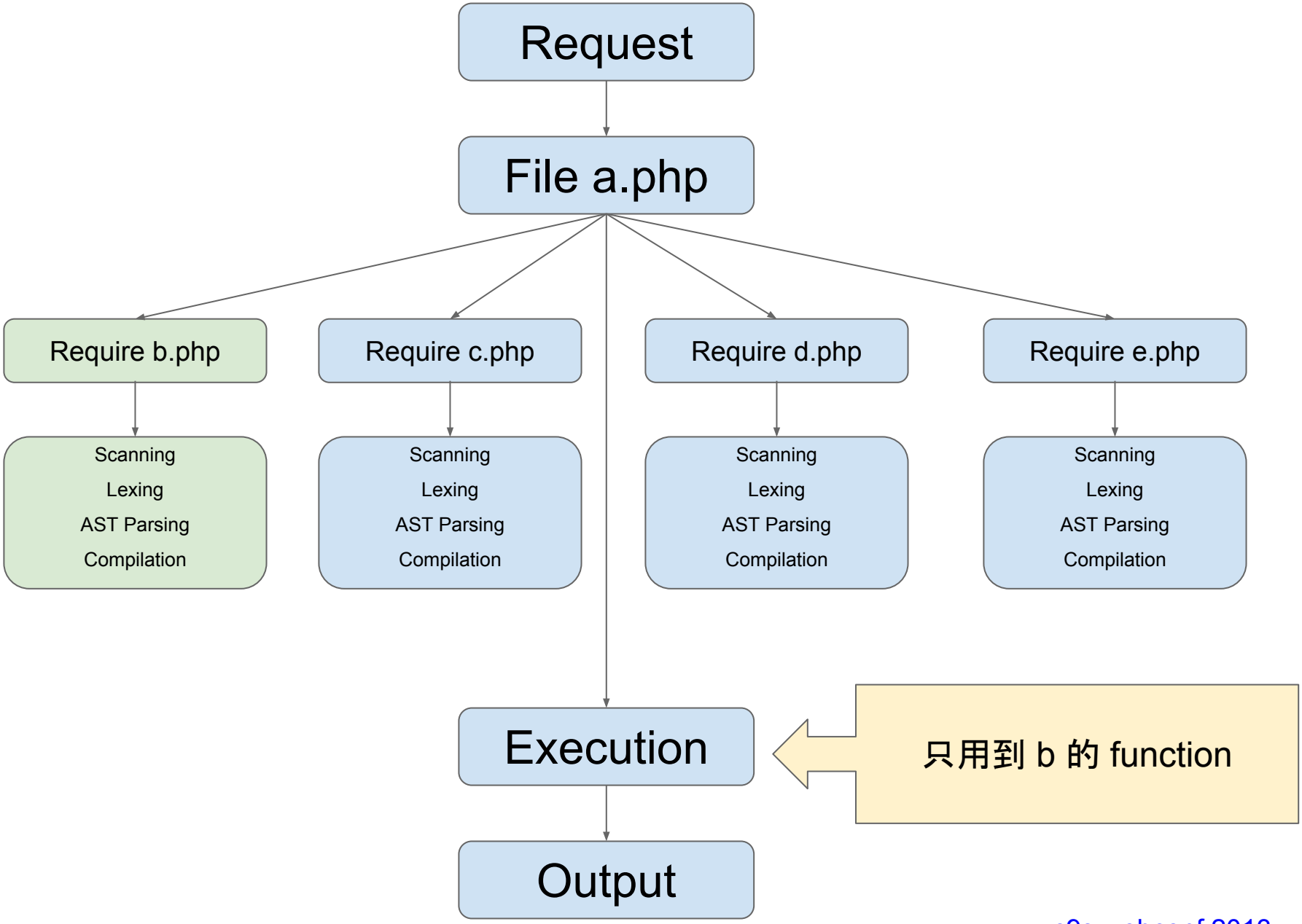


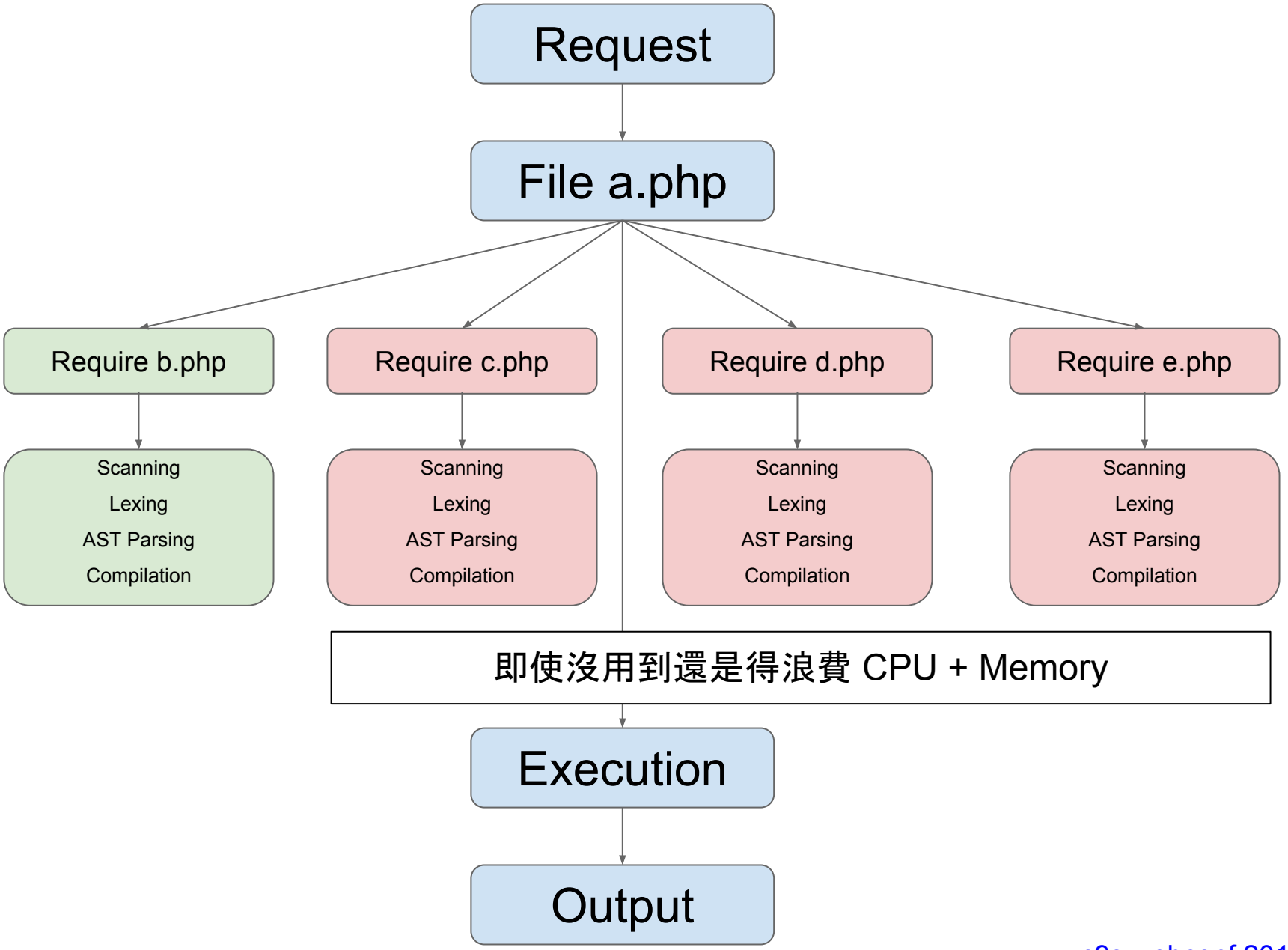
HHVM 不是有 JIT?

