

Разработка платформы дистрибуции приложений, использующих передовые технологии машинного обучения и компьютерного зрения*

Р.Р. Алькапов, П.С. Костенецкий

Южно-Уральский государственный университет

В настоящее время существует множество работ в области машинного обучения и компьютерного зрения, которые представлены в научных работах и/или реализованы в виде программных пакетов. Для того, чтобы использовать данные работы необходимо иметь квалификацию в области компьютерных наук. Реализация полноценного приложения (включая интерфейс пользователя, работу с серверами) представляет из себя дополнительные затраты для разработчиков. В данной работе предложена технология, позволяющая решить эту проблему. Разработчику необходимо реализовать обработку данных своим алгоритмом в пределах одной функции – интерфейса платформы, который однозначно определяет модель данных, а именно формат входных и выходных данных. Реализованный интерфейс, исходный код с алгоритмом и описанием его программных зависимостей представляют из себя приложение с точки зрения разработчика. Данное приложение загружается на платформу, где на основе интерфейса платформа будет предоставлять пользователю возможность обработки его данных. Пользователь имеет доступ к просмотру всех приложений на платформе и выбирает определенное, подходящее для решения его задачи. За развертку и предоставление функционала после загрузки приложения ответственность несёт платформа. Таким образом платформа предоставляет унифицированный интерфейс пользователя с единым репозиторием приложений для запуска и программный интерфейс разработчиков.

Архитектура предложенной платформы представлена на рис. 1 и состоит из 6 основных компонентов. На данном рисунке блоками представлены компоненты платформы, а стрелками указаны зависимости по принципу: блок у начала стрелки управляет/выполняет запросы к блоку на конце стрелки.

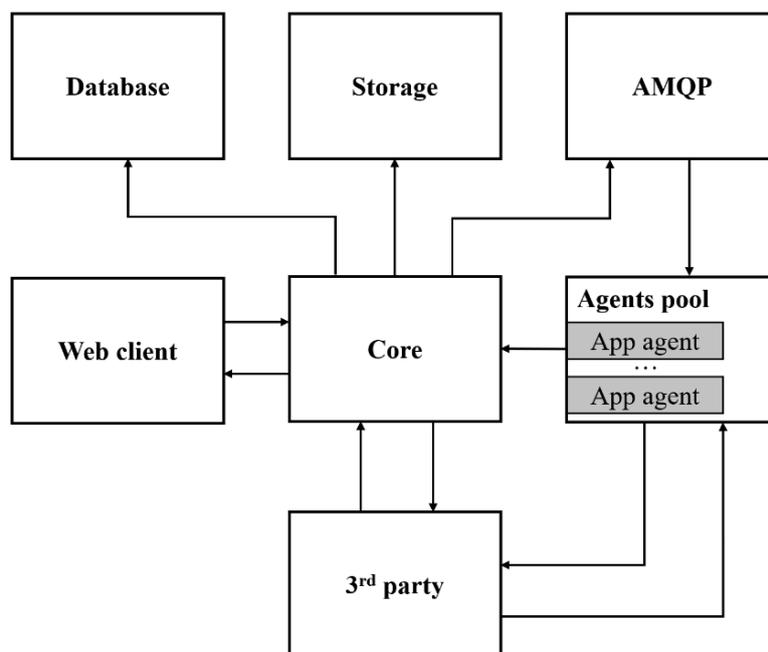


Рис. 1. Архитектура платформы дистрибуции приложений.

* Разработка выполнена при финансовой поддержке Фонда содействия инновациям в рамках договора 13944ГУ/2019 от 26.04.2019.

Веб-клиент (Web client) представляет собой конечную точку использования системы как со стороны как пользователей, так и разработчиков. Веб-клиент разработан в стиле Single Page Application [1] с помощью веб-фреймворка Angular. Для реализации использовались язык программирования TypeScript, с использованием языка разметки HTML и языка формального описания внешнего вида документов Stylus.

Ядро (Core) представляет собой веб-сервер, который отвечает за предоставления доступа к ресурсам системы и реализует бизнес логику платформы. Веб-сервер спроектирован в архитектурном стиле RESTful API [2] с использованием фреймворков Flask и Flask-Restful на языке Python 3. Организация безопасности осуществляется с помощью Json Web Token (RFC 7519). Взаимодействие между пользователями и системой осуществляется с использованием защищенного протокола HTTPS.

База данных (Database) предоставляет быстрый доступ к динамическим данным платформы. В качестве базы данных была выбран NoSQL база данных MongoDB [4], предоставляющая данные в формате BSON, который совместим с форматом Json, использующийся во всех частях программной платформы, в том числе при обмене данными с веб-клиентом. Для реализации базы данных используется официальный облачный сервис разработчиков – MongoDB Atlas, позволяющий использовать базу данных по принципу MongoDB as a Service. Сервис позволяет использовать преимущества данной базы данных, такие как горизонтальное масштабирование, репликация без дополнительных затрат по конфигурации аппаратного обеспечения.

Хранилище данных (Storage) предоставляет быстрый доступ к статическим данным платформы, который отличаются большим размером по сравнению с динамическими данными. В качестве хранилища данных используется Amazon S3. Для обеспечения политики безопасности S3 предоставляет механизмы контроля доступа (Access control list – ACL).

Очередь сообщений, организованная по принципу протокола Advanced Message Queuing Protocol [3] (AMQP) предоставляет отказоустойчивый и асинхронный процесс обработки пользовательских данных, а именно соединяет собой ядро и пользовательские приложения. В качестве программной реализации используется брокер сообщений RabbitMQ, развернутый с использованием сервиса CloudAMQP.

Пул агентов (Agents pool) представляет из себя множество агентов для разных приложений. Каждый агент представляет собой программную оболочку, которая несет логику для взаимодействия с очередью сообщений и ядром, предоставляя безопасный обмен пользовательских данных и управляет пользовательским приложением: подготавливает среду для запуска, создает необходимые каталоги для обработки пользовательских данных, которые не могут быть идентифицированы на уровне приложения, загружает результаты на ядро. Для изоляции приложений они помещены в docker-контейнеры, что обеспечивает дополнительный слой безопасности. Экземпляр каждого приложения может быть запущен множество раз для организации параллельной обработки пользовательских данных. Пользовательские приложения не изолированы виртуальной средой контейнера, где они развернуты и могут осуществлять доступ к сторонним приложениям (3rd party). Реализация и тестирование производились на языке Python 3.

Каждый компонент системы развернут на отдельном сервере с использованием облачных решений для уменьшения затрат и рисков, связанных с эксплуатацией оборудования. Данный подход позволяет обеспечить такие свойства платформы как масштабируемость и отказоустойчивость.

Предложенный стек технологий позволяет реализовать эффективное и отказоустойчивое приложение, которое удобно с точки зрения пользователей, не имеющих достаточной квалификации в области компьютерных наук и разработчиков.

Литература

1. Emmet S. Spa Design and Architecture: Understanding Single Page Web Applications. Manning Publications Company, 2015. 312 p.
2. Richardson L., Ruby S. RESTful Web Services. O'Reilly Media, 2008. 454 p.
3. Keig A. Instant RabbitMQ Messaging Application Development How-to. Packt Publishing, 2013. 54 p.
4. Nayak A., Dasadia C. MongoDB Cookbook, 2nd Edition. Packt Publishing, 2016. 370 p.