|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ***

***НА ТЕМУ:***

**Система прогнозирования нагрузки СХД с интеллектуальной подсистемой настройки**

Студент \_\_ИУ-5 82Б\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** \_\_**А.М.Ким\_\_\_\_\_\_\_\_\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель ВКР **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_А.А.Максаков\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

Консультант **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_М.В.Черненький\_\_**

(Подпись, дата) (И.О.Фамилия)

Нормоконтролер **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_Ю.Н.Кротов \_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

2024 г.

АННОТАЦИЯ

Расчётно-пояснительная записка квалификационной работы бакалавра содержит 78 страниц. С приложениями объем составляет 69 страниц. Работа включает в себя 22 таблицы и 13 иллюстраций. В процессе выполнения было использовано 12 источников.

Объектом разработки является система прогнозирования нагрузки СХД с интеллектуальной подсистемой настройки. Для разработки данной системы прогнозирования необходимо решить следующие задачи:

* Изучена предметную область, связанную с системами хранения данных и методами прогнозирования их нагрузки. Провести анализ существующих решений и технологий, включая использование алгоритмов машинного обучения для прогнозирования и оптимизации нагрузки.
* Сформулированы требования к разрабатываемой системе, включающие необходимость высокой точности прогнозирования, производительности, интеграции с существующими системами, отказоустойчивости и удобства использования.
* Спроектировать архитектуру системы, включающую фронтенд на React, бэкенд на Python/Django и базу данных на PostgreSQL. Определить ключевые компоненты системы, такие как модули сбора данных, модуль прогнозирования и интеллектуальная подсистема настройки.
* Разработать система прогнозирования нагрузки СХД, включающую алгоритмы машинного обучения для анализа данных и прогнозирования, а также интерфейс для настройки параметров и визуализации результатов. Провести тестирования, которые смогут подтвердить высокую точность и надежность системы.

Цель работы заключается выполнении всех поставленных задач, которые помогут разработать и спроектировать систему прогнозирования нагрузки СХД с интеллектуальной подсистемой настройки.

Пояснительная записка содержит 3 приложения.

СОДЕРЖАНИЕ

[АННОТАЦИЯ 2](#_Toc168320824)

[СОДЕРЖАНИЕ 3](#_Toc168320825)

[СПИСОК ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ 5](#_Toc168320826)

[ВВЕДЕНИЕ 6](#_Toc168320827)

[1 ПОСТАНОВКА ЗАДАЧ РАЗРАБОТКИ 7](#_Toc168320828)

[1.1 Общетехническое обоснование разработки 7](#_Toc168320829)

[1.1.1 Постановка задачи проектирования 7](#_Toc168320830)

[1.1.2 Описание предметной области 8](#_Toc168320831)

[1.1.3 Выбор критериев качества 9](#_Toc168320832)

[1.1.4 Анализ прототипов и аналогов 16](#_Toc168320833)

[1.1.4.1 HPE System Reporter 17](#_Toc168320834)

[1.1.4.2 SolarWinds Storage Resource Monitor 18](#_Toc168320835)

[1.1.4.3 Сравнение систем 20](#_Toc168320836)

[1.1.5 Требования предъявляемые к разрабатываемой системе 28](#_Toc168320837)

[1.1.5.1 Функциональные требования 28](#_Toc168320838)

[1.1.5.2 Требования к надежности 28](#_Toc168320839)

[1.1.6 Исследование моделей прогнозирования 29](#_Toc168320840)

[1.1.6.1 Постановка задачи 29](#_Toc168320841)

[1.1.6.2 Random Forest 29](#_Toc168320842)

[1.1.6.3 Support vector machine (SVM) 30](#_Toc168320843)

[1.1.6.4 Линейная регрессия 31](#_Toc168320844)

[1.1.6.5 Метрики оценки 32](#_Toc168320845)

[1.1.6.5.1 MAE 32](#_Toc168320846)

[1.1.6.5.2 MSE 33](#_Toc168320847)

[1.1.6.5.3 Коэффициент детерминации 33](#_Toc168320848)

[1.1.6.6 Исследование 34](#_Toc168320849)

[1.1.6.7 Сравнение результатов 36](#_Toc168320850)

[2 КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ 38](#_Toc168320851)

[2.1 Технологическая часть 38](#_Toc168320852)

[2.1.1 Технологии фронтенда 38](#_Toc168320853)

[2.1.1.1 Язык программирования JavaScript 38](#_Toc168320854)

[2.1.1.2 Фреймворк React 38](#_Toc168320855)

[2.1.2 Технологии бэкенда 40](#_Toc168320856)

[2.1.2.1 Язык программирования Python 40](#_Toc168320857)

[2.1.2.2 Фреймворк Django 41](#_Toc168320858)

[2.1.3 СУБД 42](#_Toc168320859)

[2.1.4 Модель прогнозирования 46](#_Toc168320860)

[2.2 Конструкторская часть 47](#_Toc168320861)

[2.2.1 Архитектура системы 47](#_Toc168320862)

[2.2.2 Пользовательский интерфейс 49](#_Toc168320863)

[ЗАКЛЮЧЕНИЕ 54](#_Toc168320864)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 55](#_Toc168320865)

[ПРИЛОЖЕНИЕ А ГРАФИЧЕСКАЯ ЧАСТЬ 57](#_Toc168320866)

[ПРИЛОЖЕНИЕ Б ТЕХНИЧЕСКОЕ ЗАДАНИЕ 68](#_Toc168320867)

[ПРИЛОЖЕНИЕ В ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ 73](#_Toc168320868)

# СПИСОК ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

1. **СХД** – система хранения данных;
2. **ПО** – програмное обеспечение;
3. **Python** – высокоуровневый язык программирования;
4. **JS** – JavaScript высокоуровневый язык программирования;
5. **DOM** – Document Object Model;
6. **AJAX** – Asynchronous JavaScript and XML;
7. **React** – фреймворк для веб-разработки;
8. **СУБД** – система управления базами данных;
9. **БД** – база данных;
10. **PostgreSQL** – база данных;
11. **ORM** – object-relational mapping;
12. **HTTP** – hypertext transfer protocol;
13. **TCP** – transmission control protocol;
14. **REST API** – representational state transfer application programming interface;
15. **SVM** – support vector machine;
16. **Фронтенд** – клиентский модуль;
17. **Бэкенд** – серверный модуль;
18. **Фреймворк** – программная платформа;
19. **Чекбокс** – метка включить/выключить.

ВВЕДЕНИЕ

Актуальность данной работы обусловлена стремительным ростом объемов данных в современном мире. С каждым годом количество генерируемой информации увеличивается, что связано с рядом факторов. Во-первых, увеличение числа интернет-пользователей в мире способствует генерации большего объема данных. Во-вторых, рост популярности контента высокого качества, требует больших мощностей для их хранения и обработки.

Системы хранения данных (СХД) играют ключевую роль в управлении этими объемами информации. Они не только обеспечивают хранение данных, но и гарантируют их доступность и безопасность, что является критически важным для бизнес-процессов современных компаний. Перегрузка таких систем может привести к серьезным сбоям, вплоть до полной потери данных, что недопустимо в бизнес-среде.

В связи с этим возникает острая необходимость в прогнозировании нагрузок на системы хранения данных. Эффективное прогнозирование позволяет не только оптимизировать использование ресурсов, но и предотвратить длительные простои, минимизируя тем самым потери данных и финансовые издержки. Для решения этой задачи активно используются современные методы искусственного интеллекта и машинного обучения, которые позволяют анализировать большие объемы информации и делать точные прогнозы в реальном времени.

1 ПОСТАНОВКА ЗАДАЧ РАЗРАБОТКИ

1.1 Общетехническое обоснование разработки

1.1.1 Постановка задачи проектирования

Для организации современного взаимодействия между компонентами системы прогнозирования нагрузки СХД с интеллектуальной подсистемой настройки необходимо разработать программное обеспечение.

Система должна предоставлять возможность пользователям выбирать параметры СХД, таблицы из базы данных, изменять уровни прогнозирования, выбирать режимы скользящего окна, указывать интервалы и их количество, а также выбирать уровень предсказания. Также предусмотрена возможность нахождения глобального минимума и построения облака точек.

Важной частью системы является возможность изменения уровня прогнозирования через интерфейс приложения. Пользователи смогут задавать параметры и SQL-запросы для изменения уровня прогнозирования, что обеспечит гибкость и адаптивность системы. Дополнительно, система будет поддерживать режим скользящего окна, который можно будет настроить как автоматически, так и вручную, указывая необходимые интервалы.

Помимо этого, пользователи должны иметь возможность визуализировать результаты прогнозирования на графике. Это включает отображение исторических данных, прогнозируемых уровней нагрузки и облака точек, если оно выбрано. Визуализация данных позволит пользователям быстро оценивать текущую ситуацию и принимать обоснованные решения на основе полученных прогнозов.

Таким образом, задача проектирования может быть сформулирована следующим образом:

* разработать интерфейс для указания параметров СХД и выбора таблиц из базы данных;
* реализовать возможность изменения уровня прогнозирования;
* обеспечить выбор режима скользящего окна: автоматический и ручной;
* включить опции для определения уровня прогнозирования;
* визуализация графиков, отображающие исторические данные, прогнозируемые уровни, облако точек и прогноз;
* реализовать возможность развертывания системы на любой платформе.

1.1.2 Описание предметной области

Система прогнозирования нагрузки СХД с интеллектуальной подсистемой настройки представляет собой высокотехнологичное решение, предназначенное для анализа и предсказания нагрузки на СХД. В условиях стремительного роста объемов данных, обусловленного развитием информационных технологий и цифровой трансформацией, вопросы эффективного управления и прогнозирования нагрузки на СХД приобретают особую значимость. Современные СХД являются критически важными компонентами ИТ-инфраструктуры организаций, обеспечивая хранение, обработку и доступ к огромным массивам данных [1].

Основная цель системы прогнозирования нагрузки СХД заключается в предотвращении перегрузок и сбоев, которые могут привести к потере данных, снижению производительности и нарушению бизнес-процессов. Для достижения этой цели разрабатываемая система должна обеспечивать точное прогнозирование нагрузки на основе анализа исторических данных и текущих параметров работы СХД. Прогнозирование позволяет заранее выявлять потенциальные проблемы и принимать превентивные меры для их предотвращения, что способствует повышению надежности и эффективности работы СХД.

Система будет состоять из нескольких ключевых компонентов, включая фронтенд, бэкенд и базу данных. Фронтенд представляет собой пользовательский интерфейс, через который пользователи будут взаимодействовать с системой, настраивать параметры прогнозирования и просматривать результаты анализа. Бэкенд отвечает за обработку данных, выполнение алгоритмов прогнозирования и взаимодействие с базой данных. База данных будет хранить исторические данные о нагрузке на СХД.

На рисунке 1 показано взаимодействие оператора (пользователя) и СХД с системой.

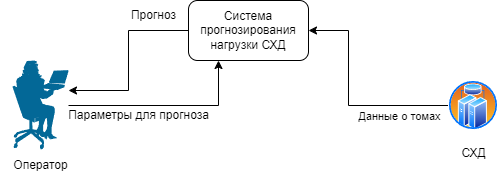


Рисунок 1 − Предметная область

Для корректной работы системы прогнозирования нагрузки СХД необходимо прописать следующие ограничения предметной области:

* Каждый оператор вводит несколько параметров для прогноза;
* Система предоставляет оператору несколько прогнозов в зависимости от значений параметров;
* Каждый оператор может получить несколько прогнозов;
* СХД предоставляет несколько данных о томах.

**1.1.3 Выбор критериев качества**

Для разрабатываемой системы прогнозирования нагрузки СХД приоритетными являются следующие критерии качества:

* Точность прогнозирования. Система должна обеспечивать точные прогнозы нагрузки на СХД, чтобы позволить эффективное планирование емкости и ресурсов;
* Производительность. Помимо прогнозирования нагрузки, система должна помогать оптимизировать использование ресурсов хранения данных, чтобы избежать излишних расходов;
* Интеграция. Возможность интеграции с другими системами управления хранилищем данных для автоматизации процессов управления емкостью;
* Отказоустойчивость. Система должна предоставлять информацию о возможных рисках перегрузки или недостатка ресурсов, позволяя оперативно принимать меры по их устранению;
* Масштабируемость. Система должна быть способной масштабироваться для учета роста объема данных и изменений в рабочих нагрузках;
* Быстродействие. Важно, чтобы система обеспечивала быстрое формирование прогнозов нагрузки на основе актуальных данных;
* Отчётность. Система должна предоставлять отчеты о точности прогнозов и мониторинге нагрузки на СХД для обеспечения прозрачности процессов управления;
* Удобство интерфейса. Интерфейс системы прогнозирования должен быть интуитивно понятным для администраторов СХД [2].

Назначим весовые коэффициенты обозначенным критериям. Результаты приведены в таблице 1.

**Таблица 1 − Проранжированные критерии качества**

| **№ п/п** | **Название критерия** | **Весовой коэффициент** |
| --- | --- | --- |
| 1 | Точность прогнозирования | 8 α |
| 2 | Производительность | 7 α |
| 3 | Интеграция и совместимость | 6 α |
| 4 | Отказоустойчивость | 5 α |
| 5 | Масштабируемость | 3 α |
| 6 | Быстродействие | 3 α |
| 7 | Отчётность | 2 α |
| 8 | Удобство интерфейса | α |

Далее для каждого критерия качества распишем соответствие значения и описания показателей по пятибалльной шкале (таблицы 2−9).

**Таблица 2 − Вербально-числовая оценка для критерия «Точность прогнозирования»**

| **Точность прогнозирования** | |
| --- | --- |
| **Оценка** | **Описание** |
| 1 | Система прогнозирования практически не предоставляет точных прогнозов, что серьезно затрудняет управление нагрузкой |
| 2 | Прогнозы системы часто ошибочны, что затрудняет эффективное управление нагрузкой. Требуется значительное улучшение точности предсказаний |
| 3 | Система прогнозирования в целом точна, но иногда допускает существенные ошибки, которые могут затруднять управление нагрузкой |
| 4 | Система прогнозирования обычно точна, и ошибки в предсказаниях несущественны. Прогнозы хорошо соответствуют реальной нагрузке |
| 5 | Прогнозы системы практически всегда точны и позволяют эффективно управлять нагрузкой. Ошибки в предсказаниях практически отсутствуют |

**Таблица 3 − Вербально-числовая оценка для критерия «Производительность»**

| **Производительность** | |
| --- | --- |
| **Оценка** | **Описание** |
| 1 | Система прогнозирования нагрузки предоставляет крайне неточные прогнозы, что серьезно затрудняет планирование и управление ресурсами |
| 2 | Прогнозы системы часто неточны, что затрудняет планирование и управление ресурсами. Требуется значительное улучшение точности прогнозов |
| 3 | Система прогнозирования обеспечивает достаточно точные прогнозы, но иногда допускает ошибки, которые могут требовать корректировок в управлении ресурсами |
| 4 | Прогнозы системы обычно точны и достаточно надежны для планирования ресурсов. Возможны незначительные отклонения, но они не существенно влияют на работу системы |
| 5 | Система прогнозирования нагрузки предоставляет точные и своевременные прогнозы, которые позволяют эффективно планировать и управлять ресурсами. Ошибки в прогнозах практически отсутствуют |

**Таблица 4 − Вербально-числовая оценка для критерия «Интеграция и совместимость»**

| **Интеграция и совместимость** | |
| --- | --- |
| **Оценка** | **Описание** |
| 1 | Система прогнозирования нагрузки практически не интегрируется с другими системами, имеет серьезные проблемы совместимости, что серьезно затрудняет ее использование в комплексной информационной инфраструктуре |
| 2 | Интеграция системы прогнозирования нагрузки с другими системами затруднена, совместимость с платформами ограничена, требуются значительные усилия для решения конфликтов и проблем интеграции |
| 3 | Система прогнозирования нагрузки интегрируется с другими системами, но требует определенных усилий для обеспечения совместимости. Возможны некоторые ограничения или конфликты при взаимодействии |
| 4 | Интеграция системы прогнозирования нагрузки с другими системами обычно проходит гладко, совместимость с основными платформами хорошая, возможны небольшие проблемы, которые решаются без значительных усилий |
| 5 | Система прогнозирования нагрузки легко интегрируется с другими системами, обеспечивает высокую степень совместимости с различными платформами и не вызывает конфликтов при взаимодействии с другими компонентами информационной инфраструктуры |

