



## Table des matières

1).....	1
2).....	1
3).....	1
4).....	1
5).....	2
6).....	2

### **1)**

#### mod\_php :

Extrait du site [www.php.net](http://www.php.net)

PHP est un langage de script côté serveur, orienté Web et multi plate-forme.

Au départ c'était simplement un moteur de livre d'or, puis il s'est développé encore et encore jusqu'à la version 3 qui était vraiment un langage de développement Web puissant

Du point de vue des auteurs:

Depuis la version 4, il est assez robuste pour des applications web professionnelles. Il est très puissant et supporte nativement toutes les principales bases de données, et les autres via ODBC. Il est un poil plus rapide que ASP sur systèmes M\$.

Il existe d'autres extensions comme apc qui accélère les traitements de 50 à 400% (en fonction du code écrit)

#### CGI (Common Gateway Interface) :

Le CGI est un protocole de communication entre un serveur (apache, Nginx...) et une page php. Elle permet l'exécution d'un nouveau processus à chaque requête, permettant ainsi la génération dynamique des pages.

Dans le cas de CGI, chaque requête lance une nouvelle instance de CGI, qui appellera le programme à exécuter. Le binaire cgi recrée à chaque appel le contexte de l'[environnement d'exécution](#) et ne permet pas de limiter le nombre de processus simultanés. Le nombre de processus simultanés sera donc dépendant du nombre de processus simultanés du serveur web.

Avec FastCGI, les applications générant les [pages dynamiques](#) peuvent se situer sur un ou des serveur(s) différent(s) du ou des serveur(s) hébergeant le service HTTP. Une variable est introduite permettant de déterminer le nombre minimum et maximum de processus CGI à exécuter, indépendamment du nombre de processus HTTP maximum.

Le FastCGI est une version améliorée du CGI. On a tout simplement ajouté une librairie `spawnfcgi` entre le serveur et la page php qui est tenue en C. Cette librairie programmée en C permet de faire tourner un démon et ainsi ne pas avoir besoin de recréer une nouvelle instance à chaque fois.

### Fpm-php :

FPM (FastCGI Process Manager) est une implémentation alternative à PHP FastCGI avec quelques fonctionnalités additionnelles particulièrement utiles pour les environnements à haute charge.

Ces fonctionnalités incluent :

- Gestion avancée des processus avec stop/start doux (graceful) ;
- Possibilité de démarrer des processus avec différents uid/gid/chroot/environnement, écoutant sur différents ports et utilisant différents php.ini (remplace le `safe_mode`) ;
- Journalisation stdout et stderr ;
- Redémarrage d'urgence en cas de destruction accidentelle du cache opcode ;
- Support de l'upload accéléré ;
- "slowlog" - journalisation des scripts (pas juste leurs noms, mais leur backtrace PHP également, utilisant `ptrace` ou équivalent pour lire le processus distant) qui s'exécutent de manière anormalement lente ;
- [fastcgi\\_finish\\_request\(\)](#) - fonction spéciale pour terminer la requête et vider toutes les données tout en continuant d'exécuter une tâche consommatrice (conversion vidéo par exemple) ;
- Naissance de processus fils dynamic/static ;
- Informations sur la SAPI (similaire à `mod_status` d'Apache) ;
- Fichier de configuration basé sur `php.ini`

## 2)

Pour mettre en place un serveur Nginx je fais tout simplement :

```
apt install nginx
```

Maintenant je dois installer php-fpm :

```
apt install php-fpm
```

Je modifie le fichier de configuration du serveur :

```
server {
    listen 80;
    listen [::]:80;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;

    root /var/www/html;

    # Add index.php to the list if you are using PHP
    index index.html index.htm index.nginx-debian.html
index.php;

    server_name localhost;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a
404.
        try_files $uri $uri/ =404;
    }
}
```

```
# pass PHP scripts to FastCGI server
#
location ~ /\.php$ {
    include snippets/fastcgi-php.conf;
#
#       # With php-fpm (or other unix sockets):
#       fastcgi_pass unix:/run/php/php7.3-fpm.sock;
#       # With php-cgi (or other tcp sockets):
#       fastcgi_pass 127.0.0.1:9000;
}

# deny access to .htaccess files, if Apache's document
root
# concurs with nginx's one
#
#location ~ /\.ht {
#    deny all;
#}
}
```