Zend Engine Compilation

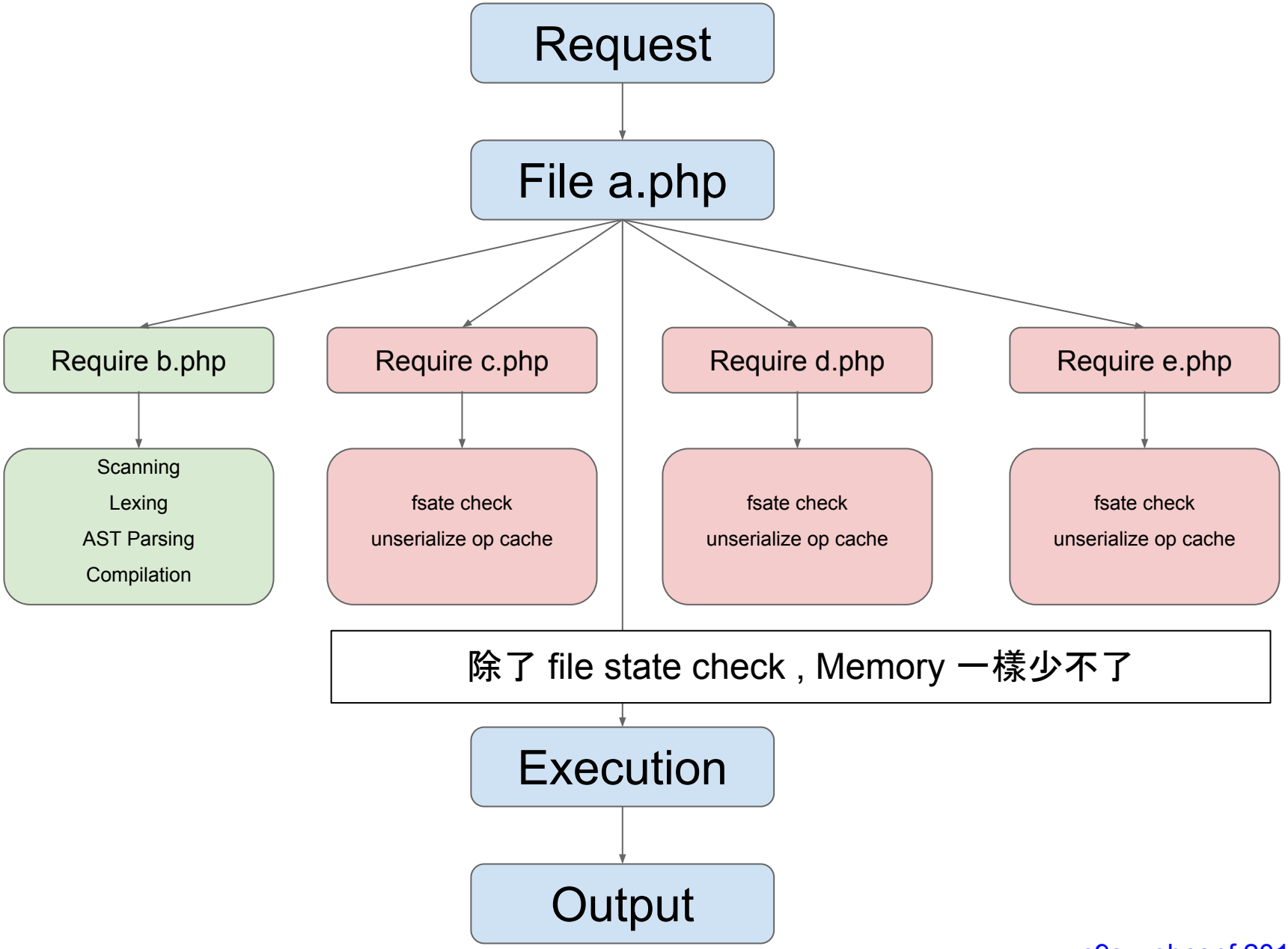



Request
File a.php
Require b.php
Require c.php
Require d.php
Require e.php
Scanning
Lexing
AST Parsing
Compilation
Scanning
Lexing
AST Parsing
Compilation
Scanning
Lexing
AST Parsing
Compilation
Scanning
Lexing
AST Parsing
Compilation
Execution
Output


Request
File a.php
Require b.php
Require c.php
Require d.php
Require e.php
Scanning
Lexing
AST Parsing
Compilation
Scanning
Lexing