**Таблица 5 − Вербально-числовая оценка для критерия «Отказоустойчивость»**

| **Отказоустойчивость** | |
| --- | --- |
| **Оценка** | **Описание** |
| 1 | Система практически не обладает отказоустойчивостью, частые отказы и сбои серьезно влияют на работоспособность системы, приводя к потере данных и длительным простоям |
| 2 | Система имеет серьезные проблемы с отказоустойчивостью, часто приводящие к простоям и потере данных, требует значительных улучшений для обеспечения надежной работы при отказах |
| 3 | Система обладает базовыми механизмами отказоустойчивости, способна частично восстанавливаться после отказов, но требует ручного вмешательства для восстановления работоспособности после серьезных сбоев |
| 4 | Система демонстрирует хорошую отказоустойчивость, способна эффективно обрабатывать отказы в работе отдельных компонентов, имеет механизмы резервирования и восстановления, что позволяет минимизировать влияние отказов на работоспособность системы |
| 5 | Система обладает высокой отказоустойчивостью, способна автоматически восстанавливаться после отказов, имеет резервированные ресурсы и механизмы аварийного восстановления, что позволяет предотвращать простои и существенно снижать риск потери данных |

**Таблица 6 − Вербально-числовая оценка для критерия «Масштабируемость»**

| **Масштабируемость** | |
| --- | --- |
| **Оценка** | **Описание** |
| 1 | Система практически не обладает масштабируемостью, не способна адекватно реагировать на изменения нагрузки, что приводит к серьезным простоям, сбоям и недоступности сервиса при увеличении нагрузки |
| 2 | Система имеет ограниченные возможности масштабирования, часто сталкивается с проблемами при увеличении нагрузки, требует значительных улучшений для эффективного расширения и обеспечения стабильной работы при высоких нагрузках |
|  | Система обладает базовыми механизмами масштабируемости, способна увеличивать свою производительность при необходимости, но может потребовать ручного вмешательства и перенастройки для оптимального функционирования при значительных изменениях нагрузки |
| 4 | Система обладает хорошей масштабируемостью, способна эффективно расширяться при увеличении нагрузки, хотя требует некоторого ручного вмешательства для добавления ресурсов. Масштабирование происходит плавно и без существенных простоев |
| 5 | Система обладает высокой масштабируемостью, способна легко и эффективно масштабироваться как вертикально (добавление ресурсов на одном уровне), так и горизонтально (добавление новых узлов или серверов). Система способна автоматически масштабироваться в зависимости от изменяющихся нагрузок без значительного вмешательства операторов |

**Таблица 7 − Вербально-числовая оценка для критерия «Быстродействие»**

| **Быстродействие** | |
| --- | --- |
| **Оценка** | **Описание** |
| 1 | Система неспособна обеспечить своевременное и точное прогнозирование нагрузки, что приводит к значительным задержкам и неточным результатам, делая ее практически бесполезной для оперативного управления ресурсами |
| 2 | Система демонстрирует заметные задержки в предсказании нагрузки, особенно при больших объемах данных или в условиях высокой изменчивости нагрузки |
| 3 | Система способна предсказывать нагрузку в разумные сроки, хотя время отклика может увеличиваться при росте объема данных или неожиданных пиковых нагрузках |
| 4 | Система обеспечивает быстрое прогнозирование нагрузки с хорошей точностью, демонстрируя стабильное время отклика на запросы при умеренных и переменчивых нагрузках |
| 5 | Система способна предсказывать нагрузку практически мгновенно, с высокой точностью и стабильностью даже при экстремальных изменениях нагрузки. Время отклика на запросы остается незначительным, независимо от объема данных |

**Таблица 8 − Вербально-числовая оценка для критерия «Отчётность»**

| **Отчётность** | |
| --- | --- |
| **Оценка** | **Описание** |
| 1 | Система не предоставляет отчётов о прогнозах нагрузок или предоставляет слишком неполные, неточные или непонятные отчёты, что делает невозможной оценку её работы |
| 2 | Система предоставляет ограниченные или неполные отчёты о прогнозах нагрузок, что затрудняет оценку её производительности и качества прогнозов |
| 3 | Система предоставляет базовые отчёты о прогнозах нагрузок, включающие основные метрики производительности. Отчёты создаются периодически, но могут быть несвоевременными или не полностью информативными |
| 4 | Система предоставляет информативные отчёты о прогнозах нагрузок, содержащие основные метрики производительности и качества прогнозов. Отчёты создаются регулярно, но могут быть несколько ограничены в деталях |
| 5 | Система предоставляет подробные и точные отчёты о прогнозах нагрузок, включая метрики точности, степень уверенности в прогнозах, а также их сравнение с фактическими данными. Отчёты генерируются автоматически, регулярно и вовремя |

**Таблица 9 − Вербально-числовая оценка для критерия «Удобство интерфейса»**

| **Удобство интерфейса** | |
| --- | --- |
| **Оценка** | **Описание** |
| 1 | Интерфейс системы крайне неудобен для использования, затрудняет выполнение задач и может вызывать путаницу у пользователей из-за неясной структуры, отсутствия необходимых функций или низкого качества дизайна |
| 2 | Интерфейс системы затрудняет выполнение задач по настройке и анализу прогнозов из-за неясной навигации, ограниченных возможностей или неудовлетворительного дизайна |
| 3 | Интерфейс системы предоставляет базовые возможности для настройки и анализа прогнозов, но требует дополнительного времени на изучение. Некоторые элементы управления могут быть неочевидными или скрытыми |
| 4 | Интерфейс системы в целом удобен и понятен, но может потребоваться небольшое время для освоения. Большинство функций доступны без лишних усилий, но некоторые аспекты могут быть менее интуитивными |
| 5 | Интерфейс системы интуитивно понятен и предоставляет полный спектр функций для настройки и анализа прогнозов. Все элементы управления понятны и легко доступны, что позволяет пользователям эффективно выполнять задачи |

1.1.4 Анализ прототипов и аналогов

Для управления объёмом данных на хранилищах и предотвращения их переполнения используются системы уведомлений о пороговых значениях заполнения. Они позволяют предупреждать администраторов о приближении к предельным значениям заполнения, что помогает предотвратить переполнение и потерю данных [3]. Это особенно важно для баз данных, файловых хранилищ и других систем, где объём данных может быстро нарастать. Сравнение различных систем уведомлений о пороговых значениях заполнения может включать в себя анализ их способностей к настройке пороговых значений, гибкости интеграции с другими системами мониторинга, а также способности обрабатывать большие объёмы данных. Однако ключевой особенностью разрабатываемой системы прогнозирования нагрузки СХД является то, что аналогов практически нет. Поэтому, для дальнейшего сравнения рассмотрим похожие системы.

#### 1.1.4.1 HPE System Reporter

HPE System Reporter является инструментом для мониторинга и отчётности, который используется в управлении системами хранения данных, предоставляемых Hewlett Packard Enterprise (HPE) [4]. Этот инструмент позволяет операторам систем получать подробные данные о производительности и использовании ресурсов в их хранилищах данных.

Основные преимущества HPE System Reporter включают:

1. Мониторинг производительности: System Reporter предоставляет данные о производительности системы хранения, что позволяет операторам оптимизировать работу системы и своевременно выявлять потенциальные проблемы.
2. Управление пороговыми значениями и оповещениями: Инструмент позволяет настроить пороговые значения для различных параметров системы хранения, таких как использование дискового пространства или производительность I/O операций. При достижении этих порогов система может автоматически отправлять оповещения, что помогает предотвратить перегрузку системы или её сбой.
3. Отчётность: System Reporter предоставляет обширные возможности для создания отчётов, которые могут быть настроены для отображения различной информации о состоянии системы хранения. Эти отчёты могут помочь в анализе тенденций использования и планировании будущих расширений системы хранения.

HPE System Reporter представляет собой мощный инструмент для операторов данных, который помогает обеспечивать высокую производительность и доступность систем хранения данных, а также помогает в принятии обоснованных решений на основе данных о текущем состоянии инфраструктуры хранения.

На рисунке 1 представлен интерфейс системы.

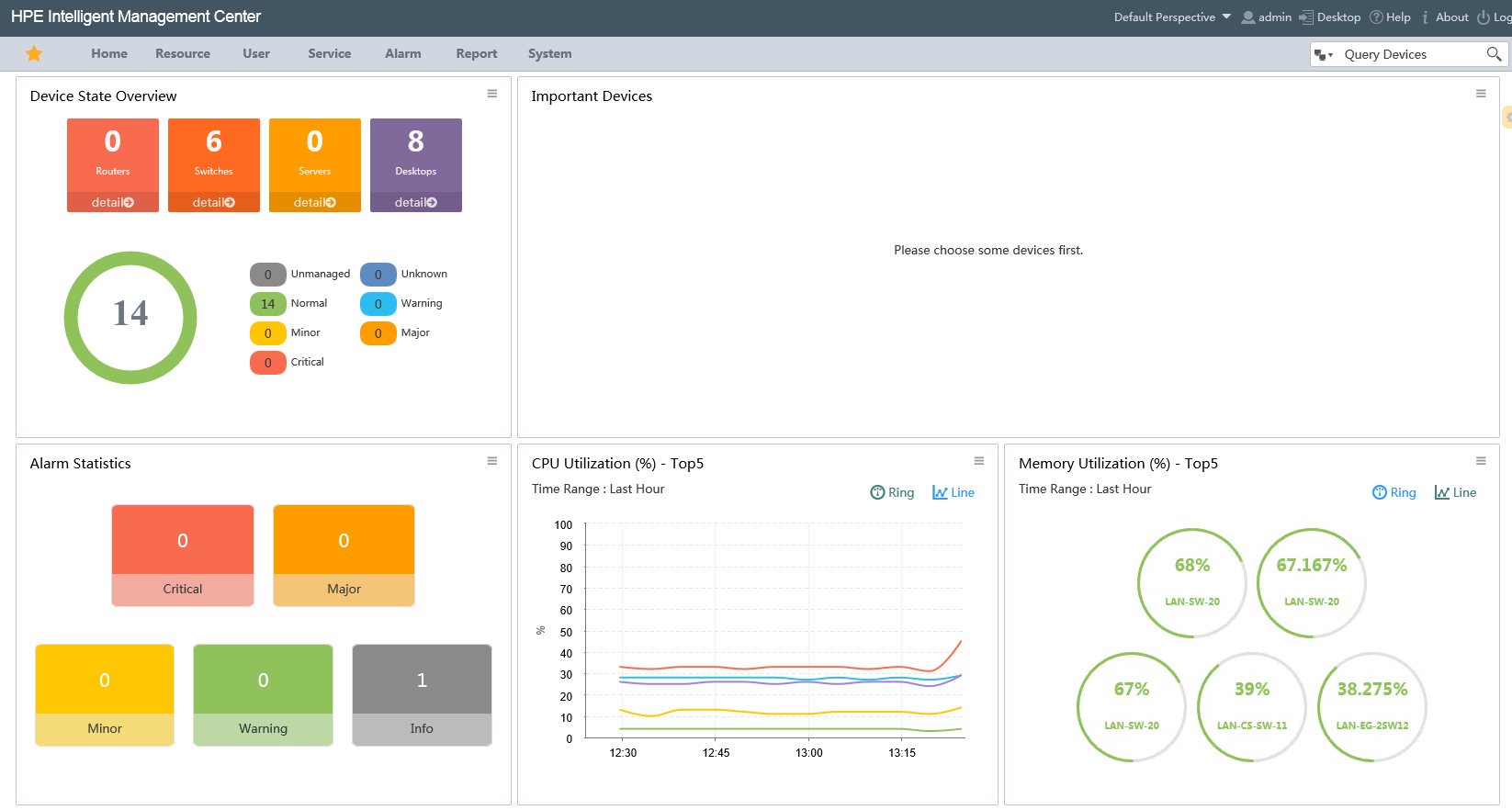


Рисунок 2 – Интерфейс пользователя

#### 1.1.4.2 SolarWinds Storage Resource Monitor

SolarWinds Storage Resource Monitor — это инструмент для мониторинга и управления системами хранения данных, разработанный компанией SolarWinds [5]. Он предназначен для системных администраторов и операторов, предоставляя подробные данные о производительности и использовании ресурсов в их хранилищах данных.

Основные преимущества SolarWinds Storage Resource Monitor включают:

1. Мониторинг производительности. Storage Resource Monitor

обеспечивает мониторинг в режиме реального времени для различных аспектов систем хранения данных, включая использование емкости, производительность ввода/вывода и доступность ресурсов. Это позволяет операторам выявлять узкие места и оптимизировать работу системы для повышения эффективности.

1. Управление пороговыми значениями и оповещениями. Инструмент

позволяет настраивать пороговые значения для ключевых параметров системы хранения. При достижении этих порогов система автоматически отправляет оповещения, что помогает операторам быстро реагировать на потенциальные проблемы и предотвращать сбои системы.

1. Отчётность и аналитика: Storage Resource Monitor предоставляет мощные возможности для создания и настройки отчётов. Эти отчёты позволяют анализировать текущие тенденции использования ресурсов и планировать будущие расширения системы хранения. Инструмент также включает визуальные панели и графики для лучшего понимания данных.
2. Прогнозирование использования. Система использует машинное обучение для прогнозирования будущих потребностей в емкости на основе исторических данных. Это помогает операторам заранее планировать обновления и предотвращать нехватку ресурсов.

SolarWinds Storage Resource Monitor является мощным инструментом для обеспечения высокой производительности и доступности систем хранения данных, а также для принятия обоснованных решений на основе детализированных данных о состоянии инфраструктуры [6].

На рисунке 3 представлен интерфейс SolarWinds Storage Resource Monitor.

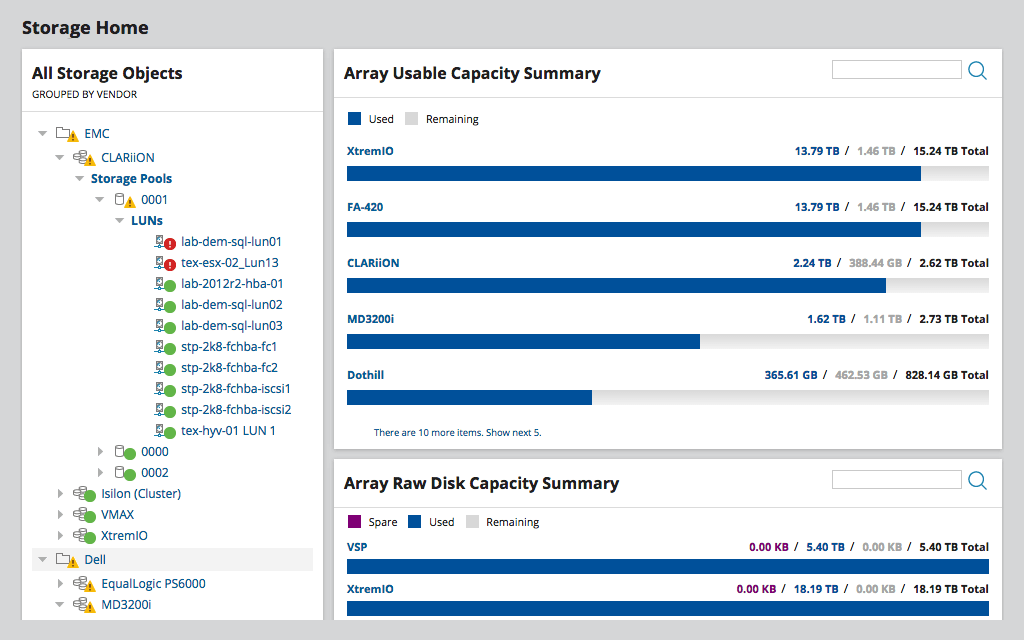


Рисунок 3 – Интерфейс SolarWinds Storage Resource

#### 1.1.4.3 Сравнение систем

На основе представленных критериев в таблице 1 проведем сравнение систем для обоснования надобности реализации системы прогнозирования нагрузки СХД с интеллектуальной подсистемой настройки. В таблице 10 представлены критерии с их обозначениями и единицами измерения.

