TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Информационных технологий и робототехники
Направление подготовки 09.04.04 Программная инженерия
Отделение школы (НОЦ) Информационных технологий

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
Разработка протокола передачи данных для системы управления потоками данных

УДК 004.415:004.724.2:519.872

Студент

Группа	ФИО	Подпись	Дата
8ПМ7И	Кайда Анастасия Юрьевна		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Савельев Алексей Олегович	к.т.н.		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ОСГН ШБИП	Потехина Нина Васильевна			

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ООД ШБИП	Горбенко Михаил Владимирович	к.т.н.		

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Губин Евгений Иванович	к.ф.-м.н.		

Томск – 2019 г.

TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа __Информационных технологий и робототехники_____
Направление подготовки (специальность) __09.04.04 Программная инженерия_____
Отделение школы (НОЦ)__Информационных технологий_____

УТВЕРЖДАЮ:
Руководитель ООП
_____ _____ ___Губин Е.И.__
(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Магистерской диссертации

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8ПМ7И	Кайда Анастасия Юрьевна

Тема работы:

Разработка протокола передачи данных для системы управления потоками данных	
Утверждена приказом директора (дата, номер)	1436/С

Срок сдачи студентом выполненной работы:	06.06.2019

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	1. База Научных Знаний (DKB), 2. Система интеграции MInT MS, 3. Образцы метаданных, 4. Авторизованный доступ к сервисам CERN (LXPLUS), 5. Техническое задание.
Перечень подлежащих исследованию, проектированию и разработке вопросов	1. Анализ существующих решений, протоколов транспортного и прикладного уровней, а также программных брокеров сообщений; 2. Анализ и описание архитектуры системы интеграции MInT MS; 3. Формализация протокола передачи данных;

	4. Описание сценариев взаимодействия компонентов DKB и MInT MS, 5. Разработка базовой сигнальной версии протокола; 6. Оценка производительности предложенного решения; 7. Оценка конкурентоспособности разработки, расчет затрат на проведение исследования; 8. Анализ условий труда исполнителей проекта.
Перечень графического материала	1. DFD-диаграмма потоков данных в цепочках ETL-процессов 2. Диаграммы последовательности взаимодействия супервизора и исполнителей 3. Схема коммуникации с помощью сигнальной версии протокола

Консультанты по разделам выпускной квалификационной работы

(с указанием разделов)

Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Потехина Нина Васильевна
Социальная ответственность	Горбенко Михаил Владимирович
Раздел на английском языке	Диденко Анастасия Владимировна

Названия разделов, которые должны быть написаны на русском и иностранном языках:

Аналитический обзор научной, нормативной и технической документации

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	01.03.2019

Задание выдал руководитель / консультант (при наличии):

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Савельев Алексей Олегович	к.т.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ПМ7И	Кайда Анастасия Юрьевна		

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Школа Информационных технологий и робототехники
Направление подготовки (специальность) 09.04.04 Программная инженерия
Уровень образования магистратура
Отделение Информационных технологий
Период выполнения весенний семестр 2018/2019 учебного года

Форма представления работы:

Магистерская диссертация

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	06.06.2019

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
10.04.2019	Раздел 1. Аналитический обзор научной, нормативной и технической документации	15
17.04.2019	Раздел 2. Архитектура системы управления потоками данных	20
24.04.2019	Раздел 3. Протокол передачи данных для системы управления потоками данных MInT MS	20
06.05.2019	Раздел 4. Результаты	20
08.05.2019	Раздел 5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
16.05.2019	Раздел 6. Социальная ответственность	10

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Савельев Алексей Олегович	к.т.н.		

СОГЛАСОВАНО:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Губин Евгений Иванович	к.ф.-м.н.		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
8ПМ7И	Кайда Анастасия Юрьевна

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Магистр	Направление/специальность	09.04.04 Программная инженерия

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:	
1. *Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих*	*Стоимость материальных ресурсов определялась согласно прейскурантам компаний* *Оклад руководителя – 33664 р.* *Оклад инженера – 21760 р.*
2. *Нормы и нормативы расходования ресурсов*	*Накладные расходы 16 %;* *Районный коэффициент 30%;* *Норма амортизации ПЭВМ 33,33%;* *Норма амортизации ПО 20%*
3. *Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования*	*Коэффициент отчислений на уплату во внебюджетные фонды 30%*
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. *Оценка коммерческого и инновационного потенциала НТИ*	*Анализ потенциальных потребителей результатов исследования, оценка качества и перспективности проекта по технологии QuaD, SWOT-анализ*
2. *Разработка устава научно-технического проекта*	*Инициация проекта: определение заинтересованных сторон проекта, целей и результатов проекта*
3. *Планирование процесса управления НТИ: структура и график проведения, бюджет, риски и организация закупок*	*План проекта, определение трудоемкости выполнения работ, разработка графика проведения научного исследования, расчет бюджета разработки*
4. *Определение ресурсной, финансовой, экономической эффективности*	*Описание потенциального эффекта*
Перечень графического материала *(с точным указанием обязательных чертежей):*	
1. *Оценочная карта для QuaD-анализа разработки* 2. *Матрица SWOT разработки* 3. *Иерархическая структура работ проекта* 4. *Диаграмма Ганта* 5. *График проведения разработки* 6. *Бюджет затрат* 7. *Основные мероприятия по снижению рисков*	

Дата выдачи задания для раздела по линейному графику	01.03.2019

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ОСГН ШБИП	Потехина Нина Васильевна			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ПМ7И	Кайда Анастасия Юрьевна		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8ПМ7И	Кайда Анастасия Юрьевна

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Магистр	Направление/специальность	09.04.04 Программная инженерия

Исходные данные к разделу «Социальная ответственность»:	
1) Характеристика объекта исследования (технология, алгоритм, методика) и область его применения.	Разработка представляет собой протокол передачи данных для внутренней системы управления потоками данных Базы Знаний, реализующий коммуникацию между элементами системы. Работа ведется в условиях учебной аудитории с использованием персонального компьютера и подключенных периферийных устройств.
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности. 1.1. Специальные (характерные для проектируемой рабочей зоны) правовые нормы трудового законодательства. 1.2. Организационные мероприятия при компоновке рабочей зоны.	– Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ – Федеральный Закон от 27.07.2006 N 152-ФЗ (ред. от 25.07.2011) «О Персональных Данных» – ГОСТ 12.2.032-78 ССБТ
2. Профессиональная социальная безопасность. 2.1. Анализ вредных и опасных факторов, которые может создать объект исследования. 2.2. Анализ вредных и опасных факторов, которые могут возникнуть на рабочем месте. 2.3. Обоснование мероприятий по защите исследователя от действия опасных и вредных факторов.	Микроклимат: – СанПиН 2.2.4.548–96 – СанПиН 2.2.2/2.4.1340-03 Шум: – ГОСТ 12.1.003-2014 ССБТ – ГОСТ 12.1.029-80 ССБТ Освещенность, естественный и искусственный свет: – СанПиН 2.2.1/2.1.1.1278–03 Поражение электрическим током: – ГОСТ 12.1.019-2017 Умственное перенапряжение: – ТОИ Р-45-084-01
3. Экологическая безопасность. 3.1. Анализ влияния объекта исследования на окружающую среду.	Несущими угрозу окружающей среде являются следующие объекты: – Люминесцентные лампы. – Компьютерное оборудование.

3.2. Анализ влияния процесса исследования на окружающую среду. 3.3. Обоснование мероприятий по защите окружающей среды.	Утилизацию следует проводить согласно: – ГОСТ Р 56397-2015 – ГОСТ 12.3.031-83
4. Безопасность в чрезвычайных ситуациях. 4.1. Анализ вероятных ЧС, которые может инициировать объект исследований. 4.2. Анализ вероятных ЧС, которые могут возникнуть на рабочем месте при проведении исследований. 4.3. Обоснование мероприятий по предотвращению ЧС и разработка порядка действия в случае возникновения ЧС.	Наиболее вероятной ЧС является пожар в здании, в т.ч. из-за короткого замыкания электропроводки. Необходимо принять ряд предупреждающих мер. В случае возникновения ЧС, вызвать противопожарную службу, эвакуировать людей и, согласно плану эвакуации, покинуть помещение. Для тушения локальных очагов возгорания возможно использование огнетушителей типа ОУ-5.
Перечень графического материала:	
1. План эвакуации учебного корпуса №22 (КЦ ТПУ)	

Дата выдачи задания для раздела по линейному графику	01.03.2019

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ООД ШБИП	Горбенко Михаил Владимирович	к.т.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ПМ7И	Кайда Анастасия Юрьевна		

Планируемые результаты обучения

Код	Результат обучения
Общие по направлению 09.04.04 «Программная инженерия»	
Р1	Проводить научные исследования, связанные с объектами профессиональной деятельности
Р2	Разрабатывать новые и улучшать существующие методы и алгоритмы обработки данных в информационно-вычислительных системах
Р3	Составлять отчеты о проведенной научно-исследовательской работе и публиковать научные результаты
Р4	Проектировать системы с параллельной обработкой данных и высокопроизводительные системы
Р5	Осуществлять программную реализацию информационно-вычислительных систем, в том числе распределенных
Р6	Осуществлять программную реализацию систем с параллельной обработкой данных и высокопроизводительных систем
Р7	Организовывать промышленное тестирование создаваемого программного обеспечения
Профиль «Технологии больших данных»/ «Big data solutions»	
Р8	Исследовать и анализировать большие данные, создавать их модели и интерпретировать структуры данных в таких моделях
Р9	Понимать принципы создания, хранения, управления, передачи и анализа больших данных с использованием новейших технологий, инструментов и систем обработки данных в высокопроизводительных сетях
Р10	Применять теорию распределенной системы управления базами данных к традиционным распределенным системам реляционных баз данных, облачным базам данных, крупномасштабным системам машинного обучения и хранилищам данных

РЕФЕРАТ

Выпускная квалификационная работа 108 с., 14 рис., 23 табл., 48 источников, 2 прил.

Ключевые слова: интеграция данных, протокол передачи данных, гетерогенные данные, ETL, база знаний.

Объектом исследования является процесс разработки протокола передачи данных для системы управления потоками данных в Базе Научных Знаний эксперимента ATLAS.

Цель работы – формализация и разработка базовой сигнальной версии протокола передачи данных для системы управления потоками данных.

В процессе исследования был проведен анализ протоколов транспортного и прикладного уровней согласно модели OSI, а также существующие брокеры программных сообщений. Были рассмотрены общие вопросы обработки и хранения больших объемов данных.

В результате исследования были формализованы требования к протоколу передачи данных для системы управления потоками данных. Спроектирована архитектура системы управления потоками данных, разработаны сценарии взаимодействия процессов-исполнителей. Реализована базовая сигнальная версия протокола.

Степень внедрения: данная работа проводилась в рамках грантов РФФИ и РНФ, на данном этапе, по результатам выполненных работ, оформляется акт о внедрении.

Область применения: научные мегапроекты в различных областях знаний, в частности, в области физики высоких энергий (эксперименты CERN).

Экономическая эффективность/значимость работы: данная разработка обеспечивает автоматизацию процесса сбора и обработки метаданных из внешних хранилищ с последующей загрузкой в централизованное хранилище Базы Научных Знаний, что является одной из задач по обеспечению инструментами автоматизации сбора данных и эффективного поиска метаданных учеными-физиками.

В будущем планируется расширение количества сигналов протокола, реализация пакетной обработки данных для всех типов процессов, параллелизация потоков обработки данных.

Оглавление

Перечень условных обозначений

БАК	Большой адронный коллайдер
БД	База данных
AMI	ATLAS Metadata Interface
AMQP	Advanced Message Queuing Protocol
ATLAS	A Toroidal LHC ApparatuS
CDS	CERN Document Server
DKB	Data Knowledge Base
ETL-процесс	от «Extract, Transform, Load», процесс извлечения, преобразования и загрузки данных
HOL	Head-of-line
HTTP	HyperText Transfer Protocol
MInT MS	Metadata Integration and Topology Management System
PDU	Protocol Data Units, единица измерения информации (данных), которой оперирует протокол
pyDKB	от Python Data Knowledge Base, Python-библиотека для DKB
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
STOMP	Streaming Text Oriented Messaging Protocol
SCTP	Stream Control Transmission Protocol

Введение

Современные научные эксперименты класса «мегасайенс», в т.ч. на уникальных научных установках – это крупные дорогостоящие международные проекты, сопряженные с вопросами сбора, обработки и хранения сверхбольших объемов данных. Одной из таких уникальных научных установок является Большой адронный коллайдер (БАК) и установленные на нем детекторы – ATLAS, ALICE, LHCb, CMS и т.д. Эксперимент ATLAS – A Toroidal LHC ApparatuS, – проводимый на одноименном детекторе, является крупнейшим экспериментом Европейской организации по ядерным исследованиям (CERN) и одним из лидирующих современных научных экспериментов по уровню интенсивности обработки данных в мире.

Решение задачи упрощения поиска и автоматизации процессов сбора, обработки и хранения метаданных для гетерогенных вычислительных инфраструктур требуют новых подходов, акцентирующих внимание на работе со сверхбольшими массивами данных. В рамках научной коллаборации ТПУ и НИЦ «Курчатовский институт» для решения поставленной задачи было принято решение о создании единой Базы Научных Знаний (DKB) научного эксперимента ATLAS.

Неотъемлемой частью DKB является система интеграции метаданных, отвечающая за извлечение, преобразование и загрузку данных, а также за управление потоками данных в системе.

Целью данной магистерской диссертации является разработка протокола коммуникации для внутренней системы управления потоками данных DKB.

1 Аналитический обзор научной, нормативной и технической документации

За 25 лет существования эксперимента ATLAS сложилась сложная цифровая инфраструктура, в том числе включающая в себя хранилища метаданных разных типов: информация о научных данных, полученных с экспериментальной установки или смоделированных с помощью специального ПО, информация о процессах производства и анализа данных, о публикациях сообщества и т.д. Ввиду отсутствия средств автоматизации поиска связанных наборов метаданных из различных баз (как реляционных, так и нереляционных), возникает необходимость осуществления поиска вручную, требующего от ученых больших временных затрат. Оптимизация процесса путем использования заранее сформированных мультизапросов (запросов, включающих обращение более чем к одному источнику информации) не является решением на сверхбольших объемах гетерогенных данных, поскольку выполнение такого рода запросов занимает значительное время, а их модернизация и адаптация для новых задач требует значительных усилий. Также, в условиях вероятного возникновения ограничения уровня доступа к той или иной базе данных для конкретного пользователя, выполнение определенных запросов может оказаться невозможным без использования прокси-систем с достаточными правами доступа. Система интеграции метаданных DKB, призванной решить задачу поиска и автоматизации процессов сбора, обработки и хранения метаданных, требует разработки специального протокола. В данном разделе освещаются общие вопросы обработки и анализа больших данных, а также протоколы передачи данных транспортного и прикладного уровней и программные брокеры сообщений, с использованием одного из которых описывается предложенное решение.

1.1 Общие вопросы обработки и анализа больших данных

В современном мире скорость роста объемов данных породила понятие больших данных. Под этим понятием подразумеваются массивы данных таких

объемов, которые невозможно обработать традиционными способами в силу ряда причин:

1) Ограниченность вычислительных ресурсов.

2) Сложность детектирования бесконечных процессов обработки данных (процессов, «зависших» из-за нехватки ресурсов и заблокировавших уже занятые ими ресурсы на неопределенное время).

3) Ограниченная пропускная способность традиционных систем (наличие т.н. «бутылочных горлышек»).

Ключевыми признаками являются 3V – объем, вариативность, скорость (от англ. volume, variety, velocity). В некоторых случаях, добавляется 4-й параметр – достоверность (veracity).

Большие объемы данных требуют новых подходов касаемо вопросов их обработки, анализа и хранения. Вместо одной машины используются вычислительные кластеры, формирующие сложные гетерогенные вычислительные инфраструктуры. В то же время, постоянно возрастающая скорость генерирования новых данных предъявляет высокие требования к пропускной способности как систем хранения, так и систем обработки данных.

Вариативность данных, в числе прочего, проявляется в их неоднородности. Работа с данными в условиях их гетерогенности является одной из сложнейших задач, требующей индивидуального подхода к решению в каждом конкретном случае [1].

1.2 Протоколы передачи данных

Протоколом является соглашение, определяющее и регламентирующее процесс обмена данными между элементами системы.

Для выявления требований к разработке были проанализированы протоколы транспортного уровня (TCP, UDP, SCTP), а также протоколы прикладного уровня (HTTP, AMQP, STOMP) согласно модели OSI [2]. Стоит отметить, что важно проанализировать не только протоколы прикладного уровня, которые могли бы стать основой решения, но и наиболее

распространенные протоколы транспортного уровня, их принципиальные различия и области применения. Современные протоколы прикладного уровня строятся над протоколами транспортного уровня (например, HTTP поверх TCP).

TCP является протоколом транспортного уровня из стека TCP/IP. Протоколом TCP реализуется ретрансмиссия, предотвращающая утерю пакетов, что отличает его от UDP, который допускает потерю датаграмм. Данный протокол применяется на серверах, ориентированных на подключение (Connection-Oriented Concurrent Server). Для установки соединения используется трехступенчатое квитирование (рукопожатие – процесс обмена определенными флагами для установки соединения). Сервер обслуживает несколько клиентов одновременно, однако обслуживание происходит через единый поток данных. При запросе на подключение сервер генерирует процесс-наследник, работающий с отправившим запрос клиентом.

UDP – протокол транспортного уровня наряду с TCP. В отличие от TCP, UDP-соединение не является надежным, поскольку не гарантирует доставку всех датаграмм, однако широко используется в случаях, когда приоритетом является минимизация коммуникационной задержки и джиттера (фазового дрожания цифрового сигнала данных). Сервер обрабатывает клиентские запросы последовательно. Данный протокол широко используется в CIS-серверах (например, серверах онлайн-RPG игр) – утеря или нарушение последовательности доставки датаграмм не является критическим нарушением работы сервера. Напротив, отсутствие рентрансмиссии позволяет протоколу не задерживать процесс обмена данными [3].

SCTP. Протокол транспортного уровня. Отличается от TCP четырехступенчатым квитированием (рукопожатием), поддержкой многопоточности (в отличие от TCP, реализует асинхронный обмен данными с помощью нескольких потоков данных), а также реализацией отсутствующих в TCP механизмов предотвращения блокировок HOL. В данном протоколе потоком является последовательность пользовательских сообщений, в то время как в TCP потоком является последовательность байтов [4].

HTTP. Протокол прикладного уровня семейства TCP/IP, разработанный для обеспечения коммуникации между веб-браузером и веб-сервером. HTTP является протоколом без сохранения состояния: сервер не сохраняет никаких данных между двумя запросами, однако обеспечивает поддержку пользовательских сессий. Первая версия протокола – HTTP 0.9 – была разработана Тимом Бернерсом-Ли в виде однострочного протокола, однако актуальная версия – HTTP 1.1 – включает в себя не только строку запроса, но и заголовки, легко расширяемые любым клиентом или сервером. В силу расширяемости конструкция запроса становится довольно громоздкой – инициируемый браузером интернет-запрос несет в себе дополнительные 500-800 байт метаданных, что в условиях интенсивной обработки данных может стать проблемой. Стоит учесть, что 500-800 байт являются минимальной планкой [5].

STOMP – текст-ориентированный протокол прикладного уровня для связующего программного обеспечения, ориентированного на обработку сообщений. STOMP является протоколом, основанным на фреймах, смоделированных, в свою очередь, на основе HTTP. Каждый фрейм состоит из команды, а также набора заголовков и тела, наличие которых опционально. Несмотря на текст-ориентированность протокола, STOMP также поддерживает передачу бинарных сообщений. Кодировка сообщений, команд и заголовков по умолчанию – UTF-8.

STOMP-сервер смоделирован в виде набора адресатов, которым могут быть отправлены сообщения. Адресат представляет собой неопределенную строку, синтаксис которой варьируется в зависимости от конфигурации сервера. STOMP также не определяет семантику процесса доставки – обмена сообщениями – адресату, семантика варьируется от сервера к серверу и даже от адресата к адресату. Неопределенная семантика позволяет поддерживать широкий спектр серверов.

STOMP-клиент выполняет роль пользовательского агента, который может действовать в двух режимах (в том числе, одновременно):

В режиме продьюсера, отправляющего адресату на сервер сообщение с использование фрейма SEND;

В режиме консьюмера, отправляя фрейм SUBSCRIBE определенному адресату на сервер и получая от сервера сообщение с помощью фрейма MESSAGE.

