



AS NOVAS TECNOLOGIAS HACK E HHVM DO FACEBOOK E O WORDPRESS



Por Amon Caldas



APRESENTAÇÃO

- ▶ Mestrando em Computação e Sistemas
- ▶ Pós graduado em Engenharia de Software
- ▶ Graduado em Sistemas para Internet



Trabalha com engenharia de software, desenvolvimento web e mobile. Atualmente é bolsita da CAPES e atua no grupo de pesquisas GANGES com foco em Sistemas de Informações Geográficas (SIG) e análise de trajetórias.



AGENDA

- ▶ O que é?
- ▶ Por que surgiu a Hack?
- ▶ Histórico
- ▶ Por que se ligar?
- ▶ Conhecendo a HHVM
- ▶ Conhecendo a Hack
- ▶ Incompatibilidades e novo paradigma
- ▶ Já posso usar com WordPress?
- ▶ Demonstração



O QUE É HACK?



É uma linguagem de programação que interopera com PHP e apresenta um modelo de transição gradual com o conceito de anotação de tipos.



O QUE É HHVM?

É uma máquina virtual (HipHop Virtual Machine) para executar programas escritos em Hack e PHP.



POR QUE O FACEBOOK CRIOU UMA LINGUAGEM?

- ▶ Grandes empresas de software tendem a lançar suas próprias linguagens (go, C#, Swift).
- ▶ A empresa relata problemas de eficiência com o PHP
- ▶ Há dificuldades para se automatizar testes e detectar defeitos em tempo de desenvolvimento com PHP

HISTÓRICO DO FACEBOOK COM PHP



- ▶ 2008 - Começa a relatar problemas de overhead e inicia o trabalho em uma solução
- ▶ 2010 - Anuncia o tradutor (para c++) e compilador PHP chamado HipHop Compiler
- ▶ 2010 - A extensão para PHP chamada XHP é criada (frag. XMLs, XSS e tempo de desenvolvimento)

HISTÓRICO DO FACEBOOK COM PHP



- ▶ 2012 - Anuncia testes com uma máquina virtual para interpretação de PHP
- ▶ 2013 - Completa a migração para a nova máquina virtual com ganhos significativos de eficiência.
- ▶ 2014 (março) - Anuncia a linguagem Hack
- ▶ 2014 (novembro) - Anuncia o Hack Transpiler



POR QUE SE LIGAR NA HHVM

- ▶ HHVM suporta o WordPress a partir da versão 3.9
- ▶ HHVM melhora significativamente a performance do site



POR QUE SE LIGAR EM HACK?

- ▶ A maior rede social do mundo está migrando para Hack e já usa HHVM
- ▶ Hack aperfeiçoa (?) o PHP e facilita a depuração da aplicação



CARACTERÍSTICAS DA HHVM

- ▶ Utiliza uma abordagem de compilação JIT aliado a cache em SQLite (Já o OPcache usa a memória)
- ▶ Mantém a flexibilidade de desenvolvimento do PHP
- ▶ Suporta checagem de tipos em tempo de desenvolvimento.

CARACTERÍSTICAS DA HHVM



- ▶ Suporta os conceitos de tipos latentes e tipos por anotação.
- ▶ É capaz de inferir tipos através da observação do código.
- ▶ Introduz o conceito de modos, com o qual é possível definir o nível de rigor ou flexibilidade aplicado na checagem.



CARACTERÍSTICAS DA HHVM

- ▶ É de código aberto (licenças PHP e Zend).
- ▶ O analisador verifica os reflexos da alteração de um trecho de código em toda a aplicação hack.
- ▶ A verificação informa onde essas alterações impactaram e como.
- ▶ Suporta FastCGI, sendo compatível com Apache, Nginx, Lighttpd e outros.



QUANTO MAIS RÁPIDA É A HHVM?

Depende de vários fatores, mas...