**Таблица 10 − Критерии качества**

| **Критерий** | **Обозначение** | **Единица измерения** |
| --- | --- | --- |
| Точность прогнозирования | K1 | (количественный критерий) |
| Производительность | K2 | (качественный критерий) |
| Интеграция и совместимость | K3 | (качественный критерий) |
| Отказоустойчивость | K4 | (качественный критерий) |
| Масштабируемость | K5 | (качественный критерий) |
| Быстродействие | K6 | (качественный критерий) |
| Отчётность | K7 | (качественный критерий) |
| Удобство интерфейса | K8 | (количественный критерий) |

Оценим системы на основе критериев, представленных выше. В таблицах 11−13 показаны результаты оценки.

**Таблица 11 – Результат оценки HPE System Reporter**

| **Критерий** | **Оценка** |
| --- | --- |
| Точность прогнозирования | 9 |
| Производительность | Хорошая |
| Интеграция и совместимость | Отсутствует |
| Отказоустойчивость | Средняя |
| Масштабируемость | Хорошая |
| Быстродействие | Хорошая |
| Отчётность | Средняя |
| Удобство интерфейса | 8 |

**Таблица 12 – Результат оценки SolarWinds Storage Resource Monitor**

| **Критерий** | **Оценка** |
| --- | --- |
| Точность прогнозирования | 95 |
| Производительность | Средняя |
| Интеграция и совместимость | Средняя |
| Отказоустойчивость | Высокая |
| Масштабируемость | Средняя |
| Быстродействие | Хорошая |
| Отчётность | Хорошая |
| Удобство интерфейса | 4 |

**Таблица 13 – Результат оценки системы прогнозирования нагрузки СХД с интеллектуальной подсистемой настройки**

| **Критерий** | **Оценка** |
| --- | --- |
| Точность прогнозирования | 93 |
| Производительность | Высокая |
| Интеграция и совместимость | Полная |
| Отказоустойчивость | Высокая |
| Масштабируемость | Средняя |
| Быстродействие | Высокая |
| Отчётность | Полная |
| Удобство интерфейса | 7 |

После оценки систем обозначим варианты в таблице 14.

**Таблица 14 – Варианты систем**

| **Критерий** | **Оценка** |
| --- | --- |
| HPE System Reporter | В1 |
| SolarWinds Storage Resource Monitor | В2 |
| Система прогнозирования нагрузки СХД с интеллектуальной подсистемой настройки | В3 |

Далее необходимо представить качественные критерии в виде количественных для корректного сравнения. Их представления оформлены в виде таблиц 15−17.

**Таблица 15 − Вербально-числовая шкала для К2 (производительность)**

| **Качественная оценка** | **Количественная оценка** |
| --- | --- |
| Высокая | 1 |
| Хорошая | 0,7 |
| Средняя | 0,3 |
| Низкая | 0 |

**Таблица 16 – Вербально-числовая шкала для К3 (интеграция и совместимость)**

| **Качественная оценка** | **Количественная оценка** |
| --- | --- |
| Полная | 1 |
| Средняя | 0,5 |
| Отсутствует | 0 |

**Таблица 17 – Вербально-числовая шкала для К4 (отказоустойчивость)**

| **Качественная оценка** | **Количественная оценка** |
| --- | --- |
| Высокая | 1 |
| Средняя | 0,7 |
| Низкая | 0,4 |
| Отсутствует | 0 |

Проведем нормализацию критериев по методу полной нормализации [7] в соответствии с формулой (1).

|  |  |
| --- | --- |
|  | (1) |

где – значение критерия;

– минимальное значение критерия;

– максимальное значение критерия.

Распределим оценки по вариантам систем, что представлено в таблице 18.

**Таблица 18 – Количественные оценки по вариантам систем**

| **Код критерия** | **Критерий** | **Варианты систем** | | |
| --- | --- | --- | --- | --- |
| **В1** | **В2** | **В3** |
| К1 | Точность прогнозирования | 0 | 1 | 0,3 |
| К2 | Производительность | 0,7 | 0,3 | 1 |
| К3 | Интеграция и совместимость | 0 | 0,5 | 1 |
| К4 | Отказоустойчивость | 0,7 | 1 | 1 |
| К5 | Масштабируемость | 0,5 | 0,5 | 0,5 |
| К6 | Быстродействие | 0,5 | 0,7 | 1 |
| К7 | Отчётность | 0,7 | 0,7 | 1 |
| К8 | Удобство интерфейса | 1 | 0 | 0,7 |

При помощи Парето-оптимальности найдем варианты систем, которые не являются Парето-оптимальными, результат представлен в таблице 19.

**Таблица 19 − Сравнение вариантов на Парето-оптимальность систем**

| **Вариант системы** | **Вариант системы** | | |
| --- | --- | --- | --- |
| **В1** | **В2** | **В3** |
| В1 | 0 | 0 | 0 |
| В2 | 0 | 0 | 0 |
| В3 | 0 | 0 | 0 |
| Результат сравнения | 0 | 0 | 0 |
| Парето-оптимальность | Да | Да | Да |

Определим весовые коэффициенты критериев, используя метод балльной оценки. Для этого определим величину α, рассчитав значения весовых коэффициентов по формуле 2.

|  |  |
| --- | --- |
|  | (2) |

где *п* – общее количество весовых коэффициентов;

*ai* – весовые коэффициенты.

Подсчитаем значения локальных критериев для каждой программы с помощью метода взвешенной суммы, используя формулу 3 и представим результат в таблице 20.

|  |  |
| --- | --- |
|  | (3) |

где *kij* –коэффициент нормализации;

*ai* – весовые коэффициенты.

**Таблица 20 − Значения локальных критериев методом взвешенной суммы**

| **Код критерия** | **Коэффициент важности локального критерия ()** | **Значения локальных критериев** | | |
| --- | --- | --- | --- | --- |
| **В1** | **В2** | **В3** |
| K1 | 0,224 | 0 | 0,224 | 0,067 |
| K2 | 0,196 | 0,137 | 0,059 | 0,196 |
| K3 | 0,168 | 0 | 0,084 | 0,168 |
| K4 | 0,140 | 0,098 | 0,140 | 0,140 |
| K5 | 0,112 | 0,056 | 0,056 | 0,056 |
| K6 | 0,084 | 0,042 | 0,059 | 0,084 |
| K7 | 0,056 | 0,039 | 0,039 | 0,056 |
| K8 | 0,028 | 0,028 | 0,020 | 0,020 |
|  | | 0,400 | 0,681 | 0,787 |

Получив результат, можно сделать вывод, что В3 (система прогнозирования нагрузки СХД с интеллектуальной подсистемой настройки) является самым оптимальным вариантом. Поэтому эта система является актуальной и имеет преимущества перед аналогами.

1.1.5 Требования предъявляемые к разрабатываемой системе

#### 1.1.5.1 Функциональные требования

Поскольку разрабатываемая система основном предназначена для операторов, следящих за нагрузкой томов СХД, то функциональные требования к системе такие:

* указание параметров СХД;
* возможность выбирать таблицу из БД во frontend;
* возможность изменение уровня прогнозирования из БД во frontend;
* выбор режима скользящего окна (автоматический или ручной);
* при ручном вводе указывать интервал и его количество;
* выбор уровня предсказания;
* выбор альтернативного варианта прогнозирования;
* выбор нахождения глобального минимума, для предотвращения ошибок, при необходимости;
* выбор построения облака точек по необходимости;
* визуализация графика.

#### 1.1.5.2 Требования к надежности

При возникновении критических ошибок система должна сохранять свою работоспособность, а также рабочие файлы для последующего восстановления. Поскольку при возникновении критических ошибок система обеспечивает надежное сохранение своей работоспособности даже при возникновении критических ошибок, все рабочие файлы автоматически должны резервироваться и сохраняться для последующего восстановления, что гарантирует минимальные потери данных и позволяет продолжить работу без значительных проблем [8].

### 1.1.6 Исследование моделей прогнозирования

#### 1.1.6.1 Постановка задачи

Для разработки системы прогнозирования нагрузки СХД с интеллектуальной подсистемой настройки заказчик заранее выдвинул 3 модели для прогнозирования временных рядов: Random Forest, SVM и линейная регрессия. Задача состояла в том, чтобы исследовать какая модель без особой настройки больше подходит для системы. Также заказчиком были выданы исторические данные, чтобы провести исследование. В пункте 3.1.3 показаны примеры данных и описаны параметры.

#### 1.1.6.2 Random Forest

Random Forest – это мощный ансамблевый метод машинного обучения, используемый для задач классификации и регрессии. Он состоит из множества отдельных деревьев решений, которые работают совместно, чтобы улучшить общую производительность модели [9]. Основная идея метода случайных лесов заключается в построении большого числа деревьев решений на случайных подмножествах данных и признаков, а затем усреднении (или голосовании) результатов этих деревьев для получения окончательного прогноза.

Преимущества метода случайных лесов:

* высокая точность. Благодаря использованию множества деревьев, случайные леса часто достигают высокой точности на тестовых данных;
* стабильность и устойчивость к переобучению. Путем усреднения результатов множества деревьев метод случайных лесов снижает риск переобучения, присущий отдельным деревьям решений;
* работа с большими наборами данных и большим числом признаков. Метод случайных лесов хорошо масштабируется и может обрабатывать большие объемы данных с многочисленными признаками;
* интерпретируемость. Хотя отдельные деревья могут быть сложными для интерпретации, важность признаков можно оценить на основе их вклада в разделение узлов.

Ограничения метода случайных лесов:

* временная и вычислительная сложность. Обучение множества деревьев и выполнение прогнозов могут быть вычислительно затратными, особенно для очень больших наборов данных;
* память. метод требует значительных объемов памяти для хранения множества деревьев.

#### 1.1.6.3 Support vector machine (SVM)

Метод опорных векторов (Support Vector Machine, SVM) — это мощный алгоритм машинного обучения, который широко используется для задач классификации и регрессии [10]. SVM стремится найти оптимальную гиперплоскость, которая наиболее эффективно разделяет данные различных классов в многомерном пространстве признаков.

Преимущества метода опорных векторов:

* высокая точность. SVM часто достигает высокой точности классификации, особенно в случаях, когда данные линейно разделимы или могут быть разделены в более высокоразмерном пространстве;
* эффективность на данных с большим числом признаков. SVM хорошо работает с данными, содержащими большое количество признаков;
* гибкость в выборе ядра. Возможность выбора и настройки различных ядер позволяет SVM справляться с различными задачами классификации;
* устойчивость к переобучению. Путем настройки параметра регуляризации SVM может избегать переобучения и хорошо обобщать данные.

Ограничения метода опорных векторов:

* выбор ядра и параметров: Эффективность SVM сильно зависит от правильного выбора ядра и его параметров, что требует экспериментов и настройки;
* вычислительная сложность на больших наборах данных: SVM может быть вычислительно затратным при работе с очень большими наборами данных, так как обучение требует значительных ресурсов;
* интерпретируемость: в некоторых случаях модели SVM могут быть трудными для интерпретации, особенно при использовании нелинейных ядер.

#### 1.1.6.4 Линейная регрессия

Линейная регрессия — это один из наиболее простых и широко используемых методов статистического анализа и машинного обучения, применяемый для прогнозирования и установления зависимости между переменными [11]. Она основывается на предположении, что существует линейная связь между зависимой переменной (целевая переменная) и одной или несколькими независимыми переменными (признаки).

Преимущества линейной регрессии:

* простота и интерпретируемость. Линейная регрессия проста в понимании и интерпретации, что делает её доступной для широкого круга пользователей;
* быстрота вычислений. Модель линейной регрессии легко и быстро обучается, даже на больших наборах данных;
* эффективность на линейных зависимостях. Линейная регрессия эффективно работает, если существует линейная зависимость между переменными.

Ограничения линейной регрессии:

* линейность. Линейная регрессия предполагает линейную связь между переменными, что не всегда соответствует реальным данным;
* чувствительность к выбросам. Модель может быть чувствительна к выбросам, которые могут значительно повлиять на результаты;
* мультиколлинеарность. Высокая корреляция между независимыми переменными может привести к нестабильным оценкам коэффициентов модели.

#### 1.1.6.5 Метрики оценки

Для оценки качества моделей машинного обучения и их прогнозов используют различные метрики. При разработке системы прогнозирования нагрузки СХД рассмотрим три основные метрики: среднюю абсолютную ошибку (MAE), среднеквадратичную ошибку (MSE) и коэффициент детерминации (R²). Каждая из этих метрик имеет свои особенности и преимущества для анализа производительности моделей [12].

#### 1.1.6.5.1 MAE

Средняя абсолютная ошибка (Mean Absolute Error, MAE) измеряет среднюю величину ошибок между предсказанными и фактическими значениями. Она рассчитывается как среднее арифметическое абсолютных разностей между предсказанными и истинными значениями.

Преимущества MAE:

* прозрачность и легкость интерпретации;
* линейность: каждое отклонение имеет одинаковое влияние на итоговую метрику, что позволяет легко понимать и интерпретировать ошибки.

Недостатки MAE:

* не учитывает направление ошибок (перепредсказание или недопредсказание).

#### 1.1.6.5.2 MSE

Среднеквадратичная ошибка (Mean Squared Error, MSE) измеряет среднее квадратическое отклонение предсказанных значений от фактических. Она более чувствительна к большим ошибкам по сравнению с MAE, так как квадратичная функция значительно увеличивает вес больших ошибок.

Преимущества MSE:

* высокая чувствительность к большим ошибкам, что помогает выявить модели с крупными отклонениями;
* полезна при настройке моделей, где необходимо минимизировать большие ошибки.

Недостатки MSE:

* чувствительность к выбросам может исказить общую оценку модели;
* единицы измерения MSE отличаются от единиц исходных данных, что может затруднить интерпретацию.

#### 1.1.6.5.3 Коэффициент детерминации

Коэффициент детерминации (R²) оценивает долю дисперсии зависимой переменной, объясняемой моделью. Он показывает, насколько хорошо предсказанные значения совпадают с фактическими, и варьируется от 0 до 1. Значение R² близкое к 1 указывает на высокое качество модели.

Преимущества R²:

* Интуитивно понятная интерпретация: показывает долю объясненной дисперсии.
* Позволяет сравнивать качество различных моделей на одном наборе данных.

Недостатки R²:

* Может быть завышен для моделей с большим количеством параметров.
* Не всегда указывает на улучшение модели, особенно если модель

переобучена.

#### 1.1.6.6 Исследование

После написания кода были проведены эксперименты на исторических данных, предоставленные заказчиком. Далее рассмотрим результаты каждой модели по отдельности.

На рисунках 4-5 представлены результаты эксперимента Random Forest.