Основными принципами STOMP являются простота использования и функциональная совместимость. Простота реализации как на клиентской, так и на серверной частях на широком перечне языков программирования позволяет снять некоторые ограничения архитектуры серверов, такие как имя адресата, специфика семантики и особенности реализации.

В общем виде фрейм представлен следующим образом:

```
COMMAND
header1:value1
header2:value2

Body^@
```

Каждый фрейм начинается со строки-команды, которая завершается сигналом конца строки (End-of-line, или EOL) и, опционально, управляющим символом возврата каретки. Заголовки используются с большинством команд и делятся на три основных типа: content-length, content-type, receipt (указывающий номер сообщения и адресата). Тело сообщения поддерживается только командами SEND, MESSAGE и ERROR.

Экранирование строковых литералов в C-подобных строках используется для кодирования любого возврата каретки, перевода строки или обозначения двоеточия, которые содержатся в заголовках, представленных в кодировке UTF-8. При декодировании заголовков фрейма должны быть распознаны следующие управляющие символы:

1) «\r» – возврат каретки,

2) «\n» – конец строки (End-of-line, EOL, line feed),

3) «\c» – двоеточие

4) «\\» – символ переноса строки « \ ».

Согласно спецификации протокола [6], любая неопознанная escape-последовательность должна вызвать фатальную ошибку протокола

AMQP – бинарный протокол прикладного уровня, разработанный по запросу JP Morgan Chase компанией iMatrix. На 2004-й год, обработка очередей сообщений и вопрос организации топологий процессов оставался открытым, что послужило движущей силой к созданию решения, обеспечивающего динамическое связывание данных в режиме реального времени от любого «издателя» к любому из «подписчиков», желающих получить данные.

Как и STOMP, AMQP базируется на фреймах, в которые входит обязательных 8-битный заголовок, в котором содержатся стандартные указания доставки, информирующие о том, как передать сообщение по AMQP сети, а также опциональные расширенный заголовок и тело. AMQP поддерживает пользовательские сессии и обеспечивает отслеживание состояния процесса передачи сообщений (в том числе в организованных цепочках обработки данных).

Спецификация AMQP подразделяет понятие «сообщения» на два: аннотированное сообщение и простое сообщение (без аннотирующего метаописания в начале и конце сообщения). При этом простое сообщение подразделяется на три секции: стандартные свойства, свойства приложения и данные приложения [7].

AMQP, равно как и STOMP, поддерживается ограниченным набором брокеров сообщений. В отличие от STOMP, AMQP имеет несколько конкретных реализаций в виде программных брокеров – например, RabbitMQ и Apache ActiveMQ. Оба протокола являются достаточно громоздкими решениями (в том числе STOMP, несмотря на свою относительную «легковесность») в условиях отсутствия необходимости маршрутизации.

1.3 Связующее программное обеспечение, ориентированное на обработку сообщений

Немаловажным при разработке протокола является анализ существующих решений среди программных брокеров сообщений (или диспетчеров очередей), а также поддержки ими существующих протоколов прикладного уровня.

Большинство брокеров сообщений реализуют паттерн Producer-Consumer (генератор-подписчик, или продьюсер-консьюмер).

Были рассмотрены три различных широко используемых брокера сообщений, два из которых являются реализацией AMQP (RabbitMQ и Apache ActiveMQ).

RabbitMQ является горизонтально масштабируемым программным брокером сообщений, имеющим ряд ограничений при разбиении топиков (категорий) сообщений на партиции (фрагменты). Данный брокер не поддерживает многоразовое чтение одного сообщения и хранит сообщение до первого прочтения. Для работы RabbitMQ требуется среда выполнения (runtime) Erlang [8].

Apache ActiveMQ, как и RabbitMQ, поддерживает протоколы STOMP и AMQP. Одним из преимуществ данного брокера является реализация клиентских библиотек на большинстве популярных языков программирования (Python, Java, C++ и т.д.). Данный брокер обеспечивает надежную доставку сообщений, однако демонстрирует крайне низкие показатели скорости передачи сообщений при интенсивном потоке данных, что является критически важным фактором выбора брокера для системы с интенсивной обработкой больших объемов данных. Также ActiveMQ, как и RabbitMQ, реализует одноразовое чтение сообщения и не хранит сообщения для последующего использования.

Apache Kafka. Горизонтально масштабируемый распределенный программный брокер сообщений. В отличие от RabbitMQ, отличается высокой производительностью при работе со сверхбольшими объемами данных. Более того, Apache Kafka включает в себя API для реализации интеграции данных

(Kafka Connect) и потоковой обработки сообщений (Kafka Streams). Стоит отметить, что Apache Kafka имеет критическую зависимость от Apache ZooKeeper [9] – централизованного сервиса для распределенных систем и хранилищ данных типа ключ-значение.

Среди существующего программного обеспечения, для решения задачи интеграции метаданных в DKB, разработчиками предпочтение отдается Kafka по причине того, что данный брокер подходит в качестве основы для построения архитектуры, реализующей быстрый обмен данными в потоковом режиме с возможностью многоразового чтения сообщений. При этом разработчики акцентируют внимание на существовании логов – Kafka не удаляет сообщение после первого прочтения. Напротив, множество подписчиков одного топика может получить полный лог и обработать его (или выполнить повторную обработку, если процесс был завершен с ошибкой). В то время как часть программного брокера сообщений Kafka похожа на предшествующие варианты логикой работы, данный брокер также имеет дополнительный уровень – уровень агрегации логов. Также Kafka поддерживает разделение топиков на партиции для обеспечения масштабируемости, а также репликацию партиций по всему кластеру для большей надежности и отказоустойчивости системы.

1.4 Выводы по аналитическому разделу

Проблема обработки и хранения больших данных в распределенных гетерогенных вычислительных инфраструктурах остается актуальной и по сей день в силу ряда причин. В первую очередь, это невозможность использования традиционных подходов. Несмотря на разнообразие программных брокеров сообщений в ряду связующего программного обеспечения, ориентированного на обработку сообщений, наиболее популярные брокеры, такие как RabbitMQ и Apache ActiveMQ, не поддерживают многоразовое чтение сообщений и, соответственно, не включают в себя встроенную систему логирования, в отличие от Apache Kafka.

Наиболее известные протоколы прикладного уровня, – STOMP и AMPQ – ориентированные на обработку сообщений, не поддерживаются Apache Kafka. Данный вывод подчеркивает необходимость создания собственного способа коммуникации между процессами внутри системы интеграции данных.

2 Архитектура системы управления потоками данных

2.1 Организация работы с данными в брокере сообщений Apache Kafka

Согласно предыдущему разделу, Apache Kafka [10] является распределенным программным брокером сообщений, реализующим паттерн producer/consumer. Это говорит о том, что из двух существующих подходов для связующего программного обеспечения, ориентированного на обработку сообщений, – обработки очереди сообщений, где каждое сообщение распределяется по определенным процессам, и т.н. «вещания» (broadcasting), когда весь пул сообщений доступен всем подписчикам – реализуется именно второй. Для последующего описания принципов работы данного брокера важно обозначить следующие ключевые термины:

1) Топик – категория сообщений в системе брокера Apache Kafka. Ближайший аналог топика – таблицы в реляционных БД.

2) Партиция – фрагмент топика. Любой топик Apache Kafka может быть разделен на партиции.

3) Продьюсер – одна из базовых сущностей Apache Kafka, отвечающая за размещение сообщений в топиках (функция записи). Высокоуровневое представление продьюсера показано на рисунке 1.

4) Консьюмер – одна из базовых сущностей Apache Kafka, отвечающая за чтение сообщений в топиках (функция чтения).

Данный брокер является мультиконсьюмерным, что позволяет параллельно взаимодействовать нескольким подписчикам с одним продьюсером. Каждый консьюмер может извлекать любой набор сообщений в топике (например, осуществляя запрос на полное чтение топика или ограничиваясь набором из n сообщений).

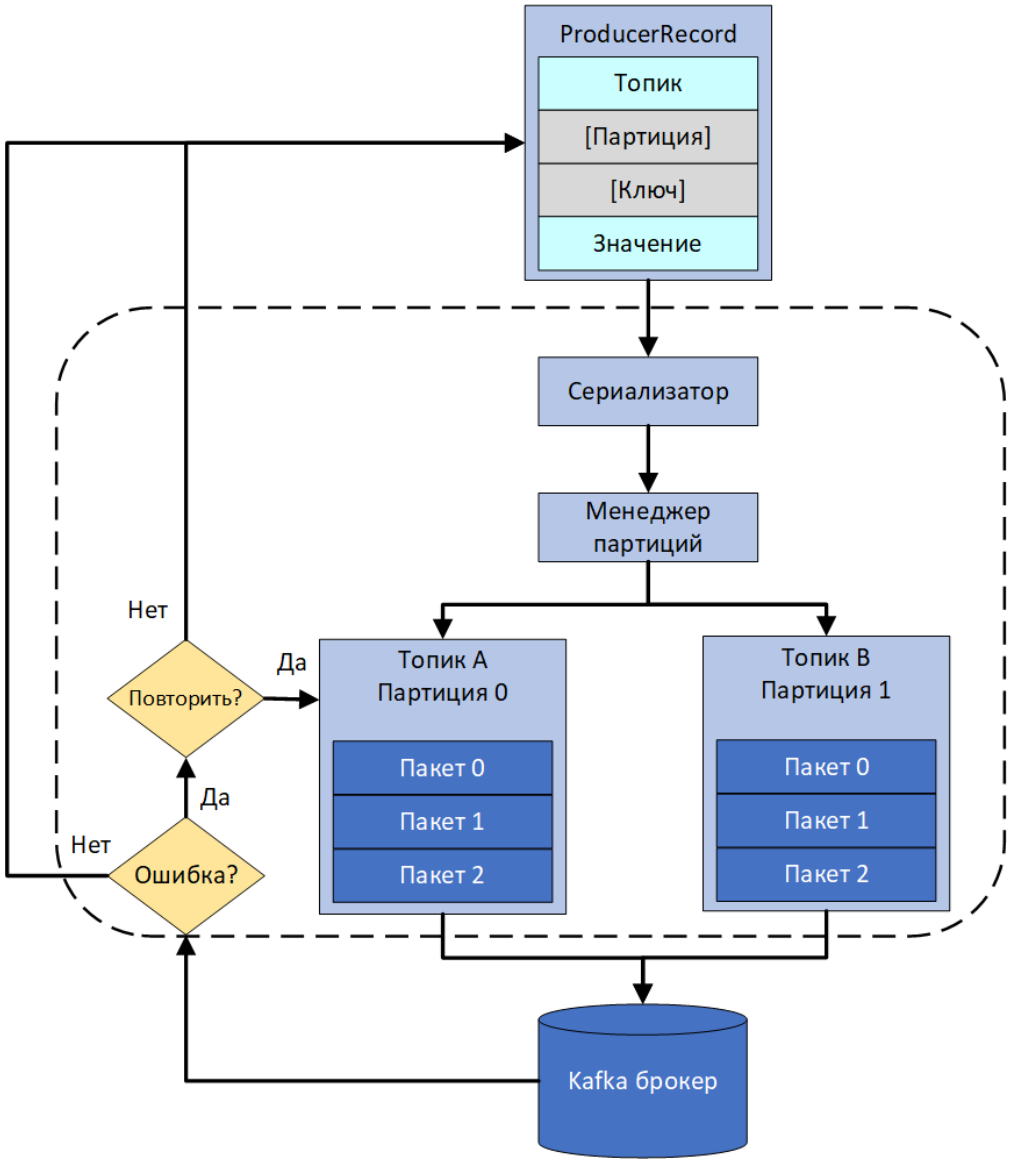

Рисунок 1. Высокоуровневое представление Kafka-продьюсера

Альтернативой использования продьюсера и консьюмера напрямую является Kafka Connect API (Sink/Source). Connect API является своеобразным решением «из коробки» для тех случаев, когда у внешних хранилищ данных уже существуют свои специфичные коннекторы, и реализует такие опции как параллелизация, обработка ошибок, поддержка различных типов данных (например, JSON) и стандартных REST API. Таким образом, использование Kafka Connect API ускоряет процесс разработки.

2.2 Построение топологий на базе Kafka Streams

Kafka Streams – клиентская библиотека для потоковой обработки данных, хранящихся в Kafka. Данная библиотека не имеет дополнительных внешних зависимостей, кроме, непосредственно, самого брокера Apache Kafka, что соответствует требованиям к системе. При горизонтальном масштабировании процессов обработки сообщений (с разделением на партиции) Kafka гарантирует строгую упорядоченность передаваемых сообщений.

Каждое приложение, поддерживающее потоковый режим – реализованное на базе Kafka Streams – выполняется как минимум одной топологией. При этом любая топология представляет собой направленный ациклический граф – серию выполняемых операций и переходов, через которые проходит каждое событие от ввода исходных до вывода конечных данных.

Стоит учитывать, что даже простой ETL-процесс является сложной комбинацией из Sink/Source коннекторов и Streams-приложения, которое, в свою очередь, содержит один или несколько узлов топологического графа преобразования данных. Помимо процессоров, выполняющих операции с данными, также существуют Source-узлы топологии, извлекающие данные из внешних топиков и передающие их на последующую обработку, а также Sink-узлы топологии, которые получают данные от последнего процессора топологии размещают подготовленные для загрузки в конечное хранилище данные в специальном топике, откуда, в свою очередь, происходит, посредством коннектора, процесс загрузки данных. Любая топология начинается с Sink-процессора и завершается Source-процессором, как продемонстрировано на рисунке 2.

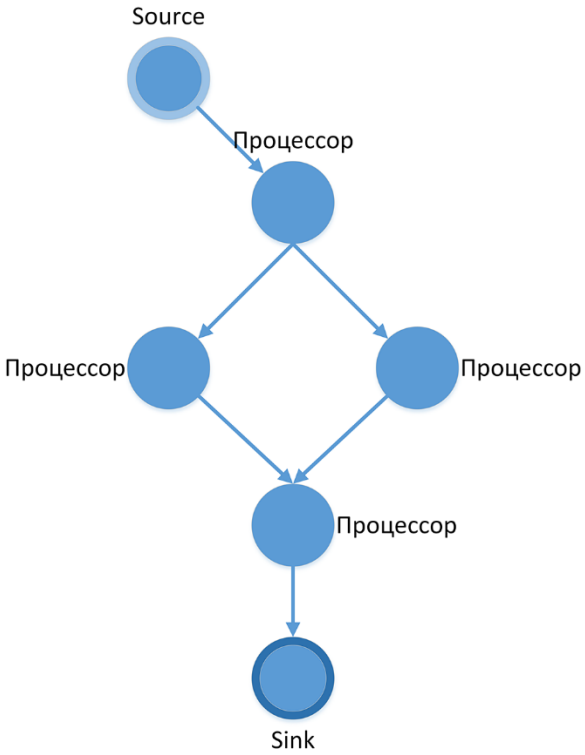

Рисунок 2. Пример базовой топологии

Масштабирование топологий реализуется за счет мультипоточного выполнения внутри одной сущности Kafka Streams. Аппарат Kafka Streams распараллеливает выполняемую топологию, разделяя ее на «таски» (от англ. task – задание) в соответствии с партициями исходного топика (source) и независимыми подграфами, автоматически выделяемыми в заданной пользователем топологии [9]. В рамках одного потока единовременно выполняется один «таск»; если количество сформированных «тасков» превышает количество потоков, задаваемое в конфигурации самого приложения, то в каждом потоке формируется очередь из «тасков», каждый из которых активируется для обработки одного сообщения, после чего снова переходит в режим ожидания своей очереди.

Помимо мультипоточного режима работы, масштабирование реализуется за счет механизма «групп подписчиков» (consumer group). Если запущено несколько экземпляров одного приложения, они регистрируются как участники общей группы подписчиков, и за каждым из них закрепляется набор партиций

исходного топика, с которыми каждый экземпляр приложения работает точно так же, как и в случае запуска одного экземпляра. Наборы партиций, закрепленные за отдельными экземплярами приложения, не пересекаются.

2.3 Система управления потоками данных MInT MS

MInT MS (аббревиатура от Metadata Integration and Topology Management System) – подсистема DKB, отвечающая за организацию топологий ETL-процессов по интеграции метаданных и управление ими. Полная архитектура MInT MS и DKB представлена на рисунке 3.

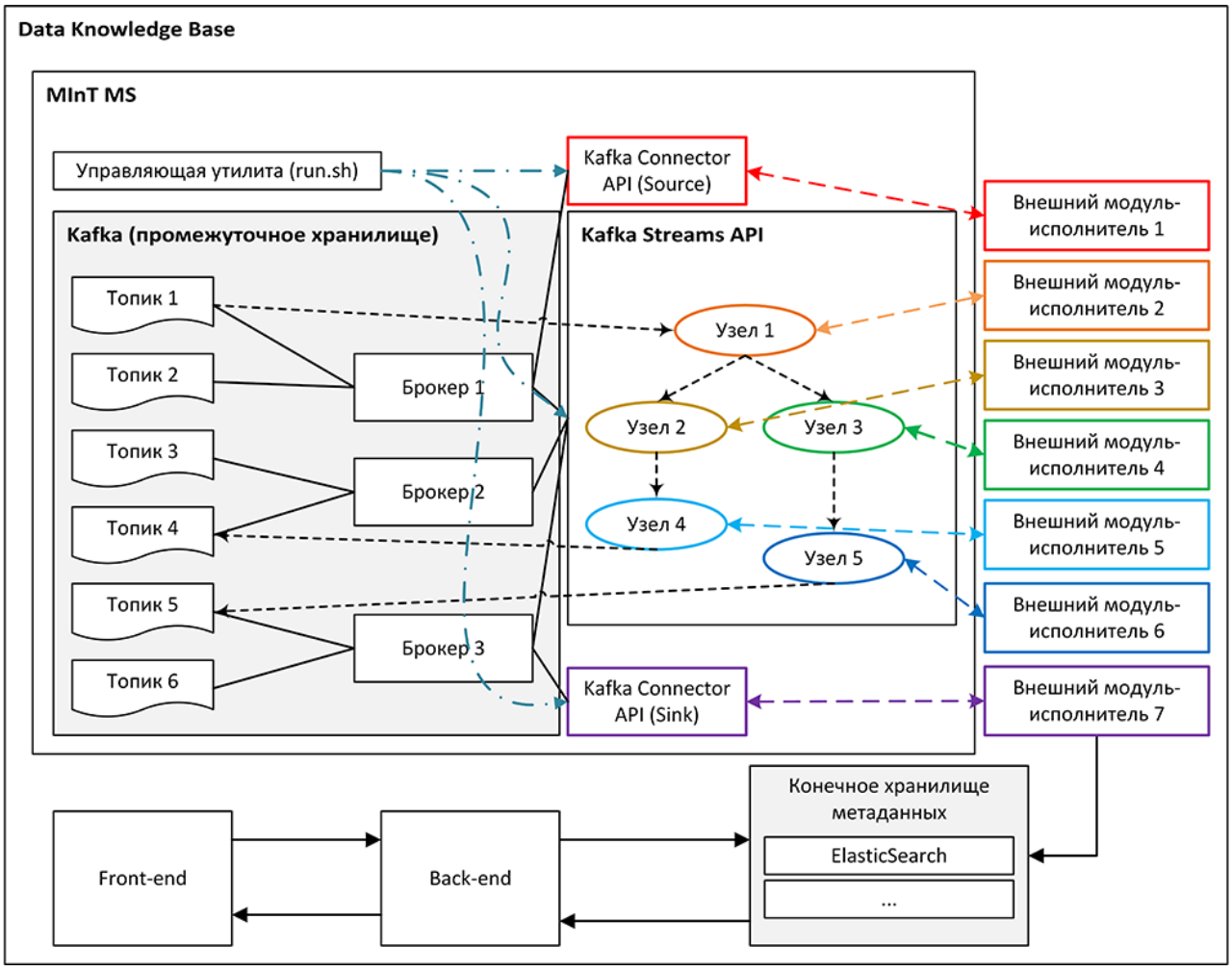

Рисунок 3. Архитектура системы управления потоками данных MInT MS для Базы Научных Знаний (DKB)

Поскольку хранилища данных существуют автономно, возникла необходимость реализации инструментов автоматизации сбора и обработки

метаданных с последующей загрузкой в конечное хранилище. В силу того, что перечень внешних хранилищ инфраструктуры научного проекта может варьироваться, зависимость между компонентами должна быть минимальной. Именно поэтому было принято решение реализации каждого из модулей извлечения данных, их обработки или загрузки в конечное хранилище в виде ETL-процессов. Несколько процессов образуют ETL-цепочки и, согласно топологии Kafka, каждая цепочка обработки начинается с процесса извлечения данных и завершается их загрузкой. При этом каждый процесс реализует только одну функцию.

