

Parlons peu, parlons tech :Headless Drupal avec Phalcon



Contexte

Imaginez que vous décrochiez le contrat de l'année : Bingo.

Ce contrat concerne un site internet à fort trafic, pour vous la solution est incontestablement Drupal.

Pendant l'échange avec votre client, il vous affirme qu'il souhaite refaire son site internet, que Drupal lui paraît être une très bonne solution et enfin que son site est très visité avec des pics de connexion ponctuels mais intenses.

Plusieurs questions se posent donc à vous : Est-ce que Drupal est capable de gérer à lui seul les centaines de milliers de visites mensuelles ? Au-delà de ces visites, est-ce que Drupal sera capable de gérer des pics de connexion dans une échelle de temps très fine ? Vous avez les bonnes questions mais n'avez pas encore la bonne réponse, si ce n'est que Drupal va être utile.

Revenons aux origines de Drupal

Drupal est un CMS, ce qui signifie que sa fonction première est de gérer du contenu, la réalité est que malgré toutes les fonctions qu'il possède, la gestion de contenu est ce qu'il sait faire de mieux. Nous allons donc l'utiliser pour cet aspect et substituer les fonctions qu'il fait de manière moins efficaces par des solutions plus légères et rapides.

Nous allons donc « préparer » Drupal à accueillir les données dont nous aurons besoin, nous allons également préparer les connexions permettant par la suite de récupérer ces données pour s'en servir.

Drupal devient alors une sorte de réservoir sur lequel nous créerons les tuyaux nous permettant de récupérer les données que nous souhaitons exploiter.

Rendons Drupal « headless »

Nous allons rendre Drupal « headless » en l'utilisant seulement pour son back-end et remplacer le front-end par un framework adapté à nos besoins. Loin de l'opération chirurgicale que sous-entend ce remplacement, c'est une pratique courante. Ces frameworks sont bien souvent choisis pour leur capacité à fournir une plus grande flexibilité de création. Les plus connus étant Symfony, AngularJS, ou encore NodeJS.

Nous allons alors exploiter son point fort et nous délester de ses points faibles qui limitent la capacité de Drupal à répondre à un grand volume de requêtes rapidement en utilisant une solution plus optimisée que jamais.

Dans le cas de notre site internet à fort trafic, nous allons utiliser un « framework PHP » un peu moins connu que ceux précédemment annoncés, celui-ci a un avantage indéniable : Il a été pensé pour la performance.

La solution : Phalcon !

Qu'est-ce que c'est ?

Phalcon est un framework PHP programmé en tant qu'extension C permettant d'intégrer le modèle MVC à votre projet. On peut voir cette « extension » comme étant une bibliothèque de classes pré-configurées intégrées au serveur et prêtes à l'emploi. Le fait que ces classes soient pré-compilées permettra de gagner du temps lors de l'exécution de la chaque tâche puisque l'on va se passer de l'interprétation, ce qui aura pour effet de répondre aux requêtes plus rapidement, et donc de pouvoir répondre à un nombre de requêtes plus élevé à chaque seconde.

Dans le cadre du site à très fort trafic, chaque dixième de seconde est d'une importance capitale et chaque retard de réponse à une requête se répercutera sur la requête suivante. Vous l'aurez compris, dans le cas où il y aurait plusieurs milliers de requêtes par minutes, les temps de chargement peuvent rapidement s'avérer très importants, sachant qu'il est primordial de rester sous la barre des 3 secondes pour fournir une expérience utilisateurs satisfaisante.

Comment fonctionne Phalcon ?

Lorsque l'internaute aura besoin d'obtenir une page, il va déclencher une requête au serveur. Sur celui-ci, Phalcon prendra la main et jouera le rôle de connecteur. Il va récupérer les données de cette page stockées au format JSON sur le back-end du site. Suite à cela, il enverra ces données à la vue qui va se charger de fusionner ces données et retourner la page complète à l'internaute.

Pourquoi Phalcon ?