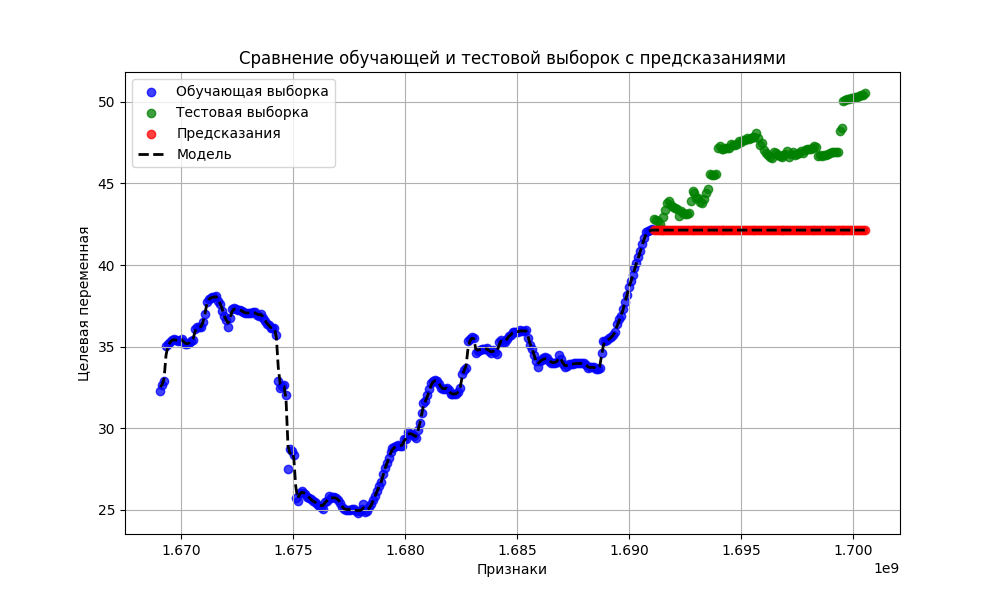


Рисунок 4 – Визуализация прогнозирования модели Random Forest

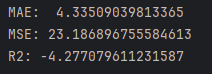


Рисунок 5 – Результаты оценивания Random Forest по метрикам

На основе полученных результатов можно сделать вывод, что модель имеет плохую производительность, то есть плохо предсказывает данные на тестовой выборке.

На рисунках 6-7 представлены результаты эксперимента SVM.

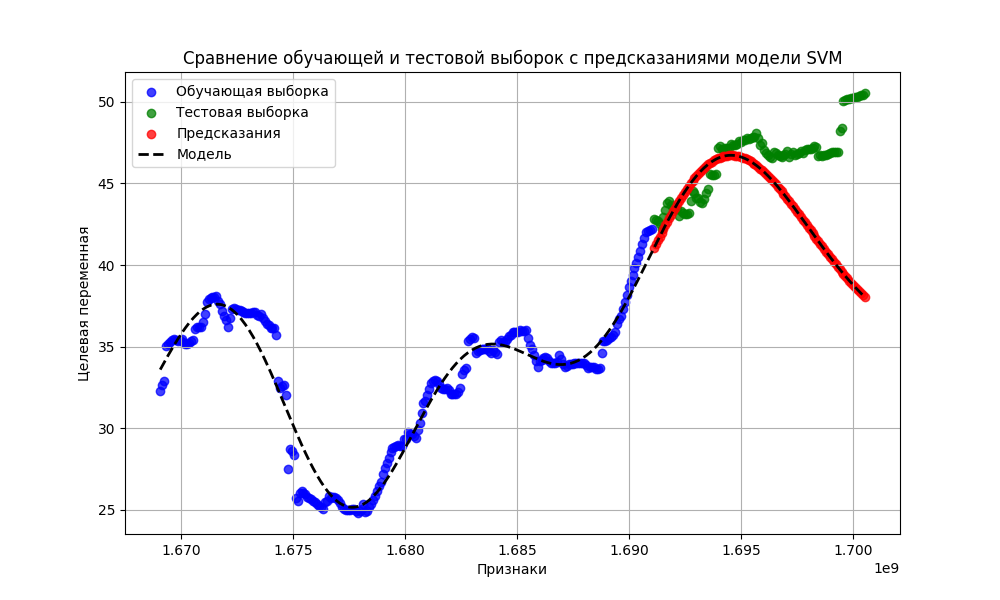


Рисунок 6 – Визуализация прогнозирования модели SVM

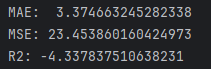


Рисунок 7 – Результаты оценивания SVM по метрикам

На основе полученных результатов можно сделать вывод, что модель справляется немного лучше, чем Random Forest, но все также имеет огромные ошибки, что видно на графике визуализации прогнозирования модели.

На рисунках 8-9 представлены результаты эксперимента линейной регрессии.

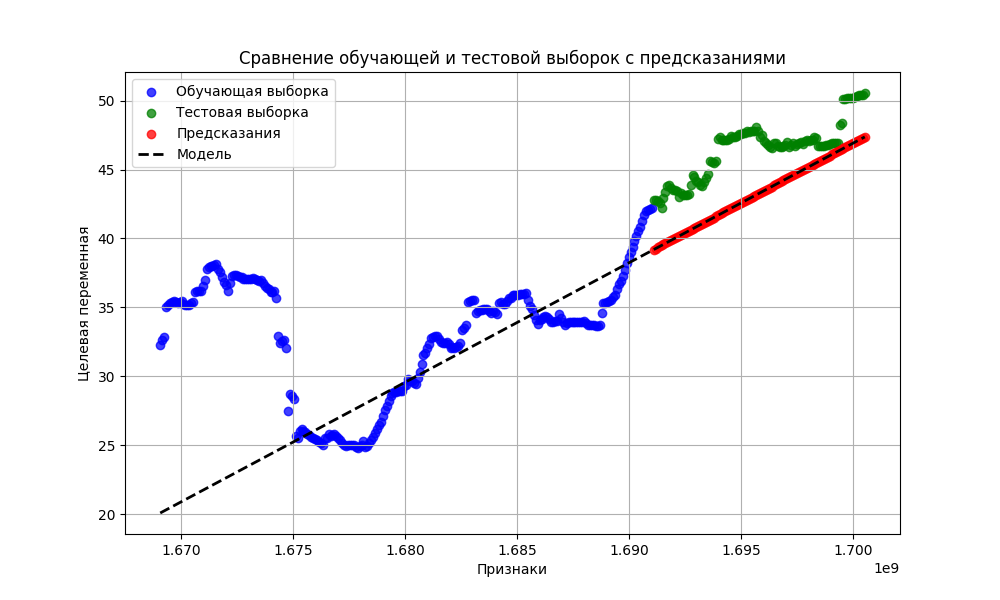


Рисунок 8 – Визуализация прогнозирования линейной регрессии

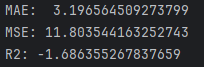


Рисунок 9 – Результаты оценивания линейной регрессии по метрикам

#### 1.1.6.7 Сравнение результатов

В таблице 21 проведём сравнительный анализ результатов по метрикам.

Пусть:

* В1 – Random Forest;
* В2 – SVM;
* В3 – линейная регрессия.

**Таблица 21 – Сравнение результатов эксперимента**

| **Вариант модели** | **Метрика** | | |
| --- | --- | --- | --- |
| **MAE** | **MSE** | **R2** |
| В1 | 4.3 | 23.2 | -4.3 |
| В2 | 3.4 | 23.5 | -4.3 |
| В3 | 3.2 | 11.8 | -1.7 |

При помощи Парето-оптимальности найдем варианты моделей, которые не являются Парето-оптимальными, результат представлен в таблице 22. Учтем, что чем меньше метрика – тем лучше.

**Таблица 22 − Сравнение вариантов на Парето-оптимальность моделей**

| **Вариант системы** | **Вариант модели** | | |
| --- | --- | --- | --- |
| **В1** | **В2** | **В3** |
| В1 | 0 | 0 | 0 |
| В2 | 0 | 0 | 0 |
| В3 | 1 | 1 | 0 |
| Результат сравнения | 1 | 1 | 0 |
| Парето-оптимальность | Нет | Нет | Да |

Получив результаты, можно сделать вывод, что линейная регрессия справляется лучше, чем Random Forest и SVM, однако также, как и они, совершает ошибки, поэтому в дальнейшем нужно производить препроцессинг данных.

2 КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

2.1 Технологическая часть

### 2.1.1 Технологии фронтенда

#### 2.1.1.1 Язык программирования JavaScript

JavaScript — это ключевой язык программирования для создания динамических веб-приложений. С момента своего создания в 1995 году он значительно эволюционировал и стал важнейшим инструментом в арсенале веб-разработчиков. JS является высокоуровневый, интерпретируемый языком программирования, предназначенный для добавления интерактивных элементов на веб-страницы. В отличие от статических языков разметки, таких как HTML и CSS, JS позволяет внедрять динамику и реагировать на действия пользователя, что делает веб-страницы более живыми и функциональными.

Основные возможности JS включают манипуляции с DOM и использование AJAX. DOM (Document Object Model) представляет собой программный интерфейс для HTML и XML-документов. JS предоставляет средства для динамического изменения структуры, стиля и содержания веб-страниц. С его помощью можно добавлять, удалять и изменять элементы, делая страницы более интерактивными и адаптивными к действиям пользователя.

Еще одна важная функция JS — это AJAX (Asynchronous JavaScript and XML), техника, позволяющая загружать данные с сервера без перезагрузки страницы. JavaScript позволяет отправлять асинхронные запросы к серверу и обрабатывать ответы, что обеспечивает динамическое обновление содержимого страницы и значительно улучшает пользовательский опыт.

#### 2.1.1.2 Фреймворк React

React — это библиотека для создания пользовательских интерфейсов, разработанная и поддерживаемая компанией Facebook. Она была впервые представлена в 2013 году и с тех пор стала одной из самых популярных технологий в веб-разработке. React позволяет создавать интерактивные и динамичные веб-приложения с использованием компонентного подхода.

React занял важное место в экосистеме фронтенд-разработки благодаря своей гибкости и производительности. Он предоставляет разработчикам мощные инструменты для создания сложных пользовательских интерфейсов с высокой степенью повторного использования кода. React широко используется в крупных проектах, таких как Facebook, Instagram и WhatsApp, что подтверждает его эффективность и надежность.

React имеет ряд преимуществ:

1. Компонентный подход – каждый компонент представляет собой независимую и повторно используемую часть пользовательского интерфейса. Это позволяет разработчикам разбивать приложения на небольшие, управляемые блоки, которые можно легко комбинировать и тестировать;
2. Виртуальный DOM – представляет собой легковесную копию реального DOM, которая обновляется в памяти перед тем, как изменения будут применены к реальному DOM. Этот процесс позволяет React эффективно управлять обновлениями интерфейса, уменьшая количество дорогостоящих операций с реальным DOM и обеспечивая высокую производительность и отзывчивость приложений;
3. Однонаправленный поток данных – он делает управление состоянием приложения более предсказуемым и удобным. Данные передаются от родительских компонентов к дочерним через свойства (props), что облегчает отслеживание и отладку состояния приложения. Это также способствует лучшей организации и структуре кода;
4. Огромное сообщество и богатая экосистема: Существует множество

библиотек и инструментов, которые расширяют возможности React и облегчают разработку. Например, такие библиотека, как Redux, помогают управлять состоянием приложения. А активное сообщество разработчиков постоянно создает и поддерживает новые инструменты и библиотеки, что делает React еще более мощным и гибким.

### 2.1.2 Технологии бэкенда

#### 2.1.2.1 Язык программирования Python

Python – это высокоуровневый, интерпретируемый язык программирования общего назначения, известный своей простой и понятной синтаксисом. Он был создан Гвидо ван Россумом и впервые выпущен в 1991 году. Python поддерживает множество парадигм программирования, включая процедурное, объектно-ориентированное и функциональное программирование.

Основные преимущества Python:

1. Удобство, гибкости и скорости разработки – все это достигается

благодаря простому синтаксису и богатой стандартной библиотекой;

1. Активное и большое сообщество – python обладает одним из самых

крупных и поддерживающих сообществ разработчиков, что обеспечивает доступ к большим возможностям и понятной документации;

1. Обширная экосистема – существует огромное множество библиотек и фрейморков для python, что позволяет охватить практически все сферы разработки.

Примеры использования Python:

* веб-разработка;
* научные исследования;
* аналитика данных;
* машинное обучение;
* игровая разработка.

#### 2.1.2.2 Фреймворк Django

Django – это высокоуровневый веб-фреймворк на языке Python, который позволяет быстро создавать надежные и масштабируемые веб-приложения. Он был разработан для упрощения процесса создания сложных и многофункциональных сайтов. Django следует философии «Не повторяй себя» (DRY, Don’t Repeat Yourself) и «Согласованность важнее конфигурации» (Convention over Configuration).

Основные характеристики Django:

* полный стек;
* быстрая разработка;
* безопасность;
* гибкость;
* активное сообщество.

Преимущества фреймворка:

1. Быстрая разработка и развертывание. Django позволяет быстро создавать веб-приложения благодаря своему полному стеку инструментов и шаблонам, что особенно важно для стартапов и проектов с ограниченными сроками.
2. Удобство и простота. Благодаря четкой документации и интуитивно понятному интерфейсу, Django легко изучить и использовать, что снижает порог входа для новых разработчиков.
3. Автоматизация задач. Django включает административную панель, которая автоматически генерируется на основе моделей данных. Это упрощает управление контентом и данными, позволяя разработчикам сосредоточиться на более важных задачах.
4. Совместимость с различными базами данных. Django поддерживает работу с различными базами данных, включая PostgreSQL, MySQL, SQLite и Oracle, что делает его гибким инструментом для разработки.
5. Расширяемость. Django легко расширять и интегрировать с другими библиотеками и фреймворками. Существуют многочисленные плагины и расширения, которые могут быть добавлены для расширения функциональности приложения.

**2.1.3 СУБД**

PostgreSQL (часто сокращаемая как Postgres) — это мощная, открытая реляционная система управления базами данных (СУБД), которая подчеркивает расширяемость и соответствие стандартам SQL. Она была разработана для управления большими объемами данных и поддержки сложных запросов, обеспечивая высокую надежность и производительность.

Основные характеристики PostgreSQL:

1. Расширяемость. PostgreSQL позволяет разработчикам добавлять новые функции, такие как типы данных, функции, операторы, агрегаты, индексы и языки процедур.
2. Поддержка стандартов. PostgreSQL соответствует многим стандартам SQL, что делает его совместимым с большинством других реляционных СУБД.
3. Открытый исходный код. PostgreSQL распространяется под лицензией PostgreSQL, которая является свободной и открытой, что позволяет использовать, изменять и распространять программное обеспечение без ограничений.

Рассмотрим таблицы, хранящиеся в БД. На рисунке 10 изображен пример таблицы data.

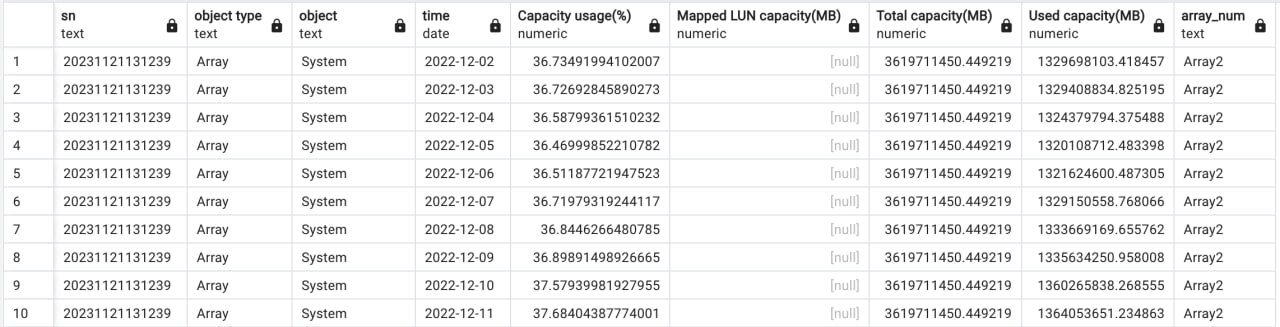


Рисунок 10 – Пример таблицы data

В таблице data хранятся исторические данные нагрузки томов СХД, распишем каждый параметр:

* sn – серийный номер хранилища данных;
* object type – тип объекта;
* object – название объекта;
* time – время записи данных;
* Array\_num – название системы хранения.