На рисунке 4 продемонстрирован пример цепочек обработки данных из систем AMI и ProdSys с последующим извлечением метаданных из Rucio, а также ES Analytics.

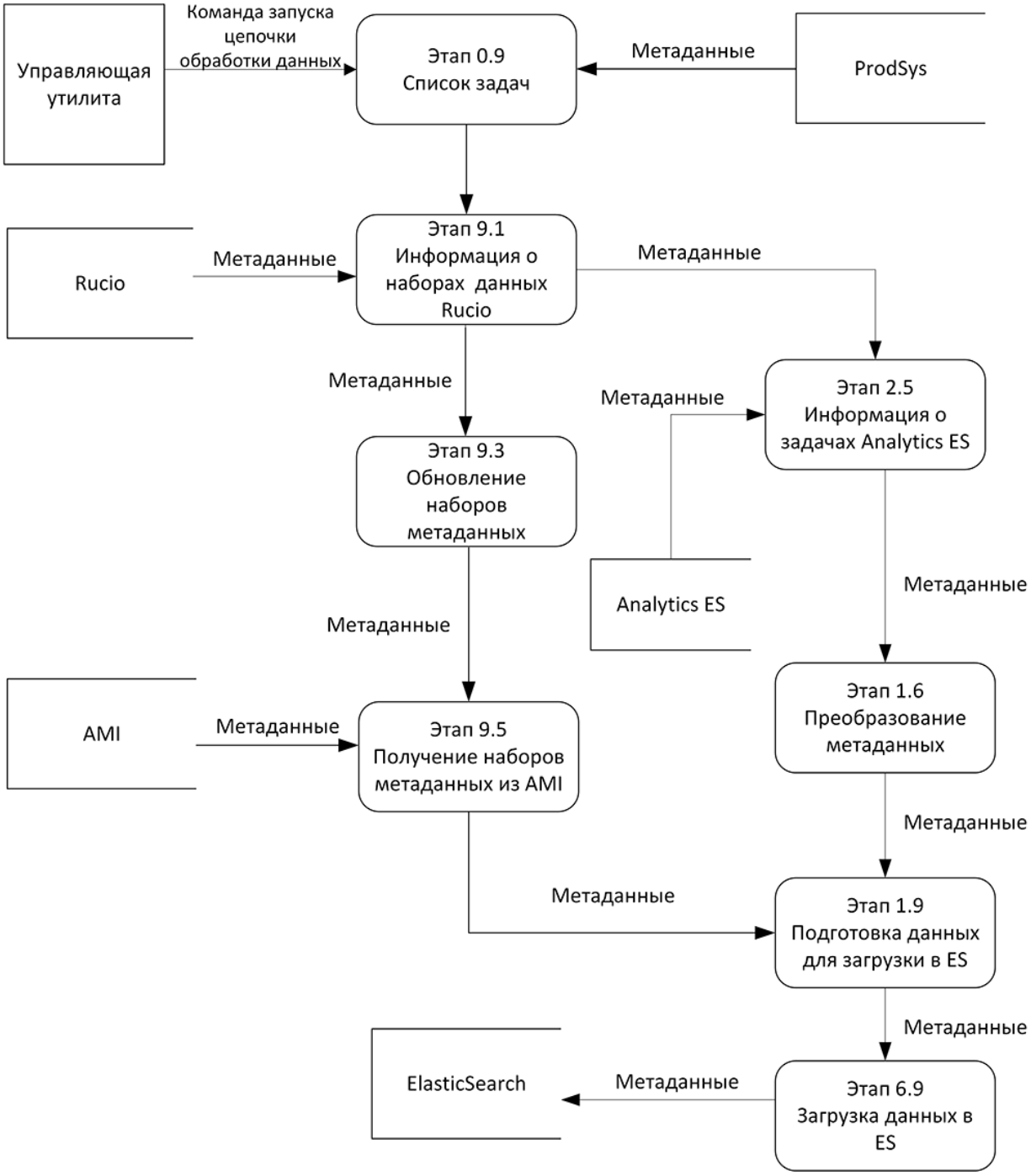

Рисунок 4. DFD-диаграмма ETL-процесса обработки метаданных

3 Протокол передачи данных для системы управления потоками данных MInT MS

3.1 Спецификация требований к протоколу передачи данных

Протоколом является соглашение, определяющее и регламентирующее процесс обмена данными между элементами системы. Протокол передачи данных для MInT MS является протоколом 7-го уровня согласно модели OSI (прикладной уровень), поскольку регламентирует сообщение между внешними программами-исполнителями и управляющей программой – супервизором. Единицей измерения информации (данных) являются сообщения, которыми обмениваются процессы.

Приоритетами разрабатываемого протокола являются: обеспечение целостности передаваемых данных, предотвращение скопления сообщений в каналах, минимизация накладных расходов на передачу сообщений и простота реализации. Основное назначение протокола – обмен текстовыми данными между двумя участниками без посредников.

Подобно TCP протоколу, разрабатываемый протокол должен соответствовать следующим требованиям:

1) Надежность. Каждое сообщение должно быть передано хотя бы один раз. Подтверждение получения сообщений, повторная передача (ретрансмиссия) в случае ошибки, а также соблюдение предустановленного лимита времени ожидания.

2) Упорядоченность и неизбыточность передачи сообщений. Первое отправленное сообщение поступает и обрабатывается первым (FIFO). Повторная отправка сообщения производится только в том случае, если оно еще не было обработано всеми ожидающими сообщение процессами.

Также были сформулированы требования:

1) Высокая скорость передачи данных.

2) Адаптивность. Возможность передачи пользовательских параметров.

3) Минимизация передачи дополнительных метаданных.

3.2 Описание протокола передачи данных

3.2.1 Определение терминов протокола

Сообщение – набор символов, не содержащий спецсимволов и заканчивающийся специальным разделителем – символом EOM; единица измерения информации (данных), или PDU, которыми обмениваются процессы.

Пакет сообщений – завершающаяся символом EOB последовательность однотипных сообщений, предназначенных для одновременной обработки одним и тем же процессом.

Супервизор – программа, отвечающая за инициализацию ETL-процесса и передачу данных между узлами работы с данными.

Исполнитель – программа, осуществляющая работу с данными и представляющая из себя логический узел ETL-процесса. Существует три типа исполнителей, соответствующих трем этапам ETL-процесса: извлечение данных (Source Connector), преобразование данных (Processor) и загрузка данных (Sink Connector).

Событие – изменение состояния процесса исполнителя: запуск или завершение.

Сигнал, или маркер – специальный символ или набор ASCII-символов, использующийся для осуществления коммуникации между программой-супервизором и программой-исполнителем.

End-of-message, или EOM – маркер конца сообщения.

End-of-process, или EOP – маркер завершения процесса.

End-of-Batch, или EOB – маркер конца пакета.

Топология – однонаправленный ациклический граф, состоящий из процессов преобразования метаданных в потоковом режиме.

3.2.2 Принятые к реализации соглашения

Кодировка сообщений, передаваемых с помощью протокола – UTF-8. Данная кодировка является своеобразным компромиссом среди прочих, обеспечивая компактность наравне с ASCII (в том случае, если файл содержит в себе только символы латиницы), но также может включать в себя любые другие символы Юникод, допуская некоторое увеличение размера сообщения.

За один раз на обработку может быть отправлено не более одного сообщения. Следующее может быть отправлено не раньше, чем будет получено подтверждение о завершении обработки сообщения, отправленного ранее.

Ответ исполнителя (в процессорах и Source-коннекторах) после процесса обработки может содержать неограниченное количество сообщений. Длина входных и выходных сообщений не ограничена.

Маркер является односимвольным. Такое решение было принято, в первую очередь, для уменьшения затрачиваемых вычислительных мощностей при определении сообщений с указанными пользовательскими маркерами. В том случае, если маркер не соответствует требованиям – переданный пользователем маркер не является односимвольным или является пустым в тех случаях, когда он таковым быть не может (маркер EOM), – вызывается обработчик исключений, записывает ошибку в логи, и программа завершается.

Некоторые процессы-исполнители для ускорения обработки данных могут функционировать в режиме «пакетной» обработки входных данных, при которой несколько сообщений обрабатываются одновременно. В текущей реализации пакетный режим предусмотрен только для Sink-коннекторов (в тех случаях, когда размер пакета равен 1, пакетный режим отключается). Реализация этого режима позволила значительно оптимизировать процессы загрузки данных в конечные хранилища с помощью интерфейсов массовой загрузки данных (bulk load). Source-коннекторы и процессоры работают с каждым сообщением отдельно.

3.2.2.1 Механизмы прерывания соединения

В текущей версии механизмы прерывания работы компонент MInT MS реализуются только посредством передачи процессу-супервизору сигнала SIGINT, при этом соединение с исполнителем прерывается в связи с закрытием канала передачи данных (Unix pipe). Принудительное завершение процессов фиксируется в логах. Для того чтобы обеспечить целостность передаваемых протоколом сообщений вне зависимости от этапа процесса интеграции метаданных в DKB в последующих версиях планируется реализация дополнительного маркера End-of-operation, передаваемого от супервизора процессу-исполнителю, сигнализирующего о намерении пользователя приостановить работу. После получения такого сигнала запускается сценарий завершения обработки данных таким образом, чтобы последнее сообщение было извлечено полностью и отправлено в нужный топик до завершения работы приложения.

3.2.2.2 Шифрование данных

Шифрование и дешифрование больших объемов данных – передаваемых с использованием протокола сообщений – требует дополнительных вычислительных ресурсов, чем может создать дополнительную нагрузку на вычислительный кластер. Было принято решение, что в условиях, предоставленных для развертывания DKB и ее компонентов, на базе вычислительных ресурсов CERN, строгое разграничение уровня доступа к данным (при условии соблюдения правил компьютерной безопасности CERN [11]) является достаточным условием для обеспечения безопасности передаваемых метаданных.

Также архитектура системы предусматривает возможность реализации шифрования на уровне сообщений путем шифровки/дешифровки каждого отправляемого сообщения с помощью существующих библиотек, таких как OpenSSL.

Дополнительное усиление шифрования может быть достигнуто при использовании асимметричного шифрования с передачей открытых ключей в начале каждой сессии, однако это потребует дополнительного вмешательства в инфраструктуру системы.

3.2.3 Сценарии коммуникации протокола

Исходя из модульной архитектуры системы, ETL-компоненты которой существуют независимо друг от друга, где каждый компонент выполняет только одну функцию из трех возможных (извлечение данных, их преобразование и загрузка в конечное хранилище), сценарии взаимодействия супервизора Kafka можно разделить на три типа по типу процесса-исполнителя:

1) Source-коннектор. При запуске исполнителя выполняется операция получения данных из внешнего источника. Полученные данные передаются супервизору в виде одного или нескольких сообщений, каждое из которых заканчивается маркером EOM. По завершении передачи данных исполнитель передает сигнал EOP и завершает свою работу. UML-диаграмма последовательности для разового получения данных из внешнего источника изображена на рисунке 4.

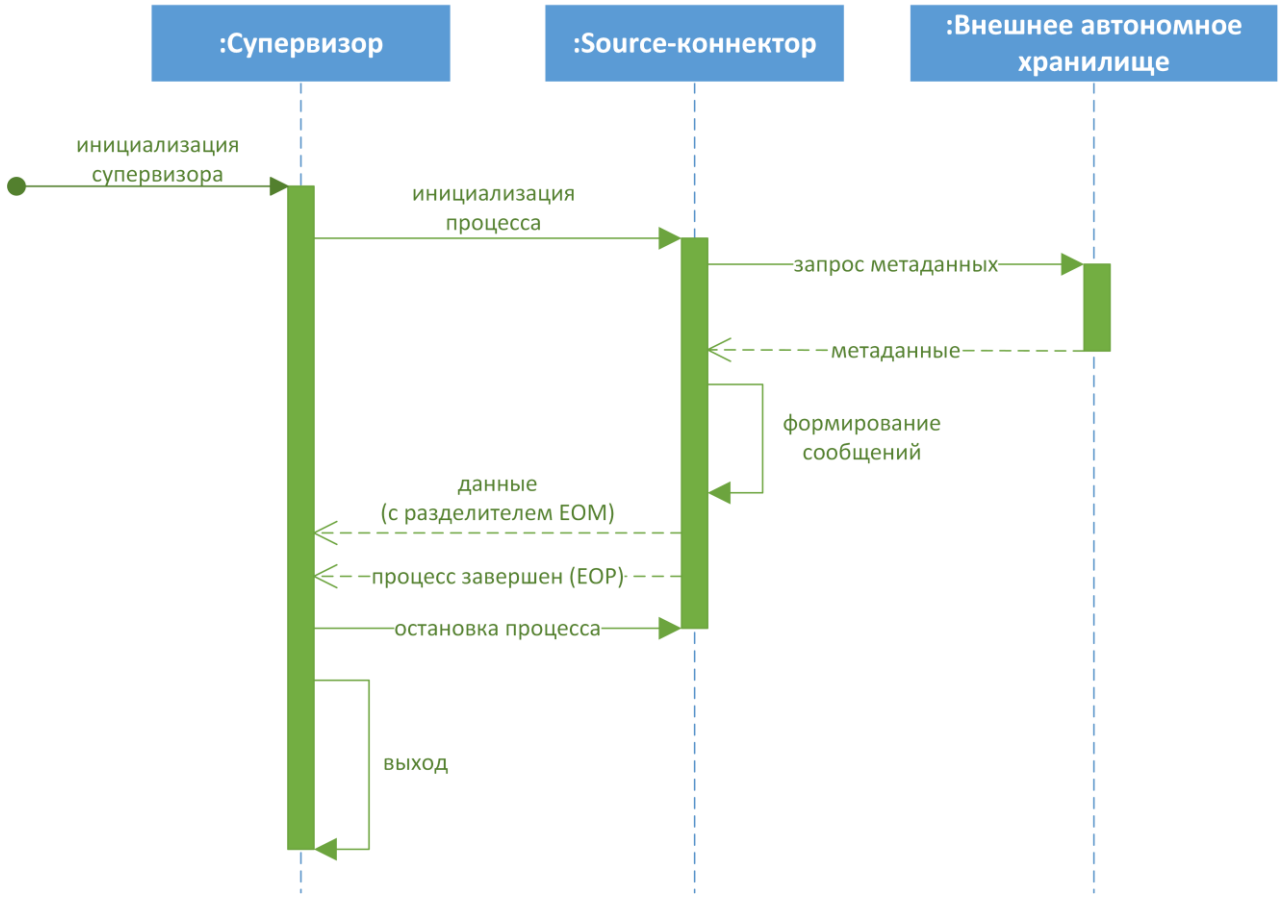

Рисунок 5. UML-диаграмма последовательности для ETL-процессов извлечения данных из внешнего хранилища с использованием Source-коннектора

2) Процессор. При запуске исполнитель переходит в режим ожидания входных данных. При появлении новых данных для обработки супервизор передает исполнителю одно сообщение, завершающееся маркером EOM, на обработку, и переходит в режим ожидания результатов обработки. Исполнитель выполняет обработку полученного сообщения и передает результаты обработки в виде нуля и более сообщений, каждое из которых завершается маркером EOM. После передачи последнего сообщения, передается также сигнал EOP. Супервизор, получив сигнал EOP, передает на обработку следующее сообщение. UML-диаграмма последовательности для обработки одного сообщения отображена на рисунке 5.

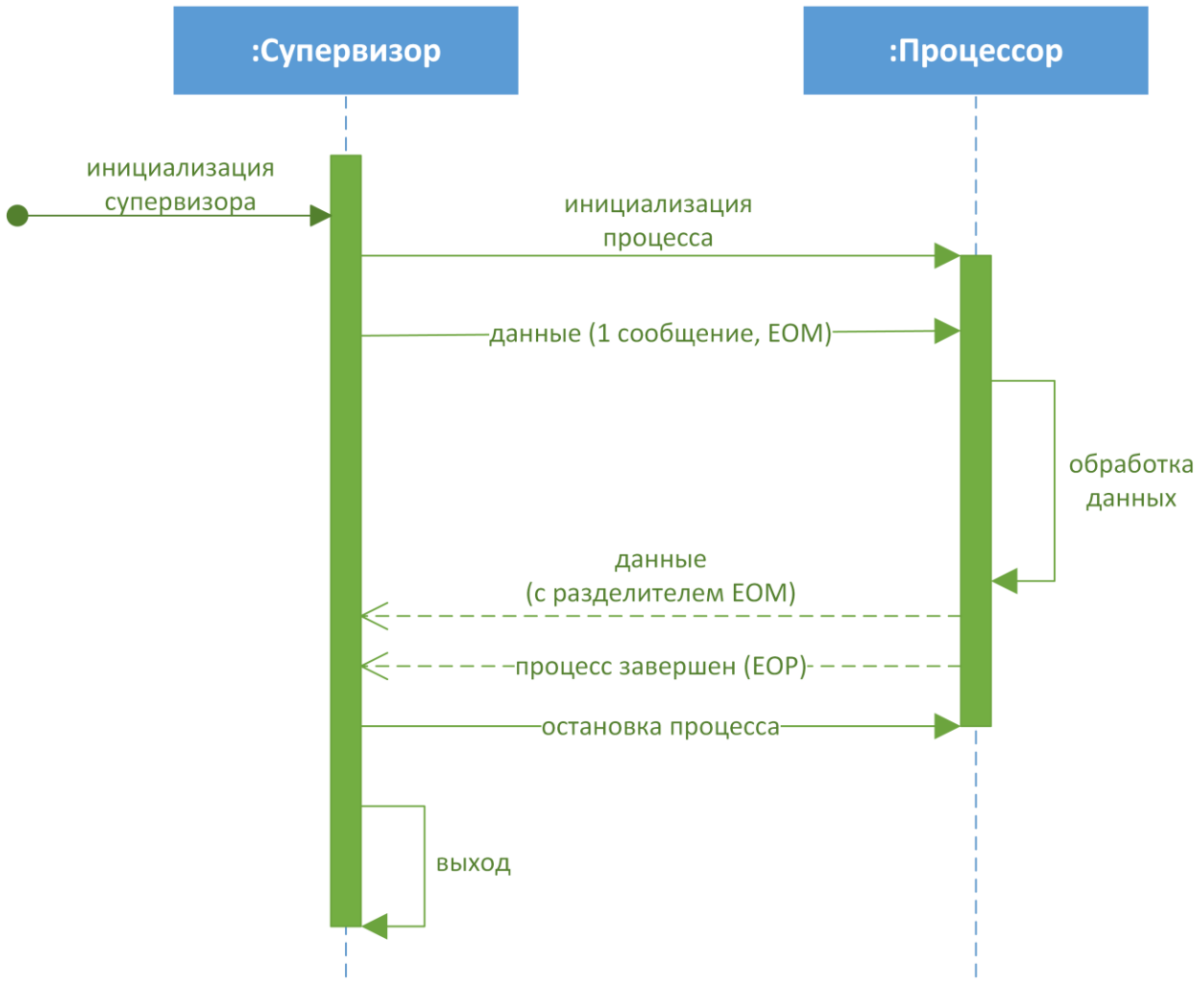

Рисунок 6. UML-диаграмма последовательности для ETL-процессов преобразования данных с использованием процессора

3) Sink-коннектор. При запуске исполнитель переходит в режим ожидания входных данных. При появлении новых данных для обработки супервизор передает исполнителю один пакет сообщений (каждое из которых завершается маркером EOM), завершающийся маркером EOB, на обработку, и переходит в режим ожидания завершения обработки. Исполнитель передает полученные данные на интерфейс массовой загрузки данных конечного хранилища Базы Научных Знаний, и по завершении передачи отправляет супервизору сигнал EOP. Получив сигнал EOP, супервизор передает исполнителю следующий пакет сообщений. UML-диаграмма последовательности для загрузки одного пакета отображена на рисунке 6.

