TRALIOS
Linux Server Management

PHP sicher, performant und skalierbar betreiben

Dipl.-Inform. Dominik Vallendor ▪ 26.09.2012

Über mich

TRALIOS
Linux Server Management

- Dominik Vallendor
- Studium der Informatik in Karlsruhe
- Seit 1995: Internet / Linux
- Seit 1999: PHP-Entwicklung
- Seit 2002: Selbständig
- Seit 2010: Tralios IT GmbH: Betrieb von Linux-basierten Web/Mailservern

Übersicht

TRALIOS
Linux Server Management

- Der Webseiten-Aufruf
- Verschiedene Möglichkeiten, PHP aufzurufen
- PHP Sicherheit
- PHP-Einstellungen / Die PHP.INI
- Fragen & Diskussion

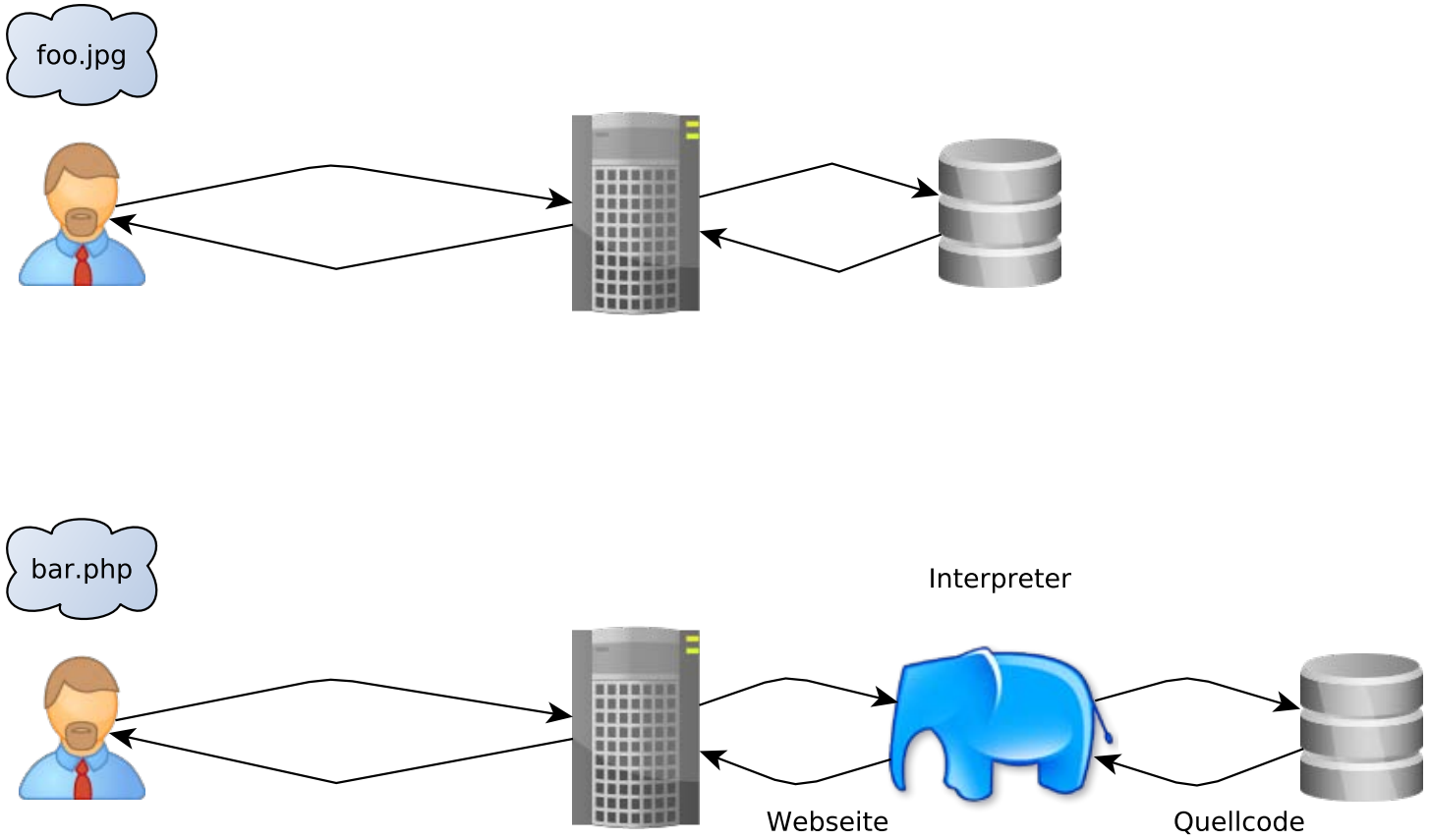
Webseiten-Aufruf

TRALIOS
Linux Server Management

- Anfrage des Users an Server
- Webserver (z.B. Apache, NGINX) nimmt Anfrage entgegen
- Statische Inhalte (Bilder, CSS) werden vom Webserver ausgeliefert
- Dynamische Inhalte: Aufruf eines sog. Handlers durch den Webserver
- Handler (PHP) liefert Content zurück, wird vom Webserver an den User ausgegeben

