**ПРИЛОЖЕНИЕ А**

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

|  |  |  |
| --- | --- | --- |
|  |  | УТВЕРЖДАЮ |
|  |  | Руководитель ОП  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Долженкова М.Л.  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г. |

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на разработку по теме

Оптимизация процесса обновления витрин данных КХД Энергосбыт с использованием Apache Spark

|  |  |  |
| --- | --- | --- |
|  |  | СОГЛАСОВАНО |
|  |  | Преподаватель МДК.05.01  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Долгих Д.К.  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

2025

Содержание

[Введение 2](#_Toc212583314)

[1 Термины и определения 3](#_Toc212583315)

[2 Перечень сокращений 4](#_Toc212583316)

[3 Основные сведения о разработке 5](#_Toc212583317)

[3.1 Наименование разработки 5](#_Toc212583318)

[3.2 Цель и задачи 5](#_Toc212583319)

[3.3 Сведения об участниках разработки 5](#_Toc212583320)

[3.4 Сроки разработки 5](#_Toc212583321)

[3.5 Назначение разработки 6](#_Toc212583322)

[3.5.1 Функциональное назначение 6](#_Toc212583323)

[3.5.2 Эксплуатационное назначение 6](#_Toc212583324)

[4 Описание предметной области 7](#_Toc212583325)

[5 Требования к результатам разработки 13](#_Toc212583326)

[5.1 Требования к функциональным характеристикам 13](#_Toc212583327)

[5.2 Требования к показателям назначения 13](#_Toc212583328)

[5.3 Требования к технологическому стеку 14](#_Toc212583329)

[5.4 Требования к видам обеспечения 14](#_Toc212583330)

[5.4.1 Требования к математическому обеспечению 14](#_Toc212583331)

[5.4.2 Требования к информационному обеспечению 15](#_Toc212583332)

[5.4.3 Требования к метрологическому обеспечению 15](#_Toc212583333)

[5.4.4 Требования к техническому обеспечению 15](#_Toc212583334)

[5.5 Требования к надежности 16](#_Toc212583335)

[5.6 Требования к безопасности 17](#_Toc212583336)

[5.7 Требования к патентной чистоте 17](#_Toc212583337)

[5.8 Требования к перспективам развития 18](#_Toc212583338)

[6 Состав и содержание работ 19](#_Toc212583339)

[7 Требования к документированию 20](#_Toc212583340)

[8 Требования к приемно-сдаточным процедурам 21](#_Toc212583341)

# Введение

Данный документ представляет техническое задание на разработку системы обновления аналитических витрин данных корпоративного хранилища данных (КХД) «Энергосбыт».

Техническое задание распространяется на разработку системы обновления аналитических витрин данных КХД, предназначенной для автоматизации и ускорения процесса обновления витрин данных, используемых для бизнес-аналитики в компании «Энергосбыт». Эта система способствует оптимизации ETL-процессов путем переноса ресурсоемких вычислений на кластер распределенной обработки данных Apache Spark. Она позволяет значительно сократить время подготовки данных и обеспечить аналитиков и руководство компании актуальной информацией для принятия оперативных решений. Предполагается, что администрировать и развивать данную систему будут инженеры данных, а основными потребителями результатов ее работы станут бизнес-аналитики компании.

Настоящий документ предназначен для заказчика и содержит описание разрабатываемой системы, требования к ее функциям, архитектуре, источникам данных, стадиям и этапам разработки. Является руководствующим документом для разработчика.

# Термины и определения

Apache Spark – фреймворк с открытым исходным кодом для реализации распределённых вычислений в кластерных системах, обеспечивающий высокую производительность за счет обработки данных в оперативной памяти.

Витрина данных – срез корпоративного хранилища данных, представляющий собой тематический набор данных, который ориентирован на определенную бизнес-область или группу пользователей и оптимизирован для аналитических запросов.

Кластер – группа объединенных по сети компьютеров (узлов), которые работают как единая система для выполнения ресурсоемких задач.

Корпоративное хранилище данных – предметно-ориентированная, интегрированная, хронологическая и неизменяемая совокупность данных, предназначенная для поддержки принятия управленческих решений.

Оркестратор – программное обеспечение, предназначенное для автоматизации, управления, координации и мониторинга выполнения сложных рабочих процессов.