Опишем пропущенные параметры. В СХД Huawei Dorado выгружаются метрики заполнения дискового пространства в виде временных рядов в БД.

Объекты типа Array обозначают отдельную систему хранения. Предполагается, что в одной кластере хранения могут быть объединены несколько систем хранения. Уникальным идентификатором объектов этого типа является поле Object, и оно содержит название объекта, которое задается вручную администратором системы хранения данных. Внутри одной СХД не может быть 2 объекта типа Array с одинаковым значением поля Object. В двух разных СХД могут быть 2 объекта типа Array с одинаковым значением поля Object.

Объекты типа Array имеют 3 поля с параметрами:

1. Total Capacity - совокупная емкость всех созданных StoragePool в данной системе хранения. Измеряется в Mb. Предполагается, что этот параметр обозначает все доступное дисковое пространство соответствующей СХД. Значение этого параметра может быть увеличено путем физического добавления новых накопителей к СХД (array upgrade). Значение этого параметра не может быть уменьшено.
2. Used Capacity - совокупная емкость всех данных, физически записанных на накопители в рамках этой системы хранения. Измеряется в Mb. Значение этого параметра складывается из физического объема данных, записанных в LUN, FS и мгновенных снимках этих LUN и FS. Значение этого параметра не может превышать Total Capacity. Значение этого параметра может увеличиваться или уменьшаться. Если Used Capacity по значению приближается к Total Capacity, то это означает о скором окончании свободного дискового пространства. При полном соответствии значение Total и Used Capacity наступает критическая неисправность СХД – нехватка свободного дискового пространства, которая приводит к простою всех сервисов и может привести к недоступности данных, уже записанных на СХД.
3. Capacity Usage - процентное соотношение Used Capacity к Total Capacity. Объекты типа Storage Pool обозначают дисковые пулы, на базе которых создаются блочные тома (LUN), файловые системы (FS) и их служебные данные (мета, мгновенные снимки). Объект типа Array может содержать более одного Storage Pool либо не содержать ни одного. Объект Storage Pool всегда существует в рамках одного Array, и не может существовать без привязки к Array.

Уникальным идентификатором объектов этого типа является поле Object, и оно содержит название объекта, которое задается вручную администратором системы хранения данных. Внутри одной СХД не может быть 2 объекта типа Array с одинаковым значением поля Object. В двух разных СХД могут быть 2 объекта типа Array с одинаковым значением поля Object.

Объекты типа Storage Pool имеют 4 поля с параметрами:

1. Total Capacity - полезная емкость данного дискового пула. Измеряется в Mb. Вычисляется из количества дисков с учетом коэффициента RAID и количества хранимых метаданных. Значение этого параметра может быть увеличено путем добавления новых накопителей (storage pool expand) из числа неиспользуемых. Значение этого параметра не может быть уменьшено.
2. Used Capacity - совокупная емкость всех данных, физически записанных на накопители в рамках этого дискового пула. Измеряется в Mb. Значение этого параметра складывается из физического объема данных, записанных в LUN, FS и их служебных данных (мгновенных снимках и метаданных). Значение этого параметра не может превышать Total Capacity. Значение этого параметра может увеличиваться или уменьшаться. Если Used Capacity по значению приближается к Total Capacity, то это означает о скором окончании свободного дискового пространства в Storage Pool’е. При полном соответствии значение Total и Used Capacity все созданные блочные тома и файловые системы блокируются для записи, но данные на них доступны для чтения.
3. Capacity Usage - процентное соотношение Used Capacity к Total Capacity.
4. Mapped LUN capacity - совокупная емкость всех блочных томов, которые были презентованы хостам. Измеряется в Мб.

В СХД Huawei Dorado используются тонкие тома (Thin LUN), которые занимают дисковое пространство только по мере записи физических данных. Пример: в момент создания блочный том размером 100Тб занимает 0 Мб дискового пространства. После его презентации хосту и записи на него 10 Гб он начинает занимать соответствующие 10 Гб дискового пространства.  
Значение Mapped LUN capacity может увеличиваться или уменьшаться. Значение этого параметра может превышать значение Total Capacity.

Пример данных, хранящихся в таблице Level, представлен на рисунке 11.

Объясним каждый параметр:

* LEVEL0 – уровень прогнозирования для 0 уровня;
* LEVEL1 – уровень прогнозирования для 1 уровня;
* LEVEL2 – уровень прогнозирования для 2 уровня;
* object – название системы хранения для каждого уровня прогнозирования.

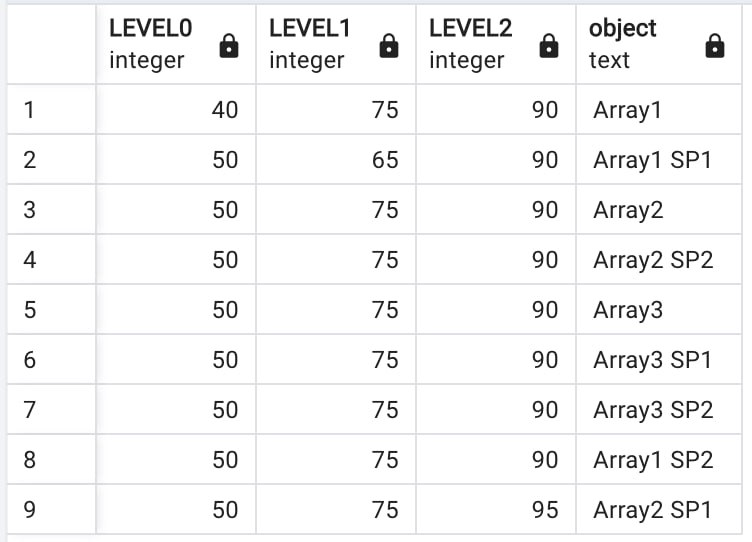


Рисунок 11 – Пример таблицы Level

### 2.1.4 Модель прогнозирования

Линейная регрессия – это метод статистического анализа, который используется для моделирования и анализа зависимостей между переменными. В линейной регрессии предполагается, что зависимость между зависимой переменной (или целевой переменной) и одной или несколькими независимыми переменными (предикторами) можно описать линейной функцией.

Основная цель линейной регрессии — найти такие параметры линейной функции (коэффициенты), которые минимизируют разницу между предсказанными значениями и фактическими значениями зависимой переменной.

Для построения модели линейной регрессии для предсказания временных рядов нужно учитывать, что временные ряды имеют временную зависимость, то есть текущее значение ряда зависит от предыдущих значений. Один из распространённых подходов заключается в использовании метода автокорреляции и лагов. Однако в нашем случае использовались скользящие окна, так как нужно было строить облака точек. Скользящее окно – это метод, используемый для анализа временных рядов и обработки данных, при котором анализируются подмножества данных (окна) определенной длины.

В системе мы можем выбрать один из двух режимом скользящего окна:

* автоматический;
* ручной (продвинутый).

Они отличаются тем, что в автоматическом режиме при помощи метода пиков, размер скользящего окна подбирается самостоятельно и постоянно меняется. В ручном (продвинутом) же мы задаем размер окна вручную, указывая интервал и его количество.

## 2.2 Конструкторская часть

### 2.2.1 Архитектура системы

Архитектура системы прогнозирования нагрузки СХД с интеллектуальной подсистемой настройки состоит из фронтенд части, реализованная на React JS, бэкенд на Python Django и БД на Postgres. Рассмотрим диаграмму развертывания на рисунке 12.

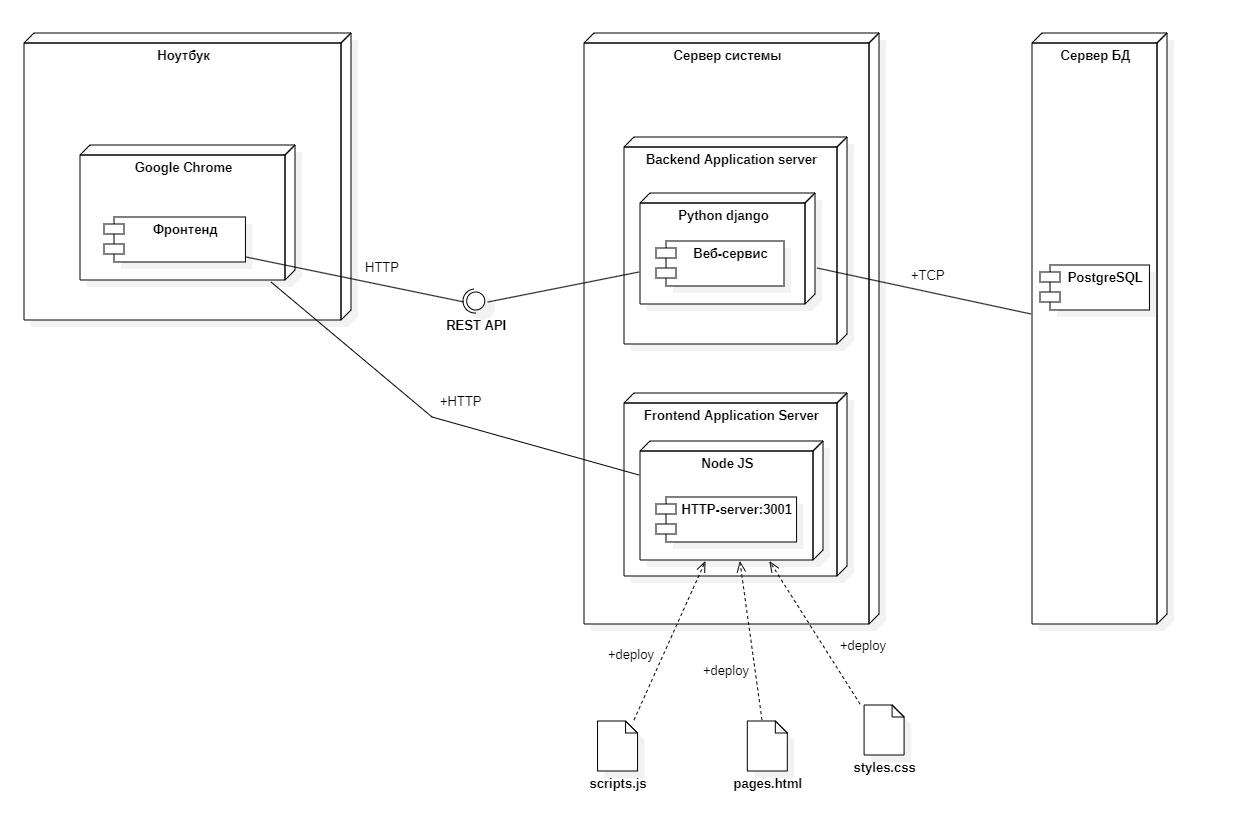


Рисунок 12 – Диаграмма развертывания

Диаграмма развёртывания изображает взаимодействие компонентов веб-приложения, включающего фронтенд, бэкенд и базу данных. Вот её описание и взаимодействие компонентов:

1. Ноутбук (Клиентская сторона):

* на клиентской стороне запущен веб-браузер (Google Chrome), в котором работает фронтенд-приложение;
* фронтенд-приложение обращается к REST API, используя HTTP-запросы.

1. Сервер системы:

Содержит два сервера приложений: Backend Application Server и Frontend Application Server.

Backend Application Server:

* использует Python Django для обработки веб-сервисов;
* обрабатывает запросы от фронтенда через REST API;
* взаимодействует с базой данных PostgreSQL по протоколу TCP.

Frontend Application Server:

* использует Node.js;
* запускает HTTP-сервер на порту 3001;
* деплой статических файлов (scripts.js, pages.html, styles.css) происходит на этот сервер.

1. Сервер базы данных (Сервер БД):

* хранит данные в PostgreSQL;
* получает запросы от Backend Application Server по протоколу TCP.

Взаимодействие компонентов:

* фронтенд-приложение в браузере на ноутбуке отправляет HTTP-запросы к REST API на Backend Application Server;
* Backend Application Server обрабатывает эти запросы, выполняет логику приложения и взаимодействует с базой данных PostgreSQL для получения данных при помощи Django ORM;
* Frontend Application Server обслуживает статические файлы (скрипты, HTML-страницы и стили CSS), которые могут использоваться фронтенд-приложением для обновления контента или интерфейса.

Таким образом, все компоненты работают совместно, обеспечивая функциональность веб-приложения, включая взаимодействие между клиентом, сервером приложений и базой данных.

Термины, которые стоит разобрать:

1. HTTP-запрос – это строка с информацией, которую отправляет клиент на сервер, для выполнения определенного действия, такого как получение веб-страницы или отправка данных на сервер.
2. TCP – это один из основных протоколов интернета, относящийся к транспортному уровню. TCP обеспечивает надёжную, упорядоченную и проверенную доставку потока данных между приложениями, работающими на узлах сети.
3. REST API – это архитектурный стиль, интерфейс, позволяющий приложениям взаимодействовать с сервером через стандартные HTTP-запросы.
4. Django ORM – это инструмент, встроенный в Django, которые позволяет разработчикам работать с БД, используя ООП, разработчики могут взаимодействовать с базой данных с помощью Python-объектов и методов, вместо написания сырых SQL-запросов.

### 2.2.2 Пользовательский интерфейс

Начальный экран представлен на рисунке 13.

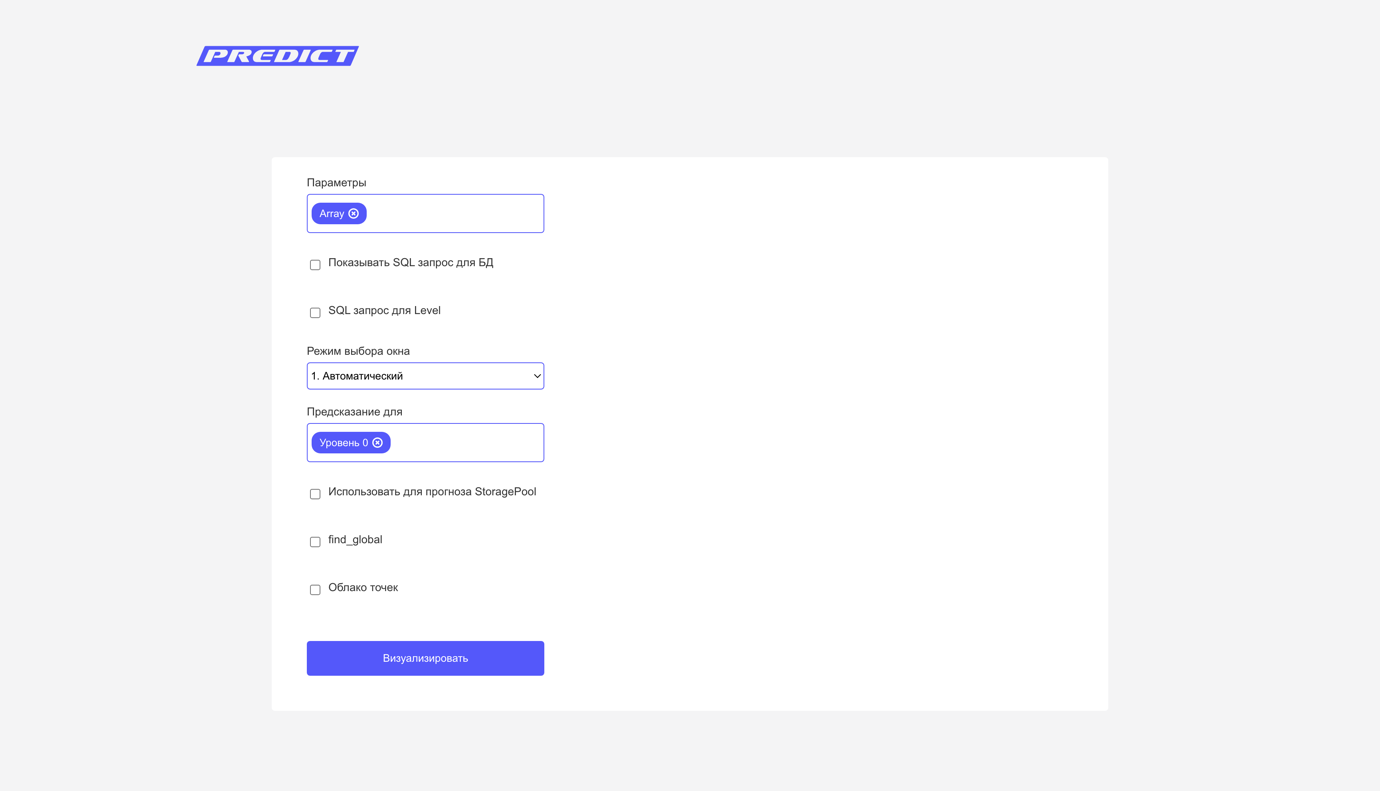


Рисунок 13 – Начальный экран

На рисунке 14 представлено поле ввода параметров. Параметры – указываются array, storagepool1 и storagepool2. Где array – это система хранения, а storagepool – это дисковые пулы в array.



