Пыхопроблемы

На похапэ можно писать довольно быстрый код. Конечно, не такой резкий, как заточенный под конкретную задачу инстанс nodejs или, не дай бог, что-нибудь для веб на няшной с ассемблерными вставками.

Можно банально придерживаться паттерна mvc и не погрязнуть в паутине спагетти-скриптов с сотнями инклюдов. Код будет хотя бы структурирован и изолирован локальными кучками говнеца. Это идеальное состояние, если большую часть рабочего времени вы добавляете в общую свалку новые, независимые друг от друга конвертики с тухлятиной.

Можно написать классы объектов, там где они необходимы и наполнить их методами. Опять же, всё ради структурирования кода, для вполне сносной и быстрой навигации по разрастающейся выгребной яме проекта.

Можно вооружиться профайлером, раскурить исходники ядра фреймворка, который вам предписало начальство, и частично переписать его, снизив время выполнения этого хитросплетения пиздеца на 80%. Вырезать конфиг веб-приложений, сделанный в xml. Уничтожить миллионы вызовов __call() и call_user_func(), от которых кровоточат глаза. Большинство макак знает, что обычное веб-приложение на похапэ инициализируется каждый раз с нуля. Поэтому уменьшить на 90% время инициализации - это очень хорошая идея.

Можно искать узкие места и куски рендерера, где хтмл генерится недостаточно быстро. Вооружиться memcached и реализовать грамотные схемы самообновляющегося блочного кеширования. Избавиться от пары дюжин лишних запросов к бд на каждый чих. Получить 80% страниц, выхлоп которых отрабатывает без запросов к бд вообще.

Можно заняться очередями сообщений и перенести на них особенно тяжёлые куски процессинга картинок, видео, музычки, почты и прочего хлама, чтобы всё упиралось в длину очереди, количество воркеров и машины, эти очереди разгребающие, а не в число клиентов и их терпение к времени отклика от сервера.

Можно навесить плюшки в виде аякса, где это уместно, и местами перенести генерацию контента вовсе на клиент, вместе с тем сэкономив десятки тяжёлых запросов на отрисовку страницы целиком.

Можно взять сверхбыстрое простое хранилище типа redis и использовать его для сегментов системы, которые создают большую плотность не очень важных запросов к бд, типа учёта баннеропоказов, трекинга статусов online и логирования всякой поебистики.

Можно придти к мысли, что mysql с её слоупочными table locks и transactional safety и с её возможностью масштабирования только при помощи анальных расширителей не очень-то, собственно, и нужна в большинстве задач. Потратить 2 месяца и перенести огромную смердящую

кучу наваленных друг на друга небольших пакетиков с говном на mongodb, на небольшой, но няшный кластер из нескольких replica sets по тройке лёгких машин. Ощутить невесомое изящество, с которой она похрустывает сотнями тысяч записей, прелесть schema-free и отсутствие дрожи в коленях, когда раньше ты запускал a lter table на рабочей копии бд, глубокой ночью, потому что оно кладёт сервер на час-другой. А потом часами напролёт в умилении смотреть на графики munin, которые резко перебежали из погранично-красной зоны в самый низ зелёной. Финально включить eaccelerator и наслаждаться запасом в сотни запросов в секунду на отдельно взятом сервере начального уровня.

Можно дополнительно озаботиться настройкой nginx, убрать из конфига логгирование для файлопомойки, включить пяток жизненно-важных параметров, указать нормальные значения для буферов. Окончательно уничтожить апач, для которого был прописан reverse proxy для некоторых урлов. Выкинуть SATA-винты на помойку. Поставить дополнительно недорогих SSD и развернуть на них кэш для самой мелкой статики.

Только это всё не нужно. Ваш сайт, результат вашей работы никогда не получит хоть какой-то нагрузки. Когда на ресурс заходит 10 человек в день, а 90% хитов совершают боты гугла, можно хуярить страницы на 50, и даже на 150 SQL-запросов, ведь все таблицы бд влезают в оперативку, и страница даже на каком-нибудь позапрошлогоднем zend framework без твиков соберётся менее, чем за секунду. Да какой там фреймворк! Какой там MVC! Проще дёргать по необходимости разнородные готовые куски, часть кода бросить голодным доширак-макакам, и склеить всё воедино лишь-бы-работало спагетти-кодом. Ведь проект нужно было сдать ещё вчера, а завтра он будет навсегда забыт. И останется крутиться на задрипанном, надолго предоплаченном vps, в cron которому прописана ежедневная перезагрузка.

Я кончил.

Автор: http://lurkmore.to/Копипаста:Программирование
Дата: 2019-09-14 14:49:46

(c) boot.nsk.ru