AST Parsing
Compilation
Scanning
Lexing
AST Parsing
Compilation
Scanning
Lexing
AST Parsing
Compilation
Execution
只用到 b 的 function
Output

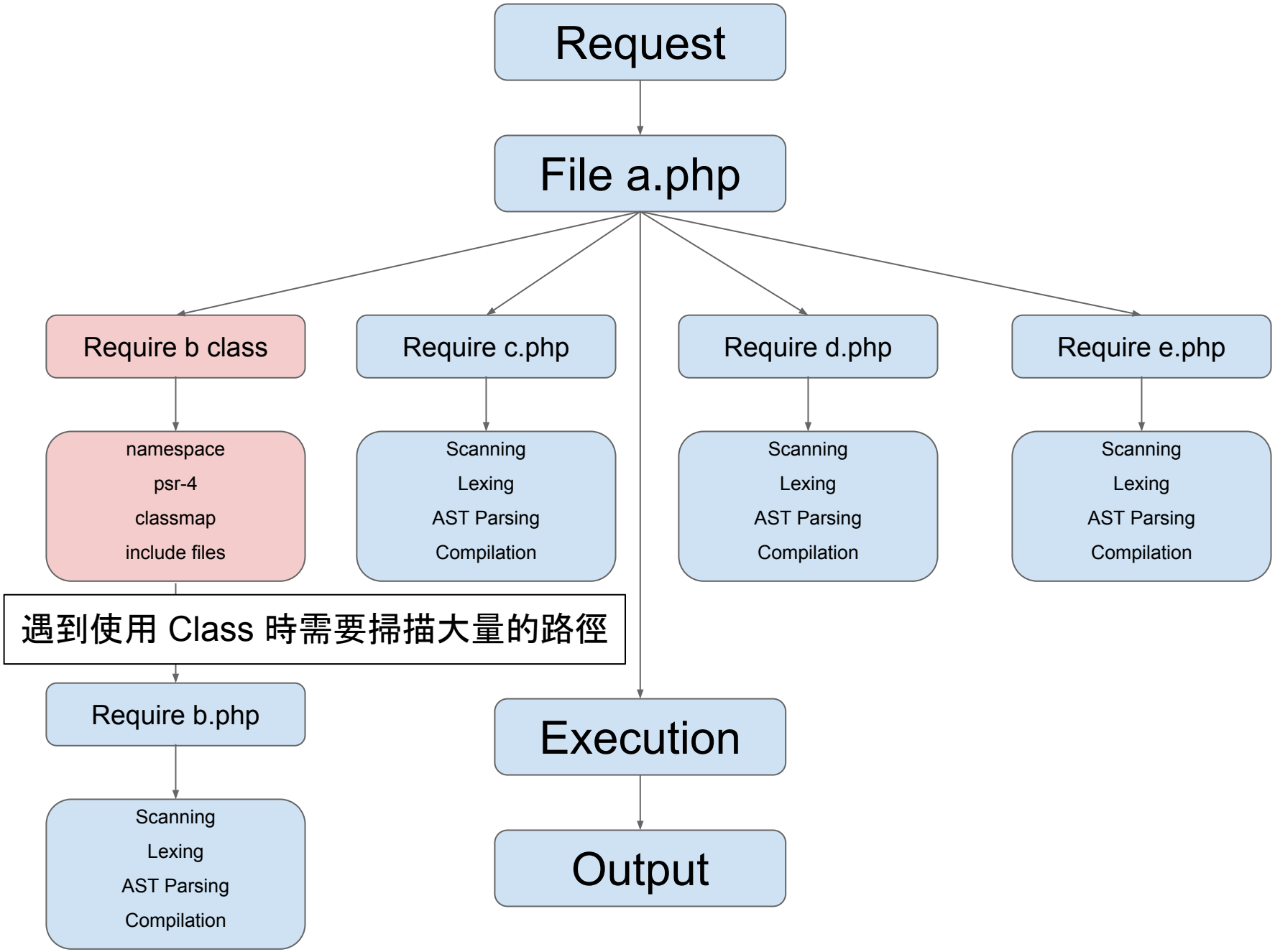

Request
File a.php
Require b.php
Require c.php
Require d.php
Require e.php
Scanning
Lexing
AST Parsing
Compilation
Scanning
Lexing
AST Parsing
Compilation
Scanning
Lexing
AST Parsing
Compilation
Scanning
Lexing
AST Parsing
Compilation
即使沒用到還是得浪費 CPU + Memory
Execution
Output