- ▶ Facebook relata ganhos em até 11.6x
- ▶ O wpengine relata ser 3x a 6x mais rápido rodando WordPress
<http://wpengine.com/2014/11/19/hhvm-project-mercury/>
- ▶ O site kradalby.no indica que HHVM foi 4.9x mais rápido usando WordPress
<https://kradalby.no/wordpress-hhvm-benchmark.html>

CARACTERÍSTICAS DA LINGUAGEM HACK



- ▶ É de código aberto sob a licença BSD.
- ▶ Concilia o ciclo de desenvolvimento rápido do PHP com a definição de tipos de forma estática.
- ▶ Permite que erros sejam checados em tempo de desenvolvimento.

CARACTERÍSTICAS DA LINGUAGEM HACK



- ▶ Permite a convivência entre código Hack e PHP
- ▶ A extensão dos arquivos pode ser `.php` ou `.hh`
- ▶ Os blocos de código iniciam com `<?hh`

CARACTERÍSTICAS DA LINGUAGEM HACK



- ▶ Permite definir o nível de rigor ou flexibilidade (*strict*, *partial*, *decl* e *unsafe*) aplicado na checagem
- ▶ É estaticamente tipada
- ▶ Código estaticamente tipado interopera com o dinamicamente tipado
- ▶ Não pode ser “misturada” com HTML



MODOS HACK

- ▶ *Strict* → todos os tipos devem ser anotados
- ▶ *Partial* (padrão) → checa todos os tipos mas permite código com tipo não anotado
- ▶ *Decl* → não checa os tipos
- ▶ *Unsafe* → permite delimitar blocos de código não tipados ou inseguros dentro dos modos *strict* e *partial*

CÓDIGO PHP



```
<?php
```

```
class MinhaClasse {  
    public function alpha(){  
        return 1;  
    }  
    public function beta(){  
        return 'olá teste';  
    }  
}
```

```
function minhaFuncao(MinhaClasse $minhaInst){ //retorno não definido  
    if(condition)  
    {  
        return $minhaInst->alpha(); //retorna um inteiro  
    }  
    else  
    {  
        return $minhaInst->beta(); //retorna uma string  
    }  
}
```

CÓDIGO HACK



<?hh

```
class MinhaClasse {  
    public function alpha() : int {  
        return 1;  
    }  
    public function beta() : string {  
        return 'olá teste';  
    }  
}
```

```
function minhaFuncao(MinhaClasse $minhaInst) : string {  
    //corrija-me  
    return $minhaInst->alpha(); // o analisador detecta um erro aqui  
}
```

HACK - INFERÊNCIA



```
<?hh
```

```
class MinhaClasse {  
    public function fazer_algo() : Vector<int> {  
        $v = Vector {1, 2}; //tipo ainda não resolvido  
        $v[] = "a";  
        return $v; //neste ponto o verificador aponta erro  
    }  
}
```



TIPOS ADICIONADOS

- ▶ Nullable → (?int, ?myclassname)
- ▶ Mixed → (tipo variado)
- ▶ Touples → (tuplas)
- ▶ Collections → (Vector, Set, Pair e etc)
- ▶ Generics → (class myClass<T>)
- ▶ Type



ALGUMAS DAS INCOMPATIBILIDADES COM PHP

- ▶ goto e if: ... endif
- ▶ AND, OR e XOR (use &&, || e ^)
- ▶ Passagem por referência: function foo(&\$x)
- ▶ @ como silenciador de erros
- ▶ Globals: global \$x;
- ▶ Misturar HTML e Código Hack

HACK: NOVO PARADIGMA



- ▶ Hack foca na consistência, performance, escalabilidade, checagem de tipos e detecção de erros.
- ▶ PHP foca na flexibilidade, liberdade e velocidade de desenvolvimento.



- ▶ Esse novo paradigma pode sofrer resistência da comunidade PHP
- ▶ Hack facilita o desenvolvimento de projetos mais complexos e com arquiteturas mais rígidas.

JÁ POSSO USAR COM O WORDPRESS?



- ▶ A máquina virtual HHVM sim, com algumas ressalvas.
- ▶ A linguagem Hack não, pois ainda não é compatível com WordPress
- ▶ Hack numa aplicação que interaja com WordPress pode funcionar

PRINCIPAIS INCOMPATIBILIDADES DO HACK COM O WORDPRESS?