Рисунок 14 – поле ввода параметров

На рисунке 15 показан чекбокс «Показать SQL запрос для БД», при нажатии на который появляется поле ввода SQL запроса для БД.

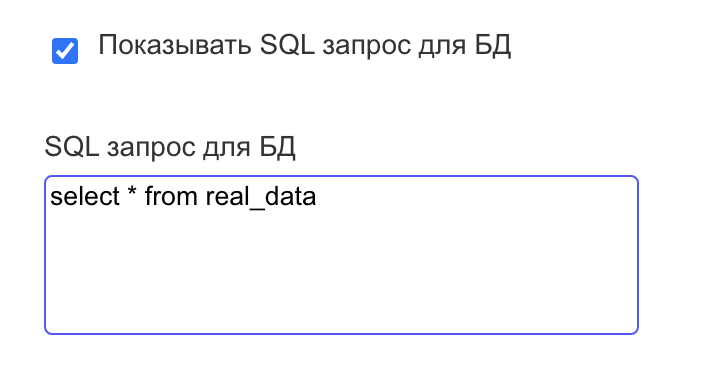


Рисунок 15 – поле ввода SQL запроса для БД

На рисунке 16 показан чекбокс «SQL запрос для Level», при нажатии на который появляется окно ввода SQL запрос для таблицы Level.

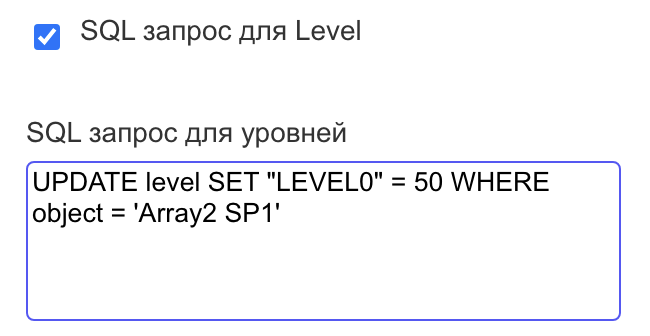


Рисунок 16 – поле ввода SQL запроса для таблицы Level

Далее на рисунке 17 представлен режим выбора окна. Как было написано ранее, скользящее окно имеет 2 режима: автоматический и ручной.

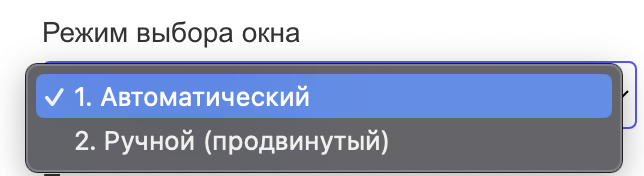


Рисунок 17 – Режим выбора окна

На рисунке 18 изображен select, в котором мы можем выбрать уровень прогнозирования.

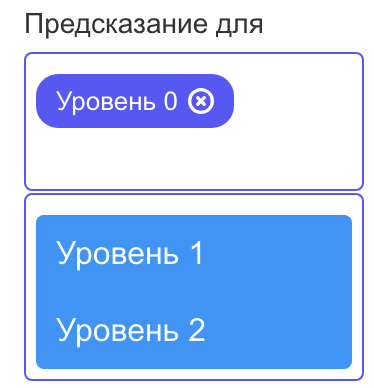


Рисунок 18 – выбор уровня прогнозирования

На рисунке 19 изображены чекбоксы: «Использовать для прогноза StoragePool», «find\_global» и «Облако точек». Чекбокс «Использовать для прогноза StoragePool» был добавлен по запросу заказчика. Эта опция позволяет учесть в прогнозе Array нагрузку на 100%, если хотя бы один дисковый пул заполнен. Чекбокс «find\_global» становится доступным только в автоматическом режиме скользящего окна. Этот метод напрямую влияет на размер окна, поскольку он определяет глобальный минимум, от которого строится прогноз. Необходимость данного метода возникает, когда исторические данные демонстрируют значительное падение на всем участке. Чекбокс «Облако точек» предназначен для построения облака точек, отображающего все предсказания при использовании скользящего окна.

Далее при нажатии на кнопку «визуализировать» появляется график с историческими данными, прогнозом, уровнями прогнозирования и возможно облако точек при нажатии на соответствующий чекбокс. На рисунке 20 представлен вариант, когда без изменений параметров в начальном окне нажата кнопка визуализации.

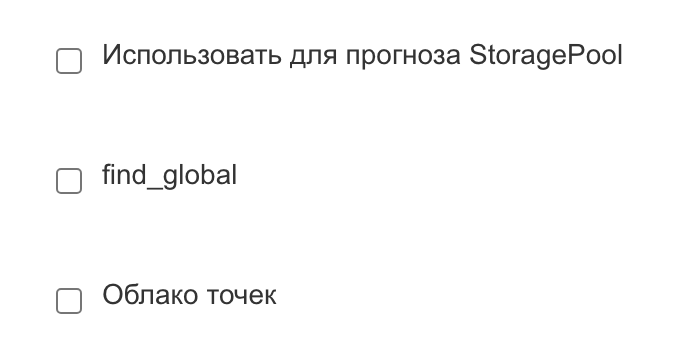


Рисунок 19 – Чекбоксы «Использовать для прогноза StoragePool», «find\_global» и «Облако точек»

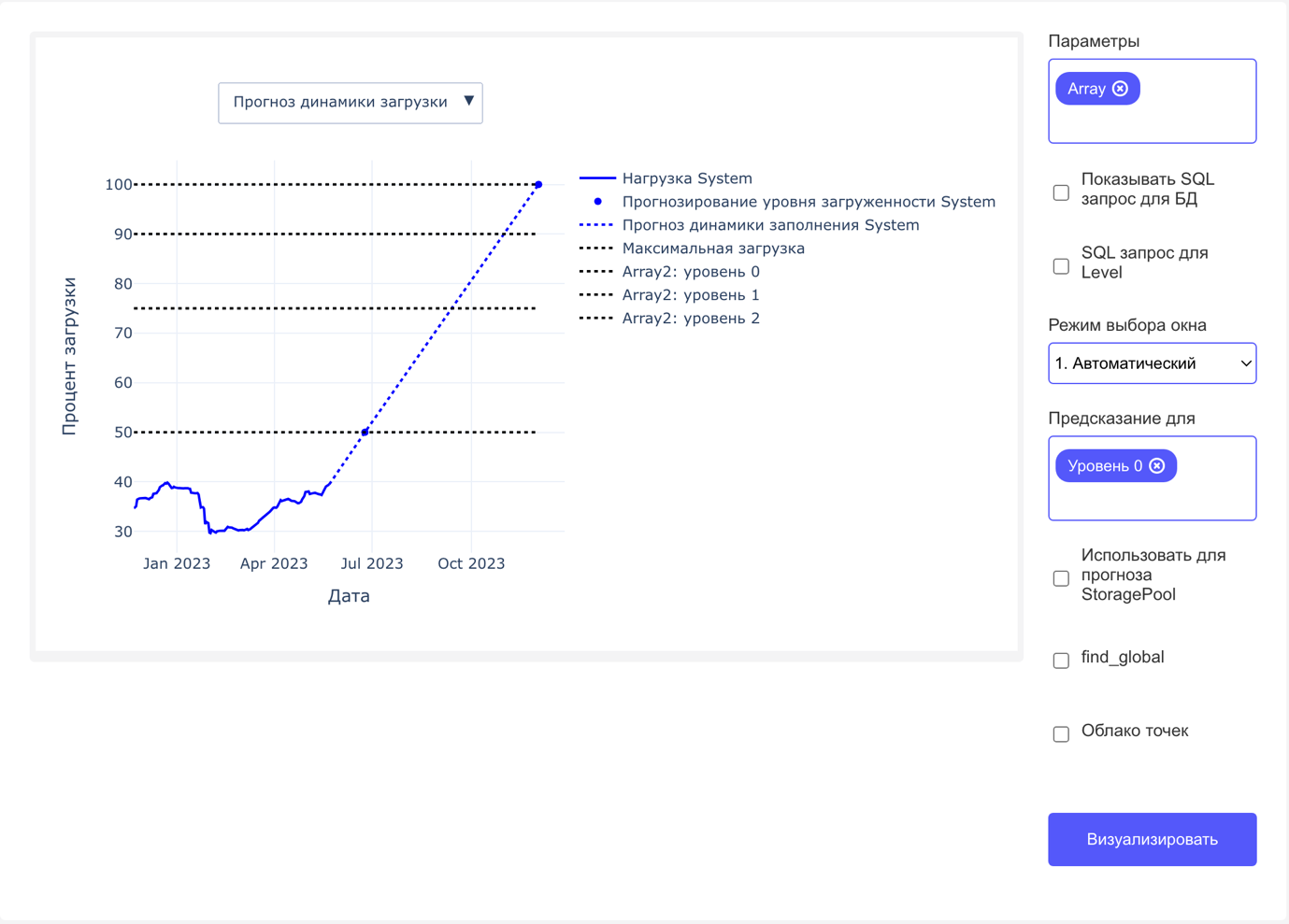


Рисунок 20 – Экран после нажатия на кнопку «визуализировать»

В приложении А находятся рисунки, на которых изображены результаты изменения параметров.

ЗАКЛЮЧЕНИЕ

При разработке системы прогнозирования нагрузки СХД с интеллектуальной подсистемой настройки были получены следующие результаты:

* Была изучена предметная область, связанная с системами хранения данных и методами прогнозирования их нагрузки. Проведен анализ существующих решений и технологий, включая использование алгоритмов машинного обучения для прогнозирования и оптимизации нагрузки.
* Были сформулированы требования к разрабатываемой системе, включающие необходимость высокой точности прогнозирования, производительности, интеграции с существующими системами, отказоустойчивости и удобства использования.
* Спроектирована архитектура системы, включающая фронтенд на React, бэкенд на Python/Django и базу данных на PostgreSQL. Определены ключевые компоненты системы, такие как модули сбора данных, модуль прогнозирования и интеллектуальная подсистема настройки.
* Разработана система прогнозирования нагрузки СХД, включающая алгоритмы машинного обучения для анализа данных и прогнозирования, а также интерфейс для настройки параметров и визуализации результатов. Проведены тестирования, подтвердившие высокую точность и надежность системы.

Полученное информационно-программное изделие обладает возможностями расширения за счет интеграции дополнительных платформ и алгоритмов. Это позволит системе адаптироваться к новым условиям и сохранять конкурентоспособность на рынке.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Конституция Российской Федерации: офиц. Текст.- М.: ЭКСМО, 2019. – 36с.
2. ГОСТ от 07.02.2011 N 3-ФЗ (ред. от 01.04.2019). [Электронный ресурс] // URL:http://www.consultant.ru. (Дата обращения 05.05.2024).
3. Григорьев Ю. А., Ревунков Г. И. Банки данных: Учеб. для вузов. – М.: Изд-во МГТУ им. Н. Э. Баумана, 2002. – 320 с.
4. Евсеев А. В., Мышенков К. С. Проектирование информационных систем: Учебное пособие. – М.: Изд. комплекс МГУПП, 2006. – 190 с.
5. Липаев В. В. Системное проектирование сложных программных средств для информационных систем. – М.: СИНТЕГ, 2002. - 224 с.
6. Официальный сайт СУБД PostgreSQL [Электронный ресурс] // postgresql.org URL: https://www.postgresql.org (дата обращения: 10.04.2019).
7. Электронная библиотека. [Электронный ресурс] – URL: http://www.zipsistes.ru/ (дата обращения: 05.05.2024).
8. Методические рекомендации по подготовке и защите выпускной квалификационной работы бакалавра. / Кротов Ю.Н. [Электронный ресурс] – URL: https://drive.google.com/file/d/1pEcfTr3xDdJ81Hxz2F6GcbtNV1n3dan6/view. (дата обращения: 05.05.2024).
9. Официальный сайт System Reporter overwiew [Электронный ресурс] // HPE URL: https://www.hpe.com/psnow/resources/ebooks/a00114824en\_us\_v2/sr\_overview.html (дата обращения 04.05.2024).
10. System Reporter threshold alerts conscept [Электронный ресурс] // techhub URL: https://techhub.hpe.com/eginfolib/storage/docs/3PARStorStoreServStorage/3PARStorServManagementConsole/SSMC3\_7\_OLH/\_help\_start.html#general/toc-s-system-reporter-alert-concepts.html (дата обращения 04.05.2024).
11. СХД (Система хранения данных) / itglobal [Электронный ресурс] – URL: https://itglobal.com/ru-ru/company/glossary/shd-sistema-hraneniya-dannyh/ (Дата обращения 04.05.2024).
12. Сеть хранения данных / Wikipedia [Электронный ресурс] – URL: https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D1%8C\_%D1%85%D1%80%D0%B0%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F\_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85 (дата обращения 05.05.2024).

ПРИЛОЖЕНИЕ А ГРАФИЧЕСКАЯ ЧАСТЬ

В графическую часть выпускной квалификационной работы входят:

А.1. Пример экранной формы с параметром Array.

А.2. Пример экранной формы с параметрами Array, StoragePool1 и StoragePool2.

А.3. Пример экранной формы со включённым режимом «Показывать SQL запрос для БД».

А.4. Пример экранной формы со включённым режимом «SQL запрос для Level».

А.5. Пример экранной формы с обоими включёнными режимами.

А.6. Пример экранной формы со включённым режимом «Использовать для прогноза StoragePool».

