

Performances PHP

Audit et optimisation LAMP

Julien Pauli

Cyril Pierre de Geyer

Guillaume Plessis

Préface d'Armel Fauveau

© Groupe Eyrolles, 2012, ISBN : 978-2-212-12800-0

EYROLLES



Préface

Amener un projet web à maturité est un processus complexe, dont la gestion des performances est un aspect décisif. En effet, ces dernières importent autant, si ce n'est plus, que les choix techniques et méthodologiques qui ont été faits. La raison en est simple mais elle échappe pourtant souvent aux développeurs et aux chefs de projets : les performances sont directement perceptibles par l'utilisateur, sous la forme d'une sensation de vitesse, mais aussi de disponibilité. D'où l'effet désastreux d'un serveur LAMP qui n'arriverait plus à traiter les demandes entrantes !

C'est un fait, l'utilisateur se soucie peu des choix architecturaux et n'en a généralement pas même connaissance ; en revanche, il n'est jamais indifférent aux performances, bonnes ou mauvaises...

Voilà pourquoi le succès de l'expérience de navigation de l'utilisateur repose sur vous, qui lisez ce livre. La rapidité et la bonne tenue de votre application (à savoir des temps de réponse courts et une continuité de service sans faille) sont des conditions nécessaires – voire suffisantes – à l'adoption de votre projet. Il est donc important d'apprendre à soulager l'empreinte mémoire, la charge CPU, les accès disques, le réseau et la bande passante, qui sont autant de maillons d'une chaîne délicate et fragile : l'architecture du système. Et c'est précisément le rôle d'un audit et le but de l'optimisation que d'en prendre soin.

Mais comment faire ? Il n'existe pas de solution prête à l'emploi et il ne s'agit pas d'une science exacte. La maîtrise des performances est un art qui s'affine avec le temps et l'expérience ; elle nécessite une remise en question permanente, *a fortiori* dans un contexte web où les usages et les technologies évoluent sans cesse. La maîtrise des performances passe donc par l'apprentissage et la bonne compréhension des différentes briques logicielles qui composent votre socle technique. Elle implique également la prise en main d'outils et la mise en œuvre de bonnes pratiques. Il vous faudra notamment connaître et comprendre le fonctionnement, le rôle et les points

faibles des différents rouages du système, pour être en mesure d'optimiser les performances de ce dernier sans en compromettre ni la stabilité ni la robustesse.

C'est précisément ce que cet ouvrage, rédigé par des auteurs passionnés, vous propose de découvrir. Après une présentation de l'architecture LAMP, vous apprendrez à écouter votre plate-forme serveur par le biais de l'analyse des fichiers journaux et de divers outils de monitoring. Vous découvrirez comment mener à bien des tirs de charge de grande envergure avec des données fictives pour simuler des situations de production réelles. Vous apprendrez à compiler, paramétrer et optimiser finement Apache, MySQL et PHP. Vous découvrirez comment soulager PHP de coûteuses étapes de *parsing* et saurez choisir judicieusement votre moteur de stockage MySQL. Vous apprendrez enfin à identifier avec précision les points de congestion dans votre code et vos requêtes. Les différentes techniques de réplication MySQL n'auront plus de secret pour vous.

Au-delà de votre plate-forme LAMP et des soucis de diminution de charge côté *backend*, vous vous apercevrez que les problématiques de performances et d'optimisation se répercutent également sur le réseau et au cœur du navigateur. Près de 90 % du temps passé à traiter et afficher une page web est consommé par les échanges réseau et les traitements côté client. Nouvelle frontière en termes de recherche et d'investigation, faisant l'objet de nombreuses études actuelles, l'optimisation côté *frontend* est donc primordiale. Vous apprendrez à limiter le nombre de requêtes HTTP, à réduire la taille des ressources et à mieux gérer le cache du navigateur.

Comme vous le constaterez, il est possible d'agir à chaque étape de votre projet, du code jusqu'aux différents protocoles réseau. Il est même recommandé de n'en négliger aucune.

Je me permettrais un dernier conseil : l'optimisation ne devrait intervenir qu'en toute fin de projet, lorsque celui-ci répond aux exigences fonctionnelles dictées par votre cahier des charges. Le faire trop tôt est en effet souvent source d'erreur... alors qu'il n'est jamais trop tard pour s'en préoccuper.

Bonne lecture !

