

PHP: Vom Entwicklerbaukasten zur Enterprise-Plattform

International PHP Conference

Kore Nordmann & Tobias Schlitt
26th October 2016



Hi



Tobias Schlitt
@tobySen



Kore Nordmann
@koredn



@qafoo



@tideways

PHP Has Come To Be Perfect For The Web





THE
LANGUAGE WARS

THE
LANGUAGE WARS

A NEW HOPE

A Brief History

- ▶ 1995: Set of Perl scripts by Rasmus
- ▶ 1998: PHP 3: Rewrite by **Zeev & Andi** (Zend)
 - ▶ PHP: Hypertext Preprocessor
 - ▶ C-like standard library
- ▶ 2000: PHP 4: Dedicated virtual machine (Zend Engine)
- ▶ 2004: PHP 5: Actual object model
- ▶ 2009: PHP 5.3: Sane garbage collection
- ▶ 2015: PHP 7: Massive performance improvements



We Grew Up



Evolution Of PHP

- ▶ Mixing paradigmes since 1995
 - ▶ Procedural from the beginning
 - ▶ Structs with functions since PHP 4
 - ▶ Object Orientation since PHP 5
 - ▶ We even got goto (since PHP 5.3)
- ▶ Inconsistencies in standard library
 - ▶ `str_split()` vs. `strlen()` vs. `htmlspecialchars_decode()` vs. `IntlBreakIterator` implements `Traversable`

How can PHP power 80%¹ of the web?

¹https://w3techs.com/technologies/overview/programming_language/all

THE
LANGUAGE WARS

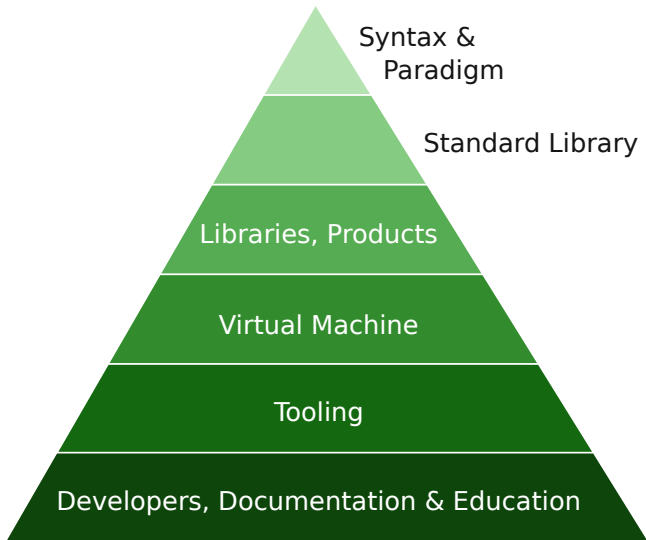
A NEW HOPE
ATTACK OF THE CLONES

There was nothing else in 2000 (except Perl)

- ▶ But then there was:
 - ▶ Ruby On Rails
 - ▶ Django & Zope (Python)
 - ▶ ASP.net
 - ▶ Java Server Faces
 - ▶ ...

How can PHP still power 80% of the web?

It Is Not About The Language



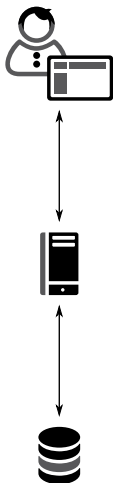
It Is Not About The Language

How does PHP power 80% of the web?

THE
LANGUAGE WARS

A NEW HOPE
ATTACK OF THE CLONES
THE FORCE AWAKENS

Where Is PHP Used?




Architecture: It's All Tradeoffs





Performance



Security



Availability

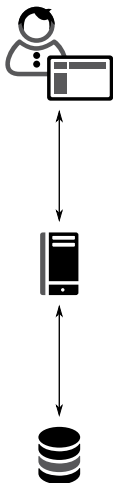


Modifiability

Performance



Where Is PHP Used?



Number Crunching ²

Language	CPU	Slower	Version
C++ (-O2)	0.973s	–	g++ 6.1.1
Java 8 (non-std lib)	1.126s	15%	1.8.0_102
Python 2.7 + PyPy	1.514s	55%	PyPy 5.4.0
Go	2.757s	183%	1.7
C++ (not optimized)	2.954s	203%	g++ 6.1.1
PHP 7.0	6.739s	592%	7.0.10
Javascript (nodejs)	7.202s	639%	4.3.1
Java 8 (see notes)	12.200s	1,153%	1.8.0_102
Ruby	13.147s	1,250%	2.3.1
Python 3.5	17.895s	1,738%	3.5.2
Python 2.7	23.691s	2,334%	2.7.12
Perl	25.562s	2,526%	5.22.2
PHP 5.6	68.784s	6,020%	5.6.17

