

# СОВРЕМЕННАЯ ВЫСОКОНАГРУЖЕННАЯ И МАСШТАБИРУЕМАЯ СИСТЕМА

Ляхор Т. В., Проволоцкий В. Н., Трофимович А. Ф., Захарьев В. А.  
Кафедра систем управления. Белорусский государственный университет информатики и радиоэлектроники  
Минск, Республика Беларусь  
E-mail: linoge@bsuir.by

## ВВЕДЕНИЕ

В настоящее время возникает необходимость в разработке высоконагруженных программных систем, предназначенных для одновременной работы десятков тысяч пользователей и способных масштабироваться при необходимости. Одной из таких систем является программное обеспечение интернет-переписи населения Республики Беларусь (ПО И-перепись).

### I. ФУНКЦИОНАЛЬНЫЕ МОДУЛИ СИСТЕМЫ

ПО И-переписи состоит из функциональных модулей изображенных на рисунке 1.



Рис. 1 – Перечень функциональных модулей ПО И-переписи

- Функциональный модуль «Авторизация» предназначен для обеспечения процессов идентификации, аутентификации и авторизации пользователя в ПО И-переписи.;
- В ходе авторизации респондентов модуль подключает подмодуль «Взаимодействие с МСИ» с целью получения подтверждения того, что пользователь зарегистрирован в МСИ;
- Модуль «Сбор событий» используется в режиме администрирования и обеспечивает управление отображением загрузки серверов ПО И-переписи, предоставление журнала отправки данных по заполненным респондентами переписным листам в ПСПД, аудит событий безопасности и ошибок функционирования системы;
- Модуль «Взаимодействие с редакторами» используется в режиме администрирования и обеспечивает управление информационным контентом, сопровождающим процесс переписи населения;
- Модуль «Взаимодействие с респондентами» используется в пользовательском ре-

жиме и предоставляет графический инструмент для заполнения переписных листов, а также для просмотра информационного контента, сопровождающего проведение переписи населения в ПО И-переписи;

- Модуль «Администрирование» используется в режиме администрирования и обеспечивает выполнение настройки параметров системы;

### II. АРХИТЕКТУРА СИСТЕМЫ

Ранее для клиент-серверных систем стандартом является трёхзвенная архитектура. Она представляет собой слой представления (браузер), слой бизнес-логики (приложение на сервере) и слой хранения данных (база данных). Однако в современных системах из-за более высоких требований к гибкости пользовательского интерфейса и усложнению слоя представления, часть бизнес логики может выноситься на сторону клиента и работать в браузере. Данные приложения называются SPA (single page application) и реализовываются на языке javascript, который выполняется в браузере у конечного пользователя. Данный подход позволяет снизить нагрузку на сервер приложений, так как страница теперь полностью формируется на стороне клиента. Также такой подход ведёт к уменьшению передаваемого трафика между клиентом и сервером и позволяет инженеру отвечающему за бизнес-логику сосредоточиться только на ней.

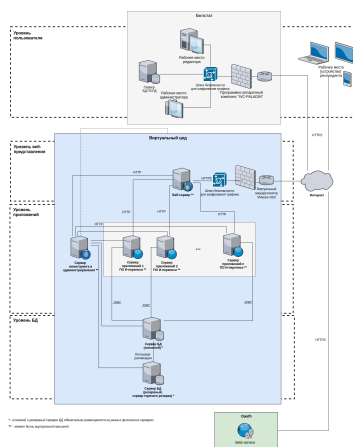


Рис. 2 – Схема комплекса технических средств

Кроме того, для работы приложения необходимы сторонние интеграции, такие как меж-

банковская система идентификации (МСИ) и подсистема сбора и передачи данных (ПСПД). МСИ необходима для авторизации и аутентификации человека, по личному идентификационному номеру и номеру мобильного телефона она позволяет аутентифицировать человека и получить данные о нём. ПСПД является системой предназначенной для осуществления переписи людей с помощью планшетов.

Основная задача ПО И-переписи это заполнения переписного листа с последующим его преобразованием к формату ПСПД и отправкой в данную систему. Для реализации клиентской части был выбран фреймворк Angular. Полученный в результате компиляции и сборки файл кладётся на web-server Nginx, который при обращении к нему пользователя отдаёт данный файл. После скачивания полученных файлов, браузер пользователя отрисовывает пользовательский интерфейс и отправляет http запросы за получением необходимых данных на web сервер.

Запросы для получения данных Nginx перенаправляет на один из шести серверов приложений, являясь в свою очередь балансировщиком нагрузки. Приложения на сервере написаны на языке Java с использованием фреймворка Spring Boot. Работа приложений заключается в сохранении и загрузке анкет, кэшировании и поиске элементов справочников, а также интеграций с МСИ и ПСПД.

В качестве слоя хранения данных используется база данных Postgres. Для надёжности хранения данных, одновременно работает 2 базы

в режиме master-slave с включенной асинхронной репликацией. В случае отказа основной базы, приложение будет работать со slave.

Для хранения пользовательских сессий и их передачи между серверами используется Hazelcast (рисунок 3).

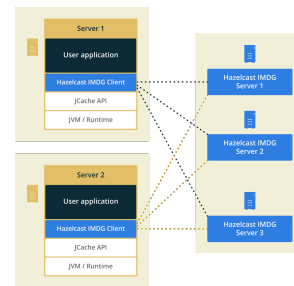


Рис. 3 – Структура работы с Hazelcast сервером

Hazelcast - это распределённое хранилище в памяти, которое позволяет быстро обращаться к элементам по ключу. Он используется для масштабирования и увеличения скорости доступа к оперативным данным, когда скорости работы с реляционными хранилищами уже не достаточно.

#### СПИСОК ЛИТЕРАТУРЫ

1. Мартин Клеппман. Высоконагруженные приложения. Программирование, масштабирование, поддержка. СПб., Питер, 2018. - 640 с.
2. Брэндан Бёрнс. Распределенные системы. Паттерны проектирования. СПб., Питер, 2019. - 224 с.
3. Ли Атчинсон. Масштабирование приложений. Выращивание сложных систем. СПб., Питер, 2018. - 256 с.