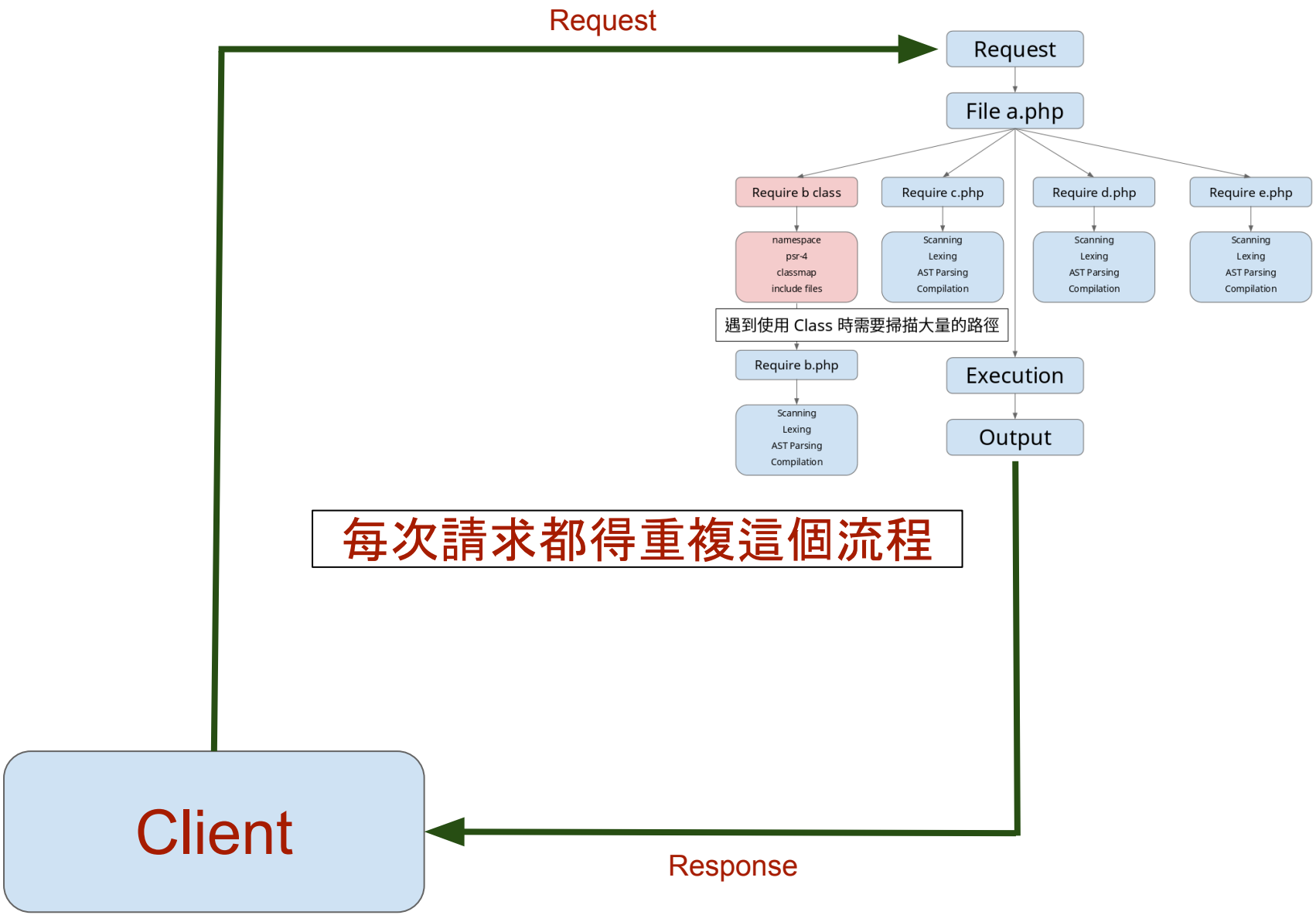
開啟
OPCache
問題就解決了？


Request
File a.php
Require b.php
Require c.php
Require d.php
Require e.php
Scanning
Lexing
AST Parsing
Compilation
fsate check
unserialize op cache
fsate check
unserialize op cache
fsate check
unserialize op cache
除了 file state check , Memory 一樣少不了
Execution
Output

更大的災難

Composer Autoload


Request
File a.php
Require b class
Require c.php
Require d.php
Require e.php
namespace
psr-4
classmap
include files
Scanning
Lexing
AST Parsing
Compilation
Scanning
Lexing
AST Parsing
Compilation
Scanning
Lexing
AST Parsing
Compilation
遇到使用 Class 時需要掃描大量的路徑
Require b.php
Scanning
Lexing
AST Parsing
Compilation
Execution
Output


Request
Request
File a.php
Require b class
Require c.php
Require d.php
Require e.php
namespace
psr-4
classmap
include files
Scanning
Lexing
AST Parsing
Compilation
Scanning
Lexing
AST Parsing
Compilation
Scanning
Lexing
AST Parsing
Compilation
遇到使用 Class 時需要掃描大量的路徑
Require b.php
Scanning
Lexing
AST Parsing
Compilation
Execution
Output
每次請求都得重複這個流程
Client
Response

c9s webconf 2013

這就是 WordPress 慢的原因

http://talks.php.net/presentations/slides/intro/wp_included.png

這就是
WordPress
~~慢的原因~~
用了 PHP 的結果

http://talks.php.net/presentations/slides/intro/wp_included.png

WordPress 4.3 will be rewritten in Node.js

http://webuilddesign.com/wordpress-4-3-will-be-rewritten-in-node-js/

Persistance Application Server

避免重複初始化

使用 Async I/O

避免 Blocking

來個 node.php 吧

Async I/O Liberaries

Async I/O Liberaries

- libevent
  - Chromium
  - memcached
- libev
  - node.js ~ 0.8
- libuv
  - node.js 0.9 ~