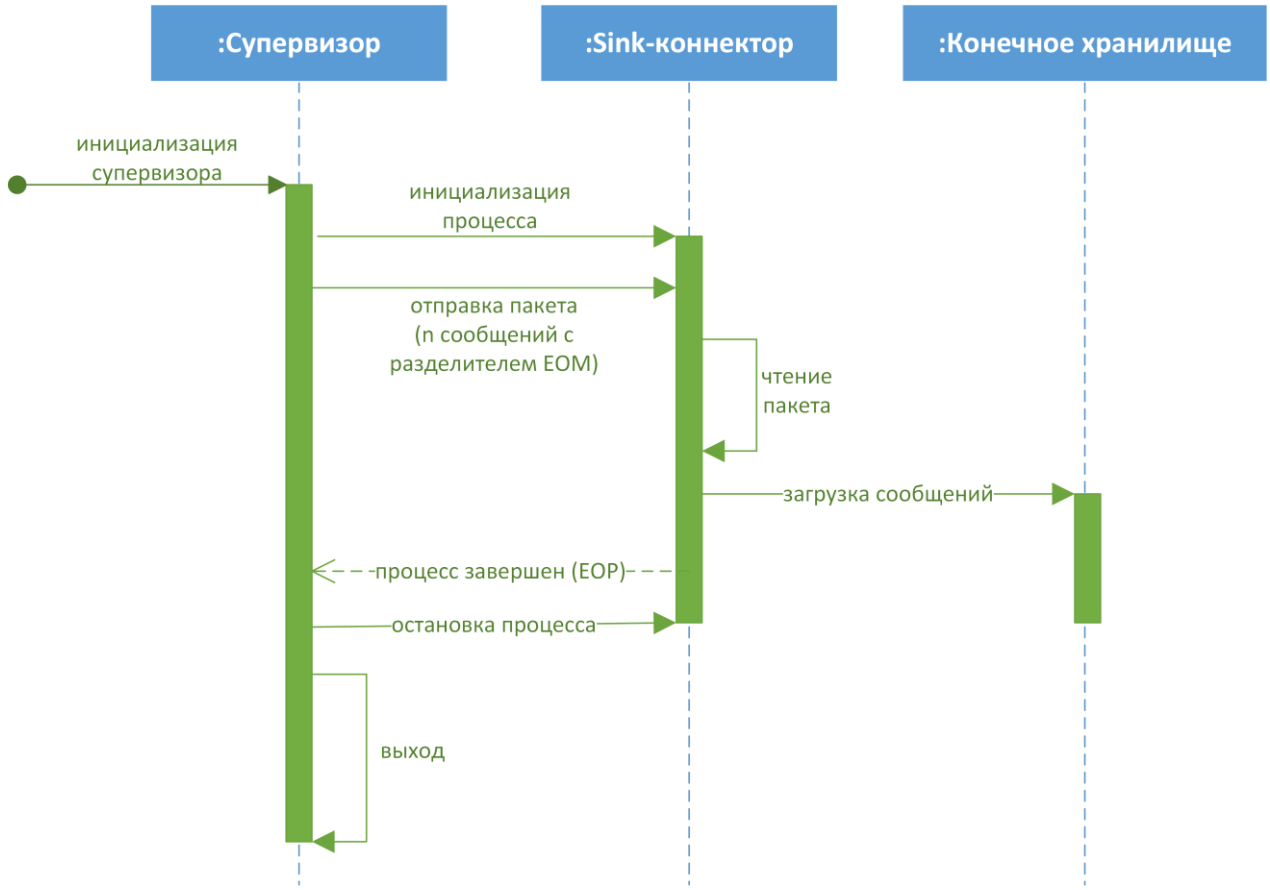

Рисунок 7. UML-диаграмма последовательности для ETL-процессов загрузки данных в конечное хранилище с использованием Sink-коннектора

Управляющая утилита запускает супервизор. Программа-супервизор (приложение, реализованное на основе базовых компонент библиотеки Apache Kafka) инициализирует ETL-процесс и передает конфигурацию специальных символов, которая будет использоваться в рамках сессии, исполнителям, заданным в конфигурации супервизора. Передача конфигурации осуществляется с помощью параметров командной строки при запуске исполнителей. Далее с заданной периодичностью Source Connector получает новые данные из внешнего источника и формирует из них сообщения для дальнейшей обработки в топологии процессоров-обработчиков. Здесь каждое сообщение последовательно проходит несколько этапов обработки (в соответствии с заданной топологией). После этого результаты обработки передаются в Sink Connector, который реализует загрузку новой информации во внутреннее хранилище DKB. Общая схема коммуникации Kafka-адаптера и внешнего

процесса преобразования данных с помощью сигнальной версии протокола представлена на рисунке 7.

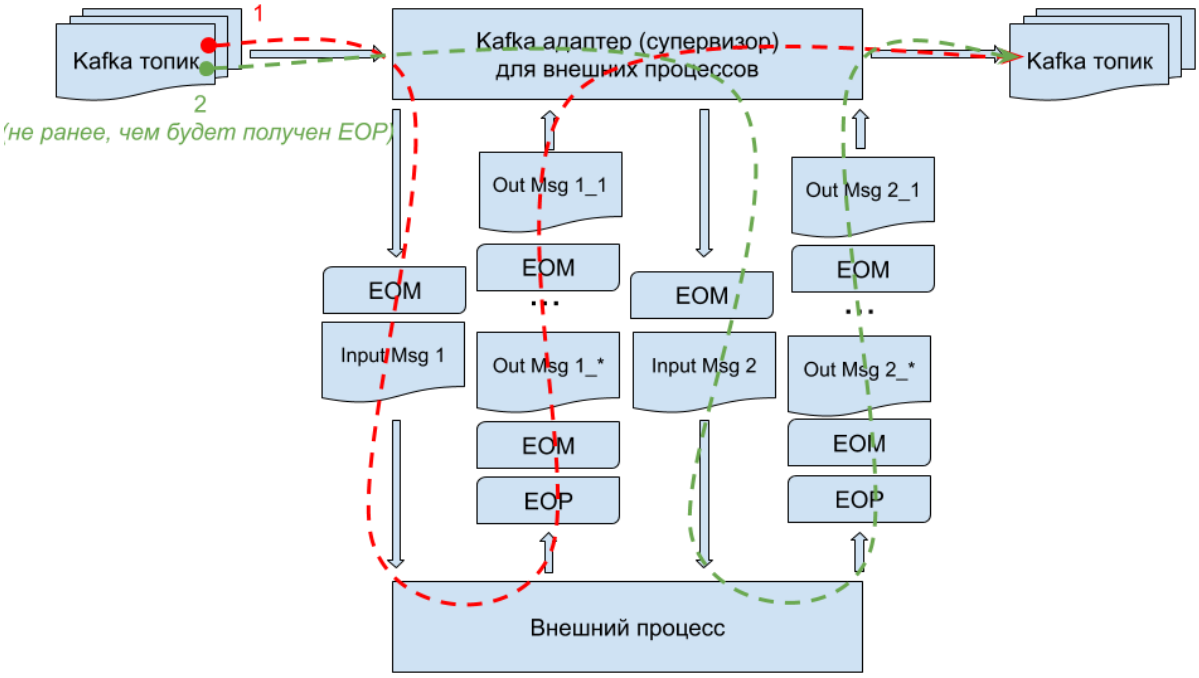

Рисунок 8. Схема коммуникации с помощью сигнальной версии протокола

3.3 Сигналы протокола передачи данных

Базовая версия сигнального протокола регламентирует процесс коммуникации программ при передаче сообщения или набора сообщений между супервизором и исполнителем, а также передачу информации супервизору о состоянии процесса-исполнителя (его завершении).

Типы сигналов и соответствующие им значения по умолчанию, выбранные в процессе формализации протокола, перечислены в таблице 1.

Таблица 1 – Типы сигналов и значения по умолчанию

Название	**Описание**	**Значение по умолчанию (ASCII)**	
		Потоковый режим	**Файловый режим**
EOM, End-Of-Message	конец передачи сообщения	NL (‘\n’)	NL (‘\n’)

EOP, End-Of-Process	конец передачи результатов работы программы-исполнителя	NUL (‘\0’)	-
EOB, End-Of-Batch	конец передачи пакета	ETB (‘\x17’)	ETB (‘\x17’)

Стоит отметить, что использование маркера конца пакета на данный момент реализовано только для Sink-коннекторов, однако планируется использование данного маркера и для остальных исполнителей. В будущем планируется расширение протокола маркером EOO (маркером завершения процесса, инициализированного пользователем – вместо прерывания KeyboardInterrupt). Также для ETL-модулей типа «Source Connector» предполагается введение дополнительного сигнального маркера GET (запроса на получение данных), который позволит не перезапускать программу-исполнителя для получения новой порции данных.

3.4 Обоснование выбора программных средств разработки

Выбор программных средств разработки обусловлен, в первую очередь, условиями, в которых была начата разработка DKB, и сформулированной концепцией. Проект претерпевал ряд изменений, в том числе переход с системы контроля версий Subversions на git. В силу того, что в CERN, на вычислительном кластере, используется Python 2.7, использование версий 3.x не является целесообразным. Python является одним из предпочитаемых языков среди разработчиков, участвующих в проектах CERN, если приоритетом является скорость разработки. Кроме того, Python обеспечивает простоту развертывания разрабатываемой системы за счет возможности реализации с минимальным количеством зависимостей от внешних модулей.

Поскольку данная разработка является компонентом большой системы, рассчитанной на конкретного конечного пользователя (ученых CERN), и

планируется в дальнейшем перенести всю систему на сервера CERN, стоит отметить, что разработка предназначена для использования в *nix системах. Данное требование, с одной стороны, накладывает некоторые ограничения, а с другой предоставляет ряд возможностей тестирования выходных данных при запуске цепочек процессов.

LXPLUS[12], сервис CERN, представляющий собой вычислительный кластер, предоставляет авторизованным пользователями доступ ко внутненней сети CERN и развернутым в ней сервисах (ElasticSearch, AMI и т.д.).

Для извлечения данных из внешних хранилищ и загрузки в конечное интегральное хранилище, ElasticSearch, были также разработаны специальные bash-скрипты.

Для создания прототипа в качестве основы для реализации высокоуровневого функционала ETL-системы была использована платформа для потоковой обработки данных Apache Kafka, обоснование выбора которой было представлено в предыдущих разделах [13].

3.5 Разработка механизмов чтения пользовательских маркеров в потоковом режиме

Большинство ETL-модулей написаны на языке Python. Для упрощения процесса их разработки основные функции, используемые в каждом таком модуле, включены в разрабатываемую библиотеку pyDKB. Данная библиотека также включает в себя класс, реализующий возможность передачи пользовательских параметров для любого из ETL-процессов обработки метаданных (в том числе, пользовательских маркеров).

Одной из проблем, с которой пришлось столкнуться в процессе разработки, являлась невозможность реализации полного функционала в pyDKB с использованием стандартных функций, а именно распознавания маркера, отличного от ASCII-символа новой строки (NL), в качестве разделителя для поступающих на вход сообщений. Функции read(), readline() и readlines() не предусматривают использование заданного пользователем разделителя, в чем

заключается одна из важных опций протокола передачи данных внутри DKB – пользователь должен иметь возможность конфигурировать при необходимости маркер EOM, а внешние процессы – распознавать маркер. Единственным допустимым в Python разделителем является непечатаемый символ новой строки – «\n». Пользовательская функция readline(), или custom_readline(), снимает ограничение языка программирования Python.

Второй проблемой в ходе реализации параметризованной функции чтения было устройство файловых дескрипторов unix-систем. По умолчанию все дескрипторы запускаются в режиме блокировки (blocking mode), при этом предустановленный размер буфера равен 4096 байт. Таким образом, процесс будет ожидать заполнения буфера, в то время как, с другой стороны, супервизор также будет ожидать ответа от исполнителя.

Для решения данной задачи, в первую очередь, используется функция poll() из модуля Python, select, реализованная через системный вызов функции poll() [14], обеспечивающей мониторинг файлового дескриптора f (где f - стандартный ввод STDIN) на возникновение флага события POLLIN, и производится установка неблокирующего ввода-вывода O_NONBLOCK.

```
poller = select.poll()
poller.register(f, select.POLLIN)
flags = fcntl.fcntl(f.fileno(), fcntl.F_GETFL)
fcntl.fcntl(f.fileno(),   fcntl.F_SETFL,   flags   |
os.O_NONBLOCK)
```

Данное решение позволит, не прибегая к посимвольному чтению (BUFSIZE=1), увеличивающему объем используемых ресурсов памяти, реализовать чтение с использованием пользовательского разделителя.

4 Результаты

4.1 Сравнительный анализ времени обработки сообщений

В ходе работы было проведено сравнение времени обработки сообщений с пользовательским разделителем, а также сравнение времени обработки управляющего символа NL стандартными средствами Python и с помощью разработанной функции библиотеки pyDKB. Тестирование проводилось на наборе одинаковых сообщений фиксированной длины, после каждого из которых следует маркер конца сообщения. В ходе исследований было выявлено, что разработанная функция custom_readline() уступает в скорости стандартной функции, но несмотря на это накладные расходы на использование custom_readline() все еще являются незначительными по сравнению со временем обработки сообщений (на некоторых этапах, где требуется обращение во внешние системы за дополнительным данными, обработка одного сообщения может занимать несколько секунд). В связи с этим в ядро библиотеки pyDKB включена опция определения EOM/EOP-маркера. В том случае, если значение остается по умолчанию, используется стандартная функция чтения.

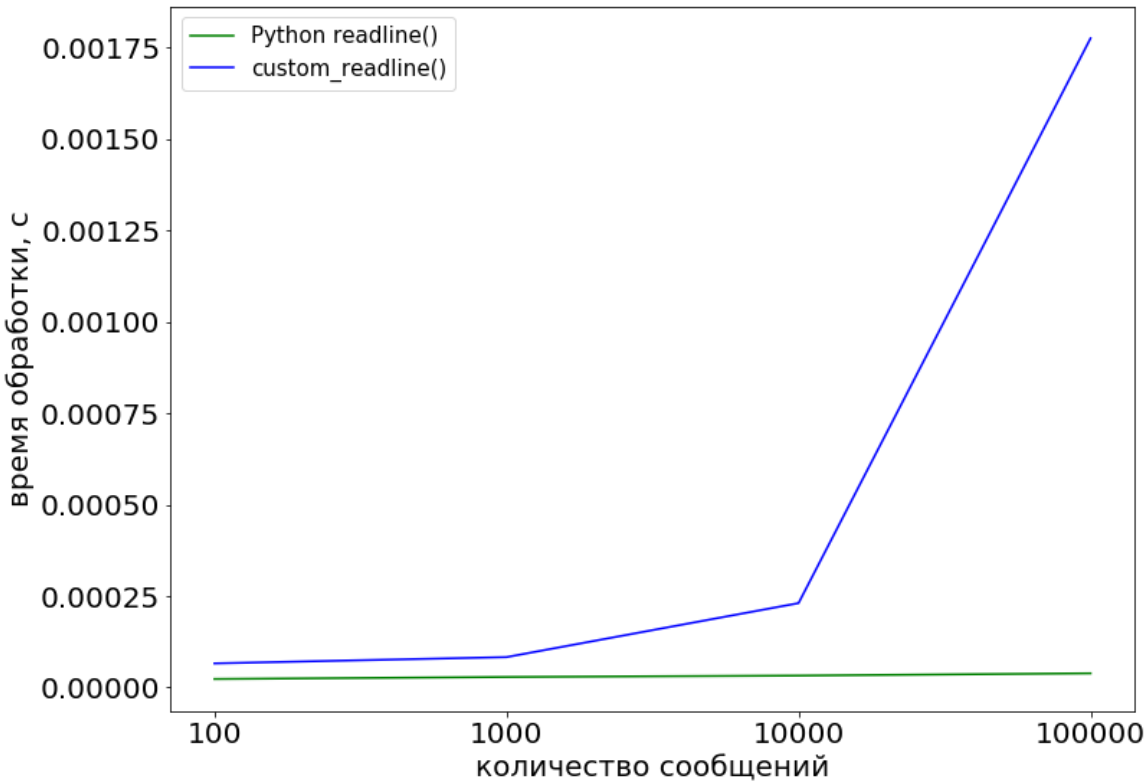

Рисунок 9. График зависимости времени обработки сообщений со стандартным разделителем «\n»

Следующая задача заключалась в оценке времени обработки сообщения, в зависимости от типа маркера, разработанной функцией чтения. Как можно увидеть на рисунке 4, время обработки сообщений в объеме до 10000 составляет не более 1 с, однако далее по мере увеличения количества сообщений рост показателей времени обработки варьируется. Стоит подчеркнуть, что в текущей версии протокола все пользовательские маркеры должны быть односимвольными.

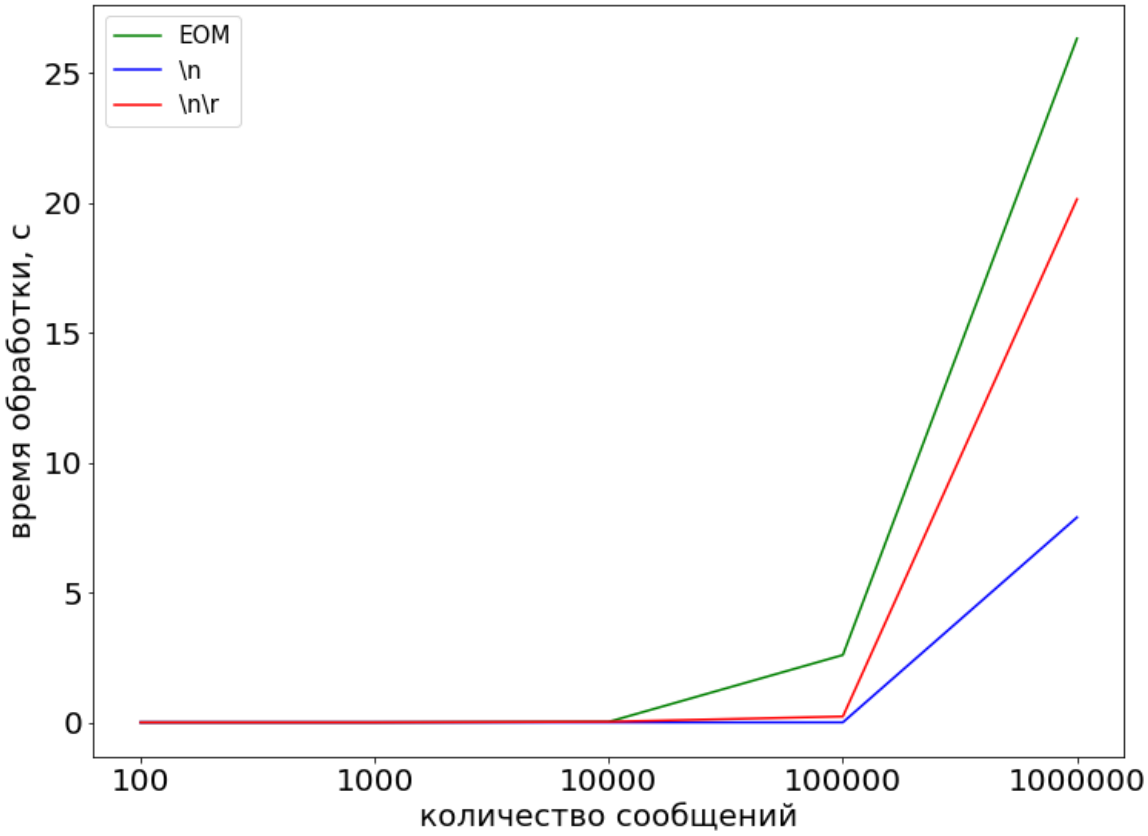

Рисунок 10. График зависимости времени обработки сообщений с пользовательским разделителем функцией custom_readline()

4.2 Обсуждение результатов и апробация

В ходе проведенного исследования были получены следующие результаты:

1) Формализация базовой версии сигнального протокола передачи данных для системы управления потоками данных MInT MS.

2) Реализация маркеров EOM и EOP с возможностью передачи пользовательских параметров.

Данная магистерская диссертация посвящена базовой сигнальной версии протокола. Протокол находится в стадии разработки. На данном этапе идет работа над реализацией пакетной обработки сообщений и формирование ETL-топологии.

Работа проводилась в рамках грантов Российского Научного Фонда №16-11-10280 и Российского Фонда Фундаментальных Исследований мол_а_вед 18-37-20003. По результатам исследований были опубликованы 3 статьи, индексированные в SCOPUS:

1) Kayda A.Y., Golosova M.V., Grigorieva M.A., Gubin M.Y. Development of DKB ETL module in case of data conversion // Journal of Physics: Conference Series. - 2018 - Vol. 1015, Article number 032055. - p. 1-5

2) Golosova M., Aulov V., Kaida A. Metadata handling for Big Data projects // Journal of Physics: Conference Series. - 2018 - Vol. 1117, Article number 012007. - p. 1-8;

3) Golosova M.V., Aulov V.A., Grigorieva M.A., Kaida A.Y. Data Knowledge Base for the ATLAS collaboration // CEUR Workshop Proceedings. - 2018 - Vol. 2267. - p. 486-492.

