Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»

Разработка системы извлечения терминов

Студент: Сапожков Андрей Максимович ИУ7-63Б

Научный руководитель: Строганов Юрий Владимирович

Цель и задачи

Цель – разработка системы извлечения многокомпонентных терминов и их переводных эквивалентов из параллельных научно-технических текстов.

Задачи:

1. Проанализировать предметную область и формализовать задачу.
2. Спроектировать базу данных и структуру программного обеспечения.
3. Реализовать интерфейс для доступа к базе данных.
4. Реализовать ПО, которое позволит пользователю создавать, получать и изменять сведения из разработанной базы данных.
5. Исследовать зависимость времени выполнения запросов от использования кеширования данных текущей сессии пользователя.

Предметная область

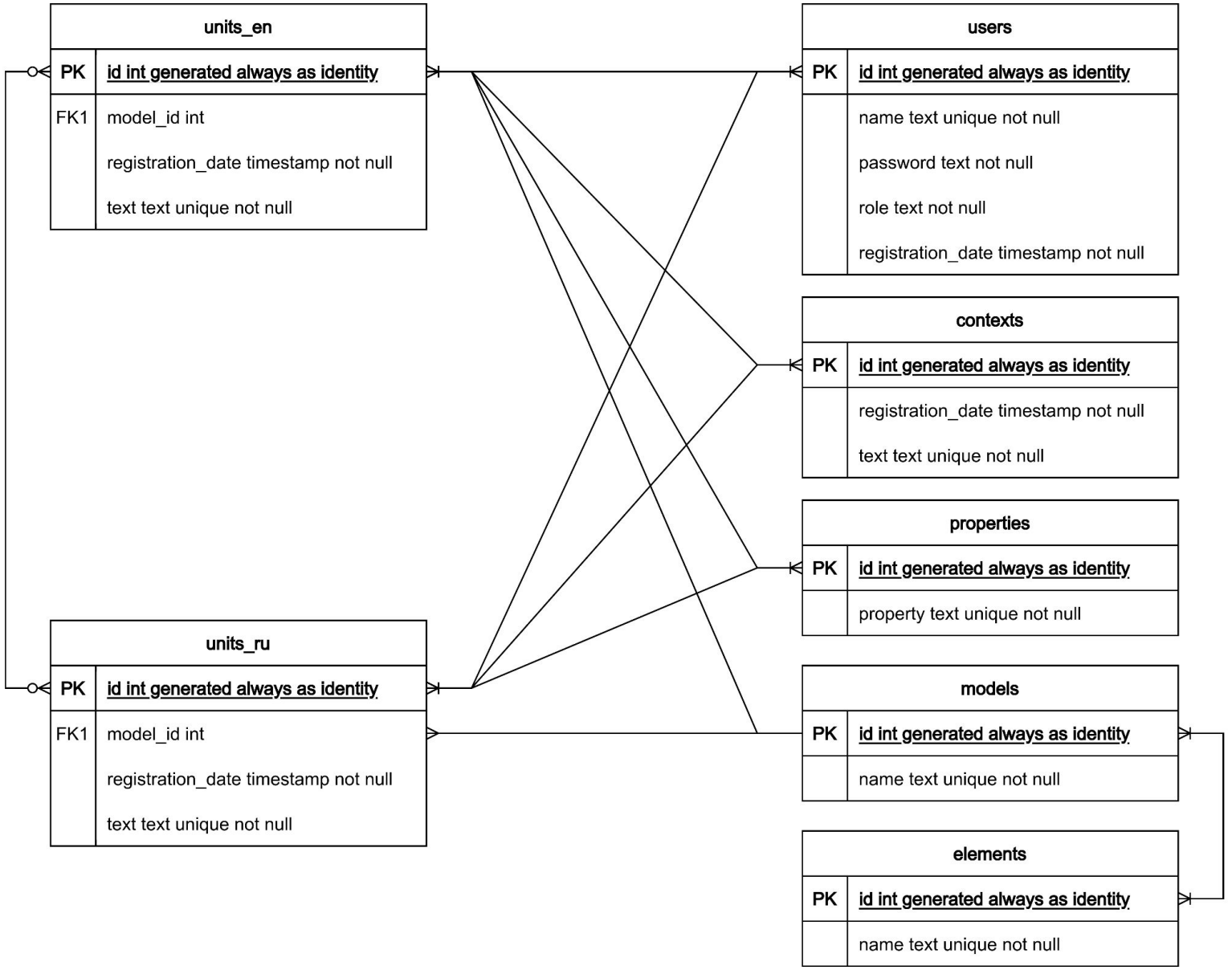
Переводчики (Google, Яндекс, DeepL) позволяют размечать тексты, но не дают работать с терминологией. Также нет возможности редактировать "разметку".

