

ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ УДАЛЕННОЙ ПЕЧАТИ

Пашук А. В., Гуринович А. Б., Волорова Н. А.

Кафедра информатики, кафедра вычислительных методов и программирования, кафедра информатики,
Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: {pashuk, gurinovich, volorova}@bsuir.by

При организации документооборота в компаниях, использующих складские помещения, очень важно эффективно организовать этот процесс, чтобы максимально сократить время, необходимое для получения нужных документов, например, при комплектации новых заказов. Аналогичная ситуация наблюдается при приемке новых товаров, на которые необходимо распечатать и наклеить этикетки или другую информацию для дальнейшего перемещения товаров по складу.

ВВЕДЕНИЕ

Работники складов крупных компаний обычно одновременно используют несколько принтеров для печати различных видов документов. Например, для печати этикеток или другой печатной продукции, которая должна быть наклеена на товары, используются специальные принтеры, поддерживающие необходимый формат бумаги. Для оформления заказов или печати бланков договоров используются обычные или промышленные многофункциональные устройства печати.

Еще одним важным аспектом работы с документацией в крупных компаниях является тот факт, что принтеры подключены к нескольким компьютерам или серверам, что усложняет процесс печати. Сотруднику необходимо скачивать нужный документ и переходить от одного рабочего места к другому для печати документов.

В рамках данного проекта было разработано решение, позволяющее упростить процесс печати и осуществлять печать с любого печатного устройства без необходимости работать с компьютера, к которому оно подключено.

I. ОПИСАНИЕ АРХИТЕКТУРЫ

Общая схема архитектуры системы удаленной печати приведена на рисунке 3. При проектировании использовался микросервисный подход, позволяющий декомпозировать сложную систему на ряд относительно самостоятельных компонентов. Достоинством данного подхода является гибкость и возможность масштабирования отдельных компонентов при необходимости (например, масштабирование микросервиса печати при увеличении количества запросов).

Каждый элемент системы представляет собой отдельный Docker [1] контейнер (или контейнеры) для упрощения процессов развертывания и масштабирования.

Система печати состоит из следующих компонентов:

- Кабинет пользователя;
- Клиентское API (Customers API);

- Платежное API (Payments API);
- Сервис Печати (Print Service);
- Сервис синхронизации (Sync Service).

Кабинет пользователя представляет собой интерфейс, через который сотрудники компании могут отслеживать состояние запросов на печать, получать информацию о количестве документов, отправленных на печать, оплатить подписку за сервис печати, проверить состояние оплаты и т.п. Данный компонент использует клиентское и платежное API для работы.

Клиентское и платежное API представляют собой вспомогательные компоненты системы и написаны с использованием фреймворка Flask (Python). В качестве базы данных для хранения информации данные сервисы используют PostgreSQL.

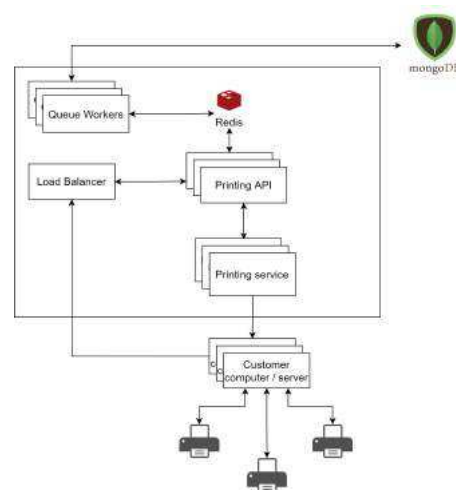


Рис. 1 – Микросервис печати

На рисунке 1 показана схема микросервиса печати, отвечающего за доставку документов на указанные принтеры. Общий принцип работы заключается в том, что на компьютеры (серверы) компании, подключенные к принтерам, устанавливается специальное клиентское приложение, которое определяет доступные принтеры и синхронизирует информацию с микросервисом печати. Приложение написано на C++ и использует

ет API операционных систем для работы с принтерами.

В систему документооборота или ERP-систему компании устанавливается интеграция, которая позволяет отправлять документы на печать без скачивания файла, а отправляя их напрямую через микросервис печати, указывая при этом принтер, на котором необходимо распечатать документ.

Микросервис печати отвечает за прием запросов на печать и отправку документов на корректные клиентские устройства, подключенные к нужным принтерам. Информация о статусе запроса и другая техническая информация хранится в базе данных MongoDB [3], что позволяет при необходимости вести статистику и оповещать техподдержку при появлении проблем.

