

NGINX



Classe : SMAGR1A

Présenté par

ASSI Adou Desquite Abilbord
ALZOUMA Moussa Mahamadou
BAMBA Adama
KOUAKOU Koffi Michel Osée
KOUAME Frankie

Professeur

N'GUESSAN-BLE Stéphane

SOMMAIRE

INTRODUCTION.....	3
I- QU'EST-CE QUE NGINX ?	3
II- INSTALLER ET CONFIGURER NGINX.....	3
III- AVANTAGES ET LIMITES NGINX.....	5
1- Avantages.....	5
2- Limites.....	5
CONCLUSION	5

INTRODUCTION

Inévitables dans le monde technologique, les sites web (aussi appelé site internet) sont un ensemble de fichiers HTML stockés sur un ordinateur connecté en permanence à internet et hébergeant les pages web (serveur web) qui permettent leurs fonctionnements.

Certains serveurs sont seulement accessibles sur des réseaux privés (intranet) et hébergent des sites utilisateurs, des documents, ou applications internes à une entreprise. Tandis qu'il y a des serveurs dits libres qui sont accessibles à tous. Ainsi parmi ces serveurs web libres nous pouvons citer Apache HTTP Server, NGINX, etc. ; dans notre cas NGINX fera l'objet de notre présentation.

Nous vous parlerons de NGINX, son installation, sa configuration, ses avantages et limites par rapport au serveur web APACHE.

I- QU'EST-CE QUE NGINX ?

Le logiciel de serveur NGINX, développé par le programmeur Igor Sysoev, a été distribué en 2004. L'attention était à l'époque centrée sur le besoin d'un serveur de haute performance, pouvant servir simultanément autant de clients Web que possible, et qui ne soit pas trop gourmand en ressources.

Etant donné la propagation croissante de son serveur Web, NGINX a depuis développé, en parallèle de la version libre gratuite, une version payante nommée NGINX Plus avec une assistance intégrée et des fonctionnalités supplémentaires comme un équilibreur de charge, ou load balancer (serveur/know-how/load-balancer-repartition-de-charge-sur-un-serveur/) en anglais. NGINX présente, tout comme Apache, une interface modulaire.

Cela signifie que différentes fonctions sont proposées sur les modules correspondants, que les administrateurs choisissent d'activer ou non.

Vous avez par exemple le choix entre ces différentes fonctionnalités :

- Application Acceleration (accélération de l'application) : permet un chargement plus rapide des contenus.
- Reverse Proxying (proxy inverse) : permet d'utiliser NGINX comme proxy inverse (http, TCP, UDP) pour l'accélération des chargements Web ou l'utilisation d'un proxy email (IMAP, POP3, SMTP).
- Chiffrement TLS : permet un échange de données sécurisé.
- Gestion de la bande passante : associe la bande passante optimale pour chaque service proposé.
- Load Balancing (équilibreur de charge) : répartit les demandes de manière à décharger le serveur principal.
- Videostreaming (streaming vidéo) : offre une haute performance pour le streaming de fichiers MP4 et FLV.

Tandis qu'Apache ouvre une nouvelle tâche ou processus pour chaque requête client, un serveur NGINX est orienté vers les événements. De cette manière il peut travailler sur des demandes de manière asynchrone, économisant ainsi de la mémoire vive et du temps. Ce logiciel de serveur est compatible avec un grand nombre de systèmes d'exploitation, notamment Unix/Linux, Mac OS et Windows.

II- INSTALLER ET CONFIGURER NGINX

➤ Installation sous Windows

Téléchargeons Nginx et plaçons le dans un dossier de notre choix (**D:\Nginx** par exemple)

Afin de spécifier le "**DocumentRoot**" voulu (répertoire racine des fichiers Web),

Il nous faut modifier le fichier **nginx/conf/nginx.conf** afin que la section server (**listen 80**) possède l'emplacement voulu.

```
location / {  
    root D:\gitlocal\web; # <-- ligne modifiée  
    index index.html index.htm;  
}
```

➤ Configuration sous Windows

Il faut tout d'abord télécharger et installer PHP.

