Визитка

ДМИТРИЙ НЕВАР, ведущий инженер интернет-провайдера «Универсум Бит». Сфера интересов – от ИТ до исторической реконструкции и фехтования. В последнее время занят организацией семинаров для системных администраторов

Mod_pizza

Фирменное блюдо от хостинга «Джино»*

Ежедневно мы получаем сотни мегабайт информации из Сети. Миллионы файлов в месяц, сотни музыкальных композиций, десятки фильмов. Мы подсели на этот безграничный источник информации. Однако редко задумываемся, где именно физически хранится данная информация. И потому расскажу о хостинге

Недавно мне удалось пообщаться с представителями одного из известных российских хостинг-провайдеров – «Джино», рассказавших об основных проблемах, с которыми им приходится сталкиваться.

В первую очередь это проблема обеспечения быстродействия и безопасности данных клиентов виртуального хостинга – наиболее востребованного и доступного вида размещения сайтов на данный момент. Самое «узкое место» здесь – это веб-сервер, который принимает все запросы от посетителей сайта и берет на себя задачу по формированию и передаче контента. Типовым решением является веб-сервер Apache, но из-за проблем с производительностью и быстродействием данное решение не самое оптимальное. На высоконагруженных системах время обработки запроса и выдача ответа от веб-сервера заметно увеличивается. С этим явлением можно бороться различными способами: например, связкой Apache + Nginx или выбором более подходящего MPM-модуля.

MPM-модуль (Multi-Processing Module) – сердце Apache, которое обеспечивает управление сетевыми соединениями и начальную обработку запросов. Фактически MPM-модуль выполняет всю работу между моментом поступления запроса и его обработкой, например, PHP-интерпретатором.

Традиционно большинство хостинг-провайдеров используют prefork. Он обеспечивает простейшую базовую модель: в памяти висит один или несколько процессов, и при поступлении нового запроса, если существующие процессы заняты, делается ветвление главного процесса на два и более. Мы понимаем, что каждое ветвление – это затраты процессорного времени, а соответственно издержки на время проведения операции. Пока процессов мало, это не страшно, но, как только накапливается критическая масса, время на ветвление значительно увеличивается, снижая общую производительность системы.

Есть альтернатива – worker. Но и с ним не все так гладко. Хоть он и является, по сути, многопоточным и выполняет каждый поток отдельно, это не отменяет проблемы безопасности, которая, согласитесь, крайне важна. Сейчас объясню, что с ней не так. Worker использует для взаимодействия между потоками общие части памяти и общие блоки. Следовательно, для обеспечения безопасности приходится использовать особые ухищрения и ограничивать все модули системы, например mod_php. Но главным недостатком данных модулей является то, что они запускают процессы от имени одного пользователя, что само по себе в рамках виртуального хостинга небезопасно.

В качестве универсального средства можно использовать suexec, но этот способ не решает полностью проблему: PHP и другие скрипты работают в режиме CGI. При обработке каждого скрипта suexec запускает отдельный процесс, что сильно (минимум в два раза) замедляет обработку и ограничивает функционал.

Существуют другие MPM-модули, призванные решить проблему с безопасной обработкой запросов без использования CGI, такие как mod_itk и mod_peruser. Код этих модулей основан на коде prefork и worker и является их довольно грубой переделкой. Для решения проблемы с безопасностью здесь используются сомнительные решения, в простонародье именуемые «костылями». Кроме того, эти модули не рассчитаны на большое число запросов – катастрофически растет потребление памяти, ведь каждый процесс перестаёт быть универсальным и становится заточенным только под определенного пользователя.

Специалисты «Джино», неудовлетворенные подобной ситуацией, решили разработать с нуля и внедрить собственный MPM-модуль – mod_pizza, который представляет собой сложный и хорошо проработанный механизм, рассчитанный на высокую нагрузку. Кроме обеспечения безопасности, оптимизированы и полностью переписаны многие функции, участвующие в обработке запросов и инициализации процессов, в связи с чем улучшены показатели потребления памяти и увеличена отзывчивость Apache в целом. Реализованы встроенные механизмы защиты от простейших DDOS-атак, ограничения скорости скачивания статического контента, ограничения нагрузки на процессор, времени выполнения скриптов.

Во время старта mod_pizza порождает множество легковесных процессов – так называемых Безработных, готовых

в любой момент стать Рабочими – получить команду, принимать запросы к определенному сайту и обрабатывать их с правами владельца этого сайта. Таким образом обеспечивается невозможность доступа Рабочим к данным других пользователей сервера. Это полностью решает традиционную проблему безопасности, присущую Apache: владелец любого сайта, зная структуру папок на сервере, мог получить доступ к информации своих «соседей».

Команды на инициализацию и сами запросы раздает специальный служебный процесс – Диспетчер. Он принимает внешние соединения и обрабатывает все запросы. В его задачу входит получение запроса, определение, к какому пользователю он пришел, и работа по инициализации и передаче запроса рабочей группе, привязанной к этому пользователю. Этот процесс имеет доступ к статистике обработки запросов и решений по инициализации Рабочих. Передачи запросов или же их блокировку и задержку он принимает на основании ее анализа. Таким образом, в mod_pizza реализованы защита от простейших DDOS-атак, балансировка и возможность начальной обработки, которая заключается в проверке правильности запроса и выдачи ошибки, если, например, домен, к которому пришел запрос, не обслуживается этим сервером.

Существуют и другие служебные процессы. Например, Бухгалтер – процесс, ведущий учет всех распоряжений Диспетчера, количества отработанных запросов и собирающий массу другой важной статистической информации.

Благодаря своей стройной внутренней архитектуре mod_pizza обладает многими преимуществами по сравнению с другими MPM. Он позволяет надежно и абсолютно безопасно работать с mod_perl, mod_wsgi/mod_python, mod_ruby и другими подобными модулями. По скорости mod_pizza не уступает, а часто и превосходит оригинальный модуль prefork, выполняющий скрипты без разделения прав, может обслуживать тысячи одновременных соединений. Очень эффективно устроена работа с памятью: для обслуживания даже очень большого количества пользователей и доменов достаточно лишь одного запущенного экземпляра Apache.

Почему «пицца»?

Когда мы покупаем пиццу, мы режем ее на кусочки, разделяя между собой. Одну пиццу едят сразу несколько человек, не залезая в общую тарелку. По тому же принципу работает и mod_pizza: разделяет ресурсы сервера между пользователями, полностью изолируя их друг от друга.

Mod_pizza дает возможность ограничивать выделяемые пользователю ресурсы в рамках виртуального хостинга. Скачки нагрузки на одном аккаунте никак не отражаются на других пользователях. Пользователи и администраторы могут мгновенно изменять настройки без перезапусков и остановок. Администраторы могут в реальном времени следить, кто из пользователей потребляет больше ресурсов, чьи скрипты зависают, кому не хватает выделенных процессов-Рабочих.

В настоящее время mod_pizza успешно работает на всем парке серверов виртуального хостинга «Джино», обеспечивая функционирование более 50 000 доменов только лишь в одной зоне .ru.

***

К чему я это все рассказываю? К тому, что сейчас в «Джино» прорабатывается вопрос о выпуске mod_pizza в виде продукта, доступного всем желающим. И я, как, думаю, и многие из вас, с нетерпением жду, когда же они выпустят модуль отдельным и, главное, открытым проектом. А до этого времени ощутить все прелести данной технологии я смог, благодаря переносу некоторых своих ресурсов на хостинг «Джино». EOF

** На правах рекламы*

Рисунок 1. Упрощенная схема работы mod_pizza