Как было сказано выше, данный компонент системы может быть масштабирован при увеличении нагрузки.

Еще одним важным микросервисом является сервис синхронизации, позволяющий синхронизировать данные в базах данных (рисунок 2). На рисунке 3 видно, что система имеет три основных хранилища данных - две базы PostgreSQL и одну базу MongoDB. Сервис синхронизации отвечает за обновление статистики использования для каждого пользователя сервиса и другие технические задачи.

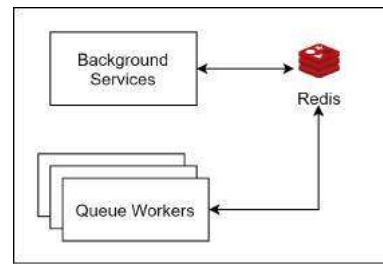


Рис. 2 – Микросервис синхронизации данных

II. ЗАКЛЮЧЕНИЕ

В рамках данного проекта была спроектирована и разработана эффективная система удаленной печати. За время эксплуатации разработанная система показала свою отказоустойчивость за счет использования микросервисного подхода (на данный момент нагрузка составляет 3-5 тысяч документов в день).

1. Empowering App Development for Developers | Docker [Electronic resource] / Mode of access: <https://www.docker.com/>. – Date of access: 22.10.2021.
2. PostgreSQL: The world's most advanced open source database [Electronic resource] / Mode of access: <https://www.postgresql.org/>. – Date of access: 23.10.2021.
3. MongoDB: The most popular database for modern apps [Electronic resource] / Mode of access: <https://www.mongodb.com/>. – Date of access: 23.10.2021.

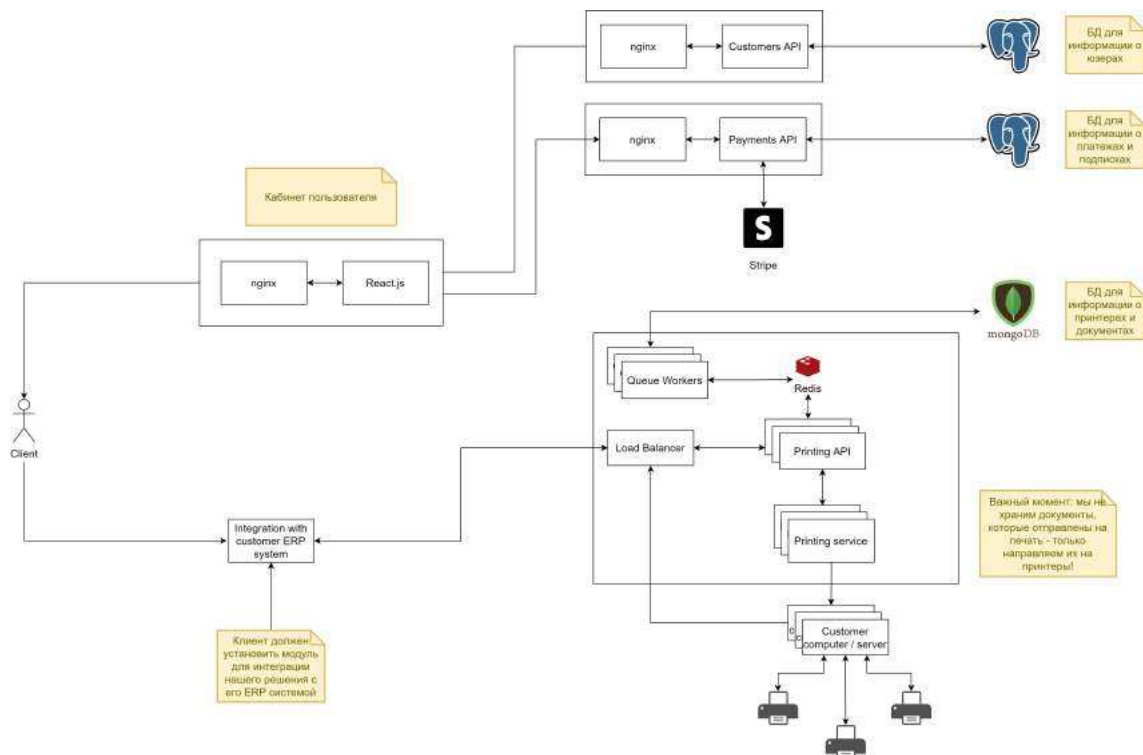


Рис. 3 – Архитектура разрабатываемой системы печати