Armel Fauveau

Co-fondateur et directeur technique de GLOBALIS

Co-fondateur de l'AFUP (Association française des utilisateurs de PHP)

Table des matières

Avant-propos 1

Pourquoi ce livre ?	1
À qui s'adresse ce livre ?	1
Structure de l'ouvrage	2
Remerciements	3

CHAPITRE 1

Présentation de l'architecture LAMP 5

Les différents composants LAMP	6
Le noyau Linux	6
Le serveur web Apache	6
Le langage PHP	7
Le serveur de base de données MySQL	7
Communication entre les composants	8
L'application témoin : une application Drupal	8
L'environnement	9
Démarche de test et outil utilisé	10

CHAPITRE 2

Analyse et administration des logs 13

À quoi sert la journalisation, comment marche-t-elle ?	13
Configurations recommandées	14
Pendant le développement ou l'audit	14
En production	14
Synthèse et recommandations de gestion de la journalisation	14
Archivage par rotation des fichiers de logs	15
Les journaux d'Apache	16
Localisation et configuration	16
Les journaux d'accès d'Apache	18
<i>Quelques manipulations sur les fichiers de logs d'accès</i>	20
<i>Utiliser les erreurs d'Apache</i>	21

Les journaux de MySQL	22
Configurer la journalisation des requêtes lentes et/ou sans index	23
Le fichier de log binaire pour la réplication	25
Activer le fichier de journalisation général de MySQL	25
Les journaux de PHP	26
Le journal d'erreurs PHP	26
Les différents niveaux d'erreurs	27
<i>Configuration via le fichier php.ini</i>	27
<i>Configuration via Apache</i>	29

CHAPITRE 3

Superviser l'activité 31

Enregistrement et suivi graphique avec Munin	32
Installation et prise en main de Munin	33
<i>Installation du serveur</i>	34
<i>Installation d'un nœud</i>	35
<i>Les plug-ins de Munin</i>	36
<i>Mise en avant des métriques de suivi utilisées</i>	36
Supervision avec Nagios	40
<i>Installation de Nagios</i>	41
<i>Configuration de Nagios</i>	43
<i>NRPE, la base pour une supervision distante</i>	43
<i>Monitoring système</i>	44
<i>Monitoring des processus Apache et MySQL</i>	47

CHAPITRE 4

Figurer, restaurer et charger une base de données MySQL 51

Figurer un état de votre base de données.	51
Quel type de sauvegarde pour votre base de données ?	52
Sauvegarder la base dans un fichier texte : la sauvegarde logique	53
<i>Créer des sauvegardes logiques avec mysqldump</i>	53
<i>Créer des sauvegardes logiques avec phpMyAdmin</i>	55
<i>Créer des sauvegardes logiques avec MySQL Workbench</i>	56
Restaurer votre base de données.	58
Restaurer une sauvegarde logique avec phpMyAdmin	58
Charger une base de données	59
Avec des scripts	60
Directement avec PHP	61
<i>En reproduisant des scénarios</i>	61
<i>Avec Selenium IDE</i>	61

CHAPITRE 5

Scénarios de test et tirs de charge 63

Scénarios de test et outils	63
Méthodes de travail	63
Élaboration d'une stratégie de tests	64
Exploitation d'Apache et de ses logs pour mettre en place des jeux de tests réalistes	65
Connaître la répartition des requêtes	66
Tir de charge simple avec Apache Bench	69
Installation et utilisation de http_load	70
Installation et utilisation de Siege	71
Création de l'unité de mesure de nos tirs de charge	72
Enregistrement et réalisation d'un scénario avec JMeter.....	74
Installation et exécution de JMeter	76
<i>Exécution de JMeter</i>	77
<i>Exécution de JMeter en ligne de commande</i>	77
<i>Exécution de JMeter derrière un proxy</i>	78
Briques de base de JMeter	78
<i>Groupe d'unités</i>	79
<i>Échantillons</i>	80
<i>Contrôleurs logiques</i>	81
<i>Récepteurs</i>	81
<i>Construire son premier plan de test</i>	82
Simulation de l'envoi de formulaires	85
<i>Contourner les jetons (tokens) bloquant les envois multiples de formulaires</i> ...	85
Éléments de configuration	87
<i>Factorisation de vos éléments</i>	87
<i>Gestion des sessions/cookies</i>	87
Mise en place du proxy JMeter	88
<i>Configuration du navigateur</i>	88
<i>Démarrage du serveur proxy JMeter</i>	88
<i>Exclusion d'éléments</i>	90