ETL-процесс – процесс извлечения данных из внешних сточников, их преобразования и загрузки в целевое хранилище.

ClickHouse – колоночная аналитическая система управления базами данных (СУБД), позволяющая выполнять аналитические запросы в режиме реального времени на структурированных больших данных.

# Перечень сокращений

КХД – Корпоративное хранилище данных.

СОВД-Spark – Система обновления аналитических витрин данных КХД «Энергосбыт» средствами Apache Spark.

ETL – Extract, Transform, Load, Извлечение, Преобразование, Загрузка.

BI – Business Intelligence, Бизнес-аналитика.

API – Application Programming Interface, Программный интерфейс приложения.

HDFS – Hadoop Distributed File System, Распределенная файловая система Hadoop.

JDBC – Java Database Connectivity, Технология для соединения Java-приложений с базами данных.

ПО – Программное обеспечение.

СУБД – Система управления базами данных.

ТЗ – Техническое задание.

YARN – Yet Another Resource Negotiator, Менеджер ресурсов кластера Hadoop.

# Основные сведения о разработке

В данном разделе предоставлены основные сведения о разработке информационной системы.

## Наименование разработки

Наименование данной разработки – «Система обновления аналитических витрин данных КХД «Энергосбыт» средствами Apache Spark».

Краткое наименование – СОВД-Spark.

## Цель и задачи

Целью разработки является реализация программного комплекса СОВД-Spark в соответствии с требованиями, указанными в настоящем документе.

Для достижения цели необходимо:

* развернуть и настроить кластер Apache Spark для выполнения распределенных вычислений;
* разработать ETL-процесс на Apache Spark для извлечения данных из источников КХД, их преобразования и агрегации;
* реализовать загрузку рассчитанных витрин данных в целевую аналитическую базу данных ClickHouse;
* обеспечить автоматизацию процесса обновления витрин по расписанию;
* провести тестирование производительности системы и корректности расчетов данных.

## Сведения об участниках разработки

В качестве исполнителя выступает студент Колледжа ФГБОУ ВО «Вятский государственный университет» учебной группы ИСПк-403-52-00 – Долгих Дмитрий Константинович;

В качестве заказчика выступает преподаватель Колледжа ФГБОУ ВО «Вятский государственный университет» Долженкова Мария Львовна – преподаватель по УП.03 Учебная практика.

## Сроки разработки

Начало разработки: хх.хх.2025.

Конец разработки: хх.хх.2025.

## Назначение разработки

В данном разделе представлено назначение разработки информационной системы.

### Функциональное назначение

Основной функцией системы СОВД-Spark является автоматизированное выполнение ETL-процесса для обновления аналитических витрин данных. Система позволяет извлекать большие объемы сырых данных из различных источников КХД, производить их очистку, обогащение, объединение и сложные агрегирующие расчеты в распределенной среде Apache Spark, загружать сформированные витрины в целевую базу данных ClickHouse.

### Эксплуатационное назначение

Система СОВД-Spark предназначена для эксплуатации в IT-инфраструктуре компании «Энергосбыт». Основными пользователями и бенефициарами системы являются бизнес-аналитики, инженеры данных и руководство компании.

# Описание предметной области

Информационные системы для бизнес-аналитики и обработки больших данных (Big Data) представляют собой комплексные программные решения, направленные на помощь компаниям в достижении их стратегических целей, будь то повышение эффективности, оптимизация затрат или рост конкурентоспособности.

Концепция сбора и анализа данных для принятия решений существует давно, однако ранее она требовала формирования отчетов вручную или выполнения медленных запросов к транзакционным базам данных, что было трудоемко, ресурсозатратно и не позволяло получать информацию оперативно. Развитие технологий распределенных вычислений в 2000-х годах открыло новые возможности для автоматизации этого процесса. Появились первые технологии для обработки сверхбольших объемов данных, пионером в этой области стала экосистема Apache Hadoop.

Основной предпосылкой для создания и развития таких систем является стремление бизнеса к более осознанному подходу к управлению на основе точных и актуальных данных. Информационные системы призваны кардинально упростить эту задачу, предоставляя инструменты для автоматизированного сбора, обработки и анализа данных.