Webseiten-Aufruf

TRALIOS
Linux Server Management



PHP-Aufruf

TRALIOS
Linux Server Management

Der Aufruf von PHP kann auf verschiedenen Wegen erfolgen:

- Apache-Modul: mod_php
- Aufruf als CGI-Skript
- FastCGI

Außerdem: PHP-Kommandozeileninterpreter

PHP-Aufruf

TRALIOS
Linux Server Management

Aufrufe von PHP unterscheiden sich u.a.

- im Konfigurationsaufwand / Komplexität des Betriebs
- in der Geschwindigkeit
- im Kontext/Rechten, in denen der Prozess ausgeführt wird
- in der Möglichkeit, PHP-Einstellungen vorzunehmen
- in der Möglichkeit, Timeouts und Prozessanzahlen festzulegen

mod_php

TRALIOS
Linux Server Management

- Aufruf erfolgt durch (Apache-)Modul
- Standard in vielen Installationen
- Sehr einfache Konfiguration

  LoadModule php5_module modules/libphp5.so

  AddHandler application/x-httpd-php .php .php5 .phtml

  DirectoryIndex index.php index.phtml
- PHP läuft immer im Kontext und mit den Rechten des Webservers

  (typischerweise User *apache* oder *httpd*)
- Gemeinsame PHP.INI

  Einzelne Einstellungen über VirtualHost-Einträge oder .htaccess änderbar

PHP als CGI

TRALIOS
Linux Server Management

- Als Handler für .php-Dateien wird ein Programm definiert
- Aufruf wie für jede andere Skriptsprache (insb. Perl) oder Linux-Programm
- PHP läuft mit Rechten des Webservers oder durch SuEXEC mit Rechten eines anderen Benutzers
- Verschiedene PHP.INIs für jeden Prozess möglich
- Für jeden Webseitenaufruf wird ein Programm gestartet $\rightarrow$ langsam!
  (Laden des Interpreters ist langsamer als eigentliche Skriptausführung)

FastCGI

TRALIOS
Linux Server Management

- = optimierter CGI-Aufruf
- PHP-Prozess beendet sich nicht, nimmt nacheinander Anfragen entgegen
  schneller als CGI, aber nicht ganz so performant wie mod_php
- PHP läuft mit Rechten des Webservers oder durch SuEXEC mit Rechten eines
  anderen Benutzers
- Verschiedene PHP.INIs für jeden Prozess möglich
- Prozessanzahl, Timeouts, etc. lassen sich feingranular steuern

suEXEC

TRALIOS
Linux Server Management

- Apache-Modul
- Ermöglicht, CGI/FastCGI-Skripte mit den Rechten definierter User auszuführen

  $\rightarrow$ dadurch kein Zugriff auf Dateien anderer Nutzer

  $\rightarrow$ Absicherung der User untereinander und gegenüber dem System
- Überprüft außerdem Datei/Programmrechte

Vergleich

TRALIOS
Linux Server Management

Welche PHP-Aufrufvariante eignet sich für welchen Zweck?

- **mod_php:** wenn nur eine Webseite pro Server & keine besonderen Zugriffsbeschränkungen von Nöten
- **FastCGI:** Shared Hosting-Umgebungen & aus Sicherheitsgründen immer zu empfehlen
- **CGI:** kein sinnvoller Einsatzzweck, da langsamer als FastCGI und keine Vorteile

Empfehlung: nach Möglichkeit immer FastCGI einsetzen

Verhalten bei zu vielen Anfragen

TRALIOS
Linux Server Management

MOD_PHP:

- Apache reicht alle Anfragen an PHP durch
- Server überlastet → sehr lange Antwortzeiten (100s und mehr)
- Theoretisch werden alle Seiten ausgeliefert, aber User warten nicht so lange
- Server ist scheinbar tot