Промежуточные результаты работы были представлены на Международной научно-практической конференции «Научная сессия ТУСУР 2018», Международной летней школе по распределенным гетерогенным вычислительным инфраструктурам NEC’17, а также на «ATLAS Software&Computing Week» в CERN. На данный момент проводится оформление актов о внедрении.

5 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

5.1 Предпроектный анализ

5.1.1 Потенциальные потребители разработки

Диссертация посвящена разработке протокола передачи данных для системы управления потоками данных внутри Базы Знаний научного эксперимента в области физики высоких энергий. В данных условиях генерируются и обрабатываются сверхбольшие объемы данных, требующие иных подходов их обработки и хранения. Задача обработки больших данных остается актуальной и по сей день. Так, в экспериментах CERN рекордная скорость потока данных составляет 4Гб/сек [15].

Потенциальным потребителем, в первую очередь, является коллаборация ATLAS, чей запрос ученых-физиков оказал влияние на возникновение научной коллаборации ТПУ и НИЦ «Курчатовского института» и инициации проекта – Базы Знаний и разработки всех внутренних компонент.

Актуальность данного раздела заключается в важности понимания коммерческой составляющей научно-технических проектов и оценки коммерческой ценности разработки.

Целью данного раздела является анализ совокупности факторов, которые определяют коммерческую привлекательность разработки, ее перспективность и успешность.

Основными задачами является оценка перспективности разработки, ее готовности к коммерциализации, выявление потенциальных угроз, а также расчет стоимости и составление графика проведения работ.

5.1.2 Технология QuaD

Технология QuaD позволяет оценить перспективность разработки на рынке и целесообразность вложения средств в научно-исследовательский

проект. Результаты оценки, проведенной в табличной форме, представлены в таблице 2.

Таблица 2 – QuaD-анализ разработки

Критерии оценки	Вес критерия	Средний балл	Макси-мальный балл	Относительное значение (3/4)	Средневзвешенное значение (5x2)
1	2	3	4	5	6
Производительность	0,07	80	100	0,8	0,04
Отказоустойчивость	0,17	90	100	0,9	0,153
Унифицированность	0,1	70	100	0,7	0,07
Безопасность	0,05	80	100	0,8	0,04
Потребность в ресурсах памяти	0,13	95	100	0,95	0,1235
Функциональная мощность	0,1	75	100	0,75	0,075
Простота эксплуатации	0,03	40	100	0,4	0,012
Масштабируемость	0,06	75	100	0,75	0,045
Конкурентоспособность продукта	0,07	50	100	0,5	0,035
Перспективность рынка	0,07	85	100	0,85	0,0595
Цена	0,1	40	100	0,4	0,04
Финансовая эффективность научной разработки	0,07	80	100	0,8	0,056
Итого	1				**0,749**

По результатам оценки качества и перспективности можно утверждать, что перспективность текущей разработки выше среднего. Улучшить данную разработку можно путем повышения качества пользовательского интерфейса и снижения уровня сложности эксплуатации. Несмотря на высокую стоимость, на данном этапе снизить данный показатель не представляется возможным.

5.1.3SWOT-анализ

SWOT-анализ разработанного протокола представляет собой двухэтапный комплексный анализ разработки. Результаты анализа представлены в таблице 3.

Таблица 3 – Матрица SWOT разработки

	Сильные стороны разработки: С1. Текст-ориентированность протокола С2. Обработка больших объемов данных С3. Перспектива расширения диапазона сигналов С4. Гибкость разработки С5. Разработка на основе реальных данных С6. Тестирование в условиях, максимально приближенных к реальным	**Слабые стороны разработки:** Сл1. Сложность конфигурирования супервизора Сл2. Техническая сложность Сл3. Сложность ядра системы Сл4. Сложность эксплуатации конечным пользователем Сл5. Высокая стоимость разработки
Возможности: В1. Потребность других научных мегапроектов в создании Базы Знаний В2. Востребованность инструментов автоматизации процесса обработки данных В3. Представление разработок лаборатории крупнейшему научному сообществу	Данная разработка является востребованным, гибким и масштабируемым средством автоматизации процессов обработки данных из независимых внешних хранилищ в базу знаний. Также может быть использована в других проектах лаборатории.	При необходимости масштабирования проекта DKB есть перспектива усложнения ядра системы, требующего дополнительных финансовых и временных затрат.
Угрозы: У1. Устаревание предустановленного ПО на серверах потенциального заказчика У2. Изоляция российского сегмента сети Интернет У3. Изменения требований к Базе Знаний потенциальным заказчиком У4. Рост требований к вычислительным мощностям оборудования У5. Несвоевременное предоставление финансового обеспечения разработки	Наступление сценариев большинства угроз маловероятно. Разрабатываемая База Знаний и все ее компоненты предусматривают изменени требований потенциального заказчика. В свою очередь, заказчик предусматривает опцию предоставления дополнительных вычислительных мощностей.	Техническая сложность разработки может повлечь за собой внесение ряда изменений с изменениями требований к Базе Знаний учеными-физиками. Конечный пользователь должен обладать специальными знаниями по эксплуатации связующего программного обеспечения.

Данная разработка обладает рядом возможностей в условиях низкой вероятности возникновения угроз. Разработка спроектирована таких образом, что сильные стороны предусматривают изменение требований к самой Базе Знаний, а также возникновению задач по масштабированию разработки.

5.1.4 Оценка готовности разработки к коммерциализации

Одной из важных задач в ходе выполнения данного раздела является оценка готовности разработки к коммерциализации. Оцениваемыми параметрами являются как научная, так и коммерческая составляющая. Таблица 4 представляет собой бланк оценки степени готовности разработки к коммерциализации.

Таблица 4 – Бланк оценки степени готовности разработки к коммерциализации

№ п/п	Наименование	Степень проработанности разработки	Уровень имеющихся знаний у разработчика
1.	Определен имеющийся научно-технический задел	4	4
2.	Определены перспективные направления коммерциализации научно-технического задела	4	5
3.	Определены отрасли и технологии (товары, услуги) для предложения на рынке	2	2
4.	Определена товарная форма научно-технического задела для представления на рынок	2	2
5.	Определены авторы и осуществлена охрана их прав	3	3
6.	Проведена оценка стоимости интеллектуальной собственности	3	3
7.	Проведены маркетинговые исследования рынков сбыта	1	1
8.	Разработан бизнес-план коммерциализации научной разработки	1	1
9.	Определены пути продвижения научной разработки на рынок	3	4
10.	Разработана стратегия (форма) реализации научной разработки	5	5
11.	Проработаны вопросы международного сотрудничества и выхода на зарубежный рынок	5	4
12.	Проработаны вопросы использования услуг инфраструктуры поддержки, получения льгот	2	2
13.	Проработаны вопросы финансирования коммерциализации научной разработки	2	3

14.	Имеется команда для коммерциализации научной разработки	3	3
15.	Проработан механизм реализации разработки	5	5
	ИТОГО БАЛЛОВ:	45/75	47/75

Поскольку данная разработка является индивидуальным проектом для уникального научного мегаэксперимента, не предполагающем дальнейший выход на рынок, коммерциализация данного продукта не является целесообразной. Более того, организация, являющаяся потенциальным потребителем, является некоммерческой. В связи с этим провести полноценную оценку перспективы коммерциализации не представляется возможным. По результатам оценки можно утверждать, что данный проект еще не готов к коммерциализации, главным образом, с точки зрения сбыта разработки и финансирования коммерциализации.

Данную разработку возможно коммерциализировать способом передачи ноу-хау.

Поскольку концепция Базы Знаний и проектируемого для нее протокола не фокусируется на конкретной коллаборации (проекте) в области физики высоких энергий, потенциальными пользователями (не только учеными-физиками, но и инженерами подобных систем) возможна передача ноу-хау. Разрабатываемая коллаборацией База Знаний, а также все ее компоненты (в т.ч. протокол – результат разработки), позиционируются как принципиально новое решение для распределенных гетерогенных вычислительных инфраструктур.

5.2 Инициация разработки

В рамках инициации разработки формулируются цели и ожидаемые результаты работы. Также определяются заинтересованные стороны разработки и возможные ограничения. Заинтересованные в данной разработке стороны представлены в таблице 5.

Таблица 5 – Заинтересованные стороны разработки

Заинтересованные стороны	**Ожидания заинтересованных сторон**
Научные коллаборации в области физики высоких энергий	Решение в области интеграции сверхбольших объемов гетерогенных данных независимо от схемы данных
Разработчики баз знаний	Обеспечение автоматизированного выполнения цепочек обработки данных с загрузкой в конечное хранилище (базу знаний)

Цели и результат проекта отображены в таблице 6.

Таблица 6 – Цели и результат разработки

Цели разработки:	Разработка сигнальной версии протокола передачи данных для системы управления потоками данных MInT MS
Ожидаемые результаты разработки:	1) Формализованное описание протокола 2) Программная реализация протокола
Критерии приемки результата разработки:	Автоматизированный обмен данными между процессами: 1) Управляющим процессом-супервизором, 2) Внешними процессами-исполнителями, 3) Процессами-коннекторами (типов Sink и Source).
Требования к результату разработки:	**Требования:**
	Формализованное описание работы протокола
	Обмен сигналами между процессами
	Бесперебойная работа всех элементов системы
	Конфигурируемость сигналов

В таблице 7 представлена рабочая группа разработки, определена роль и основные функции каждого участника в разработке.

Таблица 7 – Рабочая группа разработки

№ п/п	ФИО, основное место работы, должность	Роль в разработке	Функции	Трудо-затраты, час.
1	Савельев Алексей Олегович, ТПУ ОИТ ИШИТР, доцент	Научный руководитель	Утверждение основных разделов, выдача заданий к исполнению, координирование деятельности исполнителя	29
2	Кайда Анастасия Юрьевна, ТПУ ОИТ ИШИТР, магистрант гр.8ПМ7И	Исполнитель	Исполнение поставленных задач	228
ИТОГО:				257

Данный раздел отражает тот факт, что выполняемая работа имеет довольно большой объем. Заинтересованные стороны проекта ожидают достаточно высококачественные результаты, которые необходимо достичь исполнителю.

5.3 Планирование управления разработкой

5.3.1 Иерархическая структура работ

Иерархическая структура работ для данной разработки представляет собой детализацию укрупненной структуры работ, продемонстрированной на рисунке 11.

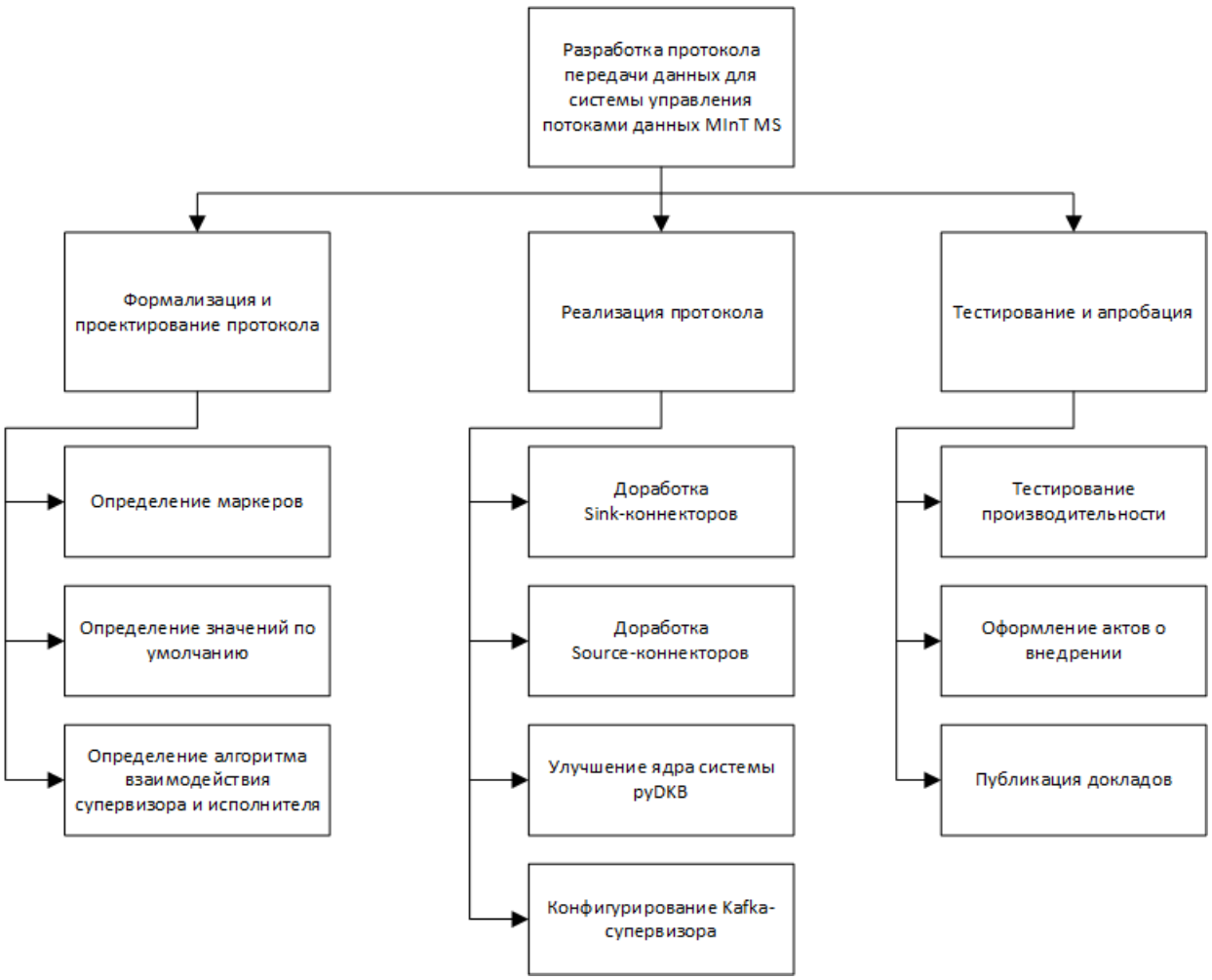

Рисунок 11. Иерархическая структура работ по проведению разработки

Задачи по созданию данной разработки разделены на три основных блока: формализация и разработка, реализация, а также тестирование и апробация. Основная часть работ фокусируется на адаптации существующих элементов системы, создания конфигурации программного брокера сообщений, а также расширения ядра системы pyDKB.

5.3.2 План разработки

Чтобы отразить ключевые события по ведению разработки, необходимо составить календарный план. План представлен в таблице 8.

Таблица 8 – Календарный план разработки

Код работы	Название	Длительность, дни	Дата начала работ	Дата окончания работ	Состав участников
1	Выбор научного руководителя магистерской работы	1	01.09.18	01.09.18	Кайда Анастасия Юрьевна
2	Составление и утверждение темы магистерской работы	2	03.09.18	04.09.18	Савельев Алексей Олегович
3	Составление календарного плана-графика выполнения магистерской работы	2	05.09.18	06.09.18	Кайда Анастасия Юрьевна, Савельев Алексей Олегович
4	Выявление требований к разработке	7	07.09.18	14.09.18	Кайда Анастасия Юрьевна, Савельев Алексей Олегович
5	Подбор и изучение литературы по теме магистерской работы	25	15.09.18	13.10.18	Кайда Анастасия Юрьевна
6	Анализ предметной области	15	15.10.18	31.10.18	Кайда Анастасия Юрьевна
7	Формализация протокола передачи данных	30	01.11.18	05.12.18	Кайда Анастасия Юрьевна
8	Разработка протокола	80	06.12.18	08.03.19	Кайда Анастасия Юрьевна
9	Тестирование	20	09.03.19	01.04.19	Кайда Анастасия Юрьевна
10	Анализ полученных результатов, сравнительная оценка производительности протокола	4	02.04.19	05.04.19	Кайда Анастасия Юрьевна, Савельев Алексей Олегович
11	Согласование выполненной работы с научным руководителем	4	06.04.19	10.04.19	Кайда Анастасия Юрьевна, Савельев Алексей Олегович

12	Выполнение других частей работы (финансовый менеджмент, социальная ответственность)	30	11.04.19	15.05.19	Кайда Анастасия Юрьевна
13	Подведение итогов, оформление работы	10	16.05.19	27.05.19	Кайда Анастасия Юрьевна, Савельев Алексей Олегович
Итого:		230			

5.3.2.1 Определение трудоемкости выполнения работ

Трудоемкость выполнения научного исследования оценивается экспертным путем в человеко-днях и носит вероятностный характер, завися от множества трудно учитываемых факторов. Для определения ожидаемого значения трудоемкости $t_{ожi}$ используется формула:

$$t_{\text{ож}i} = \frac{3t_{\text{min}i}+2t_{\text{max}i}}{5}, \qquad (1)$$

где $t_{ожі}$ – ожидаемая трудоемкость і-й работы чел.-дн;

t_{maxi} – минимально возможная трудоемкость выполнения заданной работы, чел.-дн.;

t_{mini} – минимально возможная трудоемкость выполнения заданной работы, чел.-дн.

Промежуточные расчеты представлены в таблице 9.

Таблица 9 – Временные показатели проведения разработки

Наименование работы	Исполнители работы	Трудоемкость работ, чел-дни			Длительность работ, дни	
		tmin	tmax	tож	Тр	Тк
Выбор научного руководителя магистерской работы	Кайда А.Ю.	1	2	1,4	1	1
Составление и утверждение темы магистерской работы	Савельев А.О.	1	3	1,8	2	2

Составление календарного плана-графика выполнения магистерской работы	Кайда А.Ю.	1	3	1,8	2	2
	Савельев А.О.	1	3	1,8	2	2
Выявление требований к разработке	Кайда А.Ю.	5	10	7	7	8
	Савельев А.О.	5	10	7	7	8
Подбор и изучение литературы по теме магистерской работы	Кайда А.Ю.	22	30	25,2	25	29
Анализ предметной области	Кайда А.Ю.	10	22	14,8	15	17
Формализация протокола передачи данных	Кайда А.Ю.	20	45	30	30	35
Разработка протокола	Кайда А.Ю.	60	110	80	80	89
Тестирование	Кайда А.Ю.	10	35	20	20	24
Анализ полученных результатов, сравнительная оценка производительности протокола	Кайда А.Ю.	2	7	4	4	4
	Савельев А.О.	2	7	4	4	4
Согласование выполненной работы с научным руководителем	Кайда А.Ю.	2	7	4	4	5
	Савельев А.О.	2	7	4	4	5
Выполнение других частей работы (финансовый менеджмент, социальная ответственность)	Кайда А.Ю.	20	45	30	30	35
Подведение итогов, оформление работы	Кайда А.Ю.	7	14	9,8	10	12
	Савельев А.О.	7	14	9,8	10	12

5.3.2.2 Разработка графика проведения разработки

На рисунке 12 представлена диаграмма Ганта с планом выполнения работ, где М – магистрант (Кайда Анастасия Юрьевна), НР – научный руководитель (Савельев Алексей Олегович).