Информационные системы данного типа обычно включают в себя следующие основные компоненты:

* источники данных;
* КХД;
* механизм извлечения, преобразования и загрузки данных;
* аналитические витрины данных;
* системы визуализации и отчетности;
* системы оркестрации и мониторинга.

Целевая аудитории таких систем очень широка и включает бизнес-аналитиков, формирующих отчетность; руководителей подразделений и топ-менеджмент, принимающих решения на основе данных; инженеров данных, которые разрабатывают и поддерживают инфраструктуру обработки данных.

Основные задачи, которые решают данные информационные системы:

* ускорение и автоматизация подготовки аналитической отчетности;
* обеспечение консистентности и достоверности данных;
* предоставление возможности для проведения сложного анализа без влияния на производительность операционных систем;
* снижение трудозатрат на рутинные операции по сбору и обработке информации.

На рынке существует множество аналогов и смежных технологий, от традиционных ETL-инструментов до современных фреймворков распределенных вычислений, что свидетельствует о высоком спросе и критической важности данной предметной области для современного бизнеса.

**Аналог 1 – «Apache Hadoop».**

«Apache Hadoop» – это фундаментальный фреймворк с открытым исходным кодом для распределенного хранения и обработки сверхбольших наборов данных (Big Data). Проект был запущен в 2006 году Дугом Каттингом и Майком Кафареллой, и он во многом определил развитие всей индустрии больших данных. Его ключевыми компонентами являются распределенная файловая система HDFS (Hadoop Distributed File System) и модель вычислений MapReduce. Hadoop стал стандартом для пакетной обработки данных в масштабах от терабайт до петабайт. Логотип Apache Hadoop представлен на рисунке 4.1.



Рисунок 4.1 – Логотип Apache Hadoop

Преимущества:

* высокая масштабируемость и отказоустойчивость, Hadoop разработан для работы на кластерах из тысяч обычных серверов, обеспечивая надежное хранение и обработку данных;
* вокруг Hadoop существует огромное количество проектов, которые расширяют его функциональность для решения различных задач;
* экономическая эффективность позволяет хранить и обрабатывать петабайты данных со значительно меньшими затратами по сравнению с традиционными реляционными СУБД.

Недостатки:

* низкая производительность из-за дисковых операций;
* написание MapReduce-задач является сложным процессом по сравнению с современными фреймворками, требующим от разработчика написания большого количества шаблонного кода;
* Hadoop не подходит для интерактивных запросов и обработки данных в режиме, близком к реальному времени.

**Аналог 2 – «Apache Airflow».**

«Apache Airflow» – это платформа для программного создания, планирования и мониторинга рабочих процессов. Проект был создан в Airbnb в 2014 году и позже передан в Apache Software Foundation. Airflow не является системой обработки данных, а выступает в роли **оркестратора**, управляющего сложными конвейерами данных. Задачи в Airflow описываются в виде направленных ациклических графов (DAGs) с использованием языка Python. Пример DAG в интерфейсе Airflow представлен на рисунке 4.2.

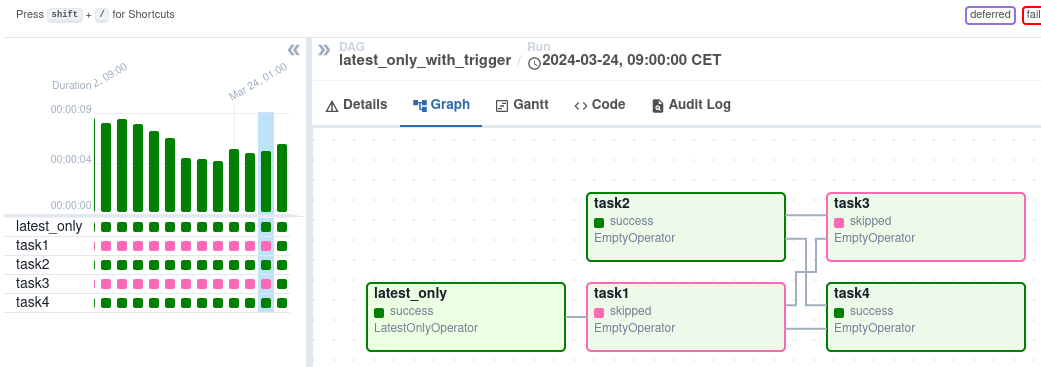


Рисунок 4.2 – Пример DAG в интерфейсе Airflow

Преимущества:

* использование Python для определения рабочих процессов позволяет создавать сложные логические зависимости, циклы и ветвления;
* мощный веб-интерфейс предоставляет удобные инструменты для визуализации конвейеров, мониторинга статуса выполнения задач, просмотра логов и управления запусками;
* существует огромное количество готовых операторов для взаимодействия с базами данных, облачными сервисами и системами обработки данных.