Словари (Thesaurus) позволяют в какой-то степени изучать терминологию, но не дают возможности дополнять её на основе размеченных текстов.

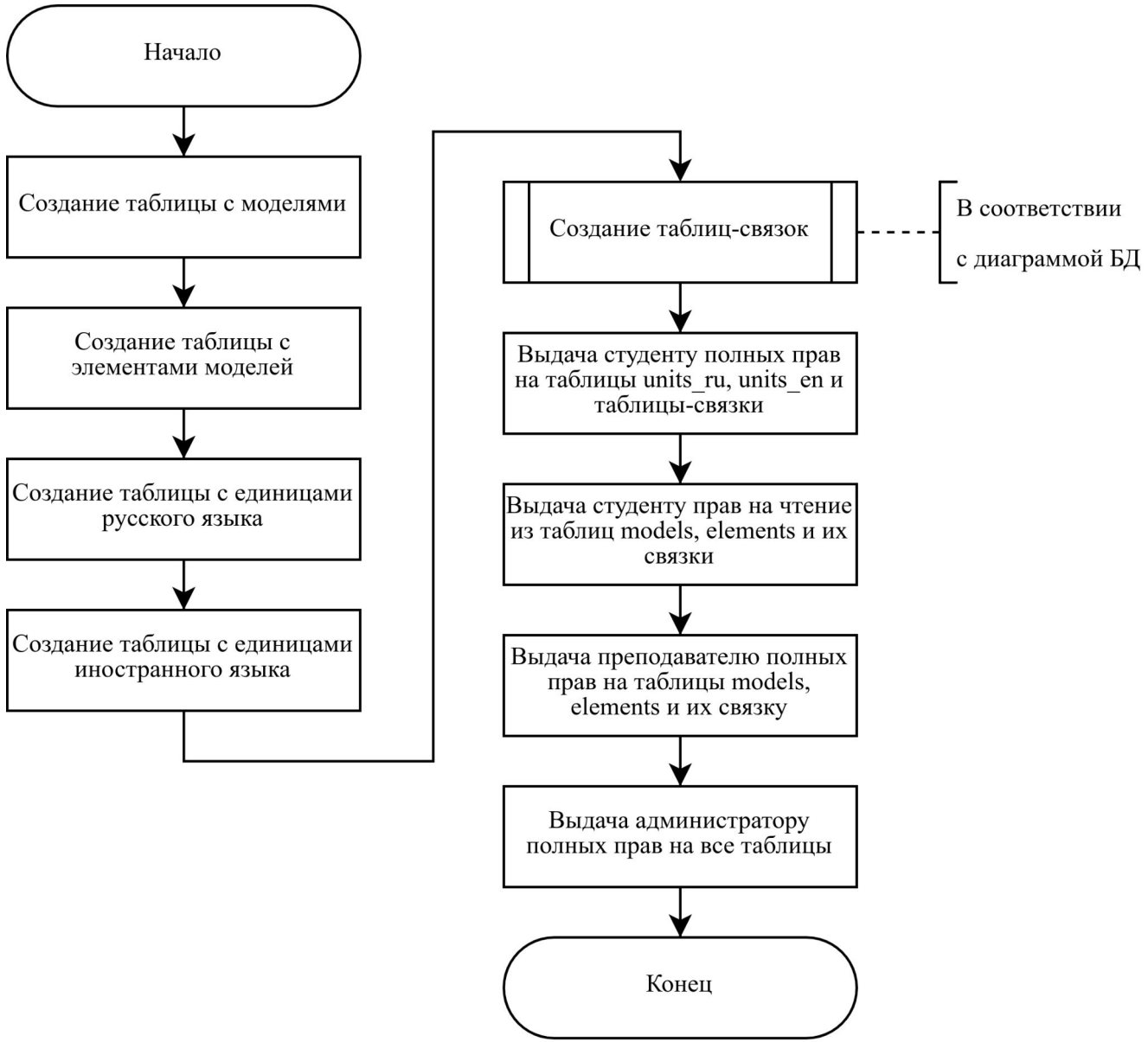



ER-модель базы данных

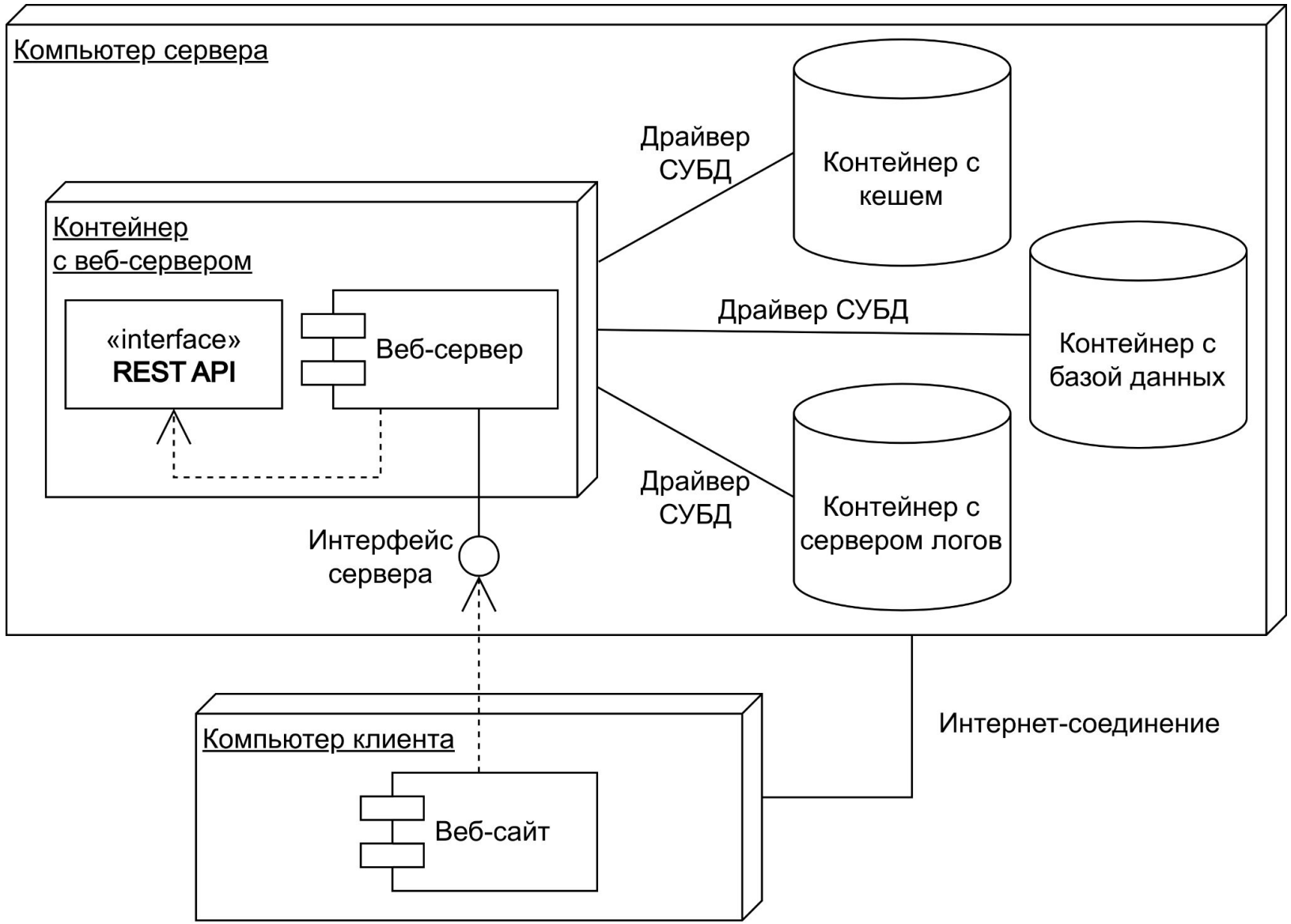
ER-диаграмма базы данных



Хранимая процедура БД



Архитектура ПО



Анализ СУБД

СУБД	**Характеристики**	**Назначение**
PostgreSQL	● Объектно-реляционная модель ● Открытый исходный код ● Сертификация ФСТЭК России	Долговременное хранение данных
Redis	● Хранилище типа ключ-значение ● Хранение данных в оперативной памяти	Кеширование данных
InfluxDB	● Ориентированность на хранение и обработку временных рядов ● Оптимизация записи данных ● Встроенные графические средства визуализации данных	Хранение логов