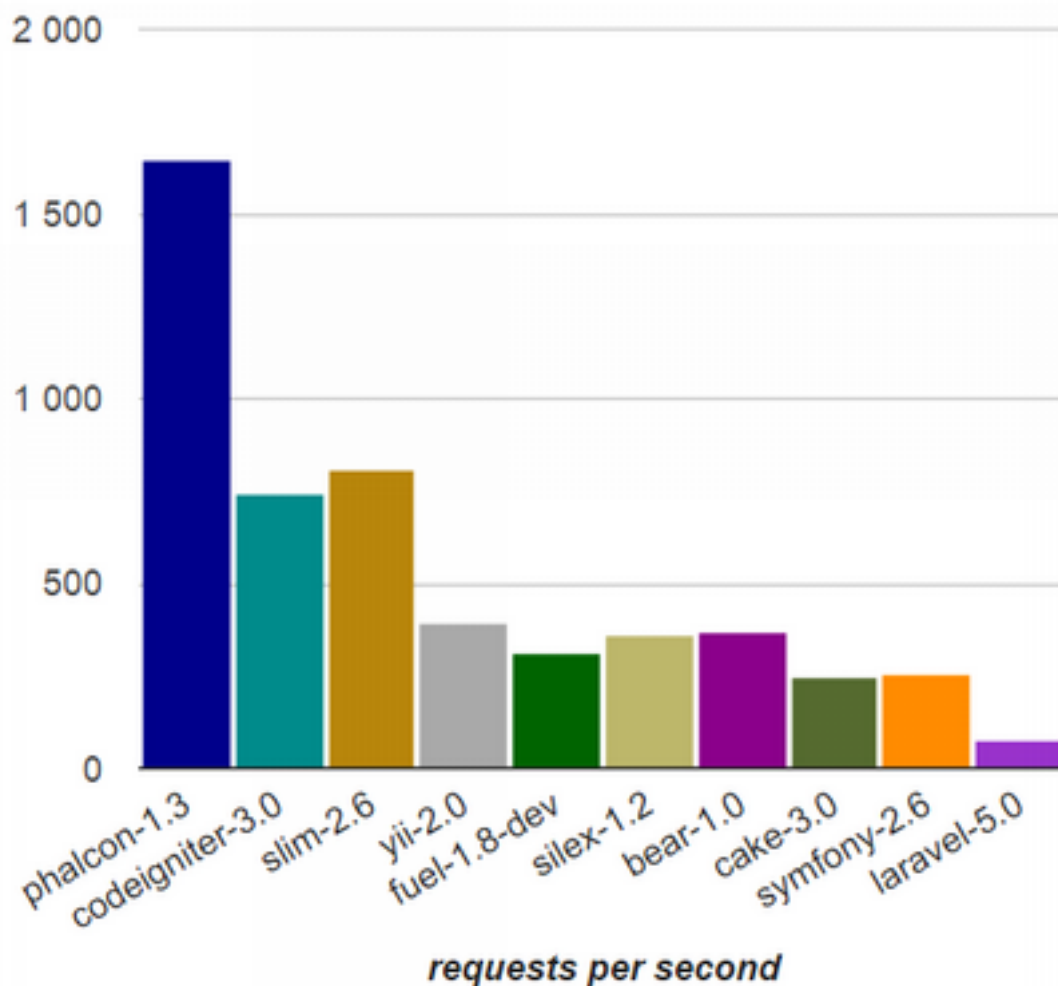
Comme mentionné plus tôt, la problématique principale de ce site est d'être capable de gérer plusieurs milliers de requêtes par minutes sans rallonger le temps de réponse à celles-ci tout en optimisant au maximum le ratio coûts d'infrastructure/performances.

Ce site internet a un nombre de visite moyen assez stable pouvant être géré par une infrastructure d'une puissance correcte ne justifiant pas l'investissement dans une infrastructure matérielle onéreuse, surtout que son utilité ne se révélerait que lors de pics de connexions, ceux-ci étant imprévisibles et rares.

La question se pose alors de savoir comment être capable à tous moment de gérer l'afflux massif d'utilisateurs sans intervention manuelle, que ce soit de manière logicielle ou matérielle. La réponse à cette question consiste à trouver un framework PHP assez rapide pour compresser le temps entre l'émission de la requête et la réception de la réponse à celle-ci sur l'appareil de l'utilisateur.

Et c'est là où Phalcon tire son épingle du jeu.

En se fiant à une analyse comparative produite par [Blog A Way Out](#), nous pouvons constater que Phalcon est bel et bien capable de gérer notre problématique. En comparaison avec Symfony, Phalcon est entre 10 et 20 fois plus rapide !



Comme nous pouvons le voir, Phalcon est capable de gérer plus de 1600 requêtes par seconde avant de commencer à montrer des signes de faiblesse, c'est déjà deux fois mieux que son concurrent direct ne pouvant gérer au maximum 700 requêtes par seconde.

Pour expliquer la performance de Phalcon, analysons les faiblesses des frameworks PHP classiques :

- Ils chargent en mémoire une grande quantité de fichiers
- Seulement un pourcentage assez réduit de chaque fichier sera utile
- Ces fichiers chargés sont compilés en bytecode qui ont besoin d'être interprétés lorsque nous souhaitons les exécuter.

Le résultat est que chaque requête fait appel à des fichiers plus lourds, partiellement utiles qui ont besoin d'être interprétés pour être utilisés. Ce sont ces points qui handicapent la rapidité d'une réponse à une requête.

La résultante de l'utilisation d'un framework PHP « classique » est que chaque requête a un poids trop important pour le serveur. C'est sur ces points que Phalcon excelle en proposant une solution à chacun de ces problèmes.

Cette performance est due à la conception même de Phalcon. En effet, Phalcon propose des classes déjà compilées, ce qui signifie que le code est chargé une seule et unique fois, suite à cela il sera prêt à être exécuté à la réception des requêtes. Les requêtes sont alors plus légères, et donc plus rapidement traitées, ce qui est idéal pour notre site à très fort trafic, même pendant les pics de fréquentation.

Conclusion : Le faucon dans nos serveurs.

Nous avons pu constater que Phalcon s'avère d'une utilité redoutable lorsqu'il est question de performance. Concernant les sites à faible trafic, l'utilisation d'un framework PHP classique reste tout à fait pertinente dans la mesure où le temps de chargement ne sera que très peu impacté.

Cependant, pour mettre en place Phalcon, à la différence des frameworks PHP classiques, il est nécessaire d'avoir un accès root au serveur du site sur lequel vous souhaitez l'implémenter.

Du côté du site à fort trafic, lors de tests de montée en charge, nous avons pu constater que Phalcon tient effectivement ses promesses tant en matière de nombre de requêtes qu'en temps de réponse, même lors de pics de trafic.