²<https://blog.famzah.net/2016/09/10/>

cpp-vs-python-vs-php-vs-java-vs-others-performance-benchmark-2016-q3/

Single Node Performance

- ▶ Performance does not match compiled code, but any other VM
- ▶ Basically no support for threads
 - ▶ Experimental async I/O support: ReactPHP³ (still single-threaded, just like node.js)
 - ▶ “Experimental” threading support⁴
- ▶ Only basic support for forks⁵

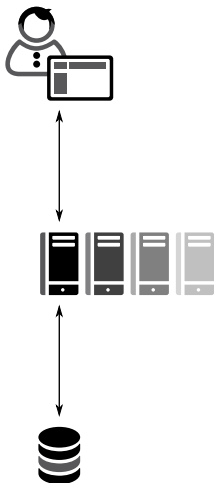
Do not use.

³<http://reactphp.org/>

⁴<https://pecl.php.net/package/pthreads>

⁵<http://docs.php.net/pcntl>

Horizontal Scalability



HTTP / REST Are Built For Scalability ⁶

LCoDC\$SS

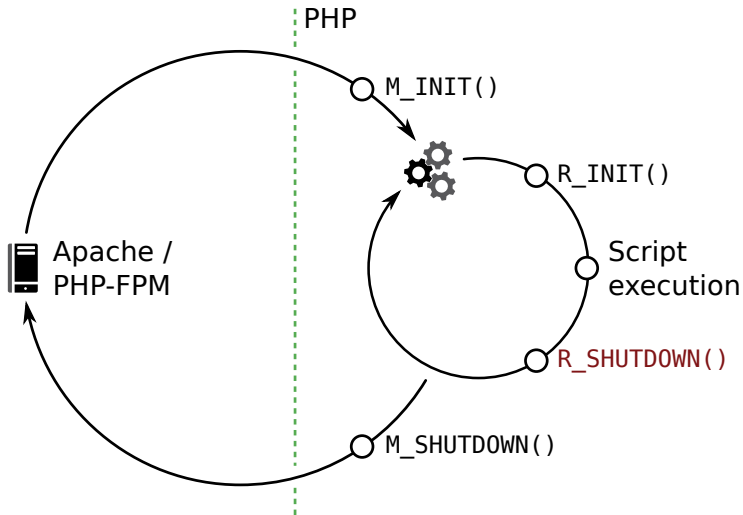
⁶<https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>