Недостатки:

* Airflow только запускает и контролирует задачи, но сами вычисления должны выполняться во внешней системе, он не может самостоятельно выполнять распределенную обработку данных;
* в инсталляциях с тысячами конвейеров планировщик Airflow может стать узким местом, требующим тонкой настройки и оптимизации;
* Airflow не предназначен для передачи больших объемов данных между задачами.

**Аналог 3 – «Apache Flink».**

«Apache Flink» – это фреймворк и распределенный движок для обработки потоковых данных. Проект зародился в академической среде и стал проектом верхнего уровня Apache в 2014 году. Ключевая особенность Flink – его архитектура, построенная по принципу «stream-first», где пакетная обработка рассматривается как частный случай потоковой. Это позволяет ему обеспечивать низкую задержку и высокую пропускную способность при обработке непрерывных потоков данных. Логотип Apache Flink представлен на рисунке 4.3.

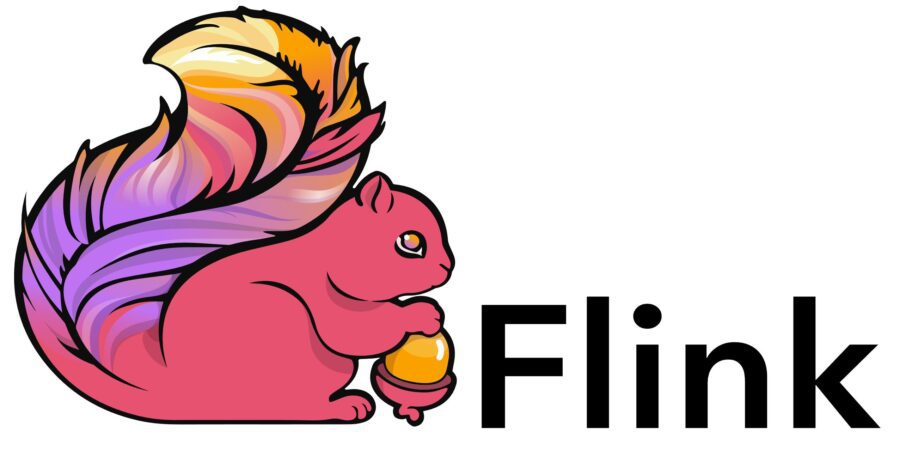


Рисунок 4.3 – Логотип Apache Flink

Преимущества:

* Flink обрабатывает события по одному, что позволяет достигать задержек в миллисекунды и идеально подходит для задач реального времени;
* продвинутое управление состоянием предоставляет мощные и надежные механизмы для хранения и управления состоянием вычислений, что критически важно для сложных алгоритмов обработки потоков;
* поддерживает разнообразные типы окон на основе времени события или времени обработки.

Недостатки:

* API и концепции Flink считаются более сложными для освоения по сравнению с Apache Spark;
* API и оптимизатор для задач пакетной обработки менее развиты, чем Catalyst Optimizer в Spark SQL;
* меньшее сообщество и экосистема, что выражается в меньшем количестве готовых библиотек, интеграций и общедоступных материалов.

На основе анализа вышеописанных систем обработки данных были выявлены различные их преимущества. Apache Hadoop предоставляет непревзойденную масштабируемость и отказоустойчивость для хранения данных. Apache Airflow является мощным и гибким инструментом для оркестрации сложных рабочих процессов. Apache Flink демонстрирует высочайшую производительность и низкую задержку в задачах потоковой обработки.

Однако для цели проекта был выявлен ряд недостатков у рассмотренных аналогов, которые делают их применение неоптимальным:

* Apache Hadoop использует устаревшую и медленную модель вычислений MapReduce, основанную на дисковых операциях, что неприемлемо для задачи оперативного обновления витрин;
* Apache Airflow не является вычислительным движком и может лишь управлять запуском задач, но не выполнять саму обработку данных;
* Apache Flink, будучи ориентированным в первую очередь на потоковую обработку, сложнее в освоении и менее оптимизирован для классических ETL-задач пакетной обработки, которые составляют основу данного проекта.