Логирование InfluxDB

В качестве СУБД для хранения логов была выбрана СУБД-ВР InfluxDB.

В InfluxDB данные представляются в виде двумерной таблицы (measurement), столбцы которой соответствуют меткам времени (timestamp).

В InfluxDB сохраняются

- запросы пользователей;
- события в бизнес-логике;
- ошибки в базе данных;
- конфигурация сервера.

Кеширование Redis

Для хранения данных пользовательских сессий использовалась нереляционная СУБД **Redis** (хранилище типа **ключ-значение** в оперативной памяти сервера).

Формат ключа:

репозиторий:**отношение**:**слой**:**язык**

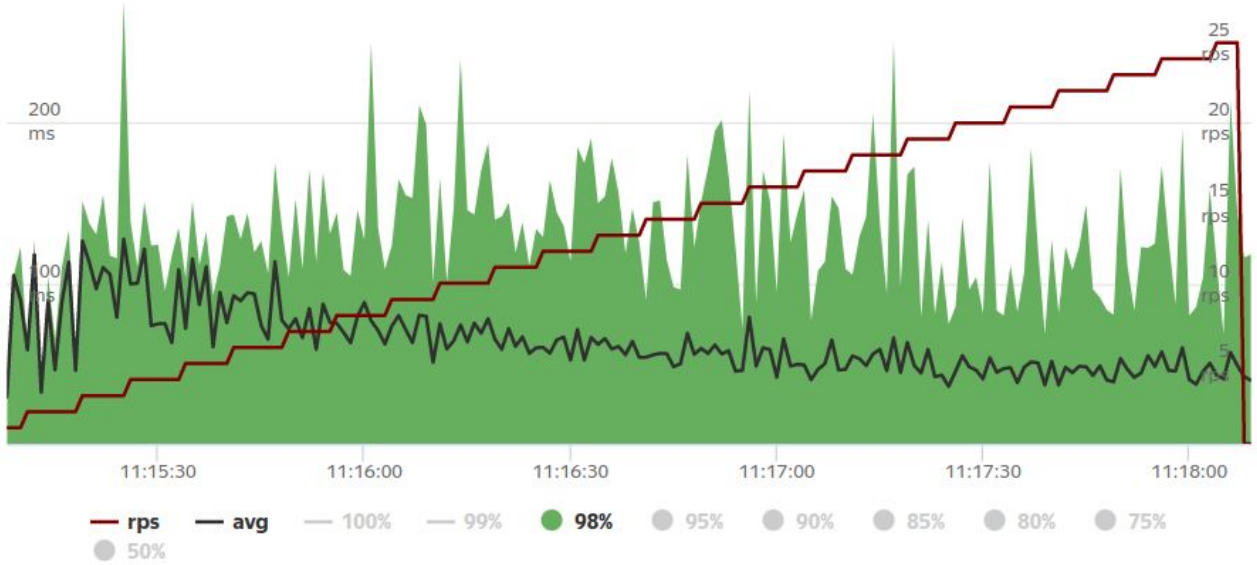
Нагрузочное тестирование

Цель – проведение нагрузочного тестирования и сравнение производительности веб-сервера при обработке запросов на сохранение терминов с использованием кеширования и без него.