Async I/O Extensions

Async I/O extensions

- libevent
  - libevent
  - event
- libev
  - ev
- libuv
  - php_ext_uv
  - hhvm-ext-uv

我們需要一個工具箱

http://www.greenbiz.com/blog/2014/02/13/better-buildings-big-data-retroficiency-wegowise

工具箱

- Async I/O - libuv
  - https://github.com/libuv/libuv
- Fast Http Parser - http-parser
  - https://github.com/nodejs/http-parser
- Fast Router - R3
  - https://github.com/c9s/r3

libuv

libuv extension

- php
  - https://github.com/RickySu/php_ext_uv
- hhvm
  - https://github.com/RickySu/hhvm-ext-uv

extension features

- tcp socket
- udp socket
- timer
- signal
- DNS Resolver
- SSL/TLS with SNI

TODO:

IPV6 , Unix Socket support

TCP Echo Server

```php
<?php

// TCP Echo Server
$loop = new UVLoop();
$server = new UVTcp($loop);
$server->listen($host, $port, function($server) {
    $client = $server->accept();
    $client->setCallback(function($client, $recv) {
        //on receive
        $client->write($recv);
    }, function($client, $status) {
        //on data sent
        $client->close();
    }, function($client) {
        //on error maybe client disconnect.
        $client->close();
    });
});
$loop->run();
```

SSL Echo Server

```php
<?php

//SSL Echo Server
$loop = new UVLoop();
$server = new UVSSL($loop);
$server->setCert(file_get_contents("server.crt"));    //PEM format
$server->setPrivateKey(file_get_contents("server.key"));  //PEM format
$server->listen($host, $port, function($server) {
    $client = $server->accept();
    $client->setSSLHandshakeCallback(function($client) {
        echo "ssl handshake ok\n";
    });
    $client->setCallback(function($client, $recv) {
        //on receive if ssl handshake finished.
        //otherwise you won't receive any data before ssl handshake finished
        $client->write($recv);
    }, function($client, $status) {
        //on data sent
        $client->close();
    }, function($client) {
        //on error maybe client disconnect.
        $client->close();
    });
});
$loop->run();
```

Timer

```php
<?php

$loop = new UVLoop();
$tickonece = [];
$tick = [];
$time = microtime(true);

$timerOnece = new UVTimer($loop);
$timerOnece->start(function($timerOnece)use(&$tickonece, $time) {
    $tickonece[] = round((microtime(true) - $time) * 10);
}, 500);

$timer = new UVTimer($loop);
$timer->start(function($timer)use(&$tick, $time) {
    $tick[] = floor((microtime(true) - $time) * 10);
    if (count($tick) > 5) {
        $timer->stop();
    }
}, 500, 500);
$loop->run();

//$tickonce => [5]
//tick => [5, 10, 15, 20, 25, 30]
```

web-util

web-util

- php-extension
  - https://github.com/RickySu/php_ext_web_util
- hhvm-extension
  - https://github.com/RickySu/hhvm-ext-web-util
- php
  - https://github.com/RickySu/php-webutil

web-util features

- native http parser
- nikic/fast-route
- R3 route
- PSR-7 compliance request object
- PSR-7 compliance response object

Http Parser

```php
<?php

$parser = new WebUtil\Http\Parser();

$parser->setOnParsedCallback(function($parsedData){
    //parsed http request data
});


$parser->feed($rawData);
```

```
Array
(
    [Request] => Array
        (
            [Method] => PUT
            [Target] => /category/2
            [Protocol] => HTTP
            [Protocol-Version] => 1.1
        )
    [Header] => Array
        (
            [Host] => localhost
            [User-Agent] => Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:40.0) Gecko/20100101 Firefox/40.0
            [Accept] => application/json, text/plain, */*
            [Accept-Language] => zh-TW,en-US;q=0.7,en;q=0.3
            [Accept-Encoding] => gzip, deflate
            [DNT] => 1
            [Content-Type] => application/json;charset=utf-8
            [Referer] => https://www.google.com/
            [Content-Length] => 62
            [Cookie] => Array
                (
                    [device_view] => mobile
                    [PHP_SESS_ID] => 1jjdfksjhdkjhd
                )
            [Authorization] => Basic DFJjkgas2873asaQA==
            [Connection] => keep-alive
        )
    [Query] => Array
        (
            [Path] => /admin/category/2
            [Param] => Array
                (
                )

        )
    [Content] => {"name":"test","enname":"family","description":"test"}
    [Content-Parsed] => Array
        (
            [name] => test
            [enname] => family
            [description] => test
        )
)
```

Async DB Query

Async MySQL Query

```php
<?php

$link = new mysqli($host, $user, $pass, $db);
$link->query("SELECT SLEEP(10), 1;", MYSQLI_ASYNC);

$callback = function($link) {
    if ($result = $link->reap_async_query()) {
        print_r($result->fetch_row());
        if (is_object($result)){
            mysqli_free_result($result);
        }
    }
};

$loop = new UVLoop();
$idle = new UVIdle($loop);
$idle->start(function(UVIdle $idle) use($link, $callback) {
    $links = $errors = $reject = array($link);
    if (!mysqli::poll($links, $errors, $reject, 0)) {
        return;
    }
    foreach ($links as $link) {
        $callback($link);
    }
    $idle->stop();
});
$loop->run();
```