- ▶ Uso de variáveis globais
- ▶ Mistura de código PHP com HTML



- ▶ É feito pelo Facebook
- ▶ Ganho de eficiência/velocidade
- ▶ É escalável
- ▶ Roda tanto PHP quanto Hack
- ▶ Tem código aberto
- ▶ Tem evoluído rapidamente



CONTRAS DA HHVM

- ▶ É feito pelo Facebook
- ▶ Há extensões não suportadas
- ▶ Nível de maturidade
- ▶ Só há suporte ao SO Linux (por enquanto)
- ▶ Para Hack só há suporte nas IDEs VIM e Emacs
- ▶ Não há suporte ao PHPMyAdmin (ainda – falta suporte ao MySQLi)



DEMONSTRAÇÃO



OBRIGADO!

Saiba mais:

- ▶ hacklang.org
- ▶ hhvm.com
- ▶ wordpress.com

Visite:



codigocriativo.com.br



fazendoasmalas.com

Contato:

- ▶ contato@codigocriativo.com.br



XHP - EXEMPLO

```
<?php
```

```
$title = 'My Project';
```

```
$head =
```

```
    <head>
```

```
    <meta charset="utf-8" />
```

```
    <title>{$title}</title>
```

```
    </head>;
```

```
$content = <x:frag> Welcome to my project! </x:frag>;
```

```
$body = <body> {$content} </body>;
```

```
echo
```

```
<x:doctype>
```

```
    <html> {$head} {$body}
```

```
    </html>
```

```
</x:doctype>;
```

```
>
```




REFERÊNCIAS

[1] Keith Adams et al. "The HipHop Virtual Machine". *Proceedings of the 2014 ACM International Conference on Object Oriented Programming Systems Languages & Applications*. ACM, 2014.

[2] Benda, Jan, Tomas Matousek, and Ladislav Prosek. "Phalanger: Compiling and running PHP applications on the Microsoft .NET platform." *.NET Technologies 2006* (2006).

[3] Zhao, Haiping, et al. "The HipHop compiler for PHP." *ACM SIGPLAN Notices*. Vol. 47. No. 10. ACM, 2012.

[4] Static Detection of Cross-Site Scripting Vulnerabilities *Software Engineering, 2008. ICSE'08. ACM/IEEE 30th International Conference on*. IEEE, 2008.

[5] TIOBE programming community index. Web site: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

[6] PHP5. Web site: <http://php.net>.

[7] Facebook. Web site: <http://facebook.com>.



REFERÊNCIAS

[8] Wordpress. Web site: <http://wordpress.org>

[9] W3Schools. Web site: <http://www.w3schools.com/php>

[10] PHC. Web site: <http://www.phpcompiler.org>

[11] PHP Compiler. Web site: <http://www.php-compiler.net>

[12] PHP bytecode Compiler. Web site:
<http://php.net/manual/en/book.bcompiler.php>

[13] Hack Language. Web site: <http://hacklang.org>

[14] Zhao, Haiping, et al. "The HipHop compiler for PHP." *ACM SIGPLAN Notices*. Vol. 47. No. 10. ACM, 2012.

[15] XHP. Web site: <https://www.facebook.com/notes/294003943919>

[16] Hack release. Web site:
<https://code.facebook.com/posts/264544830379293>



REFERÊNCIAS

[17] HHVM. Web site: <https://code.facebook.com/posts/10151170460698920>

[18] HHVM. Web site: <http://hhvm.com>

[19] Parab, Jivan S., et al. "Integrated Development Environment." *Exploring C for Microcontrollers: A Hands on Approach* (2007): 19-28.

[20] Paulson, Linda Dailey. "Developers shift to dynamic programming languages." *Computer* 40.2 (2007): 12-15.

[21] Best PHP IDE in 2014. SitePoint web site: <http://www.sitepoint.com/best-php-ide-2014-survey-results>

[22] Unsupported PHP Features in Hack. Web site: <http://docs.hhvm.com/manual/en/hack.unsupported.php>

[23] The HipHop Compiler. Web site: <https://www.facebook.com/notes/280583813919>.

<http://hhvm.com/blog/4061/go-faster>

<http://wpengine.com/2014/11/19/hhvm-project-mercury/>