Для решения поставленной задачи и создания эффективной системы СОВД-Spark был выбран фреймворк Apache Spark, который предлагает следующие ключевые преимущества перед аналогами:

* высокая производительность за счет обработки в памяти;
* Spark предоставляет единый API (Spark SQL, DataFrame API) для решения как пакетных, так и потоковых задач, что делает его гибким и удобным инструментом для большинства  
  ETL-сценариев;
* высокоуровневые API на Python и мощный оптимизатор запросов Catalyst делают разработку ETL-процессов значительно проще и быстрее, чем на Hadoop MapReduce, и более интуитивной для пакетных задач, чем на Apache Flink.

Реализация системы на базе Apache Spark позволит создать современный, производительный и масштабируемый конвейер данных, который будет значительно превосходить традиционные подходы по скорости обновления витрин и будет проще в разработке и поддержке для данной конкретной задачи, чем более нишевые инструменты.

# Требования к результатам разработки

В данном разделе представлены требования к разработке приложения, которым она должна соответствовать.

## Требования к функциональным характеристикам

Разрабатываемая система СОВД-Spark должна обладать следующими функциональными возможностями:

* извлечение и чтение данных:

обеспечение подключения к источникам данных КХД «Энергосбыт»;

реализация извлечения данных из источников за определенный период;

распараллеливание процесса чтения данных для обеспечения высокой скорости извлечения больших объемов информации;

* преобразование и агрегация данных:

выполнение процедур очистки и валидации данных, включая обработку пропусков, аномальных значений и дубликатов;

обогащение данных путем объединения информации из нескольких источников;

реализация бизнес-логики для вычисления производных показателей;

выполнение сложных агрегирующих операций для формирования итоговых витрин данных, оптимизированных для аналитических запросов;

* загрузка и публикация данных:

Загрузка результирующих витрин данных в целевую аналитическую СУБД ClickHouse;

поддержка режима полного перезаписи данных в целевых таблицах для обеспечения консистентности витрин при каждом обновлении;

обеспечение высокой производительности записи данных в ClickHouse.

## Требования к показателям назначения

Разрабатываемая система СОВД-Spark должна соблюдать следующие требования к показателям назначения:

* общее время выполнения суточного ETL-процесса не должно превышать 4 часов при объеме обрабатываемых исходных данных до 1 ТБ;
* на этапе преобразования и агрегации система должна обеспечивать среднюю скорость обработки данных не менее 300 000 записей в секунду;
* отклонение контрольных сумм по ключевым бизнес-показателям в итоговой витрине от эталонных значений, рассчитанных по исходным данным, должно составлять 0%. Любые расхождения должны быть зафиксированы в логах как критическая ошибка;
* данные в аналитических витринах должны быть доступны для пользователей не позднее 08:00 по московскому времени каждого рабочего дня;

## Требования к технологическому стеку

Для разработки логики ETL-процессов должен использоваться язык программирования Python с применением библиотеки PySpark. Этот выбор обусловлен низким порогом вхождения, простотой синтаксиса, а также широкой поддержкой и наличием большого количества библиотек для обработки и анализа данных.

В качестве основного движка для обработки данных должен использоваться Apache Spark. Он обеспечивает высокую производительность за счет вычислений в оперативной памяти и предоставляет мощный API для реализации сложных преобразований и агрегаций.

В качестве основного хранилища для сырых и промежуточных данных должна использоваться распределенная файловая система HDFS (Hadoop Distributed File System).

В качестве целевой СУБД для хранения итоговых аналитических витрин должна использоваться колоночная база данных ClickHouse. Она обеспечивает сверхбыстрое выполнение аналитических запросов, что критически важно для BI-систем.

Для автоматизации запуска ETL-процессов по расписанию и управления зависимостями между задачами должен использоваться оркестратор Apache Airflow.

Иные требования к технологическому стеку не предъявляются.

## Требования к видам обеспечения

В данном разделе приведены требования к видам обеспечения разработки.