CHAPITRE 6

Optimiser la configuration de PHP 91

Localisation du fichier de configuration	91
Directives de configuration de PHP	93
Options générales	94
Gestion du tampon de sortie de PHP	94
<i>Compression des pages avant envoi</i>	95

Données en transit sur HTTP	96
Options liées aux limitations des ressources	97
Directives liées aux données et aux répertoires	98
<i>Gestion des inclusions</i>	99
Sécurité	101
Gestion des sessions PHP	102
Ramasse-miettes	104
Gestion des extensions	104

CHAPITRE 7

Administration de PHP 105

Réduction de la quantité de code à compiler	105
Rappel sur le fonctionnement global de PHP	105
Compilation utile	106
<i>Identification des fichiers inclus</i>	106
<i>Auto-chargement de classes</i>	107
Cache d'op-codes	109
Fonctionnement d'un cache d'op-codes	110
Installation d'un cache d'op-codes	113
Configuration du cache d'op-codes APC	114
Monitoring du cache d'op-codes	115
Tir de charge sur notre application de référence	116
<i>Premier benchmark de référence</i>	116
<i>Second benchmark de référence : avec APC</i>	117
Fragmentation de la mémoire	118
Technologies de bas niveau utilisées par le cache	119
Cache utilisateur	119
Compilation et personnalisation de PHP	120
Qu'est-ce que la compilation de sources ?	120
Tir de charge sur un PHP compilé avec des optimisations	121
<i>Tir de charge du moteur</i>	121
<i>Tir de charge de notre application</i>	122
Compilation manuelle d'Apache	124

CHAPITRE 8

Améliorer son code PHP 127

Qu'est-ce que le profiling applicatif ?	127
Principe à retenir	127
Mise en pratique du profiling	128
Installation de Xdebug	128
Installation d'un outil de visualisation graphique	129

Exercice de profiling	129
Profiling de notre application Drupal	132
Mise en pratique du profiling Xhprof	134
Installation de Xhprof	134
Exercice de profiling avec Xhprof	135
Un peu plus loin avec Xhprof	137
Conclusion sur le profiling	138

CHAPITRE 9

Administration et optimisation d'Apache 139

Apache : ce qu'il faut absolument savoir	139
Comment Apache est devenu un incontournable du Web	139
Architecture et fonctionnement modulaire	142
Modules multi-processus (MPM)	144
<i>MPM prefork</i>	144
<i>MPM worker</i>	146
<i>Les autres MPM</i>	148
Intégration de PHP via mod_cgi	149
Installation de PHP en CGI	149
Tirs de charge sur un script PHP en mode CGI	151
Fonctionnement de CGI	153
Intégration de PHP via mod_php	154
Installation de mod_php	154
Tirs de charge sur un script PHP interprété par mod_php	156
Fonctionnement interne de mod_php	158
Configuration d'Apache	160
Syntaxe du fichier de configuration d'Apache	160
Portée des directives	162
Fichiers .htaccess	164
Localisation de la configuration	165
Optimisation d'Apache	166
Optimisation de la gestion des processus fils	166
<i>Tirs de charge</i>	168
Optimisations générales	170
<i>Accès disque et temps processeur</i>	170
<i>Mémoire vive</i>	173
Alternatives à Apache	175
Problème C10k	175
Présentation de Nginx	175
Installation et configuration de Nginx	176
Interactions avec PHP	177

Tirs de charge	181
----------------------	-----

CHAPITRE 10

Optimisations HTTP 183

Protocole HTTP : l'essentiel	183
HTTP/1.0	184
<i>Format d'une requête</i>	184
<i>Format d'une réponse</i>	185
HTTP/1.1	186
<i>Connexions persistantes</i>	186
<i>Pipelining</i>	188
Outils d'analyse et de manipulation des requêtes HTTP	190
<i>Curl</i>	190
<i>LiveHttpHeaders</i>	190
<i>FireBug</i>	191
<i>Yslow</i>	192
<i>TamperData</i>	192
Optimisation via le cache	194
Optimisation via le cache navigateur	194
<i>Exploitation de l'en-tête Expires</i>	195
<i>Exploitation de l'en-tête Cache-Control</i>	198
<i>Exploitation de l'en-tête Etag</i>	200
Optimisation par un reverse-proxy cache HTTP	202
<i>Exploitation du mod_cache d'Apache</i>	202
<i>Exploitation de Varnish : le cache HTTP</i>	205
Compression des documents	209
Pourquoi compresser ?	209
Compatibilité des navigateurs	210
Mise en place de la compression HTTP au sein d'Apache	211
Mise en place de la compression HTTP au sein de PHP	213
En résumé.	214
Vérifications essentielles	214
Vérifications complémentaires	215
Liste récapitulative pour optimiser le cache HTTP	216