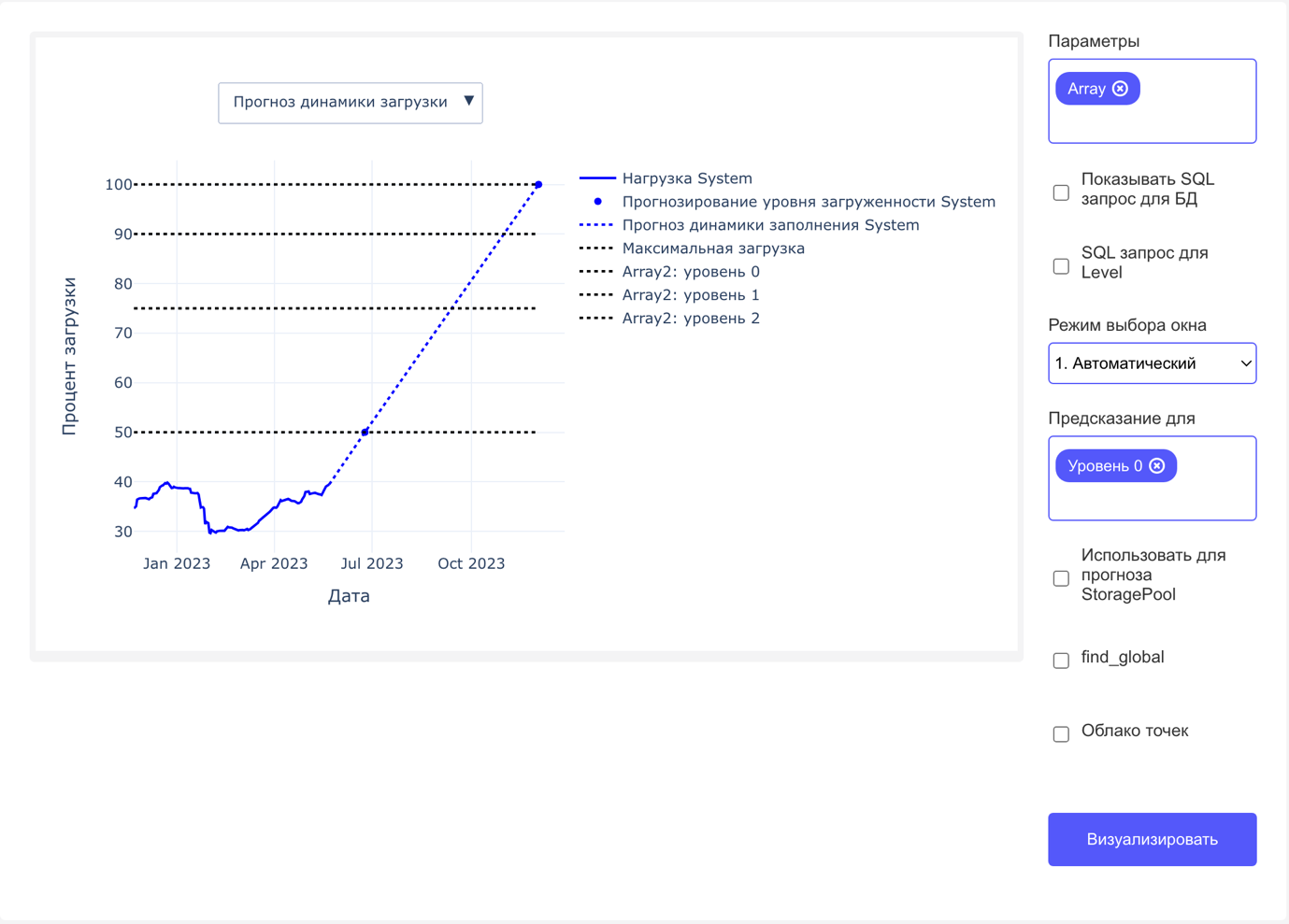
А.7. Пример экранной формы со включённым режимом «find\_global».

А.8. Пример экранной формы со включённым режимом «Облако точек».

А.9. Пример экранной формы с ручным режим выбора окна.

А.10. Пример экранной формы с увеличенным масштабом графика.

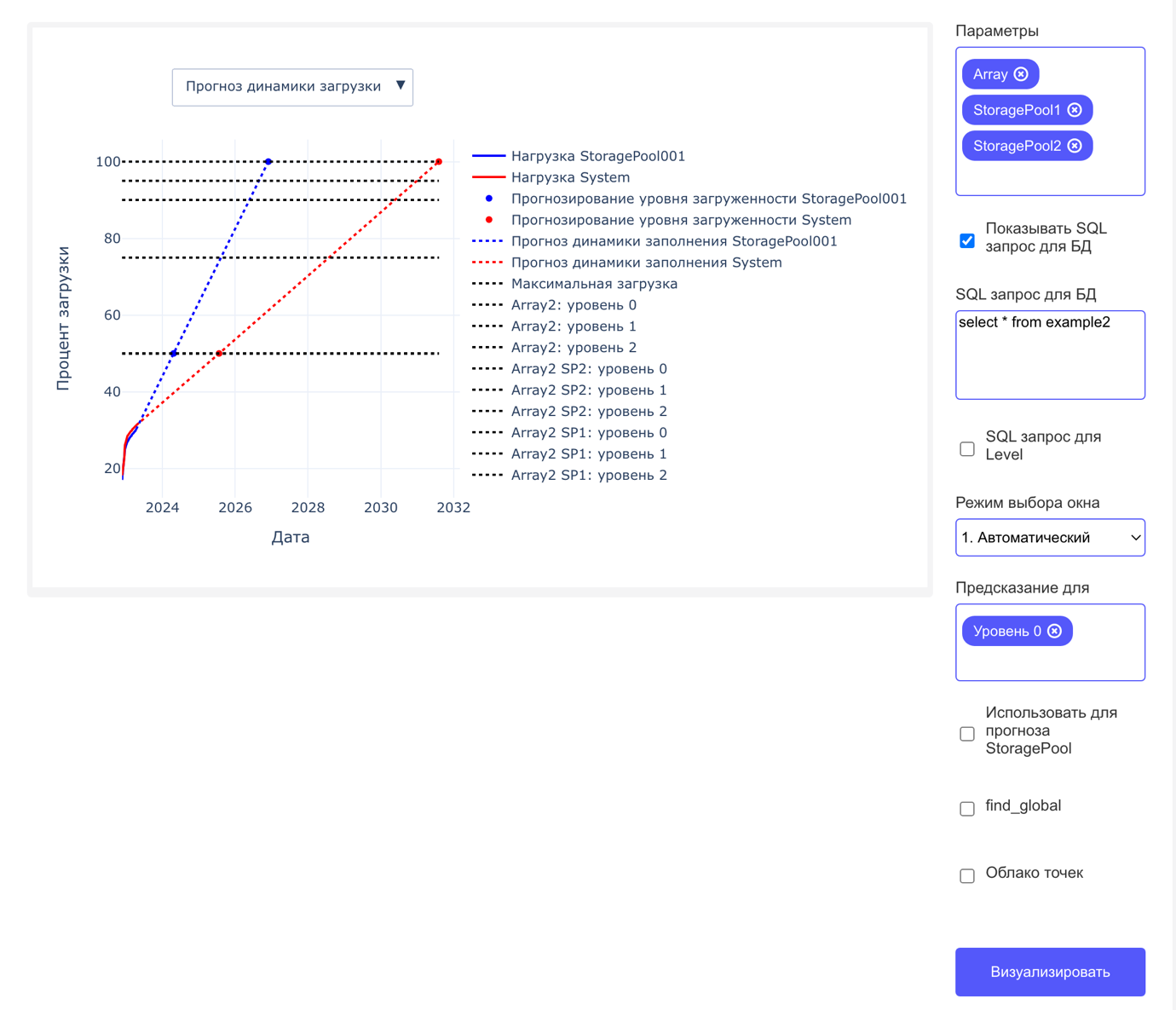
А.1 Пример экранной формы с параметром Array



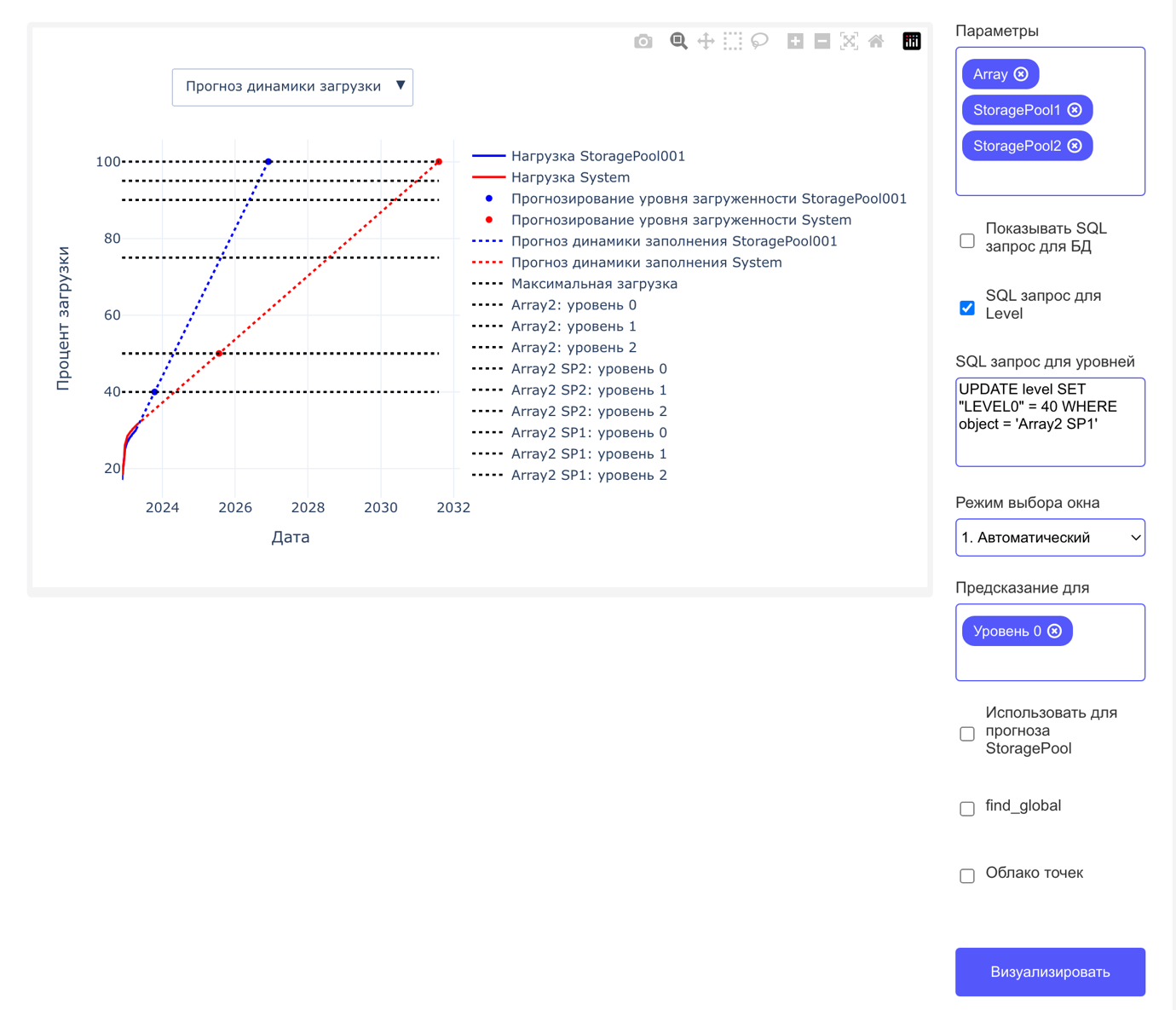
А.2 Пример экранной формы с параметрами Array, StoragePool1 и StoragePool2



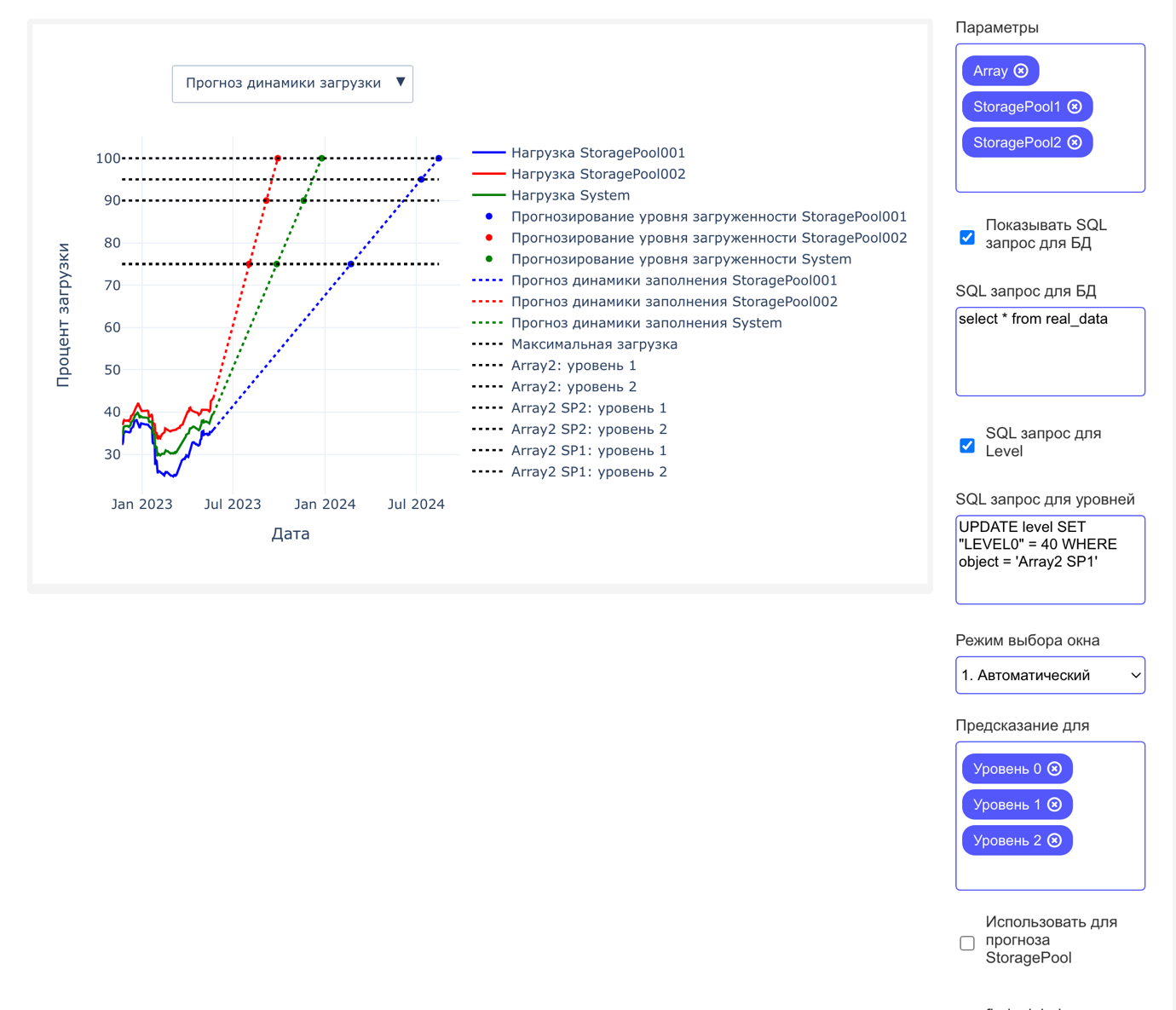
А.3 Пример экранной формы со включённым режимом «Показывать SQL запрос для БД»