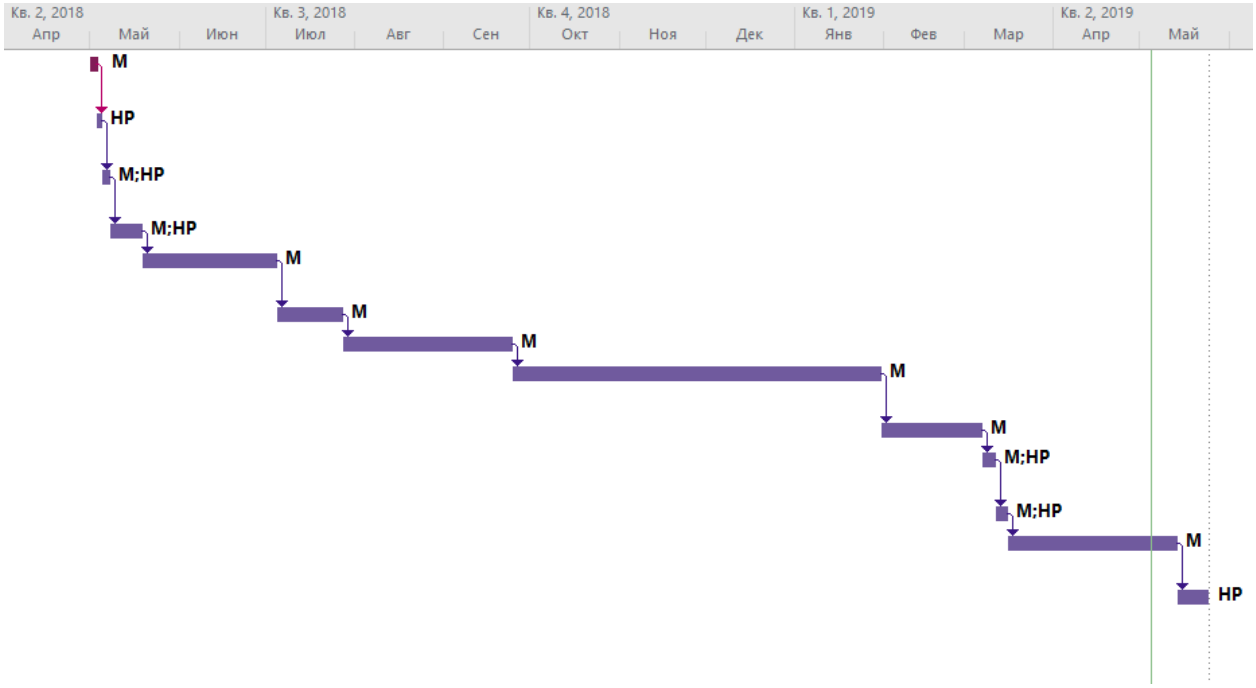

Рисунок 12. График проведения разработки

5.3.3 Бюджет разработки

Для данной разработки бюджет состоит из следующих пунктов:

1) Материальные затраты;
2) Амортизационные отчисления;
3) Основная заработная плата исполнителей темы;
4) Дополнительная заработная плата исполнителей темы;
5) Страховые отчисления;
6) Накладные расходы.

5.3.3.1 Расчет материальных затрат разработки

Для проведения исследования какие-либо специальные материалы и комплектующие не приобретались. Единая сумма на канцелярские принадлежности составляет 2500 рублей.

5.3.3.2 Расчет амортизационных отчислений

Поскольку для проведения исследований специальное дорогостоящее оборудование не приобреталось, при расчете затрат учитывается только

амортизация. Первоначальная стоимость ПК магистранта, используемого для проведения исследований, составляет 50000 рублей. Срок полезного использования данной машины – 3 года, из которых 9 месяцев машина использовалась для написания ВКР.

Норма амортизации:

$$\text{Ан} = \frac{1}{n} * 100\% = \frac{1}{3} \times 100\% = 33{,}33\% \quad (2)$$

Годовые амортизационные отчисления:

$$\text{Аг} = 50000 \times 0{,}33 = 16500 \text{ рублей} \quad (3)$$

Ежемесячные амортизационные отчисления:

$$\text{Ам} = \frac{16500}{12} = 1375 \text{ рублей} \quad (4)$$

Итоговая сумма амортизации основных средств:

$$\text{А} = 1375 \times 12 = 16500 \text{ рублей} \quad (5)$$

Таким образом затраты на амортизацию ПК составляют 16500 рублей.

Годовая лицензия программного среды разработки для Python составляет 199$ (или 13000 рублей в рублевом эквиваленте), т.е. срок полезного использования данного ПО составляет 1 год.

Ежемесячные амортизационные отчисления:

$$\text{Ам} = \frac{13000}{12} = 1083 \text{ рублей} \quad (6)$$

Итоговая сумма амортизации основных средств:

$$\text{А} = 1083 \times 12 = 12996 \text{ рублей} \quad (7)$$

Суммарная стоимость остального программного обеспечения с перманентной лицензией составляет 42000 рублей.

Норма амортизации:

$$\text{Ан} = \frac{1}{n} * 100\% = \frac{1}{5} \times 100\% = 20\% \quad (8)$$

Годовые амортизационные отчисления:

$$Аг = 42000 \times 0{,}2 = 8400 \text{ рублей} \quad (9)$$

Ежемесячные амортизационные отчисления:

$$Ам = \frac{8400}{12} = 700 \text{ рублей} \quad (10)$$

Итоговая сумма амортизации основных средств:

$$А = 700 \times 12 = 8400 \text{ рублей} \quad (11)$$

Амортизация остального программного обеспечения составляет 8400. Итого амортизация ПО составляет 21396 рублей.

Таблица 10 – Расчет затрат на амортизацию

Наименование	Затраты, руб.
Амортизация ПК	16500
Амортизация ПО	21396
Итого	**37896**

5.3.3.3 Основная заработная плата исполнителей темы

Исполнителями темы выступают научный руководитель и инженер. Оклад руководителя в ТПУ без районного коэффициента составляет 33664 рубля, оклад инженера – 21760 рублей. Баланс рабочего времени для 6-дневной недели, по которой учитывается рабочее время преподавателей и студентов, представлен в таблице 11.

Таблица 11 – Баланс рабочего времени (для 6-дневной недели)

Показатели рабочего времени	Дни
Календарные дни	365
Нерабочие дни (праздники/выходные)	66
Потери рабочего времени (отпуск/невыходы по болезни)	56
Действительный годовой фонд рабочего времени	243

Таблица 12 – Расчет основной заработной платы

Исполнители	Здн, руб.	Кпр	Кд	Кр	Тр	Зосн
Кайда Анастасия Юрьевна	931,29	0,3	0,2	1,3	228	414051,53
Научный руководитель (Савельев Алексей Олегович)	1440,76	0,3	0,5	1,3	29	97769,97
					Итого:	**511821,5**

$$\text{Здн} = \frac{\text{Зм} \times \text{М}}{F\text{д}} = \frac{21760 \times (1 + \text{Кпр} + \text{Кд}) \times \text{Кр} \times 10{,}4}{243} = 1860{,}02 \text{ руб.} \quad (12)$$

$$\text{Здн} = \frac{\text{Зм} \times \text{М}}{F\text{д}} = \frac{33664 \times (1 + 0{,}3 + 0{,}5) \times 1{,}3 \times 10{,}4}{243} = 3371{,}38 \text{ руб.} \quad (13)$$

Расчет основной заработной платы (для инженера и руководителя соответственно):

$$\text{Зосн} = \text{Здн} \times \text{Тр} \quad (14)$$

$$\text{Зосн}_{\text{инж}} = 1860{,}02 \times 228 = 414051{,}53 \quad (15)$$

$$\text{Зосн}_{\text{рук}} = 3371{,}38 \times 29 = 97769{,}97 \quad (16)$$

5.3.3.4 Дополнительная заработная плата исполнителей темы

Пусть дополнительная заработная плата составляет 15% от основной заработной платы. Тогда зарплаты инженера и руководителя соответственно будут высчитываться по формуле:

$$\text{Здоп} = \text{Зосн} \times 0{,}15 \quad (16)$$

$$\text{Здоп}_{\text{инж}} = 414051{,}53 \times 0{,}15 = 62107{,}73 \quad (17)$$

$$\text{Здоп}_{\text{рук}} = 97769{,}97 \times 0{,}15 = 14665{,}5 \quad (18)$$

Дополнительная заработная плата исполнителей равна 62107,73 и 14665,5 соответственно.

5.3.3.5 Отчисления во внебюджетные фонды (страховые отчисления

Составляют 30% от заработной платы (основная + дополнительная). Таким образом страховые взносы составляют 176578,42 рублей.

$$Отч = (Здоп + Зосн) \times 0{,}3 \qquad (19)$$

$$Отч = (511\,821{,}5 + 76\,773{,}23) \times 0{,}3 = 176578{,}42 \qquad (20)$$

5.3.3.6 Накладные расходы

Накладные расходы составляют 16% от суммы материальных затрат, затрат на специальное оборудование, затрат на основную заработную плату, затрат на дополнительную заработную плату и страховых взносов. Накладные расходы составляют 128500,664 рублей.

5.3.3.7 Формирование бюджета затрат научно-исследовательского разработки

Таблица 13 – Бюджет затрат

Наименование	Сумма, руб.	Удельный вес, %
Материальные затраты	2500	0,27
Затраты на специальное оборудование	37896	4,06
Затраты на основную заработную плату	511821,5	54,77
Затраты на дополнительную заработную плату	76773,23	8,22
Страховые взносы	176578,42	18,89
Накладные расходы	128891,06	13,79
Общий бюджет	934460,21	100

Общий бюджет разработки составляет 934460,21 рублей. Данный бюджет ниже конкурентных зарубежных разработок.

5.3.4 Риски разработки

Проведение любого научно-исследовательского проекта сопряжено с возникновением различных рисков. Предварительное определение рисков помогает своевременному принятию мер по предотвращению возникновения угроз или минимизации их последствий.

Таблица 14 – Определение рисков

№ п/п	Наименование риска	Описание риска
1	Политические	Риск отказа от продолжения сотрудничества потенциального заказчика в связи с обострением политической обстановки.
2	Технологические	Безвозвратная утеря большого процента исходных данных, на которые опирается разработка, в ходе работы.
3	Финансовые	Прекращение финансирования проекта Базы Знаний.
4	Технические	Сбой или поломка оборудования, связанного с хранилищами данных, на которые опирается разработка.

Таблица 15 – Оценка вероятности рисков

№ п/п	Наименование риска	Оценка вероятности риска (низкая, средняя, высокая)
1	Политические	Низкая
2	Технологические	Низкая
3	Финансовые	Низкая
4	Технические	Низкая

Таблица 16 – Оценка уровня потерь

№ п/п	Наименование риска	Оценка уровня потерь (низкий, средний, высокий)
1	Политические	Высокий
2	Технологические	Средний
3	Финансовые	Высокий
4	Технические	Низкий

Таблица 17 – Оценка уровня потерь

Матрица вероятности рисков/потерь				
		Уровень потерь		
		Высокий	Средний	Низкий
Вероятность	Высокая	Высокий	Высокий	Умеренный
	Средняя	Существенный	Существенный	Незначительный
	Низкая	Умеренный	Умеренный	Незначительный

Таблица 18 – основные мероприятия по снижению рисков

№ п/п	Наименование риска	Мероприятия по снижению риска
1	Политические	Заключение контракта о сотрудничестве на четко обозначенный период. Организация полноправного удаленного доступа пользователю, имеющему статус CERN User.
2	Технологические	Разграничение уровня доступа пользователей. Соблюдение протокола безопасности CERN. Создание бэкапов данных.
3	Финансовые	Своевременное принятие мер по подготовке отчетности по текущим работам и подача заявки на новые.
4	Технические	Соблюдение протокола безопасности CERN.

Основными рисками при выполнении разработки можно назвать технологические и технические риски, связанные с хранилищами данных о научном эксперименте в области физики высоких энергий, при этом наибольшую угрозу предствляют собой технологические риски. Среди прочих рисков стоит отметить, что финансовые риски связаны, в основном, с финансированием. Предотвращения данного риска возможно в случае своевременного предоставление релевантной документации в научные фонды. Политические риски наименее вероятны, однако стоит отметить, что в случае наступления предполагаемого сценария, подобные риски имеют серьезные последствия для такого крупного научного проекта, как База Знаний.

5.4 Определение потенциального эффекта разработки

Потенциальным пользователям разработки является некоммерческая организация, не финансирующая НИОКР (Базу Знаний) или любую его часть напрямую, поэтому метод оценки абсолютной эффективности исследования не подходит для данного исследования. Поскольку в ходе выполнения магистерской диссертации разрабатывался только один вариант разработки, провести оценку сравнительной эффективности исследования не представляется возможным.

Данная разработка ориентирована на конкретного потребителя/группу потребителей, которыми являются научные коллаборации в области физики высоких энергий (например, коллаборация ATLAS). Стоит отметить, что, поскольку потенциальный рынок сбыта мал, коммерциализация данной разработки остается открытым вопросом. Бланк оценки степени готовности к коммерциализации указывает на то, что готовность в коммерциализации находится на нижнем пороге уровня выше среднего.

Общая продолжительность исследования составляет 9 месяцев.

Потенциальная стоимость исследования составляет около 1 млн. рублей. При этом специальное дорогостоящее оборудование не закупалось. Данная цена является конкурентоспособной, поскольку стоимость решений для подобных вычислительных структур «из коробки», не ориентированных на конкретного потребителя, находятся в том же ценовом диапазоне.

Данный проект сопряжен с малыми рисками, однако стоит обратить внимание на технологические риски, последствия которых могут нанести существенный урон подобным исследованиям, а также при продолжении работы над данным проектом.

Проект не имеет прямых аналогов в обозначенных условиях обработки больших объемов данных в гетерогенных вычислительных инфраструктурах. Таким образом, концепция данного исследования вне области знаний, для которой используются данные база знаний и, соответственно, протокол является направляющей для будущих исследований в данной области.

6 Социальная ответственность

Объектом исследования является результат научно-исследовательской деятельности – протокол передачи данных для системы управления потоками данных. Протокол регламентирует взаимодействие между другими элементами системы – Базы Знаний и системы интеграции метаданных.

Проектирование, разработка и тестирование производится с персональной электронной вычислительной машины в условиях учебной аудитории Кибернетического центра Томского политехнического университета.

Создание Базы Знаний, для интеграции данных в которую и разрабатывается протокол, было инициировано Европейской организацией по ядерным исследованиям совместно с НИЦ «Курчатовский институт» и ТПУ. Реализация всех компонентов, в т.ч. протокола, позволит создать важный инструмент – своеобразную централизованную точку доступа к ключевой информации об экспериментах, – с использованием которого можно снизить в т.ч. риски при проведении экспериментов.

Целью данного раздела является выявление специальных правовых норм трудового законодательства, выявление потенциальных вредных и опасных факторов, мер защиты, анализ экологического влияния исследования (объекта и процесса), а также анализ возникновения возможных ЧС, мер их предупреждения и устранения последствий.

6.1 Правовые и организационные вопросы обеспечения безопасности

6.1.1 Специальные (характерные для проектируемой рабочей зоны) правовые нормы трудового законодательства

Правовое регулирование трудовых отношений между работодателем, работником и государством регулируется Трудовым кодексом Российской Федерации от 30.12.2001 N 197-ФЗ. В ТК РФ [16], в соответствии с Конституцией РФ, признаются свобода труда, выбор и согласие на него, а также выбор профессии и деятельности. Запрещаются принудительный труд, дискриминация

по какому-либо признаку. Гарантируются справедливые и достойные условия труда.

ТК РФ регламентирует порядок разрешения индивидуальных и коллективных трудовых споров, особенности труда женщин, детей и людей пенсионного возраста, права и обязанности работодателей и работников, нормы рабочего времени, порядок оплаты труда и виды компенсаций во вредных условиях труда, а также особенности социального страхования.

В соответствии со ст. 111 ТК РФ, рабочая неделя (в т.ч. шестидневная) не должна превышать 40 часов в неделю. Воскресенье является выходным днем.

В соответствии со ст. 212 ТК РФ, работодатель обязан обеспечить безопасные условия труда, а также обязательное социальное страхование работников от несчастных случаев на производстве и профессиональных заболеваний.

В соответствии со ст. 142 ТК РФ, в случае задержки выплаты заработной платы на срок более 15 дней работник имеет право, известив работодателя в письменной форме, приостановить работу на весь период до выплаты задержанной суммы, кроме ряда перечисленных случаев.

Данные о работнике, предоставляемые работодателю, обрабатываются только с согласия самого работника и охраняются Федеральным Законом от 27.07.2006 N 152-ФЗ (ре. от 25.07.2011) «О персональных данных» [17].

6.1.2 Организационные мероприятия при компоновке рабочей зоны

Согласно ГОСТ 12.2.032-78 ССБТ [18], выявлены следующие параметры рабочей зоны:

1) Согласно наименованию работы (работа за ЭВМ) при отсутствии регулирующих механизмов высоты рабочей поверхности, высота рабочей поверхности, при организации рабочего места, составляет (для женщин) 630 мм. Высота сиденья 400 мм.

2) Рабочая поверхность в соответствии с видом работ может содержать дополнительное углубление для периферийных устройств (клавиатура).

3) Рабочее место при выполнении работ сидя организуют при легкой работе, не требующей свободного передвижения работающего.

4) Конструкция рабочего места и взаимное расположение всех его элементов должны соответствовать антропометрическим, физиологическим и психологическим требованиям, а также характеру работы.

В соответствии с СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы» (с изменениями на 21 июня 2016 года) [19] были выявлены следующие правила организации работы с ПЭВМ:

1) Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300-500 лк. Освещение не должно создавать бликов на поверхности экрана. Освещенность поверхности экрана не должна быть более 300 лк.

2) При размещении рабочих мест с ПЭВМ расстояние между рабочими столами с видеомониторами (в направлении тыла поверхности одного видеомонитора и экрана другого видеомонитора), должно быть не менее 2,0 м, а расстояние между боковыми поверхностями видеомониторов - не менее 1,2 м.

3) Конструкция рабочего стула (кресла) должна обеспечивать поддержание рациональной рабочей позы при работе на ПЭВМ, позволять изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления. Тип рабочего стула (кресла) следует выбирать с учетом роста пользователя, характера и продолжительности работы с ПЭВМ.

4) Ширина и глубина поверхности сиденья должны составлять не менее 400 мм.

5) Контролируемыми гигиеническими параметрами персональных цифровых ЭВМ (в т.ч. портативных) являются: уровни электромагнитных полей (ЭМП), акустического шума, концентрация вредных веществ в воздухе, визуальные показатели ВДТ, мягкое рентгеновское излучение.

6.2 Профессиональная социальная безопасность

В данном подразделе анализируются вредные и опасные факторы, которые могут возникать при проведении исследований в лаборатории, при разработке или эксплуатации проектируемого решения.

Согласно ГОСТ 12.0.003-2015 [20] в таблице 19 представлены возможные вредные и опасные факторы. Работа по разработке программного обеспечения делится на три основных этапа: проектирование, разработка и эксплуатация.

Таблица 19 – Возможные вредные и опасные факторы

Факторы (ГОСТ 12.0.003-2015)	Этапы работ			Нормативные документы
	Проектирование	Разработка	Эксплуатация	
1. Отклонение показателей микроклимата	+	+	+	СанПиН 2.2.4.548–96 СанПиН 2.2.2/2.4.1340-03
2. Превышение уровня шума	+	+	+	ГОСТ 12.1.003-2014 ССБТ ГОСТ 12.1.029-80
3. Отсутствие или недостаток естественного света	+	+	+	СанПиН 2.2.1/2.1.1.1278–03
4. Недостаточная освещенность рабочей зоны	+	+	+	СанПиН 2.2.1/2.1.1.1278–03
5. Повышенное значение напряжения в	+	+	+	ГОСТ 12.1.019-2017

электрической цепи, замыкание которой может произойти через тело человека				

6.2.1 Анализ вредных и опасных факторов, которые может создать объект исследования

Теоретически, объект исследования (протокол передачи данных) при определенных условиях, не предусмотренных разработчиками, способен привести к перегрузкам в ПЭВМ и вызвать определенные последствия. Поскольку разработка не является осязаемым объектом и неотделима от ПЭВМ, вредные и опасные факторы, которые могут быть прямо или косвенно отнесены к разработке, относятся и к рабочему месту.

6.2.2 Анализ вредных и опасных факторов, которые могут возникнуть на рабочем месте

6.2.2.1 Отклонение показателей микроклимата

Показатели микроклимата должны обеспечивать сохранение теплового баланса человека с окружающей средой и поддержание оптимального или допустимого теплового состояния организма. Отклонение показателей от нормы в пределах допустимых величин могут вызвать локальные ощущения теплового дискомфорта и напряжение механизмов терморегуляции.

Согласно СанПиН 2.2.4.548-96 [21], санитарные правила устанавливают гигиенические требования к показателям микроклимата рабочих мест производственных помещений с учетом интенсивности энерготрат работающих, времени выполнения работы, периодов года и содержат требования к методам измерения и контроля микроклиматических условий.

Оптимальные микроклиматические условия установлены по критериям оптимального теплового и функционального состояния человека. Они

обеспечивают общее и локальное ощущение теплового комфорта в течение 8-часовой рабочей смены при минимальном напряжении механизмов терморегуляции, не вызывают отклонений в состоянии здоровья, создают предпосылки для высокого уровня работоспособности и являются предпочтительными на рабочих местах.

Категория работ определена, как наименее энергозатратная, т.е. не сопровождающаяся какой-либо физической нагрузкой (сидячий вид деятельности, умственный труд) – 1а. Оптимальные величины показателей микроклимата представлены в таблице 20.