CHAPITRE 11

Administration et optimisation de MySQL 219

MySQL, la base de données idéale pour le Web	220
Une adoption rapide	220
Un développement chahuté, mais dynamique	220
Fonctionnalités	221

<i>Support du standard ANSI SQL-99</i>	221
<i>Gestion des partitions</i>	222
<i>Schémas d'information et de performance</i>	222
<i>Gestion du cache de requêtes</i>	223
<i>Gestion de la réplication</i>	223
<i>Une architecture modulaire</i>	223
Connexion de PHP à MySQL	224
Adapter MySQL aux ressources disponibles	227
Installer MySQL sur un système 64 bits	228
Migration vers une version 5.1+ de MySQL	228
Adapter sa configuration	229
<i>Paramétrage général</i>	229
<i>Paramétrage InnoDB</i>	230
<i>Le cache de requêtes</i>	231
<i>Connexions échouées : augmenter max_connect_errors</i>	232
<i>Accélération des connexions</i>	232
<i>Bonne valeur de max_connections</i>	232
<i>Une aide précieuse : MySQLTuner</i>	233
Optimisation du schéma et des requêtes	236
Normalisation du schéma	236
Choix des bons types de données	236
Utiliser correctement les index : pas à pas	237
<i>Identifier les requêtes peu performantes</i>	237
<i>Analyser le plan d'exécution</i>	239
<i>Poser les bons index</i>	240
<i>Cas limite</i>	241
Profilage de requêtes	242
Choix du bon moteur de stockage	244
<i>Quels sont les moteurs disponibles ?</i>	244
<i>MyISAM, le plus véloce</i>	246
<i>InnoDB, le plus robuste</i>	246
<i>Blackhole, le trou noir pour vos données</i>	247
<i>Heap, rapide pour des données temporaires</i>	247
<i>Archive, idéal pour la journalisation</i>	247
<i>Un moteur pour chaque problématique</i>	248
Monter en charge grâce à la réplication MySQL	248
Maître-esclave(s) : gain en lectures	249
Maître-maître actif/actif : gain en écritures	250
Maître-maître actif/passif : haute disponibilité	251
DRBD pour les petites structures	252
Tungsten Replicator pour une réplication synchrone performante	253

Réplication MySQL : les bonnes pratiques	253
<i>Éviter certaines requêtes non sûres</i>	254
<i>Ne plus utiliser MyISAM</i>	254
<i>Empêcher l'esclave de « crasher »</i>	255
<i>Avoir un esclave au moins aussi puissant que le maître</i>	255
<i>Filtrage de la réplication : à utiliser avec précaution</i>	256
<i>Adapter le format des logs binaires</i>	256
En résumé.	257
Que veut-on optimiser dans une base de données ?	257
Comment s'effectue la mise à l'échelle ?	257
Quel type de mise à l'échelle ?	258
<i>La plus fréquente : la mise à l'échelle verticale</i>	258
<i>Plus complexe mais plus pérenne : la mise à l'échelle horizontale</i>	259
Index	263

Avant-propos

Pourquoi ce livre ?

Pourquoi écrire un livre si ce n'est une affaire de passion ? En effet, pour nous, PHP est une affaire de cœur. Nous allons vous transmettre non seulement un savoir, mais aussi une expérience et une passion.

Que ce soit dans le cadre de l'écriture (co-écriture des livres *PHP 5 avancé*, *ZF en action*, articles dans la presse) ou d'une forte implication associative (paquets Dotdeb, création de l'AFUP (Association française des utilisateurs de PHP, PHPfrance, etc.), nous travaillons depuis plus de dix ans à la promotion et au développement de PHP.

Au travers de ces expériences et de nos activités professionnelles respectives, nous avons été confrontés à de nombreuses problématiques, dont une récurrente pour toute application Web victime de son succès : l'optimisation des performances.

