



Nginx, Php-Fpm и что это вообще?

PHP-FPM - это разновидность SAPI для PHP. Чтобы понять, что это такое, стоит рассказать о понятии SAPI.

FPM

SAPI, он же Server API. В php есть несколько таких API для разных вариантов его работы:

1. CLI SAPI - в качестве консольной команды `php` для запуска наших кронов и других cli-программ (Command Line Interface)
2. apxs2 SAPI - в качестве модуля к apache2
3. CGI SAPI - в качестве запускаемого на каждом запросе CGI (сейчас так почти никто не делает)
4. FPM SAPI - Fast Process Manager, написанный для PHP разработчиками из компании Badoo и теперь поддерживаемый сообществом

Работа с FPM отличается от работы с Apache в первую очередь тем, что FPM - это только PHP. Это не веб-сервер, не что-то умное. Это наоборот - максимально простой,

легкий и быстрый менеджер процессов для PHP. В отличие от апача, он даже не использует http-протокол, а работает со специальным fastcgi-протоколом. В первую очередь FPM быстрее обрабатывает запросы благодаря его легковесности и простоте.

Во вторую очередь, FPM действительно умный менеджер процессов. Он контролирует количество работающих PHP-процессов, частоту их перезапуска для борьбы с утечками памяти (да, модули php как и всегда текут) и прочие простые вещи, необходимые для контроля сервера.

Нужно помнить, что независимо от того, какое SAPI вы используете, будь то модуль Apache, CGI или PHP-FPM - это не отнимает ни каких особенностей php. А первая его особенность:

1. Один процесс одновременно обрабатывает один запрос. Это абсолютно так же свойственно для PHP-FPM, как и для Apache.
2. Количество процессов определяет, сколько одновременно может "висеть" запросов в обработке.
3. Точно также, как и Apache, FPM подвержен DoS-атакам путем "длительных запросов". Допустим, у Вас на сервере работает не более 50-ти процессов PHP-FPM, а это значит, что если 50 пользователей одновременно начнут делать upload файла (пусть даже небольшого, но главное, чтобы они делали это медленно) -

пятьдесят первый пользователь получит ошибку 504, т.к. FPM не возьмет его запрос на обработку, пока не разберется с теми, что у него уже есть.

NGINX

Nginx - это разработанный Игорем Сисоевым http-проху-сервер (он сам чаще называет его проху-сервером, чем web-сервером). Это и есть его основное отличие от Apache (обычно к nginx приходят те, кто испытывает проблемы с Apache). Благодаря тому, что Nginx сам не выполняет никакой тяжелой работы, Игорь смог заложить в него прекрасную асинхронную событийную архитектуру.

Благодаря этой архитектуре nginx на порядки быстрее обрабатывает запросы, чем любой другой сервер и благодаря ей же потребляет при этом сильно меньше ресурсов. Как это происходит?

Один рабочий процесс nginx обрабатывает не один запрос пользователя (как apache), а тысячи этих запросов. Ввиду того, что nginx - это проху-сервер, для него не составляет никакого труда получить запрос пользователя, отправить его на backend (например php-fpm), а пока бекенд занят трудом - обрабатывать остальные запросы пользователей, когда FPM ответит Nginx-у о том что тот самый первый запрос обработан и отдаст ответ, nginx передаст ответ назад пользователю.

Nginx работает как конвейер - он просто быстро перекладывает запросы и ответы между backend и пользователями.

В эту схему отлично вписалась асинхронная работа со статическими файлами. Благодаря тому, что в современном мире с файлами можно работать почти так же асинхронно, как и с backend, Nginx отлично разделяет работу на две части: статику отдает с диска, динамику обрабатывает в PHP-FPM.

В чем выигрыш?

Возьмем тот же Apache (prefork или itk). Мы выставили у него максимальное количество рабочих процессов, равное 35. Что это значит? Это значит, что мы сможем одновременно обработать только 35 запросов пользователей и это не важно - запрос это за статикой или за динамикой. 35 всего.

У вас на странице 100 картинок+js+css-ок? Значит, большая их часть будет висеть в очереди внутри сервера Apache и ждать, когда пользователь получит предыдущие 35 ответов.

В случае с Nginx + PHP-FPM важнее всего количество процессов PHP-FPM. Мы можем поставить его таким же равным тридцати пяти. Что это значит? Это значит, что мы можем одновременно обрабатывать 35 запросов к **динамике**, запросы же к статике nginx разгребет своими силами в количестве 3000+ одновременных почти на любой слабой VPS.

Расход оперативной памяти при использовании Nginx+PHP-FPM меньше, чем на "голом" Apache, при равном количестве процессов (FPM и Apache). Скорость обработки запросов выше.

На расход CPU внутри PHP замена Apache на FPM никак не скажется. CPU в первую очередь кушает Ваш PHP-код, а пока его никто не оптимизирует, работать быстрее и экономичнее он не начнет.

Итог

1. Все проблемы PHP (процесс на запрос, не оптимальный код самого проекта) никуда не деваются.
2. Появляется возможность перелопачивать тонны запросов за статикой, которой нет в Apache.
3. Вы экономите оперативную память, что сказывается на цене оборудования или VPS.
4. Появляется море приятного функционала самого Nginx.
5. Пропадает возможность использовать htaccess, но честно скажу, конфигурация nginx куда более простая и понятная, чем htaccess.

Установка PHP и модулей на Ubuntu/Debian

В Debian/Ubuntu это делается легко и просто, зачастую не требуется ничего собирать

(/journal/ustanovka-modulei-php-na-ubuntu-debian/)

Капризы WebSocket и при чём здесь костыли

Протокол WebSocket имеет свои преимущества и свои недостатки:
детальный разбор

(/journal/problemu-websocket-v-srednei-polose-rossii/)

Curl в PHP

Curl - это библиотека для работы с URL. Она интегрирована во множество языков, в том числе и в PHP.

(/journal/curl-v-php/)

Почему timeout для curl в php необходим

Использование cURL в PHP имеет свою не очень приятную особенность - вечная блокировка процесса

(/journal/pochemu-timeout-dlia-curl-v-php-neobkhodim/)

IoT Highload: особенности и подводные камни

Особенности серверных приложений, работающих с сетью IoT-устройств на практике и в теории

(/journal/iot-highload-osobennosti-i-podvodnye-kamni/)

О безопасности, рисках и вложениях

Как сделать так, чтобы Ваш IT-отдел не стал местом утечки информации, за которую Вы отвечаете?

[\(/journal/o-bezopasnosti-riskakh-i-vlozheniiakh/\)](/journal/o-bezopasnosti-riskakh-i-vlozheniiakh/)

Продукты и услуги

[Аутсорсинг \(/system-administrator/\)](/system-administrator/).

[haCron \(/system-administrator/tech/haCron.html\)](/system-administrator/tech/haCron.html).

[Оборудование \(/hardware/\)](/hardware/).

Ресурсы

[VDS \(/vds/\)](/vds/).

[HTTP-балансировщик \(/sale/http-balancer.html\)](/sale/http-balancer.html).

[Доп. ресурсы \(/sale/\)](/sale/).

О нас

[Контакты \(/contacts.html\)](/contacts.html).

[Perfect Journal \(/journal/\)](/journal/).