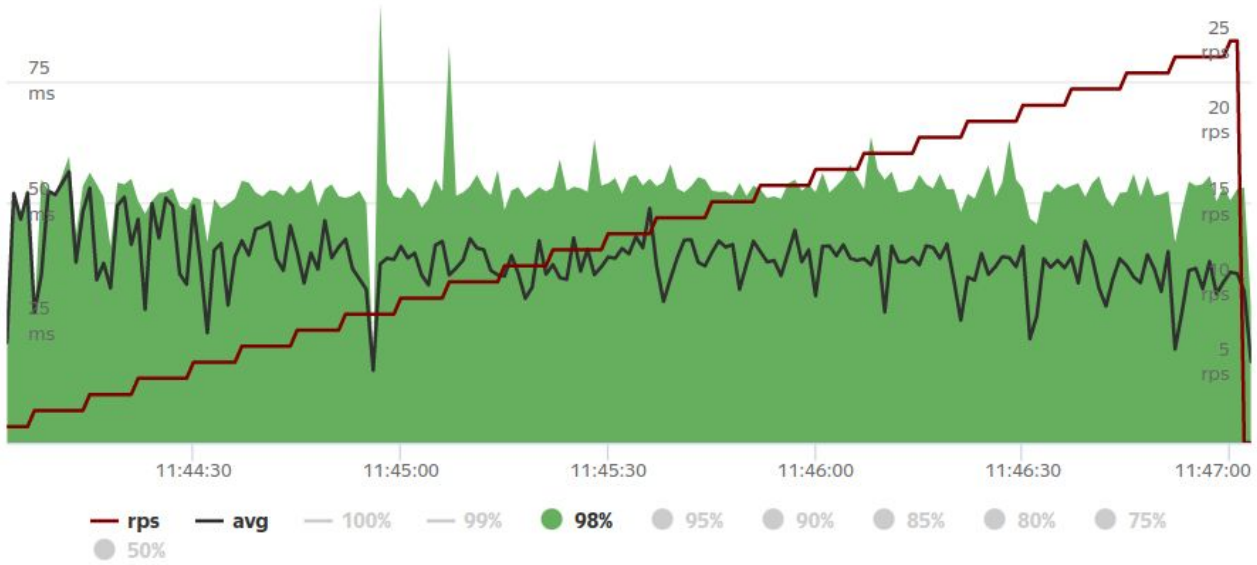
Технические характеристики:

1. Операционная система: Manjaro Linux x86-64, версия ядра 5.15.32.
2. Объём оперативной памяти: 16 Гб.
3. Процессор: Intel i5-9300H 2.4 ГГц.