Ensuite, rééditer le fichier *nginx.conf* et modifier les lignes en commentaires qui sont en rapport avec PHP FastCGI.

```
location ~ \.php$ {  
    root D:\gitlocal\web; # <-- Ligne modifiée  
    fastcgi_pass 127.0.0.1:9123;  
    fastcgi_index index.php;  
    fastcgi_param SCRIPT_FILENAME /scripts$fastcgi_script_name;  
    include fastcgi.conf; # <-- Ligne modifiée  
}
```

Démarrer PHP et Nginx pour terminer

➤ Installation sous Linux

Avant toute manipulation, il faut se connecter en utilisateur root

- Sous Debian Wheezy, pour avoir la dernière version stable de Nginx il faut d'abord ajouter les sources du dépôt DotDeb et ensuite récupérer la clé pour accéder au dépôt

```
$ vim /etc/apt/sources.list
```

- Et y ajouter les lignes suivantes :

```
Dev http://packages.dotdeb.org wheezy all  
Dev-src http://packages.dotdeb.wheezy all
```

- Récupération de la clé de dépôt

```
$ wget http://www.dotdeb.org/dotdeb.gpg  
$ cat dotdeb.gpg | sudo apt-key add -
```

- Nous pouvons ensuite installer Nginx : commande d'installation en linux

```
$ apt-get update  
$ apt-get install nginx
```

- Démarrage de Nginx

```
$ service nginx start
```

La page de démarrage de Nginx se trouve dans le dossier */usr/share/nginx/html*

➤ Configuration sous Linux

La configuration de Nginx est morcelée en plusieurs fichiers qui se situent tous dans le dossier */etc/nginx* :

- Le fichier *nginx.conf* contient la configuration générale
- Le fichier *conf.d/conf* (ou *sites-enabled/.conf*) contient la configuration des différents "servers"

III- AVANTAGES ET LIMITES NGINX

1- Avantages

- ❖ Gère un grand nombre de connexions simultanées. Nginx traite plus de 10000 connexions simultanées alors qu'Apache n'est limité qu'à seulement 256 connexions simultanées.
- ❖ Extrêmement rapide grâce au pattern REACTOR
- ❖ Léger grâce à son modèle Thread Driven
- ❖ Configuration plus facile que celle d'Apache car seuls les paramètres utiles sont disponibles
- ❖ Un serveur proxy inverse pour le protocole HTTP, HTTPS, SMTP, POP3 et IMAP
- ❖ Un équilibreur de charge et un cache HTTP
- ❖ Un proxy de frontend pour Apache et d'autres serveurs Web, exemple la combinaison entre la flexibilité d'Apache et la bonne performance de contenu statique de NGINX
- ❖ NGINX est connu pour sa haute performance, sa stabilité, son riche ensemble de fonctionnalités, sa configuration simple et sa faible consommation de ressources.

2- Limites

- ❖ Moins configurable qu'Apache
- ❖ Fonctionnalités limitées - Impossibilité de rajouter des modules via ligne de commande.
Il faut nécessairement recompiler Nginx pour avoir les modules souhaités
- ❖ Un hébergement partagé de domaines protégés par un mot de passe ou attribué en interne peut être difficile à réaliser.
- ❖ Les réécritures (rewriting) que cela implique ne peuvent qu'être mises dans le dossier principal de configuration et le serveur doit être relancé par la suite.
- ❖ Des erreurs peuvent avoir des conséquences sur tous les acteurs d'un hébergement partagé.
- ❖ Alors qu'un serveur Apache apporte quant à lui un avantage non négligeable, celui d'attribuer des fichiers .htaccess individuels séparés.

CONCLUSION

NGINX peut être un excellent moyen de servir votre application ou même être utilisé comme un proxy HTTP ou un équilibreur de charge pour vos autres serveurs Web. Comprendre son fonctionnement et son traitement donneront beaucoup de puissance et permettront équilibrer la charge de vos applications.

Selon notre étude, les deux serveurs Web NGINX et APACHE, dans leurs dernières versions, peuvent rivaliser l'un avec l'autre dans la plupart des domaines.

Pour le contenu statique, NGINX est un roi, mais pour un contenu dynamique, la différence de performance est assez mince. NGINX brille avec certaines de ses fonctionnalités plus avancées (diffusion multimédia, reverse proxy pour les protocoles non HTTP), ainsi que son soutien commercial et sa formation.

Les propriétaires de sites Web à fort trafic qui doivent servir beaucoup de contenu statique et / ou de flux multimédia préféreront probablement NGINX (ou utiliseront une combinaison d'Apache et NGINX).

Dans la plupart des autres cas d'utilisation de site Web, aussi bien Apache que Nginx feront le travail très bien. Ainsi de quelles manières, pourrions-nous maximiser les performances de NGINX ?