Layered Code on Demand Client Cached **Stateless** Server

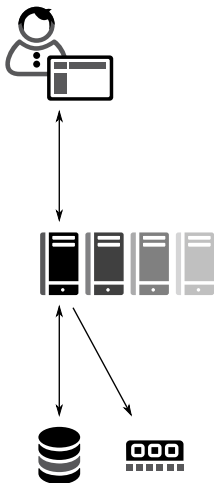
PHPs Shared Nothing Architecture

⁶<https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>

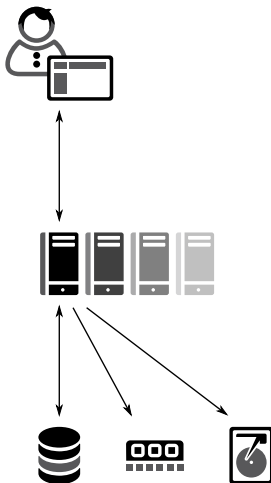
PHP Is Built For Shared Nothing



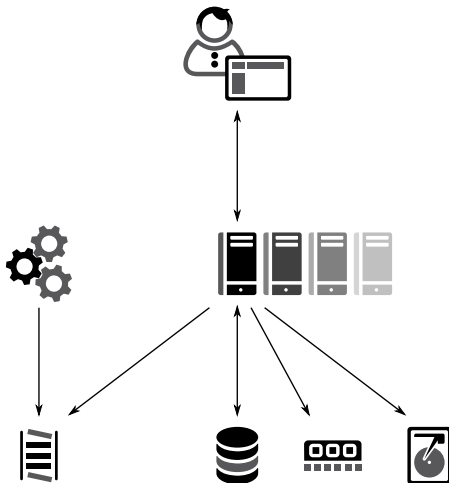
Pattern: Sessions



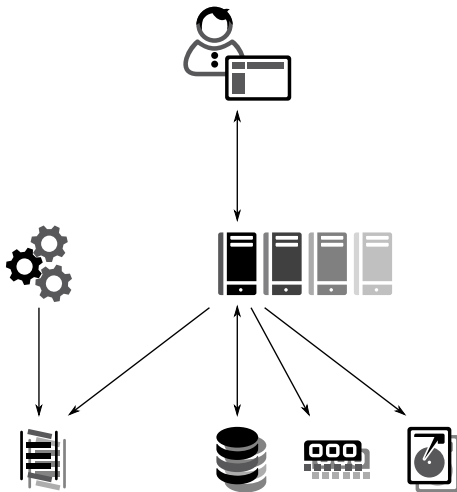
Pattern: Binary Data



Pattern: Offline Jobs



Scaling External Services




Summary

- ▶ PHP applications are usually scaled to multiple application servers from the very beginning
- ▶ Developers know the challenges and frameworks embrace them
- ▶ Servers are usually commodity hardware
- ▶ Do **not** use for:
 - ▶ Application Servers with shared object graph (Node.js?, Java, ...)
 - ▶ WebSockets (Node.js?, Go, Erlang/OTP, Java, ...)
 - ▶ Number Crunching (C, C++, Go, Java?, ...)



Performance



Security



Availability



Modifiability

A close-up photograph of a person's hand holding a heavily rusted, spherical metal object against a door handle. The hand is positioned on the left side of the frame, with the fingers wrapped around the ball. The door handle is a rectangular metal plate with a circular hole in the center, also showing signs of rust. The background is a dark, textured surface, possibly a door or wall, with some visible wear and tear. The lighting is dramatic, highlighting the textures of the skin, the rust, and the metal.


Security

- ▶ “No” issues here
- ▶ Issues were caused by default configurations leading to stupid code
- ▶ Issues are caused by unaware developers
 - ▶ Maybe related to weak dynamically typed language
- ▶ It is not trivial to write insecure code with modern frameworks. . .

Low entry barrier is a double–edged sword. . .



Performance



Security



Availability



Modifiability



Availability


Availability

- ▶ Horizontal scaling with shared nothing
 - ▶ Everything else must be highly available, too
- ▶ Resilience
 - ▶ “NullPointerException” are finally catchable since PHP 7 (almost all errors are)
 - ▶ Management processes basically do not die
- ▶ Monitoring
 - ▶ Error monitoring with libraries like Monolog⁷
 - ▶ Application performance and error metrics with Tideways, NewRelic, AppDynamics, ...

⁷<https://github.com/Seldaek/monolog>



Performance



Security



Availability



Modifiability



Modifiability

Modifiability

- ▶ Hackability

- ▶ (I think) The reason Wordpress is as big as it is. . .
- ▶ . . . like it or hate it – double-edged sword again.

- ▶ Trivial deployments

- ▶ Just put new source on server and change link to new source directory. . .
- ▶ Maybe tell opcode-cache, if fstat is disabled

- ▶ Tooling

- ▶ Composer⁸: Sane dependency management
- ▶ PHPStorm⁹: IDE with all the bits

⁸<https://getcomposer.org>

⁹<https://www.jetbrains.com/phpstorm/>

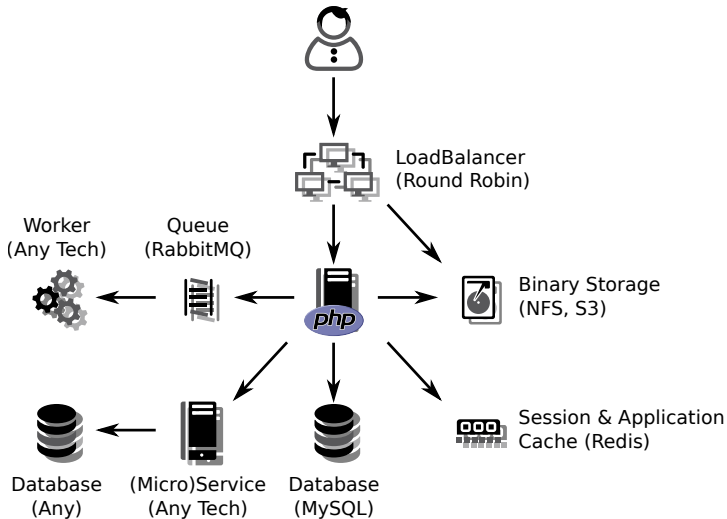
Quality Assurance

- ▶ **PHPUnit:** (Unit) Tests with PHPUnit are default for “all” libraries
- ▶ **PHPMD:** Mess detector exposes the common violations (coupling, size, complexity, ...)
- ▶ **PHP_CodeSniffer:** Verification of codings standards (PSR-2)
- ▶ **PHPCS:** Detects copy-pasted code
- ▶ **Build Systems:** Phing (ant), Phake, ...
- ▶ **CI:** TravisCI for Open Source, Jenkins, ContinuousPHP, ...
- ▶ ...



We Are Watching

Anatomy Of An Enterprise PHP Application



Relax



THE
LANGUAGE WARS

...AND THEY ALL LIVED
HAPPILY EVEN AFTER.



THANK YOU

Rent a quality expert
qafoo.com