### Требования к математическому обеспечению

Математическое обеспечение системы СОВД-Spark должно включать:

* алгоритмы инкрементального чтения данных: система должна реализовывать логику чтения только новых или измененных записей из систем-источников за прошедший период;
* алгоритмы объединения данных: система должна выполнять распределенные операции объединения больших наборов данных;
* алгоритмы очистки и стандартизации данных: должны быть реализованы процедуры для обработки пропущенных значений и фильтрации аномальных значений;
* алгоритмы агрегации данных: система должна реализовывать распределенные агрегирующие функции для расчета итоговых показателей витрин.

### Требования к информационному обеспечению

В данном подразделе приведены требования к информационному обеспечению.

#### Требования к форматам хранения данных

Для хранения данных в рамках системы СОВД-Spark должны использоваться следующие форматы:

Все данные, извлеченные из систем-источников (сырые данные), а также результаты промежуточных вычислений в ETL-процессе должны храниться в распределенной файловой системе HDFS. Основным форматом для хранения этих данных должен быть Apache Parquet.

Итоговые, агрегированные витрины данных должны храниться в целевой аналитической СУБД ClickHouse. Для хранения данных в ClickHouse должны использоваться движки таблиц семейства MergeTree, которые оптимизированы для быстрой вставки больших пакетов данных и высокопроизводительных аналитических запросов.

При передаче данных между этапами ETL-процесса внутри Apache Spark используется внутренний формат представления данных. Данные для загрузки из Spark в ClickHouse передаются через JDBC-соединение в пакетном режиме.

#### Требования к лингвистическому обеспечению

В разрабатываемом мобильном приложении будет использоваться русский язык.

### Требования к метрологическому обеспечению

Требования к метрологическому обеспечению не предъявляются.

### Требования к техническому обеспечению

Требования к вычислительной технике приведены ниже.

Требования к управляющим узлам:

* операционная система – серверная ОС семейства Linux;
* процессор – не менее 8 vCPU;
* оперативная память – не менее 32 Гб;
* свободное место на системном накопителе – не менее 100 Гб, рекомендуется  
  SSD-накопитель;
* диск для метаданных HDFS: не менее 500 Гб.

Требования к рабочим узлам:

* операционная система – серверная ОС семейства Linux, идентичная ОС на управляющих узлах;
* процессор – не менее 16 vCPU;
* оперативная память – не менее 64 Гб;
* свободное место на системном накопителе – не менее 100 Гб, рекомендуется  
  SSD-накопитель;
* диск для метаданных HDFS: не менее 2 Тб.

## Требования к надежности

Стабильная работа программы должна быть гарантирована посредством следующих организационно-технических мер:

* обеспечение непрерывного электропитания технических устройств в соответствии с требованиями ГОСТ Р 55844–2013 «Информационные технологии. Управление безопасностью информации. Общие требования»;
* контроль за вводимыми данными в соответствии с Федеральным законом № 149-ФЗ «Об информации, информационных технологиях и о защите информации»;
* регулярное обслуживание программных средств и аппаратного обеспечения в соответствии с Постановлением Министерства труда и социального развития РФ от 23 июля 1998 года № 31 «Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств»;
* систематическое тестирование программных средств на наличие компьютерных вирусов в соответствии с ГОСТ 51188–98 «Защита информации. Тестирование программных средств на наличие компьютерных вирусов».

Дополнительно, для обеспечения надежности программы следует учитывать требования Федерального закона № 127-ФЗ «О техническом регулировании», который устанавливает стандарты и требования к техническим системам для обеспечения их устойчивой работы.

## Требования к безопасности

Безопасное функционирование системы СОВД-Spark должно быть обеспечено комплексом организационно-технических мер, которые включают в себя:

* система должна поддерживать механизмы строгой аутентификации для доступа пользователей и сервисов к ресурсам кластера;
* должно быть реализовано разграничение прав доступа к данным. Система должна позволять настраивать права доступа (чтение, запись, выполнение) к файлам и каталогам в HDFS на уровне отдельных пользователей и групп;
* все сетевые коммуникации между узлами кластера в процессе выполнения Spark-заданий должны быть защищены с помощью механизмов шифрования;
* должно проводиться регулярное обновление программного обеспечения кластера для устранения известных уязвимостей. Доступ к узлам кластера по сети должен быть ограничен с помощью межсетевых экранов.