Таблица 20 – Оптимальные величины показателей микроклимата на рабочих местах производственных помещений

Период года	Категория работ по уровню энергозатрат, Вт	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Теплый	1а	22-24	21-25	60-40	0,1
Холодный	1а	23-25	22-26	60-40	0,1

Для определения температуры воздуха в помещении использовался настенный термометр. Показания температуры воздуха составляют около 23 °С. Для измерения относительной влажности воздуха использовался гигрометр механического типа. Показания гигрометра на момент измерения составили 45%. Отсюда можно сделать вывод, что микроклимат в учебной аудитории является оптимальным для работы в данное время года.

6.2.2.2 Превышение уровня шума

Превышения уровня шума является вредным фактором на рабочем месте. Постоянный шум, превышающий допустимые значения, не только воздействует

на органы слуха, но и влияет на общее самочувствие работника, способствует ослаблению организма, а также снижает работоспособность.

Согласно ГОСТ 12.1.003-2014 [22], машины, которые в процессе работы могут производить шум, неблагоприятно воздействующий на работников, следует конструировать и изготовлять с учетом последних достижений технологии и принципов проектирования, позволяющих снизить излучаемый шум.

Наибольшим шумовым событием на рабочем месте для исследователя является ПЭВМ или несколько ПЭВМ.

СанПиН 2.2.2/2.4.1340-03 регулирует допустимые значения уровней звукового давления в октавных полосах частот и уровня звука, создаваемого ПЭВМ. Данные представлены в таблице 21.

Таблица 21 – Допустимые значения уровней звукового давления в октавных полосах частот и уровня звука, создаваемого ПЭВМ

Уровни звукового давления в октавных полосах со среднегеометрическими частотами									Уровни звука в дБА
31, 5 Гц	63 Гц	125 Гц	250 Гц	500 Гц	1000 Гц	2000 Гц	4000 Гц	8000 Гц	
86 дБ	71 дБ	61 дБ	54 дБ	49 дБ	45 дБ	42 дБ	40 дБ	38 дБ	50

6.2.2.3 Отсутствие или недостаток естественного света

Отсутствие естественного света в темное время суток является естественным явлением, актуальным для г.Томска в течение рабочего дня в зимний период в силу короткого светового дня. Однако в случае, если основным источником освещения является естественный свет через светопроемы (окна), освещение должно соответствовать нормам согласно СанПиН 2.2.1/2.1.1.1278-03 [23]. В таблице 22 представлены нормируемые показатели не только естественного, но также совмещенного и искусственного освещения.

Таблица 22 – Нормируемые показатели естественного, искусственного и совмещенного освещения основных помещений общественного здания, а также сопутствующих им производственных помещений

Помещения	Рабочая поверхность и плоскость нормирования КЕО и освещенности (Г – горизонтальная, В – вертикальная) и высота плоскости над полом, м	Естественное освещение		Совмещенное освещение		Искусственное освещение				
		КЕО, %		КЕО, %						
		при верхнем или комбинированном освещении	при боковом освещении	при верхнем или комбинированном освещении	при боковом освещении	Освещенность, лк			Показатель дискомфорта, М, не более	Коэффициент пульсации освещенности, Кп, %, не более
						при комбинированном освещении		при общем освещении		
						всего	от общего			
1	2	3	4	5	6	7	8	9	10	11
Кабинеты информатики и вычислительной техники	Г-0,8 Экран дисплея: В – 1;	3,5	1,2	2,1	0,7	500	300	400	15	10

6.2.2.4 Недостаточная освещенность рабочей зоны

Основным источником освещения в учебной аудитории является искусственное освещение. Недостаточная освещенность рабочей поверхности может сказаться на здоровье работника и его производительности труда. Согласно таблице 4, общая освещенность должна всего составлять 500 лк, где 300 лк составляет искусственное освещение. При общем освещении показатель составляет 400 лк.

Равномерное освещение горизонтальной рабочей поверхности достигается при определенных отношениях расстояния между центрами светильников L(м) (L = 1,75·H), к высоте их подвеса над рабочей поверхностью H (м). В помещении для разработки исследования высота составляет 2,5.

Количество люминесцентных ламп ЛБ 40 в количестве 24 шт. Согласно ГОСТ 6825-91. «Лампы люминесцентные трубчатые для общего освещения» номинальный световой поток люминесцентные ламп ЛБ 40 составляет 3000лк.

L = 1.75* 2.5 = 4.375(м)

Ф = НСП*N = 3000*24= 72000 (люмен)

НСП - номинальный световой поток ламп, N -количество ламп

I = Ф/2π =72000/6.28 = 11465 (кандел)

E = I/ L*L = 11465/ 4.375*4.375 = 599 (люкс) (без учета потерь в плафоне светильника)

6.2.2.5 Повышенное значение напряжения в электрической цепи

К опасным факторам на рабочем месте относится поражение электрическим током, поскольку исследователь взаимодействует с электрооборудованием.

Согласно ГОСТ 12.1.019-2017 ССБТ [24], во избежание поражения работника электрическим током следует соблюдать следующие базовые принципы:

1) Проводящие части, находящиеся под опасным рабочим, наведенным, остаточным напряжением, не должны быть доступными, а

доступные проводящие части не должны находиться под опасным напряжением при нормальных условиях (при отсутствии повреждения), а также в случае единичного повреждения.

2) Защиту при нормальных условиях (защиту от прямого прикосновения) обеспечивают посредством основной защиты, а защиту при условиях единичного повреждения (защиту при косвенном прикосновении) обеспечивают посредством защиты при повреждении.

Степень опасного воздействия на человека электрического тока электрической дуги зависит от:

1) величины напряжения прикосновения, электрического сопротивления тела человека, силы тока, протекающей через него, а также величины падающей энергии электрической дуги;

2) рода (постоянный, переменный, выпрямленный) тока и частоты переменного электрического тока;

3) пути протекания тока через тело человека и площади контакта электрической дуги с поверхностью тела человека;

4) продолжительности воздействия электрического тока и электрической дуги на организм человека;

5) индивидуальных особенностей организма человека;

6) условий внешней среды.

6.2.2.6 Умственное перенапряжение

Работая за ПЭВМ, работник также находится под влиянием еще одного вредного производственного фактора, нервно-психической перегрузки в виде умственного перенапряжения.

Согласно ТОИ Р-45-084-01 [25], виды трудовой деятельности разделяются на 3 группы: группа А - работа по считыванию информации с экрана компьютера с предварительным запросом; группа Б - работа по вводу информации; группа В - творческая работа в режиме диалога с компьютером. При выполнении в течение рабочей смены работ, относящихся к различным

видам трудовой деятельности, за основную работу с компьютером следует принимать такую, которая занимает не менее 50% времени в течение рабочей смены или рабочего дня. Уровень нагрузки представлен в таблице 23.

Таблица 23 – Уровень нагрузки за рабочую смену при видах работ с компьютером

Категория работ	Уровень нагрузки за рабочую смену при видах работ с компьютером		
	группа А, количество знаков	группа Б, количество знаков	группа В, час.
III	До 60000	До 40000	До 6,0

6.2.3 Обоснование мероприятий по защите исследователя от действия опасных и вредных факторов

Для поддержания оптимального микроклимата в помещении используются средства центрального отопления и кондиционирования в зависимости от сезона. Согласно СанПиН 2.2.2/2.4.1340-03, также:

1) В помещениях, оборудованных ПЭВМ, проводится ежедневная влажная уборка и систематическое проветривание после каждого часа работы на ЭВМ.

2) Уровни положительных и отрицательных аэроионов в воздухе помещений, где расположены ПЭВМ, должны соответствовать действующим санитарно-эпидемиологическим нормативам.

Для защиты от шума, согласно ГОСТ 12.1.029-80 ССБТ [26], могут применяться следующие средства и методы:

1) Рациональное размещение рабочих мест.

2) Рациональное размещение технологического оборудования.

3) Применение малошумных современных ПЭВМ.

Поскольку применение индивидуальных средств шумоизоляции оказывает дополнительный психологический эффект и ухудшение общего

состояния работника после длительного использования, средства защиты от шума должны являться некоторым компромиссом, направленным на снижение уровня шума и, вместе с тем, сохранение комфортных условий работы.

Защитой от прямого прикосновения к токопроводящим частям электрооборудования являются:

1) Основная изоляция.
2) Безопасное расположение токоведущих частей.
3) Защитное отключение.

Окружающая среда не должна быть проводящей. При эксплуатации электрооборудования необходимо соблюдать технику безопасности.

Для III категории работ (ТОИ Р-45-084-01) по уровню нагрузки перерыв регламентирован через 1,5 - 2,0 часа от начала рабочей смены и через 1,5 - 2,0 часа после обеденного перерыва продолжительностью 20 минут каждый или продолжительностью 15 минут через каждый час работы.

6.3 Экологическая безопасность

Целью данного подраздела является выявление потенциальных опасностей объекта и процесса исследования на окружающую среду, а также разработка мер, обеспечивающая безопасность исследовательской деятельность для окружающей среды.

6.3.1 Анализ влияния объекта исследования на окружающую среду

Объект исследования (протокол передачи данных) не оказывает влияния на окружающую среду, поскольку используется только совместно с ПЭВМ. Сами ПЭВМ могут являться источниками различных загрязнений окружающей среды.

6.3.2 Анализ влияния процесса исследования на окружающую среду

Процесс исследования включает в себя работу на ПЭВМ в учебной аудитории (КЦ ТПУ), в том числе в условиях искусственного освещения, обеспечиваемого люминесцентными лампами.

Отработанная офисная техника относится к опасным отходам. При производстве компьютеров и других агрегатов применяются вещества, опасные для жизнедеятельности, например, свинец, мышьяк и др. Обычное выбрасывание техники, особенно регулярное, может нанести непоправимый вред экологии и здоровью. Согласно Административному Кодексу РФ, ст. 8.2 [27], запрещается выбрасывать технику наряду с обыкновенным мусором, причем запрет распространяется не только на физических лиц, но и на организации.

6.3.3 Обоснование мероприятий по защите окружающей среды

Согласно ГОСТ Р 56397-2015 [28], в результате технической экспертизы может быть принято следующее решение: оборудование не ремонтопригодно, признается неработоспособным и рекомендуется к списанию (замене); в случае деградационного отказа оборудования и нецелесообразности его ремонта и модернизации даются рекомендации о необходимости его списания и утилизации. Самостоятельная утилизация оргтехники запрещена, утилизация производится только в промышленных условиях. Утилизировать компьютерную технику имеют права специализированные предприятия при наличии соответствующей лицензии.

На рабочем месте программиста используются 16 люминесцентных ламп ЛБ40, Согласно ГОСТ 12.3.031-83 [29] «Работы со ртутью. Требования безопасности» п.2.1. все ртутьсодержащие отходы и вышедшие из строя приборы, содержащих ртуть, подлежат сбору и возврату для последующей регенерации ртути в специализированных организациях. В п.2.2. К работе по замене и сбору отработанных ртутьсодержащих ламп допускаются только электромонтеры. Главным условием при замене и сборе отработанных ртутьсодержащих ламп является сохранение герметичности. В п.2.13. Факт

сдачи ртутьсодержащих отходов подтверждается возвращением паспорта на вывоз отходов с отметкой о приеме представителя специализированного предприятия.

6.4 Безопасность в чрезвычайных ситуациях

6.4.1 Анализ вероятных ЧС, которые может инициировать объект исследований

В ходе проведения анализа не было выявлено ЧС, которые может инициировать объект исследования напрямую.

6.4.2 Анализ вероятных ЧС, которые могут возникнуть на рабочем месте при проведении исследований

К наиболее вероятным ЧС на рабочем месте можно отнести следующие: пожар (взрыв) в здании, авария на коммунальных системах жизнеобеспечения, землетрясение.

Наиболее вероятным ЧС является пожар. Источниками возгорания может стать электропроводка, внутренние работающие устройства ПК, взрывоопасные предметы в помещении исследователя согласно ГОСТ 12.1.044-2018 «Система стандартов безопасности труда. Пожаровзрывоопасность веществ и материалов. Номенклатура показателей и методы их определения» [30].

Поражающими факторами пожаров в помещении являются токсическое воздействие горючих материалов (в т.ч. отравление угарным газом), экстремальный нагрев среды, а также обломки и осколки при нарушении целостности конструкций здания [31].

6.4.3 Обоснование мероприятий по предотвращению ЧС и разработка порядка действий в случае возникновения ЧС

Согласно ГОСТ Р 22.3.03-94 [32], обеспечение безопасности людей в ЧС, обусловленных природными стихийными бедствиями, техногенными авариями и катастрофами, а также применением современного оружия (военные ЧС)

является общегосударственной задачей, обязательной для решения всеми территориальными, ведомственными и функциональными органами управления и регулирования, службами и формированиями, а также подсистемами, входящими в Российскую систему предупреждения и действий в чрезвычайных ситуациях (РСЧС).

Мероприятия по защите людей от источников ЧС должны планироваться в объемах, гарантирующих непревышение нормативного воздействия на них возможных поражающих факторов для расчетной ЧС.

Для защиты жизни и здоровья населения в ЧС следует применять следующие основные мероприятия гражданской обороны, являющиеся составной частью мероприятий РСЧС:

1) укрытие людей в приспособленных под нужды защиты населения помещениях производственных, общественных и жилых зданий, а также в специальных защитных сооружениях;

2) эвакуацию населения из зон ЧС;

3) использование средств индивидуальной защиты органов дыхания и кожных покровов;

4) проведение мероприятий медицинской защиты;

5) проведение аварийно-спасательных и других неотложных работ в зонах ЧС.

Мерами по предупреждению ЧС являются:

1) Соблюдение техники безопасности при работе с ПЭВМ. Использование только исправного оборудования.

2) Своевременное проведение ТО и ППР электроустановок согласно утвержденного графика и технических средств противопожарной защиты и пожаротушения.

3) Установка противопожарной сигнализации.

4) Своевременное проведение инструктажа рабочего персонала.

В случае угрозы возникновения ЧС (пожара) необходимо вызвать противопожарную службу, отключить электроэнергию и, следуя плану

эвакуации, эвакуировать находящихся в помещении людей и покинуть помещение. В случае, если очаг возгорания является небольшим, и нет угрозы поражения электрическим током, можно использовать углекислотные огнетушители ОУ-5 высокого давления с зарядом жидкой двуокиси углерода, согласно ГОСТ 8050-85 [33]. Расположение огнетушителей отмечено на плане эвакуации людей при пожаре и других ЧС из помещений учебного корпуса №22, отмеченном на рисунке 13.

Рисунок 13. План эвакуации при пожаре и других ЧС

Список публикаций и научных достижений

Участие в конференциях:

1) Diploma (First Place) - awarded to the best presentation at The 2nd International School on Heterogeneous Computing Infrastructure NEC'2017;

2) Диплом I степени за лучший доклад на Международной научно-технической конференции студентов, аспирантов и молодых ученых «Научная сессия ТУСУР», 2018 г.;

3) Диплом I степени на VIII международной научно-практической конференции «Инвестиции, строительство, недвижимость как материальный базис модернизации и инновационного развития экономики», 2018 г.;

4) Диплом I степени за доклад, представленный на XV Международной научно-практической конференции «Молодежь и современные информационные технологии», 2017 г.

Конкурсы НИР, выставки, кейсы, конкурсы научных проектов, чемпионаты, научные игры, workshop:

1) International workshop "Big Data Analysis for Smart Cities" - Best Project Diploma, 2017 г.;

2) II региональный конкурс «Открытые данные Томской области» – победа в номинации «Лучшее приложение на основе открытых данных»;

3) Диплом за III место в отборочном этапе Международного инженерного чемпионата «CASE-IN» студенческой лиги по направлению «Цифровой атом», 2019 г.;

4) Сертификат эксперта II Отборочных соревнований Digital Skills Томской области 2019.

Премии, звания, стипендии:

1) Стипендия Фонда Владимира Потанина 2018/2019;

2) Именная стипендия Президента РФ по ПНР на 2018-2019 уч. год;

3) Диплом лауреата премии Томской области в сфере образования, науки, здравоохранения и культуры 2018 г.;

4) Победа в конкурсе «Лучший студент ТПУ», 2018-2019 уч.г.;

5) Именная стипендия Правительства РФ по ПНР на весенний семестр 2017-2018 уч. года;

6) Сертификат I степени за победу в конкурсе на назначение повышенной государственной академической стипендии в номинации "За достижения в научно-исследовательской деятельности" – весна 2018/2019 учебного года;

7) Сертификат I степени за победу в конкурсе на назначение повышенной государственной академической стипендии в номинации "За достижения в научно-исследовательской деятельности" – осень 2018/2019 учебного года;

8) Сертификат III степени за победу в конкурсе на назначение повышенной государственной академической стипендии в номинации «За достижения в научно-исследовательской деятельности» – весна 2017/2018 учебного года;

9) Сертификат III степени за победу в конкурсе на назначение повышенной государственной академической стипендии в номинации «За достижения в научно-исследовательской деятельности» – осень 2017/2018 учебного года.

Научные стажировки:

1) Академический обмен в рамках программы FIRST+ (Finnish–Russian Student and Teacher Exchange Programme) между ТПУ и VAMK (г. Вааса, Финляндия);

2) Сертификат о прохождении международной летней школы ассоциации технических вузов России и Китая (Китай, г.Харбин, Харбинский Политехничейский Университет).

Участие в выполнении исследований по программам и грантам:

1) Грант РФФИ мол_а_вед 18-37-20003 «Разработка методов динамической структуризации и каталогизации метаинформации для больших научных сообществ»;

2) Грант РФФИ опн 19-011-31535 «Методика автоматизированного выявления угроз методами web mining»;

3) Грант Российского Научного Фонда №16-11-10280 «MetaMiner for BigData: Создание гетерогенной системы хранения метаинформации для научных экспериментов экзабайтного масштаба и применение методов «Машинного Обучения» для выявления нарушений при функционировании распределенных систем обработки и анализа больших данных».

Членство в научных сообществах (СНО, СИГРЭ и др.):

1) CERN Associated Member.

Публикации:

1) Kaida A., Golosova M., Grigorieva M., Gubin M. Development of DKB ETL module in case of data conversion // Journal of Physics: Conference Series. - 2018 - Vol. 1015, Article number 032055. - p. 1-5;

2) Golosova M., Aulov V., Kaida A. Metadata handling for Big Data projects // Journal of Physics: Conference Series. - 2018 - Vol. 1117, Article number 012007. - p. 1-8;

3) Golosova M.V., Aulov V.A., Grigorieva M.A., Kaida A.Y. Data Knowledge Base for the ATLAS collaboration // CEUR Workshop Proceedings. - 2018 - Vol. 2267. - p. 486-492;

4) Кайда А.Ю., Голосова М.В. Принципы организации автоматизируемых ETL-процессов на примере базы знаний научного эксперимента // Научная сессия ТУСУР–2018: материалы Международной

научно-технической конференции студентов, аспирантов и молодых ученых, в 5 частях, Томск, 16-18 Мая 2018. - Томск: В-Спектр, 2018 - Т. 3 - С. 254-256;

5) Кайда А. Ю. Социальные эффекты инвестиций в образование: общее и частное для стран ОЭСР // Инвестиции, строительство, недвижимость как материальный базис модернизации и инновационного развития экономики: материалы VIII Международной научно-практической конференции, в 2 ч., Томск, 13-15 Марта 2018. - Томск: ТГАСУ, 2018 - Т. 2 - С. 690-697;

6) Кайда А.Ю., Губин М.Ю. Импорт разнородных метаданных научного эксперимента в централизованное онтологическое хранилище // Молодежь и современные информационные технологии: сборник трудов XV Международной научно- практической конференции студентов, аспирантов и молодых ученых, Томск, 4-7 Декабря 2017. - Томск: ТПУ, 2018 - С. 46-47.

