

«Colored crabs» включает в себя шесть модулей: «Операции над числами», «Конвертация чисел», «Работа с матрицами», «Нахождение площадей фигур», «Нахождение объёмов фигур» и «Дополнительные операции (НОК/НОД)».

При распределении модулей между членами команды осуществляется разделение ответственности, исходя из их ролей и обязанностей, которые и определяет модель организации работы в команде. К моделям организации команды можно отнести следующие виды: административная модель (теория X), модель хаоса (теория Y), открытая архитектура (теория Z).

При разработке программного продукта необходимо пользоваться современными технологиями разработки, которые имеют своей целью как упрощение, так и ускорение командного взаимодействия.

#### ЛИТЕРАТУРА

1. Ещё раз про семь основных методологий [Электронный ресурс] – Режим доступа: <https://habr.com/ru/company/edison/blog/269789/>. – Дата обращения: 03.04.2022

2. Этапы разработки программного обеспечения [Электронный ресурс] – Режим доступа: <https://ru.intechcore.com/stages-software-development/> – Дата обращения: 03.04.2022

УДК 004.07

Студ. А.А. Ткачёв

Науч. рук. ст. преп. И.Г. Сухорукова  
(кафедра программной инженерии, БГТУ)

### **ВЫСОКОНАГРУЖЕННЫЕ СИСТЕМЫ В PYTHON И НОВЫЕ ТРЕНДЫ В ВЕБ-ТЕХНОЛОГИЯХ**

В современной разработке веб-сервисов, программисты часто начали сталкиваться с проблемой highloadsystem (высоконагруженных систем, HL), это связано с тем, что число пользователей сети Internet увеличивается с каждым годом и число I/Obound операций в среде тоже растёт. Поэтому начались исследования в сфере поддержки HLsystem и связывание их с последними трендами в веб-технологиях.

Последние пять лет, можно увидеть увеличение интереса к асинхронным программам, и это не зря, так как это решает задачу I/Obound операций. С использованием асинхронного кода производительность программ, можно увеличить в несколько раз. Примерами асинхронных фреймворков в Python являются Tornado (интересный факт на нём был

написан Facebook в 2010-х годах), FastApi (тренд 2021-2022 в сфере разработке микросерверов на Python).

Очевидно, что приложение не может взаимодействовать с 10 тысячами пользователями в секунду, поэтому в горизонтальном масштабировании программных систем часто применяют два подхода: бессерверные технологии и докер технологии.

Рассмотрим докер технологии. Докер можно назвать гибкой изолированной системой, которая позволяет разворачивать приложение на любом компьютере или сервере, не завися от ОС, что и даёт гибкость. Также простота написания контейнеров позволяет разработчикам легко писать свои контейнеры для проверки своих приложений на сервере.

Докеры помогли решить проблему нагруженности приложений, но как мы будем распределять трафик пользователей по системе. Эту проблему решает использование обратного прокси-сервера (Reverse Proxy) или балансировка нагрузки (Load Balancers). Обратный прокси-сервер является регулировщиком трафика и служит шлюзом между пользователями и сервером-источником вашего приложения. При этом он обрабатывает все управление политиками и маршрутизацию трафика.

Лидерами на рынке является Nginx и ApacheServer. ApacheServer в последнее время показывает не Nginx самые лучшие результаты по сравнению с Nginx, он не решает проблему 10 тысяч, также больше требует оперативной памяти и процессорных возможностей. Поэтому предпочтительнее использовать Nginx.

Теперь мы можем построить простой веб-сервис, который может принимать более 1 миллиона пользователей в секунду (рисунок 1).

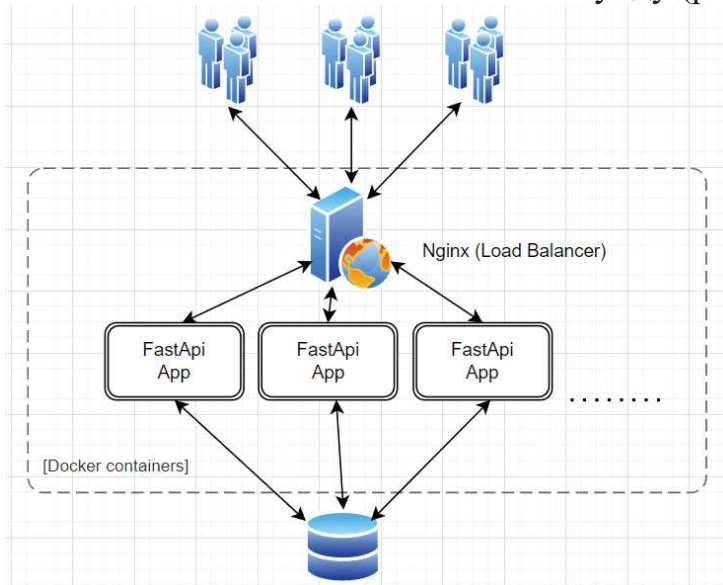


Рисунок 1 – Схема простой HL системы

Как мы можем увидеть нас ничего не ограничивает в горизонтальном масштабировании системы, кроме возможностей самого сервера запускать и обрабатывать информацию. Мы можем увеличивать каждый раз FastAPI приложения при высоких загрузках и уменьшать при низких загрузках, что даёт ещё дополнительную гибкость системы.

Пример такой схемы, можно увидеть в моем курсовом проекте «Видеохостинг “HangOut”» со стек технологий WPF, Docker, Nginx, FastAPI, Redis, Postgresq. Все выше описанные технологии позволили, создать высоконагруженную систему, которую могут использовать тысячи пользователей сети Internet, а также гарантировать надежность нашего сервиса от высоких нагрузок и DDOS атак.

#### ЛИТЕРАТУРА

1. Наш опыт с Kubernetes в небольших проектах [Электронный ресурс] – Режим доступа: [https://www.youtube.com/watch?v=XZQ7-7vej6w&ab\\_channel=HighLoadChannel](https://www.youtube.com/watch?v=XZQ7-7vej6w&ab_channel=HighLoadChannel) (дата обращения 04.04.2022)
2. ContainerizationExplained [Электронный ресурс] – Режим доступа: [https://www.youtube.com/watch?v=0qotVMX-J5s&ab\\_channel=IBMTechology](https://www.youtube.com/watch?v=0qotVMX-J5s&ab_channel=IBMTechology) (дата обращения 04.04.2022)

УДК 004.415.2

Студ. А.А. Бесман  
Науч. рук. ассист. А.В. Годун  
(кафедра программной инженерии, БГТУ)

#### **ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ УЧЕТА ХОДА ВЫПОЛНЕНИЯ ДИПЛОМНЫХ ПРОЕКТОВ НА КАФЕДРЕ ИСиТ**

Годы назад для ведения учета хода выполнения какого-либо процесса людям приходилось либо запоминать всё наизусть, что не совсем надежно из-за несовершенства человеческой памяти, либо же записывать всё на листок бумаги. С развитием компьютеров и сетей стало возможно несколько людям, подключенным к одной сети и имеющим доступ к компьютеру, подключенному к этой сети работать совместно над одним ресурсом. Это и дало толчок к развитию автоматизации процессов посредством создания компьютерных программ и приложений.

В связи с этим было решено написать веб-приложения, которое поможет как преподавателям, так и студентам без особых усилий вести отчетность по ходу выполнения дипломного проекта, что должно помочь улучшить качество выполнения дипломного проекта, т.к. данный