

МИКРОСЕРВИСНАЯ АРХИТЕКТУРА ВЕБ-ПРИЛОЖЕНИЙ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Щитляк А.Н.

Мухуров Н.И. – д.т.н., профессор

Возрастающие требования к современным веб-приложениям, такие как возможность предоставления программного интерфейса, интеграция с другими веб-приложениями, обработка большого количества запросов, масштабируемость, обеспечение необходимой скорости доступа к информации, приводят к тому, что корпоративные веб-приложения с монолитной архитектурой становятся неудобными в разработке, сложно тестируются и вводятся в эксплуатацию с большими временными задержками.

Микросервисная архитектура – это метод создания распределенных приложений в виде набора независимо разрабатываемых и развертываемых небольших сервисов, запускаемых как один или несколько изолированных процессов. Основная цель такого разделения – возможность изменения отдельного взятого микросервиса, не меняя при этом связанных с ним компонентов. Бизнес-логика приложения разбивается на отдельные части, каждая из которых представляет собой небольшое приложение, выполняющее одну бизнес-задачу. Число таких приложений ничем не ограничено и между собой они общаются, используя API, построенное, например, на основе HTTP протокола.

Структурная схема микросервисной архитектуры веб-приложения представлена на рисунке 1:



Рис. 1 - Структурная схема микросервисной архитектуры веб-приложения

Основные преимущества микросервисной архитектуры:

- использование различных языков программирования и программных средств, оптимальных для реализации каждого микросервиса;
- взаимозаменяемость микросервисов;
- независимость микросервисов друг от друга, каждый микросервис может быть развернут независимо от других служб;
- упрощение процесса масштабирования разрабатываемого веб-приложения;
- организация микросервисов как модулей вокруг отдельных функций.

Основные недостатки микросервисной архитектуры:

- сложность реализации общего поведения для всех микросервисов (авторизация запросов, агрегация данных из разных микросервисов);
- дополнительные издержки при обработке пользовательских запросов;
- необходимость учитывать, что любой из микросервисов в любой момент времени может быть недоступен.

Важно учитывать, что система на базе микросервисной архитектуры больше подвержена ошибкам из-за использования сети передачи данных для взаимодействия между микросервисами, что приводит к дополнительным издержкам при обработке пользовательских запросов и сложности реализации общего поведения для всех микросервисов (авторизация запросов, агрегация данных).

Подводя итоги, нельзя однозначно утверждать, что микросервисная архитектура – это будущее проектирования программного обеспечения, однако уже сегодня ясно, что микросервисная архитектура обладает большим потенциалом и серьезными преимуществами для разработки и реализации веб-приложений.

Список использованных источников:

1. Микросервисы, SOA и API: друзья или враги? – [Электронный ресурс]. – Режим доступа. – URL: https://www.ibm.com/developerworks/ru/library/1601_clark-trs/index.html
2. Применение микросервисной архитектуры при разработке веб-приложений – [Электронный ресурс]. – Режим доступа. – URL: <https://sibac.info/journal/student/18/87616>
3. Микросервисы – [Электронный ресурс]. – Режим доступа. – URL: <https://dev.by/lenta/anadea/mikroservisuy>