Je modifie le fichier /etc/php/7.3/fpm/php.ini :

```
cgi.fix_pathinfo=0
```

Je n'ai plus qu'à créer un fichier index.php dans /var/www/html/ et y mettre ce que je veux :

```
<?php
    phpinfo( );
?>
```

Je relance le serveur et j'actualise ma page web :

```
root@202-15:~# nginx -s reload
```

Dans ma page web je met [http://Mon\\_IP/index.php](http://Mon_IP/index.php) et je dois voir mon site web.

### **3)**

Spawn-fcgi est un paquet. Je peux donc faire des recherches dessus avec un apt show :

```
root@202-15:~# apt show spawn-fcgi
Package: spawn-fcgi
Version: 1.6.4-2
Priority: optional
Section: web
```

```
Maintainer: J  r  my Lal <kapouer@melix.org>
Installed-Size: 44,0 kB
Depends: libc6 (>= 2.4)
Homepage: http://redmine.lighttpd.net/projects/spawn-fcgi
Tag: implemented-in::c, role::program
Download-Size: 14,8 kB
APT-Sources: http://deb.debian.org/debian buster/main amd64
Packages
Description: g  n  ration (spawn) de processus FastCGI
 Ce programme permet aux processus FastCGI d'  tre s  par  s des
 processus de
 serveur web :
   - cr  ation ais  e de socket avec changement de mode (chmod) ;
   - s  paration des privil  ges sans besoin d'ex  cutable Setuid ou
     d'ex  cution d'un serveur en tant qu'administrateur ;
   - possibilit   de red  marrer un serveur web et les applications
FastCGI
   sans red  marrer les autres ;
   - ex  cution des processus dans des chroot() diff  rents ;
   - ex  cution des applications FastCGI en ne d  pendant pas du
serveur web
   en cours d'utilisation, autorisant un test de migration plus
facile
   vers d'autres serveurs web.
```

Il permet de garder en d  mon les instances qui ont   t   cr    .

Pour le mettre en place avec Nginx sur un serveur affichant "hello world" programm   en C je dois commencer par installer le paquet :

```
apt install spawn-fcgi
```

Je dois aussi importer la librairie du devkit-fcgi pour pouvoir utiliser les librairies dans le programme C :

Je t  l  charge le dossier .tar.gz et je le d  compresse

J'installe le paquet libfcgi-dev :

```
apt install libfcgi-dev libfcgi
```

Je dois faire le programme en C qui affichera hello world dans le bon chemin donn   dans le fichier default : (ici /var/www/html/)

```
#include "fcgi_stdio.h"
#include <stdlib.h>
```

```
void main(void)
{
    int count = 0;
    while(FCGI_Accept() >= 0)
        printf("Content-type: text/html\r\n"
               "\r\n"
               "<title>Hello World!</title>"
               "<h1>Hello World!</h1>");
}
```

Je le compile :

```
gcc fcgi.c -lfcgi -o programme.fcgi
```

Je dois maintenant modifier le fichier default dans /etc/nginx/sites-enabled :

```
location /programme {
    include /etc/nginx/fastcgi_params;
    fastcgi_pass 127.0.0.1:9000;
}
```

Je peux maintenant redémarrer le serveur nginx :

```
nginx -s reload
```

Maintenant je lance en démon avec la commande spawn-fcgi le fichier .fcgi compilé :

```
spawn-fcgi -a127.0.0.1 -p9000 -n programme.fsgi
```

Si je vais sur ma page web et que je tape Mon\_IP/Mon\_fichier je dois voir s'afficher "Hello World!"

## 4)

Les arguments sont transmis via des requêtes GET ou POST au programme C avec la variable QUERY\_String :

```
fastcgi_param QUERY_STRING      $query_string;
```

**5)**

**6)**