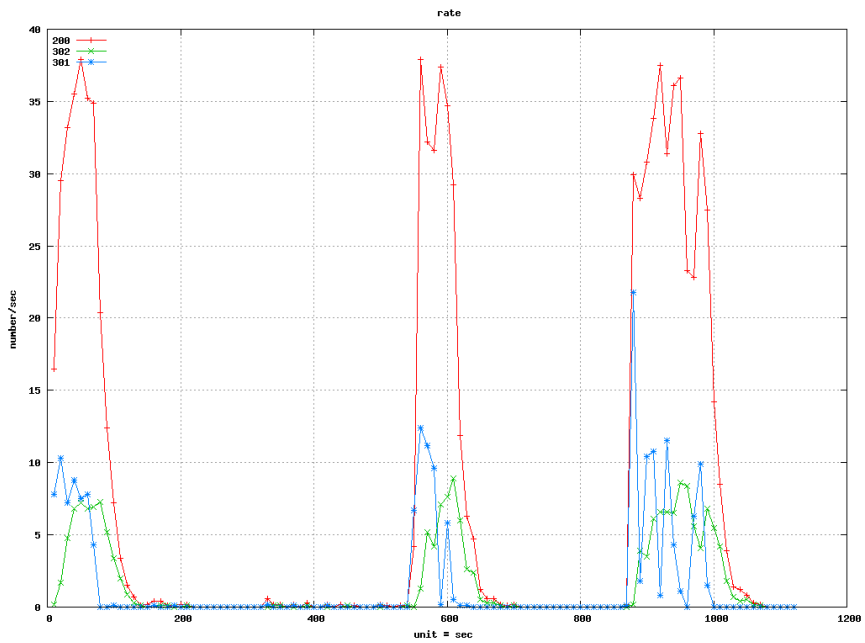
FastCGI:

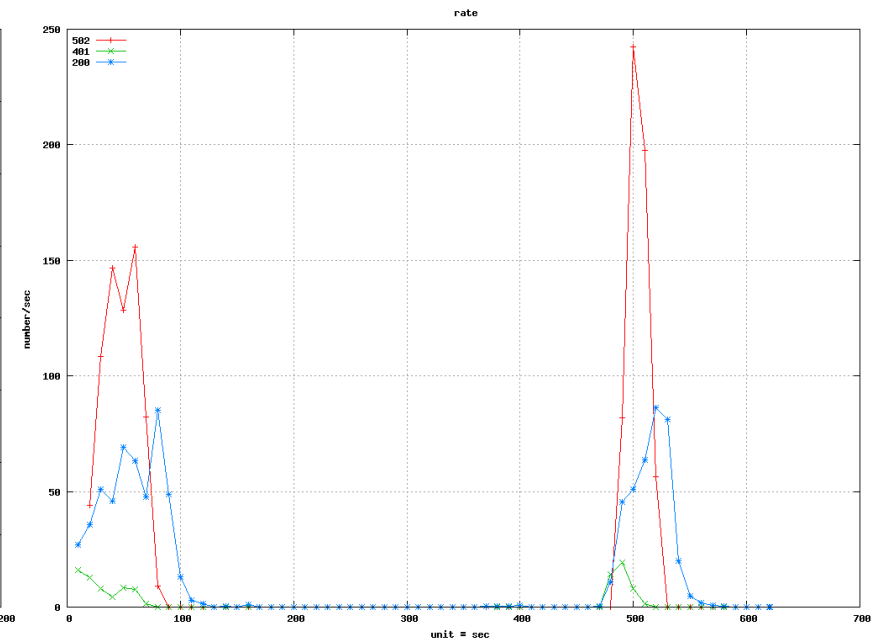
- Apache reicht nur beschränkte Anzahl Anfragen an PHP durch
- User erhält sehr schnell eine Antwort: entweder Webseite oder Fehlermeldung
- Für User besser nachvollziehbar, Fehlermeldung kann “hübsch“ gestaltet werden

Verhalten bei zu vielen Anfragen

TRALIOS
Linux Server Management

MOD_PHP vs. FastCGI:





Beispiel FastCGI-Konfiguration

TRALIOS
Linux Server Management

Apache-Konfiguration (Auszug)

```
LoadModule fastcgi_module modules/mod_fastcgi.so
FastCgiWrapper /usr/sbin/suexec2
FastCgiConfig -pass-header AUTHORIZATION -idle-timeout 60 -appConnTimeout 60
    -maxClassProcesses 5 -maxProcesses 128 -killInterval 30 -autoUpdate
Action php5-fastcgi /.fastcgi/php5-fcgi-starter
```

Beispiel FastCGI-Konfiguration

TRALIOS
Linux Server Management

Apache-Konfiguration II

```
<Location /.fastcgi/php5-fcgi-starter >
    SetHandler fastcgi-script
    Options +ExecCGI
</Location >
```