REST API

- redis
  - https://github.com/nicolasff/webdis
- mongoDB
  - http://docs.mongodb.org/ecosystem/tools/http-interfaces/
- CouchDB

Async I/O 應用

respond

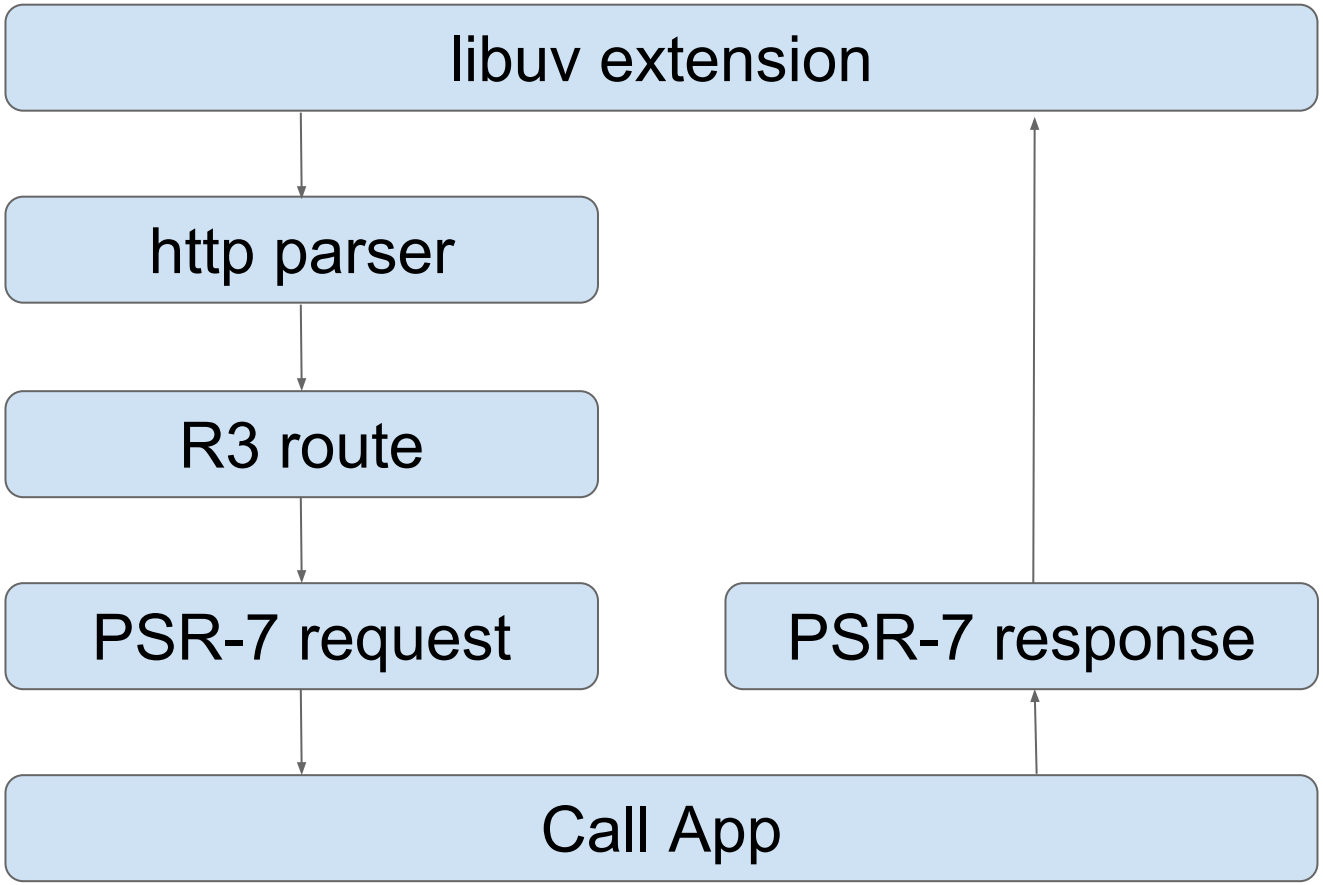
Fast, async I/O web framework for php

https://github.com/RickySu/respond

respond

```php
<?php
use WebUtil\Http\Request\ServerRequest;
use WebUtil\Http\Response\Response;

$app = new Respond\App\WebApp();

return $app->listen('0.0.0.0', 8080)
    ->get('/{id:\\d+}', function(ServerRequest $request){
        return 'match id';
    })
    ->post('/{id:\\d+}.html', function(ServerRequest $request){
        return 'match id.html';
    })
    ->put('/{id:\\d+}-{id2:\\d+}.html', function(ServerRequest $request){
        return $request->getAttribute('id');
    })
    ->defaultRequest(function(ServerRequest $request){
        return new Response('page not found', 404);
    });
```

respond flow



PHPSGI

PHP Server Gateway Interface

https://github.com/phpsgi/phpsgi

PHPSGI

A PHPSGI Server is a PHP program providing an environment for a PHPSGI application to run in.

