А. С. Качкин
(*ГГУ им. Ф. Скорины, Гомель*)

ОПТИМИЗАЦИЯ СЕРВЕРНОЙ ЧАСТИ ПРИЛОЖЕНИЯ ДЛЯ РАСЧЁТА ФИЗИЧЕСКОЙ НАГРУЗКИ ПУТЕМ РАЗДЕЛЕНИЯ ЕГО НА МИКРОСЕРВИСЫ

Монолитный сервер – самый очевидный способ построения back-end части приложения. Вся логика для обработки запросов выполняется в одном процессе. Данный подход позволяет использовать все возможности языка программирования. Возможно так же легко запустить приложение. Монолитные приложения легко масштабируются с помощью запуска дополнительных серверов. Такие приложения используются до сих пор, но их процент становится все меньше с каждым годом из-за прихода облачных технологий. Любые изменения сервера, даже самые незначительные, требуют полной перезагрузки сервера. Неудобства использования единого приложения привело к созданию архитектурного стиля микросервисов. Данный подход подразумевает разбитие приложения на набор сервисов.

Сервер приложения для расчета физической нагрузки состоит из следующих частей: модуль авторизации; модуль расчета нагрузки; модуль работы с журналом тренировок; модуль расчета питания.

Если требуется обновить один из модулей, то остальные будут недоступны в течении нескольких минут.

Согласно архитектуре микросервисов каждый из этих модулей должен быть отдельным проектом. В этом случае пользователь сможет использовать практически весь функционал приложения во время обновления одного из модулей.

Главный модуль приложения – это модуль авторизации, в нем проверяется доступ пользователя к приложению, генерируется авторизационный токен, который записывается в Redis. Redis – это главное хранилище данных back-end приложения. Он использует опера-

тивную память для хранения токенов, построения очередей запросов, а также для кэширования данных. Каждый из наших модулей имеет доступ к Redis, это сделано для того, чтобы отдельные сервисы могли проверить валидность токена, который приходит вместе с запросом, т.к. наши сервисы – отдельные приложения, которые ничего не знают друг о друге и для проверки авторизации они использую хранилище Redis для поиска токена, который был записан модулем авторизации. Данный подход сокращает риски при разработке приложений и тем самым ускоряет ее.