Beispiel FastCGI-Konfiguration

TRALIOS
Linux Server Management

VirtualHost-Eintrag

```
<VirtualHost *:80 >
ServerName www.example.com
DocumentRoot /var/www/kunde/www.example.com
SuexecUserGroup kunde kunde
ScriptAlias /.fastcgi/ /var/fastcgi/customers/42/
<FilesMatch \.php$ >
    SetHandler php5-fastcgi
</FilesMatch >
</VirtualHost >
```

Beispiel FastCGI-Konfiguration

TRALIOS
Linux Server Management

Startskript

```
#!/bin/sh
umask 007
PHPRC="/var/fastcgi/customers/42/"
export PHPRC
PHP_FCGI_MAX_REQUESTS=1000
export PHP_FCGI_MAX_REQUESTS
exec /usr/bin/php-cgi -c /var/fastcgi/customers/42/php.ini
```

PHP Sicherheit

TRALIOS
Linux Server Management

Wogegen absichern?

- Angriffe von außen durch “Hacker“
- Angriffe durch eigene Kunden
- Erschleichen von Ressourcen durch eigene Kunden
- Absicherung gegen Amok laufende Skripte

Korrekter Aufruf des PHP-Handlers

TRALIOS
Linux Server Management

- Falsch:

  AddType application/x-httpd-php5 .php5 .php4 .php

  Ermöglicht die Ausführung von datei123**.php**.jpg

  Problematisch z.B. beim Upload von Dateien durch Webseitenbesucher

- Korrekt:

  <FilesMatch "\.(php5|php4|php)$" >

  SetHandler application/x-httpd-php5

  </FilesMatch>

Die PHP.INI

TRALIOS
Linux Server Management

- Alle PHP-Einstellungen in einer einzigen Datei
- Nur manche Variablen lassen sich durch den Nutzer oder das Skript überschreiben

  .htaccess bei mod_php bzw. Aufruf von ini_set()
- Der Benutzer sollte **keine** Möglichkeit haben, eine eigene PHP.INI zu verwenden

  falls doch: Absicherung durch weitere Maßnahmen, z.B. *ulimit* notwendig
- Achtung: viele Einstellungen lassen sich umgehen; weitere Absicherung notwendig

Wichtige PHP-Einstellungen

TRALIOS
Linux Server Management

- **open_basedir**: Pfad, auf den ein Skript zugreifen darf
- **memory_limit**: Maximaler RAM-Vebrauch pro Skript
- **allow_url_fopen** / **allow_url_include**: Laden/Einbinden von Code von fremden Servern
- **register_globals**: Bereitstellen von Variablen in globalem Kontext
- **safe_mode**: “Sicherer Modus“ schränkt Zugriffe ein
- **magic_quotes_gpc**: Automatisches Escapen von Eingaben

Probleme

TRALIOS
Linux Server Management

- Viele Einstellungen lassen sich überschreiben $\rightarrow$ dadurch wirkungslos
- Einschränkungen lassen sich umgehen (z.B. open_basedir durch Verwendung von *exec()*)
- Teilweise Deaktivierung von PHP-Einstellungen notwendig, damit Skripte funktionieren, z.B. *allow_url_fopen*
- Sicherheitskritische Befehle in PHP vorhanden, insb. *eval()*

Konfigurierbare PHP-Einstellungen

TRALIOS
Linux Server Management

Im Kundencenter konfigurierbare PHP-Einstellungen, Kompromiss zwischen

- Benutzer kann eigene PHP.INI verwenden (sicherheitskritisch)
- Keine Möglichkeit, eigene Einstellungen selbst vorzunehmen

Suhosin

TRALIOS
Linux Server Management

- Schutzsystem für PHP-Installationen
- Überprüft übergebene Parameter, z.B. auf Länge oder Namen
- Ermöglicht die Deaktivierung von Funktionen, z.B. *eval()*
- Interne Checks, z.B. maximale Verschachtelungstiefe von Funktionen
- Session-Absicherung
- Erweiterte Logging Features
- ...

Fazit

TRALIOS
Linux Server Management

- Verschiedene Möglichkeiten, PHP zu nutzen

  FastCGI empfohlen, da sicher und flexibel
- Bestimmte Werte in PHP.INI sind sicherheitskritisch

  $\rightarrow$dürfen nicht durch Benutzer editiert werden
- PHP weiter absichern, z.B. durch Suhosin

Fragen & Diskussion

TRALIOS
Linux Server Management

???

Kontakt

TRALIOS
Linux Server Management

- Dipl.-Inform. Dominik Vallendor

- Tralios IT GmbH

  Pfinztalstr. 90

  76227 Karlsruhe

  Telefon: 0721 - 94269660

  Telefax: 0721 - 94269666

  E-Mail/Jabber: vallendor@tralios.de