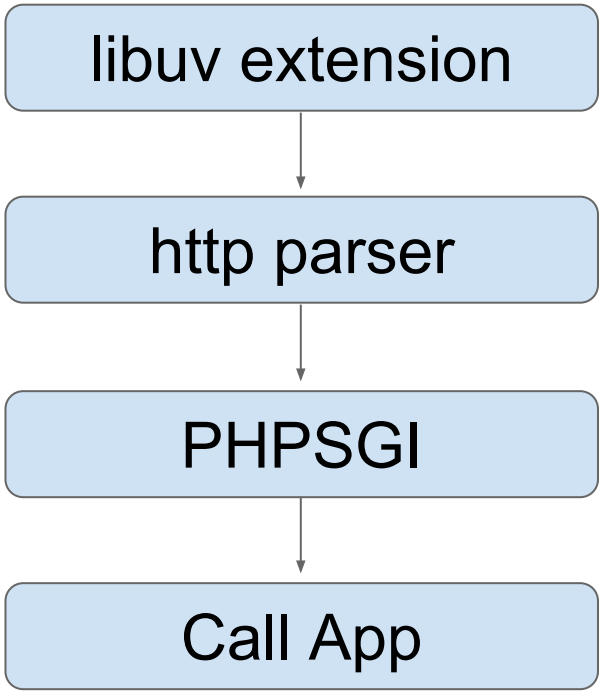
PHPSGI application

```php
<?php

$app = function(array & $environment, array $response) {
    // [response code,  response headers, body content ]
    return [ 200, [ 'Content-Type' => 'plain/text' ], 'Hello World' ];
};
```

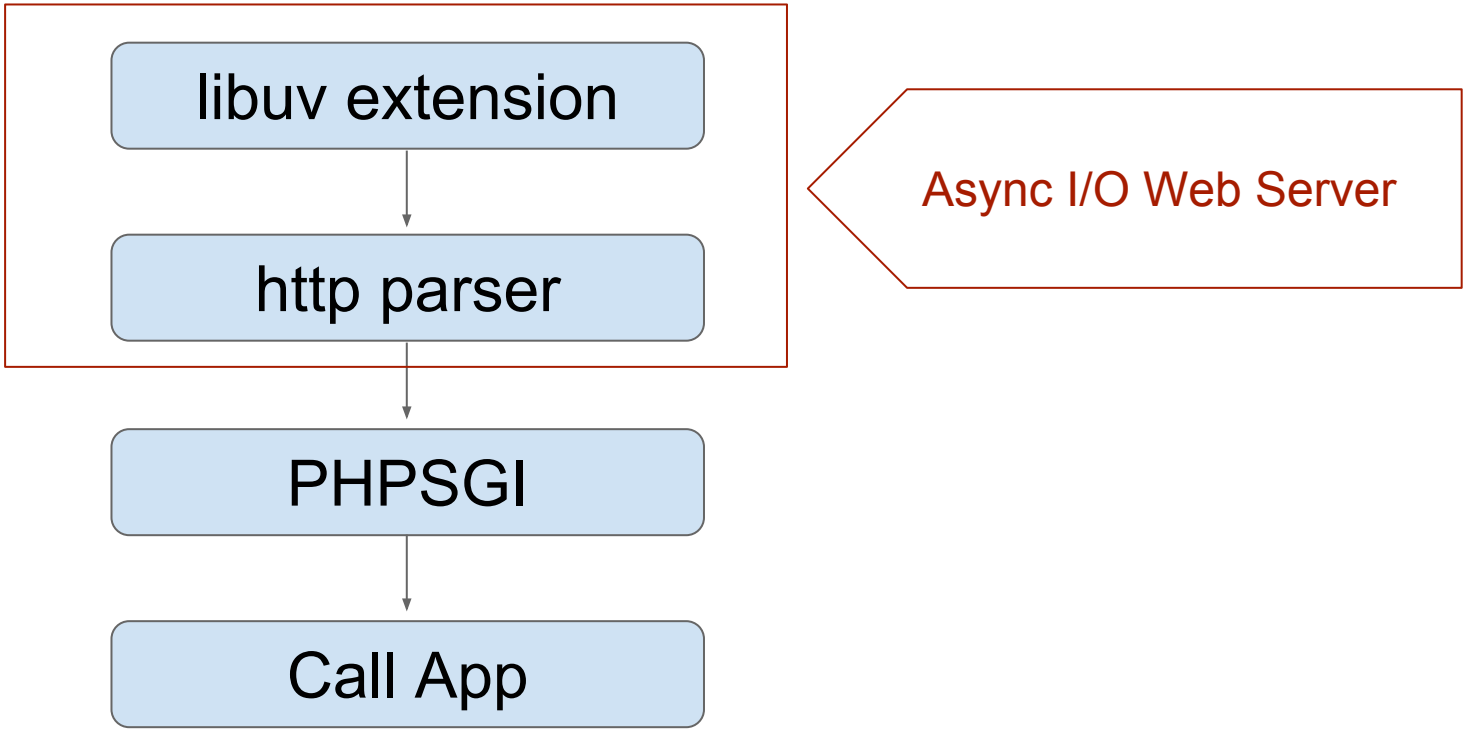
PHPSGI application

```php
<?php
use PHPSGI\App;

class MyApp implements App {
    public function call(array & $environment, array $response) {
        // [response code,  response headers, body content ]
        return [ 200, [ 'Content-Type' => 'plain/text' ], 'Hello World' ];
    }

    public function __invoke(array & $environment, array $response) {
        return $this->call($environment, $response);
    }
}
```