Открытая линейная нагрузка

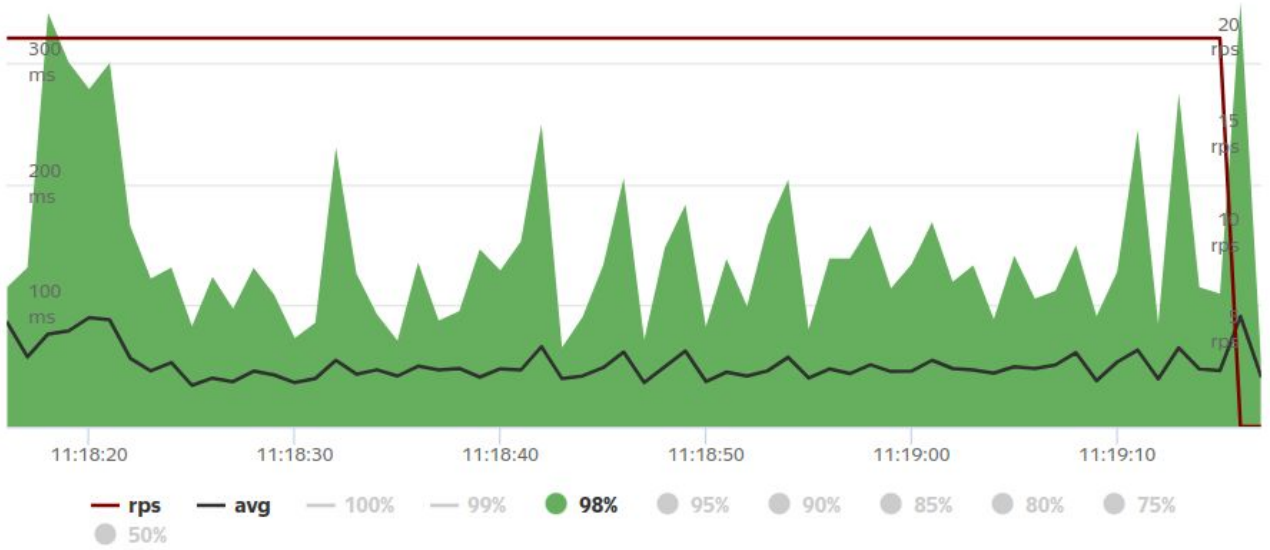


Без использования кеширования Redis

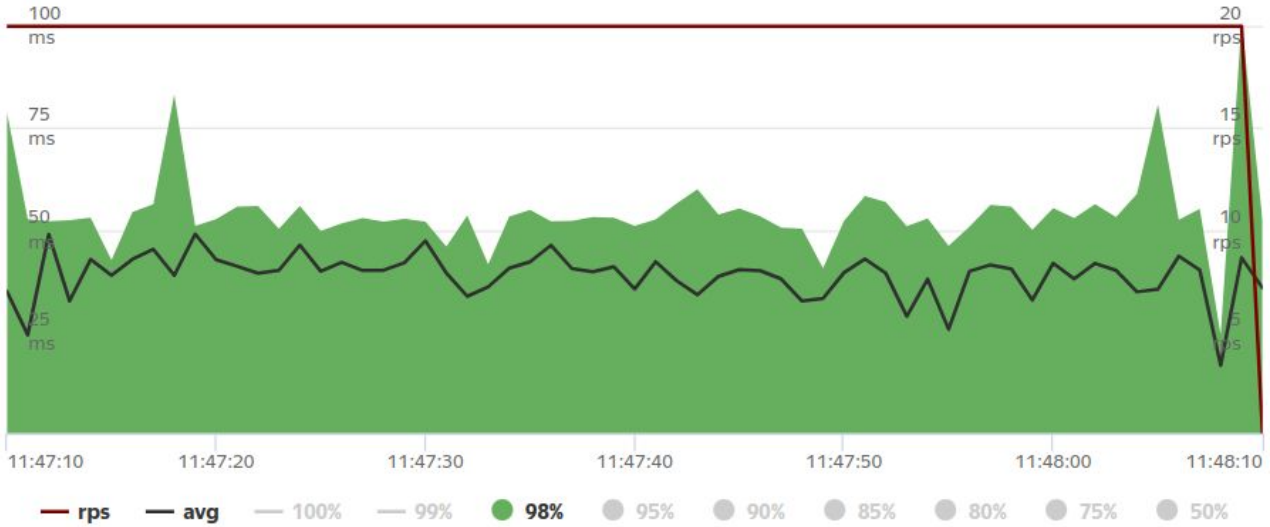


С использованием кеширования Redis

Открытая постоянная нагрузка

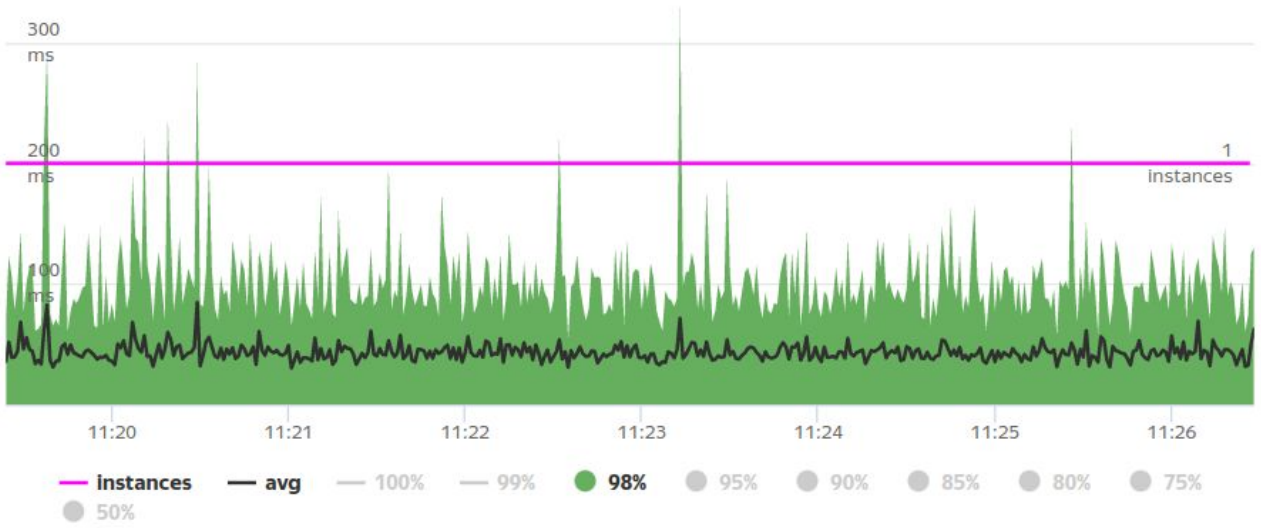


Без использования кеширования Redis

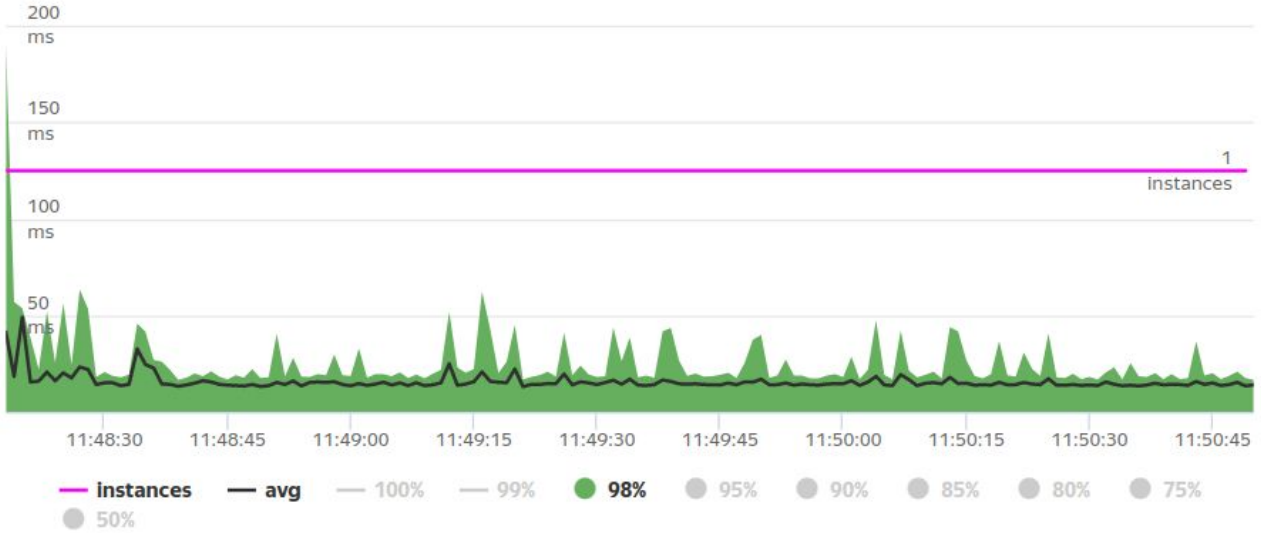


С использованием кеширования Redis

Закрытая нагрузка

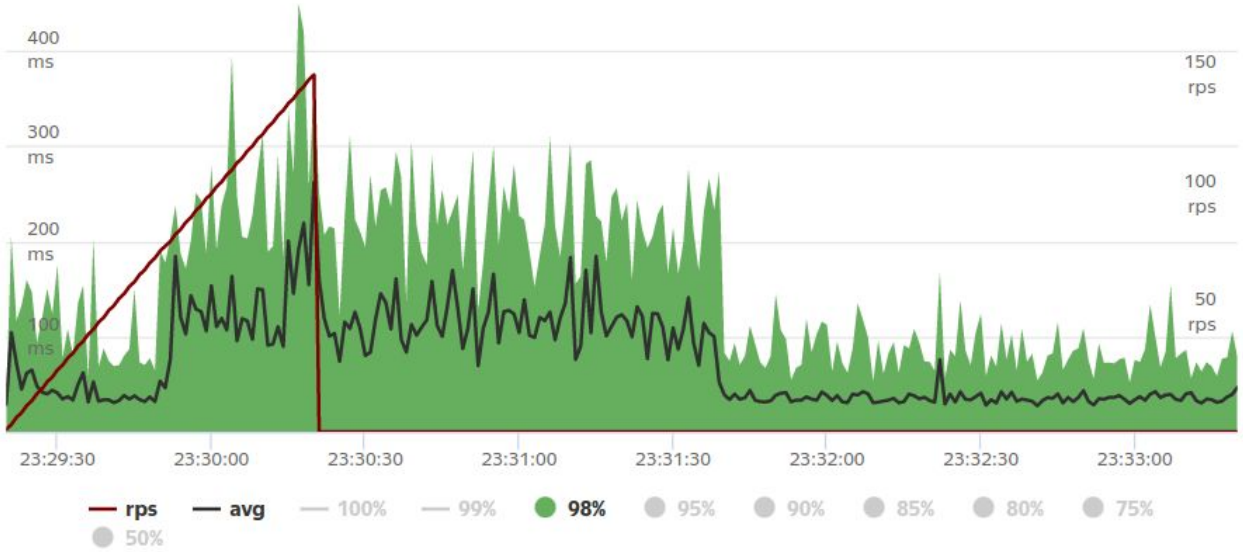


Без использования кеширования Redis

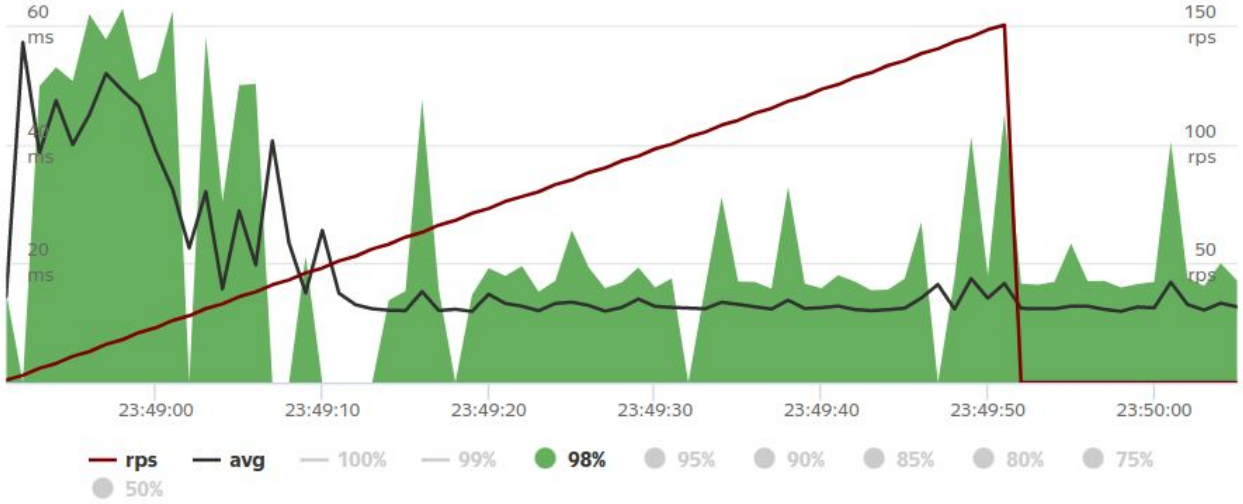


С использованием кеширования Redis

Предельная нагрузка системы



Без использования кеширования Redis максимум **33 RPS**



С использованием кеширования Redis максимум **83 RPS**

Результаты исследования

С использованием кеширования

1) при линейной открытой нагрузке среднее время ответа сервера уменьшилось с **63 мс** до **37 мс** (на **41%**);
2) при линейной постоянной нагрузке среднее время ответа сервера уменьшилось с **49 мс** до **38 мс** (на **22%**);
3) при закрытой нагрузке среднее время ответа сервера уменьшилось с **44 мс** до **15 мс** (на **66%**);
4) уменьшился разброс времён ответов, то есть сервер стал отвечать на запросы стабильнее.

Заключение

В рамках курсовой работы была разработана система извлечения многокомпонентных терминов и их переводных эквивалентов из параллельных научно-технических текстов. Для достижения этой цели были решены следующие задачи:

1. Проведён анализ предметной области и формализована задача.
2. Спроектирована база данных и структура ПО.
3. Реализован интерфейс для доступа к базе данных.
4. Реализовано ПО, которое позволяет пользователю создавать, получать и изменять сведения из разработанной базы данных.
5. Проведено исследование зависимости времени выполнения запросов от использования кеширования данных текущей сессии пользователя.