Иные требования к безопасности не предъявляются.

## Требования к патентной чистоте

Программа должна соответствовать следующим требованиям в области патентной чистоты:

* использование ПО с открытым исходным кодом: Все ключевые компоненты технологического стека являются программным обеспечением с открытым исходным кодом и распространяются под премассивными лицензиями. Данные лицензии разрешают свободное использование, модификацию и распространение ПО, в том числе в коммерческих целях, что гарантирует патентную чистоту базовой платформы;
* не допускается использование объектов интеллектуальной собственности, защищённых патентами, авторскими правами и другими правами третьих лиц без соответствующего разрешения правообладателей, что предусмотрено частью IV Гражданского кодекса Российской Федерации;
* исключается использование результатов умственной деятельности, которые нарушают права на интеллектуальную собственность, охраняемую Федеральным законом № 98-ФЗ «О коммерческой тайне» и другими нормативными актами, регулирующими защиту интеллектуальной собственности.

## Требования к перспективам развития

Разрабатываемая система СОВД-Spark должна быть спроектирована с учетом следующих перспектив развития:

* переход на потоковую обработку данных;
* интеграция с MLOps платформой;
* развитие системы оркестрации;
* контейнеризация и переход на Kubernetes.

# Состав и содержание работ

В рамках разработки программного продукта в соответствии с настоящим документом требуется выполнить перечень работ, представленных на таблице 1.

Таблица 1 – Состав и содержание работ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № этапа | Наименование этапа | Длительность | Состав работ | Результат |
| 1 | Составление ТЗ | 1 неделя | Разработка и утверждение технического задания | Техническое задание |
| 2 | Настройка рабочего окружения | 1 неделя | Подготовка и настройка рабочего окружения | Подготовленное к написанию кода рабочее место |
| 3 | Проектирование | 2 недели | Разработка и утверждение структуры программы и БД. | Утверждённая структура программы |
| 4 | Реализация структуры базы данных | 1 недели | Реализация структуры БД для её использования в разрабатываемой программе | Структура БД |
| 5 | Программная реализация | 3 недели | Реализация программы в соответствии с требованиями, указанными в ТЗ | Программа, соответствующая требованиям, указанным в ТЗ |
| 6 | Тестирование программы | 1 неделя | Тестирование программы | Найденные ошибки в работе программы |
| 7 | Доработка программы | 1 недели | Устранение ошибок и недочётов, выявленных на этапе тестирования | Программа с исправленными ошибками и недочётами |
| 8 | Подготовка сопутствующей документации | 1 неделя | Написание отчёта по проекту | Отчёт по проекту |
| 9 | Приемо-сдаточные процедуры | 1 неделя | Сдача результатов работ, проведение испытаний результатов разработки, занесение результатов в ведомость | Оценка в ведомости |

# Требования к документированию

Состав программной документации должен включать в себя:

* техническое задание – должно содержать требования к разработке системы СОВД-Spark, ее цель, задачи, сроки и назначение. Документ включает в себя описание предметной области, анализ аналогов, а также детальные требования к функциональным характеристикам, используемым технологиям, надежности, безопасности и результатам работы;
* пояснительная записка, содержащая описание результатов выполненных работ в процессе разработки;
* исходный код программы.

Всё вышеперечисленные документы должны быть написаны, с учетом, следующим требований:

* СТП ВятГУ 101–2004;
* ГОСТ 34.602–2020.

# Требования к приемно-сдаточным процедурам

Испытания проводятся комиссией, в качестве заказчика, которым является Колледж ФГБОУ ВО «Вятский государственный университет», представленный Долженковой Марией Львовной, преподавателем по УП.03 Учебная практика.

Во время испытаний комиссия проверяет работу программы в соответствии со следующими позициями:

* Набор функциональных тестов;
* Корректное функционирование заданных в техническом задании функций;
* Возможность функционирования на ПК с указанными минимальными системными требованиями.

Комиссии должны быть представлены эксплуатационные документы, разработанная программа и доклад.

Структура доклада должна отражать следующие вопросы разработки:

* краткое описание задачи;
* результаты рассмотрения предметной области, аналогов, описание проблематики;
* описание этапа проектирования, возникавших проблем и путей их решения;
* выводы по результатам работы.