PHPSGI



PHPSGI


libuv extension
http parser
PHPSGI
Call App
Async I/O Web Server

benchmark

test case

- pecl libevent http server
- libuv + http parser
- respond
- golang
- node.js

environment

- ubuntu 14.04 x64
- i7-4720HQ CPU @ 2.60GHz
- 16GB Ram
- ab -c 500 -n 50000
- ab -c 10000 -n 50000

libevent http server

```php
<?php

$base = new EventBase();
$http = new EventHttp($base);
$socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
socket_bind($socket, '127.0.0.1', 8080);
socket_listen($socket, 0);
$http->accept($socket);
$http->setDefaultCallback(function($req) {
    $buffer = new EventBuffer();
    $buffer->add("hello world");
    $req->sendReply(200, "OK");
});
$base->dispatch();
```

libuv + http parser

```php
<?php

$content = "hello world";
$message = "HTTP/1.1 200 OK\r\n" .
        "Server: LibUV Server\r\n" .
        "Content-Type: text/plain; charset=utf-8\r\n" .
        "Content-Length: " . strlen($content) . "\r\n\r\n$content";

$loop = new UVLoop();
$server = new UVTcp($loop);
$server->listen('127.0.0.1', 8080, function($server) use($message) {
    $client = $server->accept();
    $parser = new \WebUtil\Parser\HttpParser();
    $parser->setOnParsedCallback(function($parsedArray) use($client, $message) {
        $client->write($message);
    });
    $client->setCallback(function($client, $data) use($parser) {
        $parser->feed($data);
    }, function($client) use($parser) {
        $parser->setOnParsedCallback(null);
        $client->shutdown(function($client) {
            $client->close();
        });
    }, function() {

    });
});

$loop->run();
```

respond

```php
<?php
use WebUtil\Http\Request\ServerRequest;

$app = new Respond\App\WebApp();

return $app->listen('0.0.0.0', 8080)
    ->defaultRequest(function(ServerRequest $request){
        return 'hello world';
    });
```

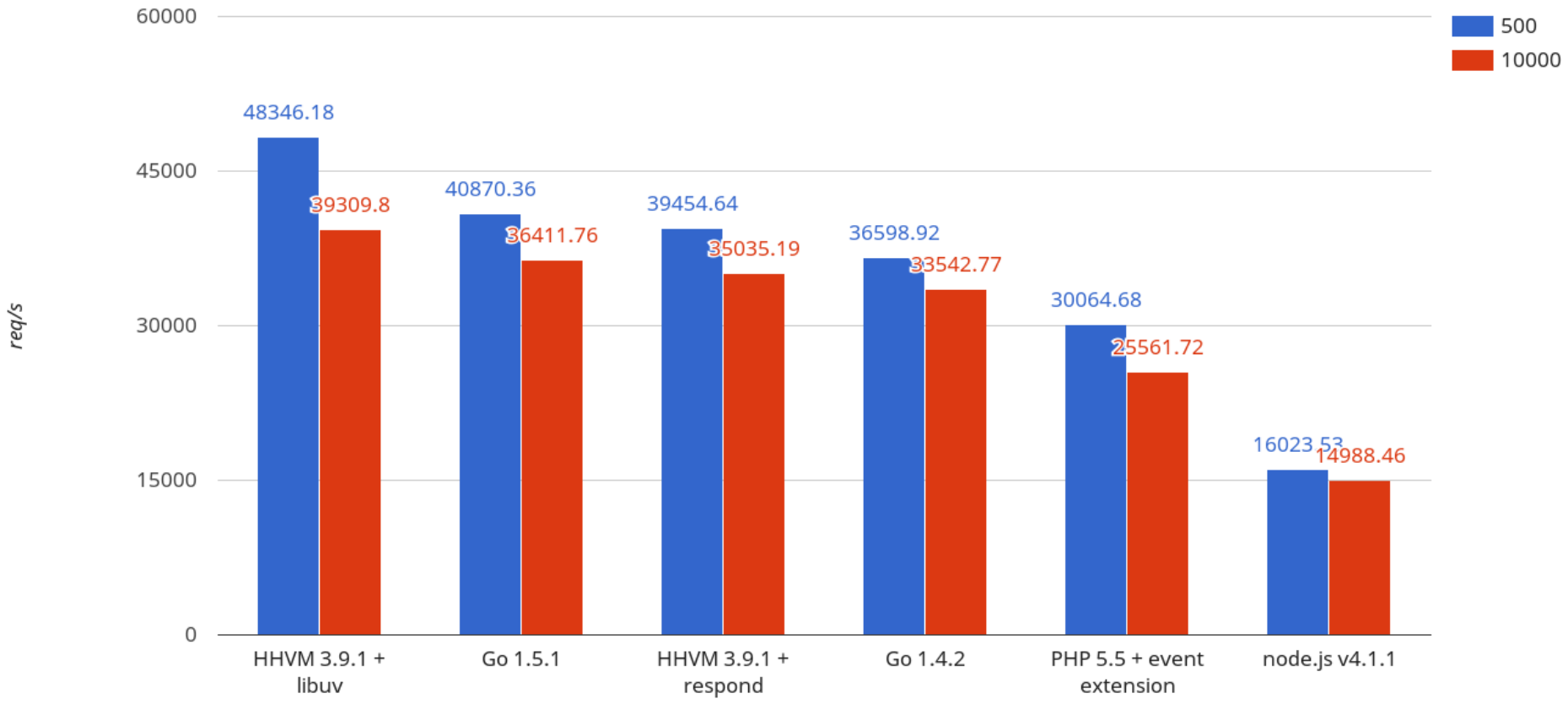
node.js web server

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('hello world');
}).listen(8080, '127.0.0.1');
```

golang web server

```go
package main

import (
    "fmt"
    "net/http"
)

func sayhelloName(w http.ResponseWriter, r *http.Request) {
    fmt.Fprintf(w, "Hello world")
}


func main() {
    http.HandleFunc("/", sayhelloName)
    http.ListenAndServe(":8080", nil)
}
```

Apache ab benchmark



有問題嗎?

Thanks