Bien qu'il n'y ait pas de solution idéale générique, il existe une méthodologie et des pistes pour multiplier les performances de la pile LAMP (Linux, Apache, MySQL et PHP).

Nous avons souhaité vous transmettre ces connaissances au travers d'une méthode d'apprentissage ludique : au fur et à mesure de votre lecture, vous pourrez mettre en place les optimisations signalées précédemment et, petit à petit, améliorer votre application.

À qui s'adresse ce livre ?

Cet ouvrage est technique, il s'adresse à des informaticiens ayant une bonne expérience de la programmation PHP et de l'administration de serveurs.

Pour l'administrateur, nous avons mis en place une approche pédagogique lui permettant de comprendre de façon fine comment fonctionne la pile LAMP, son instal-

lation, son administration et sa gestion quotidienne. Au fil des chapitres, il apprendra à optimiser finement chaque partie et à en mesurer les performances.

Quant au développeur, il apprendra au travers de cet ouvrage le fonctionnement intime de son langage fétiche, PHP, afin d'être en mesure d'optimiser ses développements. Il acquerra une bonne vue d'ensemble de la partie serveur web, au travers d'Apache, sans oublier la communication HTTP, dont l'optimisation est très importante.

Enfin, tout lecteur curieux au sujet de la pile LAMP saura trouver des informations détaillées concernant son fonctionnement et les manières de l'optimiser, de mesurer les temps de réponse, de jouer des scénarios de test, etc.

Structure de l'ouvrage

Ce livre a été décomposé en 11 chapitres, chacun présentant une partie spécifique visant à optimiser les performances d'une application LAMP.

- Le **chapitre 1** est une introduction présentant les composants LAMP, la manière dont ils sont liés et les différents points qu'il conviendra de surveiller en vue d'une possible optimisation.
- Le **chapitre 2** explique comment utiliser les logs, ces informations de journalisation que vos composants LAMP enregistrent. Vous verrez comment les trouver, les exploiter, et dans quel but.
- Le **chapitre 3** est dédié au monitoring, c'est-à-dire à la surveillance active en temps réel de vos composants LAMP. En effet, ceux-ci sont capables de vous avertir lorsqu'un problème de performance a lieu ; à vous de savoir les écouter.
- Dans le **chapitre 4**, nous étudions les informations présentes dans une base de données. Nous montrons comment les sauvegarder et les restaurer dans le but de monter une architecture parallèle sur laquelle effectuer des tests.
- Les tests de charge sont abordés au **chapitre 5** qui traite des outils nécessaires à ce travail et de leur mise en place.
- PHP fera l'objet des **chapitres 6, 7 et 8** qui le décortiqueront de l'intérieur pour vous montrer comment il fonctionne, mais aussi et surtout comment optimiser ce fonctionnement pour qu'il soit plus efficace.
- Le serveur web est un composant crucial que le **chapitre 9** présente. Comment fonctionne Apache, quelles sont les erreurs à ne pas faire, comment en tirer des performances maximales ? Nous aborderons également les alternatives à Apache.
- Dans le **chapitre 10**, nous nous efforçons de vous montrer que la communication entre les composants est d'une importance capitale en matière de performances. HTTP sera donc à l'honneur dans ce chapitre.

- Enfin, le **chapitre 11** traite du fonctionnement, de la configuration et de l'optimisation de MySQL, le serveur de bases de données Open Source le plus connu et le plus utilisé dans le domaine du Web.

Remerciements

Nous tenons à remercier tous ceux qui nous ont aidés à rédiger ce livre :

- aux familles, proches et amis pour leur soutien et leur patience pendant ces longs mois de rédaction et de réflexion ;
- aux éditions Eyrolles pour avoir cru en notre projet et l'avoir soutenu dès le départ ;
- à Valérie Glane, Armel Fauveau, Olivier Dasini, Nicolas Henion (nicolargo), Milamber et Frédéric Hardy pour leurs contributions multiples au développement des divers chapitres ;
- à, dans le désordre, Guillaume Rossolini, Stéphane Crivisier, Sébastien Lucas, Hugo Hamon, Charlotte Grafeille, Francis Nappez pour les diverses relectures qu'ils ont pu faire ;
- ... et à tous les autres dont nous n'avons pas le nom complet, que nous n'avons pas pu recontacter ou que nous avons simplement oubliés dans la précipitation juste avant l'impression de cette page.

Merci à tous, car sans vous, ce livre n'aurait peut-être pas vu le jour.