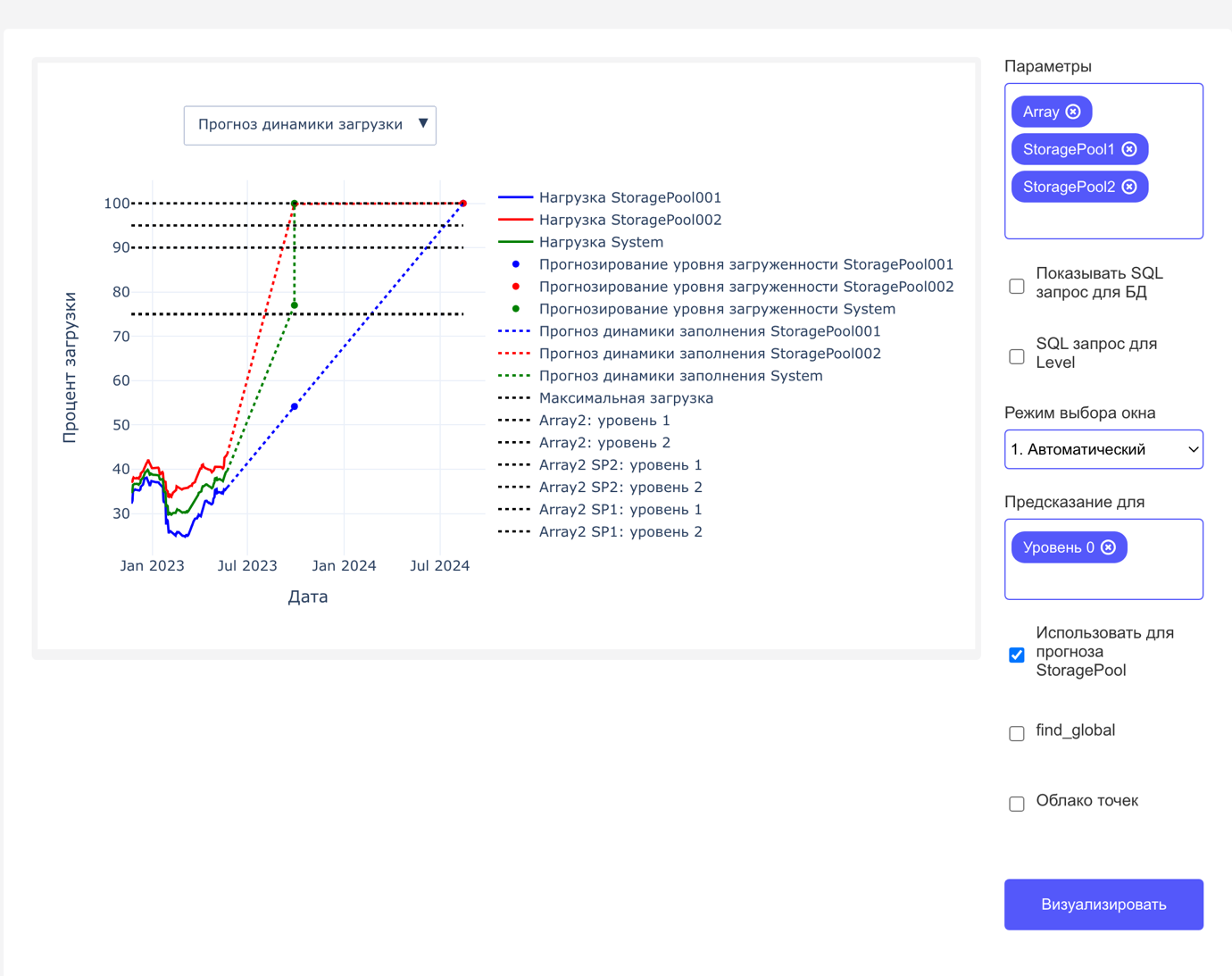
А.4 Пример экранной формы со включённым режимом «SQL запрос для Level»



А.5 Пример экранной формы с обоими включёнными режимами



А.6 Пример экранной формы со включённым режимом «Использовать для прогноза StoragePool»



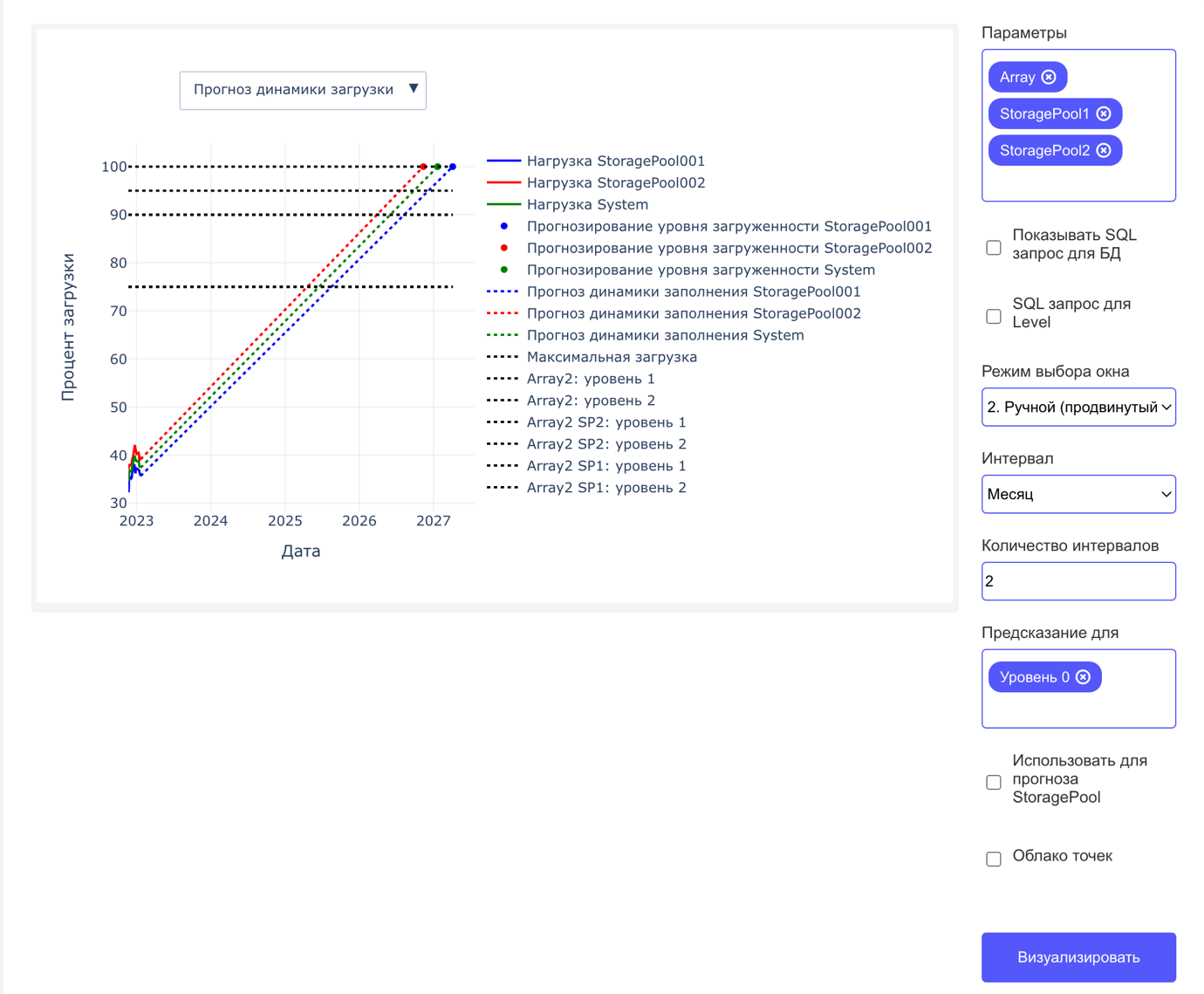
А.7 Пример экранной формы со включённым режимом «find\_global»



А.8 Пример экранной формы со включённым режимом «Облако точек»



А.9 Пример экранной формы с ручным режим выбора окна



А.10 Пример экранной формы с увеличенным масштабом графика



ПРИЛОЖЕНИЕ Б ТЕХНИЧЕСКОЕ ЗАДАНИЕ

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Кафедра «Системы обработки информации и управления»

|  |  |  |
| --- | --- | --- |
| Утверждаю  Заведующий кафедрой ИУ-5 |  | Согласовано  научный руководитель |
| \_\_\_\_\_\_\_\_\_\_\_В.И.Терехов  "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |  | \_\_\_\_\_\_\_\_\_\_А.А. Максаков  "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |

**Система прогнозирования нагрузки СХД с интеллектуальной подсистемой настройки**

Техническое задание

(вид документа)

писчая бумага

(вид носителя)

5

(количество листов)

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Ким Алексей Максимович |
| "\_\_"\_\_\_\_\_\_\_\_\_\_\_2024 г. |  |

Москва – 2024

1. **Введение**

Система прогнозирования нагрузки СХД с интеллектуальной подсистемой настройки. Будет создано 3 контейнера для работы системы – Frontend, BackEnd и БД postrgres.

1. **Основание для разработки**

Основанием для разработки является задание на выпускную квалификационную работу, подписанное руководителем выпускной работы и утверждённое заведующим кафедрой ИУ5 МГТУ им. Н.Э. Баумана 15 декабря 2023 года.

1. **Назначение и цель разработки**

Разрабатываемая система предназначена для прогнозирования перегрузки СХД (система хранилища данных), для предотвращения ошибок и повышение эффективности работы СХД.

1. **Требования к программе или программному изделию**

4.1 Требования к функциональным характеристикам:

Программа должна выполнять следующие функции:

4.1.1 Указание параметров СХД

4.1.2 Возможность выбирать таблицу из БД во frontend

4.1.3 Возможность изменение уровня прогнозирования из БД во frontend

4.1.4 Выбор режима скользящего окна (автоматический или ручной)

4.1.5 При ручном вводе указывать интервал и его количество

4.1.6 Выбор уровня предсказания

4.1.7 Выбор альтернативного варианта прогнозирования

4.1.8 Выбор нахождения глобального минимума, для предотвращения ошибок, при необходимости

* + 1. Выбор построения облака точек по необходимости
    2. Визуализировать график

4.2 Требования к надежности

При возникновении критических ошибок программа должна сохранять свою работоспособность, а также рабочие файлы для последующего восстановления.

4.3 Требования к составу технических средств

Операционная система, которая поддерживает интерпретатор Python , NodeJS и postrgres: Linux, macOS, или Windows. Интерпретатор языка программирования Python: Python 3.10. Установленные библиотеки numpy, pandas, tenacity, requests, matplotlib, plotly, scipy, scikit-learn, fastapi; также нужен NodeJS с установленным фреймворком React и библиотеками: plotly, axios, styled-component, react-spinners; и БД postgres.

**5. Требования к программной документации**

Для представления заказчику разрабатываются следующие документы:

1. Техническое задание.

2. Программа и методика испытаний.

3. Руководство пользователя – c описанием всех действий, которые пользователь может произвести, и реакцию системы на эти действия; порядок действий пользователя при зависании или сбое программы.

4. Расчётно-пояснительная записка.

**6. Технико-экономические показатели**

Требования к данному разделу не предъявляются

**7. Стадии и этапы разработки**

7.1 График приведен в соответствии с учебными неделями МГТУ им. Н.Э. Баумана 4 курса 8 семестра.

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование этапов выпускной квалификационной работы** | **Сроки выполнения этапов** |
| 1. | Задание на выполнение работы. Формулирование проблемы, цели и задач работы. | *26.01.2024*  *Планируемая дата* |
| 2. | Разработка и формулирование ТЗ | *20.03.2024*  *Планируемая дата* |
| 3. | Разработка программы и методики испытания | *20.04.2024*  *Планируемая дата* |
| 4. | Разработка второй части РПЗ “Конструкторско-технологическая часть” | *27.04.2024*  *Планируемая дата* |
| 5. | Защита макета программы (АСОиУ) | *12.05.2024*  *Планируемая дата* |
| 6. | Разработка заключения, приложений, оформление работы | *23.05.2024*  *Планируемая дата* |
| 7. | Подготовка доклада и презентации. Предзащита | *25.05.2024*  *Планируемая дата* |
| 8. | Получение заключения научного руководителя | *28.05.2024*  *Планируемая дата* |
| 9. | Допуск работы к защите на ГЭК (нормоконтроль) | *30.05.2024*  *Планируемая дата* |
| 10. | Защита работы на ГЭК | *01.06.2024*  *Планируемая дата* |

**8. Порядок контроля и приёмки**

Приём программного изделия в виде тестовых испытаний осуществляется в ходе «Защиты макетов программ – предварительной защиты ВКРБ» в период с 15 по 24 мая 2024 года в соответствие с разработанной программой и методикой испытаний.

На испытание представляются: рабочая программа, результаты выполнения и код данной программы.

ПРИЛОЖЕНИЕ В ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Кафедра «Системы обработки информации и управления»

|  |  |  |
| --- | --- | --- |
| Утверждаю  Заведующий кафедрой ИУ-5 |  | Согласовано  научный руководитель |
| \_\_\_\_\_\_\_\_\_\_\_В.И.Терехов  "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |  | \_\_\_\_\_\_\_\_\_\_А.А. Максаков  "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |

**Система прогнозирования нагрузки СХД с интеллектуальной подсистемой настройки**

Программа и методика испытаний

(вид документа)

писчая бумага

(вид носителя)

5

(количество листов)

ИСПОЛНИТЕЛЬ: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ким Алексей Максимович

"\_\_"\_\_\_\_\_\_\_\_\_\_\_2024 г.

Москва − 2024

**Аннотация**

В данном документе описываются последовательность и методы проведения испытаний при тестировании программного изделия, состав и структура технических и программных средств, необходимых для проведения испытаний, а также приводятся требования к предъявляемой документации, характеристикам программы применительно к условиям эксплуатации и требования к информационной и программной совместимости. Описывается тестовый пример и реакция системы на него.

1. **Объект испытаний**

Система прогнозирования нагрузки СХД с интеллектуальной подсистемой настройки Цель испытаний

Цель испытания – проверка функционирования всех указанных в техническом задании функций программы.

1. **Состав предъявляемой документации**

На испытания программного изделия предъявляются следующие документы:

1) Техническое задание.

2) Программа и методика испытаний.

1. **Технические требования**

* 1. **Требования к программной документации**

Комплектность программной документации должна удовлетворять разделу данного документа "Состав предъявляемой документации".

* 1. **Требования к техническим характеристикам**
     1. **Требования к составу аппаратного обеспечения**

Система будет выполнятся на ноутбуке с операционной системой Mac OS Ventura, с установленным Docker container.

* + 1. **Требование к составу программного обеспечения**

Для работы программного изделия требований к составу программного обеспечения не предъявляется.

1. **Порядок проведения испытаний**

Испытания данного программного изделие будут проводиться в следующем порядке:

1) Запуск системы.

2) Тестирование функционала системы.

* 1. **Требования к составу аппаратного обеспечения**

Требования к составу аппаратного обеспечения учитываются согласно пункту 4.2.1.

* 1. **Требования к составу программного обеспечения**

Требования к составу программного обеспечения учитываются согласно пункту 4.2.2.

1. **Методы испытаний**

|  |  |  |  |
| --- | --- | --- | --- |
| **N** | **N пункта ТЗ** | **Выполняемые действия** | **Результат** |
| 1 | 4.1.1  Указание параметра | Пользователь выбирает параметр один из трех СХД, который он хочет прогнозировать. Можно выбрать несколько | Прогнозируется по выбранному параметру |
| 2 | 4.1.2  Возможность выбрать таблицу из БД | Пользователь выбирает любую таблицу с данными. | Выбирается определенная таблица из БД с историческими данными СХД |
| 3 | 4.1.3  Возможность изменение уровня прогнозирования из БД во frontend | Пользователь вводит в специальное поле SQL запрос для изменения уровня прогнозирования | Уровень прогнозирования в БД, как отдельная таблица, меняет параметр на прописанные в SQL запросе |
| 4 | 4.1.4  Выбор режима скользящего окна | Пользователь:   1. Выбирает автоматический 2. Выбирает ручной    1. Вводит интервал    2. Вводит размер интервала | 1. Происходит выбор автоматического окна, то есть скользящее окно будет подстраиваться и находить оптимальный размер окна 2. Появляется дополнительные 2 поля ввода – интервал и размер интервала |
| 5 | 4.1.5  При ручном вводе указывать интервал и его количество | Пользователь вводит интервал и количество интервалов | После ввода интервала и его размера на всех исторических данных один размер окна |
| 6 | 4.1.6  Выбор уровня предсказания | Пользователь в меню выбирает уровень предсказания (можно выбрать несколько) | Происходит предсказание на определенный уровень загрузки в %, выбранные пользователем |
| 7 | 4.1.7  Выбор альтернативного варианта прогнозирования | Пользователь нажимает на чекбокс | Происходит смена варианта прогнозирования, то есть воспользоваться прогнозированием на основе StoragePool |
| 8 | 4.1.8  Выбор нахождения глобального минимума | Пользователь может выбрать нахождение глобального минимума, если у нас автоматическое скользящее окно | Находится глобальный минимум временных рядов, после чего автоматическое окно начинается с него, что помогает избежать ошибки out of bounds, которая возникает из-за пологих участков графика |
| 9 | 4.1.9  Выбор построения облако точек | Пользователь нажимает на чекбокс с построением облака точек | На графике будет построено облако точек |
| 10 | 4.1.10  Визуализировать график | Пользователь нажимает на кнопку визуализировать | Происходит визуализация графика с историческими данными, прогнозом, уровнями и облако точек, если оно выбрано |