Список используемых источников

1. Cielen D., Meysman A., Ali M., Introducing Data Science. – USA: Manning Publications Co, 2016. – 322 p.
2. Day J.D., Zimmermann H., The OSI reference model // Proceedings of the IEEE Volume: 71, Issue: 12, Dec. 1983. pp.1334 – 1340 – [Электронный ресурс] URL: https://ieeexplore.ieee.org/document/1457043. Дата обращения: 01.04.2019.
3. Hall E., Internet Core Protocols: The Definitive Guide. – USA: O'Reilly Media, Inc., 2000. – 472 p.
4. Ibe O.C., Fundamentals of Data Communication Networks. – US: Wiley. – 209-227 p.
5. Grigorik I., High Performance Browser Networking. – USA: O'Reilly Media, Inc., 2013. – 400 p.
6. STOMP Protocol Specification, Version 1.2 – [Электронный ресурс] URL: https://stomp.github.io/stomp-specification-1.2.html. Дата обращения: 01.04.2019.
7. OASIS Advanced Message Queuing Protocol (AMQP) Version 1.0 – [Электронный ресурс] URL: http://docs.oasis-open.org/amqp/core/v1.0/amqp-core-complete-v1.0.pdf. Дата обращения: 01.04.2019.
8. Videla A., Williams J.W., RabbitMQ in action: Distributed messaging for everyone. – US: Manning Publications, 2012. – 312 p.
9. Apache ZooKeeper – [Электронный ресурс] URL: https://zookeeper.apache.org/. Дата обращения: 01.04.2019
10. Narkhede N., Shapira G., Palino T., Kafka: The Definitive Guide. – USA: O'Reilly Media, Inc., 2017. – 12-16 p.
11. CERN Computer Security Information – [Электронный ресурс] URL: https://security.web.cern.ch/security/home/en/index.shtml. Дата обращения: 01.04.2019

12. LXPLUS Service | IT Department – [Электронный ресурс] URL: http://information-technology.web.cern.ch/services/lxplus-service. Дата обращения: 01.04.2019

13. Golosova M.V., Aulov V.A., Grigorieva M.A., Kaida A.Y. Data Knowledge Base for the ATLAS collaboration // CEUR Workshop Proceedings. - 2018 - Vol. 2267. - p. 486-492

14. poll – [Электронный ресурс] URL: http://pubs.opengroup.org/onlinepubs/007908799/xsh/poll.html. Дата обращения: 01.04.2019

15. Processing: What to record? | CERN – [Электронный ресурс] URL: https://home.cern/science/computing/processing-what-record Дата обращения: 01.04.2019

16. Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 01.04.2019)

17. Федеральный Закон от 27.07.2006 N 152-ФЗ (ред. от 25.07.2011) «О Персональных Данных»

18. ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования

19. СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы (с изменениями на 21 июня 2016 года)

20. ГОСТ 12.0.003-2015 Система стандартов безопасности труда (ССБТ). Опасные и вредные производственные факторы. Классификация

21. СанПиН 2.2.4.548-96 Гигиенические требования к микроклимату производственных помещений

22. ГОСТ 12.1.003-2014 Система стандартов безопасности труда (ССБТ). Шум. Общие требования безопасности

23. СанПиН 2.2.1/2.1.1.1278-03 Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий (с изменениями на 15 марта 2010 года)

24. ГОСТ 12.1.019-2017 Система стандартов безопасности труда (ССБТ). Электробезопасность. Общие требования и номенклатура видов защиты

25. ТОИ Р-45-084-01 Типовая инструкция по охране труда при работе на персональном компьютере

26. ГОСТ 12.1.029-80 Система стандартов безопасности труда (ССБТ). Средства и методы защиты от шума. Классификация

27. КоАП РФ Статья 8.2 Несоблюдение экологических и санитарно-эпидемиологических требований при обращении с отходами производства и потребления, веществами, разрушающими озоновый слой, или иными опасными веществами

28. ГОСТ Р 56397-2015 Техническая экспертиза работоспособности радиоэлектронной аппаратуры, оборудования информационных технологий, электрических машин и приборов. Общие требования

29. ГОСТ 4658-73 Ртуть. Технические условия (с Изменениями N 1-6)

30. ГОСТ 12.1.044-2018 Система стандартов безопасности труда. Пожаровзрывоопасность веществ и материалов. Номенклатура показателей и методы их определения

31. ГОСТ Р 22.0.07-95 Безопасность в чрезвычайных ситуациях. Источники техногенных чрезвычайных ситуаций. Классификация и номенклатура поражающих факторов и их параметров

32. ГОСТ Р 22.3.03-94. Безопасность в чрезвычайных ситуациях. Защита населения. Основные положения

33. ГОСТ 8050-85 Двуокись углерода газообразная и жидкая. Технические условия (с Изменениями N 1, 2, с Поправкой)

Приложение А

Раздел 1
The technical overview

Студент:

Группа	ФИО	Подпись	Дата
8ПМ7И	Кайда Анастасия Юрьевна		

Консультант школы ИШИТР отделения ИТ:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Соколова Вероника Валерьевна	к.т.н.		

Консультант – лингвист отделения ИЯ школы ШБИП:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Диденко Анастасия Владимировна	к.ф.н.		

1 The technical overview

1.1 The general issue of big data handling

Today, the amount of data is growing dramatically. According to the statistics, more than 90 percent of the data in the world has been generated in the last few years [1].

Because of that fact, the concept of Big Data has been developed. Big Data is also a term for any large and complex data collection that cannot be processed using traditional data management techniques. It is characterized by four Vs: volume, variety, velocity, and veracity.

Traditional data processing infrastructures were not designed for today mobile, streaming, and online world. Conventional databases are designed around slow mechanical disk drives that cannot keep up with modern workloads. Large data volumes cause new challenges, such as overloaded memory and algorithms that never stop running. There are four main following issues:

1) The memory limit of a single machine does not allow to process huge data arrays using RAM.

2) Infinite loop processing stops general data pipelines.

3) The “bottleneck” process in a system does not allow a system to work with normal average processing speed.

4) The low overall speed of data processing causes a message queue overflow.

Thus, big data requires new approaches such as distributed computing systems and data partitioning. A single machine has the limitations of RAM, in other words, there is no reason to build a single-node cluster. Modern approaches consider using cloud or grid computing to prevent some limitations and special middleware to make dependencies between system components and processes weaker [2].

1.2 Data transfer protocols

All the existing protocols are classified according to the OSI Reference Model. The OSI Reference Model is a seven-layer model where the interprocess communication is subdivided into seven independent layers. It consists of the following layers:

1) The Physical Layer provides mechanical, electrical, functional, and procedural standards to access the physical medium.

2) The Data Link Layer is responsible for preventing errors occurrence in the Physical Layer, so on that layer, the data is transferred between network entities.

3) The Network Layer provides independence from the data transfer technology and relaying and routing considerations, and it handles relaying and routing data trough as many concatenated networks as necessary.

4) The Transport Layer provides data transfer between different end-points (systems).

5) The Session Layer is responsible for application checkpoint setting and session recovering, and allocation access rights.

6) The Presentation Layer manages data encoding and data decoding, so it transforms data into the relevant ready-to-transfer form by the application request; furthermore, the data content stays the same during the encoding and decoding process.

7) The Application Layer is the highest layer of the OSI Reference model. It is responsible for interprocess communication. This layer does not provide services to any other layer [3].

To define the requirements only transport layer protocols and application layer protocols were considered. The developed protocol is an application layer protocol. Despite this fact, the major transport layer protocols should be taken into account because application layer protocols are usually built over transport layer protocols, for example, HTTP over TCP.

TCP (Transmission Control Protocol) is a transport layer protocol. TCP provides a high level of reliability. To prevent the loss of packages TCP uses retransmission. To perform retransmission, TCP connection uses three-way handshake

(with SYN, SYN+ACK, ACK flags). If the application does not receive a packet with an acknowledgment (ACK) flag from a client, a server will send a packet again. A TCP server serves multiple clients simultaneously. When a connection request arrives, the server generates a child process and it will work with the requesting clients. The well-known port is used only for connection establishment.

UDP (User Datagram Protocol) is a transport layer protocol that uses datagrams. UDP is an unreliable protocol because it does not provide the warranty that all the sent datagrams will be delivered in the same sequence as they have been sent. The prioritized property of UDP is fault tolerance. Usually, UDP is used in connectionless iterative servers (CIS) when communication delay and jitter are more important. A CIS-server serves clients sequentially – one client at a time. Software such as online games should provide to its users all the services despite the fact that some datagrams can be lost. Instead of TCP, UDP offers high performance. If reliability comes at the expense of performance, then, conversely, performance can be gained by eliminating some of the overhead associated with reliability. Moreover, if an application requires to use broadcasts or multicasts to send data to multiple hosts simultaneously, then that application will have to use UDP [4].

SCTP is an expanded version of TCP. SCTP connection uses four-hand handshake as it is presented in figure 1. In contrast to TCP three-way handshake, a four-way handshake eliminates exposure to the TCP SYN flooding attacks. The receiver of the initial (INIT) contact message in a four-way handshake does not need to save any state information or allocate any resources. Instead, it responds with an INIT-ACK message, which includes a state cookie that holds all the information needed by the sender of the INIT-ACK to construct its state. The state cookie is digitally signed.

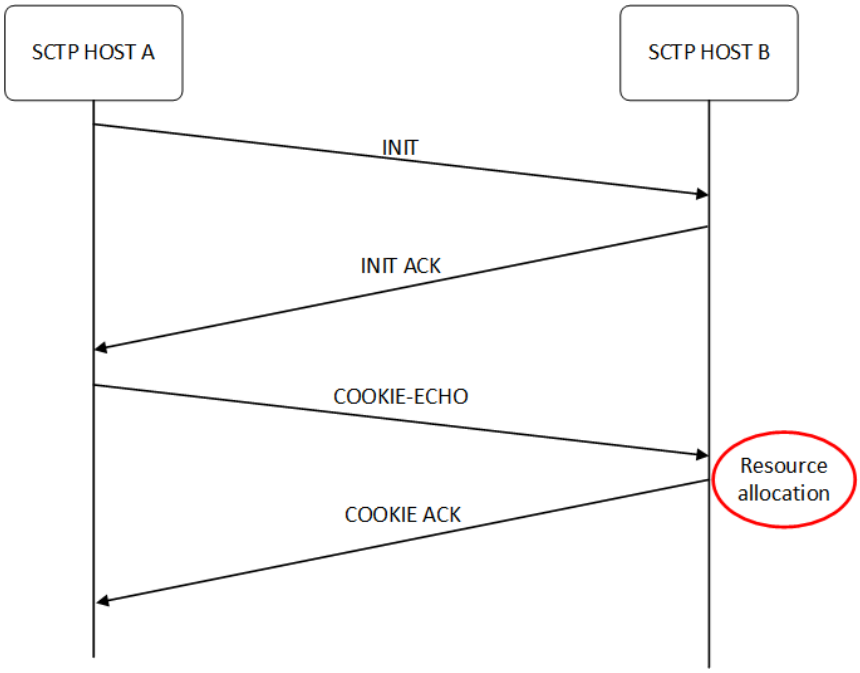

Figure 1. SCTP Four-way handshake

The main advantage of SCTP is multistreaming. In TCP, only one data stream per connection is allowed. It delivers all bytes of information one by one to keep the right order. SCTP allows multiple simultaneous data streams within one SCTP connection or association. Each message sent to a data stream can have a different final destination, but each must maintain message boundaries. This delivery scheme reduces unnecessary HOL blocking between independent streams of messages. SCTP streams are unidirectional channels, and within each channel, the data messages are usually transported in sequence. An application may also request a message to be delivered unordered, which can reduce blocking effects in case of message loss, since the reordering mechanism of one stream is not affected by that of another stream that may have to wait for a retransmission of a previously lost data chunk [5].

HTTP (The Hypertext Transfer Protocol) is a well-known application layer protocol that almost everybody uses surfing the Internet. It has had a rich history since the moment of its proposal by Tim Berners-Lee. HTTP has been established as the one-line protocol following four general features:

1) HTTP uses client/server architecture, implementing request-response communication that became one of the basic methods of a communication process between machines.

2) HTTP is an ASCII protocol, running over TCP/IP.

3) HTTP implements hypertext documents transfer.

4) HTTP closes the connection between client and server parts after every request.

After several years of evolution, modern HTTP allow pipelining (persistent HTTP) using existing connection multiple times strictly following FIFO method between multiple application requests. All modern browsers have an allowance to use it automatically as a secure and flexible solution.

The main disadvantage of this protocol is that HTTP 1.1 (current version) and HTTP 0.9 (initial version) are not the same. HTTP has evolved from a simple, one-line ASCII protocol to and an extended multiline protocol with request and response headers. Moreover, HTTP allows exchanging additional metadata. Today, almost all requests carry an additional 500-800 bytes of metadata. In the case of using HTTP-like protocol to process large volumes of data, the result can be dramatic [6].

AMQP is an application layer binary protocol created by iMatrix Corporation. JPMorgan Chase has initiated the development process. In 2004, message queuing and topologies issues were unsolved. AMQP became a powerful solution dynamically connecting the information in real time from any publisher to any interested consumer over a software bus.

However, this protocol has a critical disadvantage. AMQP does not provide a fixed ordering of messages. Thus, AMQP does not suit according to the general idea of the current R&D project [7].

STOMP is a text-oriented protocol designed as HTTP. The main advantage of this protocol is a simple way to write a client part. Moreover, even using Telnet will be enough to get access to any message broker supporting STOMP. Unfortunately, it is not supported by Apache Kafka message broker [8].

1.3 Message-oriented middleware

Before defining the requirements for the developed protocol, it is essential to analyze message-oriented middleware (MOM) that is a type of software that provides data transfer between applications using a communication channel and carrying messages as units of information. In a MOM-based communication environment, messages are usually sent and received asynchronously. Using message-based communications, applications are abstractly decoupled, senders and receivers are never aware of each other. These applications send and receive messages to and from the messaging system. In that case, the messaging system is responsible for the dataflow control; all messages should be delivered to the final destination [9].

Message queues are first-in, first-out (FIFO) data structures, which is also the natural way to process logs. Message queues organize data into user-defined topics, where each topic has its own queue.

While external processes (senders and receivers) are unaware of each other and its supervisor (the messaging system), the communication process should be regulated by the protocol developed for message-oriented middleware. The protocol should consist of a set of rules for external process state recognition. Moreover, it should provide a sustained data transfer between program elements of the system.

Message brokers are a subgroup of MOM that distributes messages from a sender to receivers. The core pattern of this kind of MOM is a Publish/subscribe pattern with three major entities: sender (publisher), a piece of data (message) and receiver (subscriber). This pattern considers that the publisher never sends any message to a specific direction, so message brokers as a central point have weak dependencies between processes.

RabbitMQ is a fault-tolerant reliable message broker that allows easy clustering implementation. RabbitMQ supports both STOMP and AMQP protocols. Moreover, RabbitMQ is the only broker implementing the AMQP open standard. Unfortunately, it does not support message rereading option that rejects to create processing nodes that will be used in several data pipelines. It also prevents idle logging implementation,

which is critical in the DKB project. In addition, RabbitMQ is too rigid for fast scalability.

This broker requires Erlang runtime environment that slows down the process of technological adoption in the developers' community [7]. The Prolog-based syntax is a big barrier in the learning process. Erlang supports concurrent object-oriented programming paradigm, and this approach causes many questions during the system design process.

Apache ActiveMQ is a message broker for remote communication between systems using the JMS (Java Message Service) specification. This broker has a wide range of advantages: completely customized security policy, Java application server integration, a number of client APIs besides Java, clustering support for scalability purposes and simple administration. It is suitable for heterogeneous API integration.

The critical disadvantage of ActiveMQ conception is a one-time delivery. RabbitMQ has the same weakness, because this broker does not keep a message inside the system once it has been delivered [10]. Fortunately, Apache has another message broker for some purposes – Apache Kafka. It is used in use cases where JMS, RabbitMQ, and AMQP may not even be considered due to volume and responsiveness, because Kafka is able to handle millions of messages per second [11].

The development of Kafka is related to LinkedIn which required a messaging system that contains features of both the monitoring and tracking systems. The initial team formulated the following goals:

1) To decouple producers and consumers by using push-pull model;

2) To provide persistence for message data within the messaging system that allows multiple consumers;

3) To optimize for high throughput of messages;

4) To allow for horizontal scaling of the system to grow as the data streams expand.

The developed publish/subscribed messaging system has a typical interface in comparison with other message brokers, but a storage layer has log-aggregation system features. Apache Kafka suits perfectly for use cases, which requires scalability.

According to the data report of August 2015, LinkedIn message system processed one trillion produced messages and over a petabyte of consumed data per day. This message broker is suitable for the following use cases:

1) Kafka tracks messages regarding actions the user is taking with frontend applications.

2) Kafka is used for messaging, where applications need to send notifications to users.

3) Kafka is suitable for collecting application and system metrics logs.

4) Kafka supports the concept of a commit log, so it monitors application streams to receive live updates.

5) Kafka processes data streams and operates on data in real time, as quickly as messages are produced. [12].

The main disadvantage is deployment complexity. It is important to note that Apache Kafka works jointly with Apache Zookeeper that is a centralized service for maintaining configuration information and name registries as well as providing distributed synchronization and various other services used by distributed systems. Unfortunately, it can be a "bottleneck" element of the system in case of writing because when writing data, ZooKeeper performs a consensus check among the ZooKeeper nodes [13].

Apache Kafka is ideally suited as the backbone of fast data architectures as a distributed, partitioned, replicated commit log service. Kafka does not delete messages once they have been read. Because of that, multiple consumers can see the whole log and process it [14].

1.4 The use of ETL as a complex solution for data integration tasks

Modern projects which operate with big data consider an option to work with heterogeneous data sources. Heterogeneous data infrastructures, especially those related to the scientific community, can be unstable and volatile; during an experiment lifetime, the number of data sources can change from time to time.

Today, Extract, Transform, and Load (ETL) techniques are the most popular ways to perform data integration solutions, and it is a proven method that many developers and companies rely on. There is a list of the following reasons to choose ETL as a core idea for the data integration system:

1) ETL has evolved over time to support emerging integration requirements for options like streaming data.

2) ETL follows the idea of message broker publish/subscribe pattern, so independent standalone modules can keep weak ties between elements.

3) ETL can provide batch scheduling and real-time capabilities.

4) ETL modules with extract or load function can play a significant role as an adapter that can increase the number of data source with which the whole system can work.

5) After extracting data, ETL uses pre-defined rules to transform the data into new formats.

6) ETL-chains can be easily changed without any changes in a system core.

Conclusion

To sum up, among the huge variety of different application layer protocols only a few suits the research task, but with some limitations. Message-oriented middleware has a subgroup with a variety of different message brokers. The majority of brokers support modern message-oriented application layer protocols, but implement one-time reading feature only. The aim of this overview is to help to formulate a list of requirements for the developed protocol.

The following ideas were highlighted:

1) Big data requires new approaches for data processing tasks.

2) On a transport layer, the preferred base is a TCP protocol, because the priority of TCP is reliability instead of performance as in UDP.

3) Modern protocols developed especially for MOM (STOMP, AMQP) cannot be accepted as a solution, because they do not work in a bind with Apache Kafka message broker, and furthermore, AMQP does not provide a fixed ordering of messages.

4) Kafka is a suitable message broker according to the main goals of the current R&D project, because it is fast and scalable, and it has a special logging layer.

5) ETL is a proven and reliable technique which keeps the integrity of the remaining components of complex systems.

References

1. How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read. – URL: https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read

2. Cielen D., Meysman A., Ali M., Introducing Data Science. – USA: Manning Publications Co, 2016. – 322 p.

3. Day J.D., Zimmermann H., The OSI reference model // Proceedings of the IEEE Volume: 71, Issue: 12, Dec. 1983. pp.1334 – 1340. – URL: https://ieeexplore.ieee.org/document/1457043.

4. Hall E., Internet Core Protocols: The Definitive Guide. – USA: O'Reilly Media, Inc., 2000. – 472 p.

5. Ibe O.C., Fundamentals of Data Communication Networks. – US: Wiley. – 209-227 p.

6. Grigorik I., High Performance Browser Networking. – USA: O'Reilly Media, Inc., 2013. – 400 p.

7. Videla A., Williams J.W., RabbitMQ in action: Distributed messaging for everyone. – US: Manning Publications, 2012. – 312 p.

8. STOMP – URL: https://stomp.github.io/

9. Chappell D.A., Enterprise Service Bus. – USA: O'Reilly Media, Inc., 2004. – 288 p.

10. Snyder B., Davies R., Bosanac D., ActiveMQ in Action. – USA: Manning Publications, 2011. – 408 p.

11. What is Kafka? – DZone Big Data. – URL: https://dzone.com/articles/what-is-kafka

12. Narkhede N., Shapira G., Palino T., Kafka: The Definitive Guide. – USA: O'Reilly Media, Inc., 2017. – 12-16 p.

13. Wampler D., Fast Data Architectures for Streaming Applications. – USA: O'Reilly Media, Inc., 2016. – 47 p.

14. Carson C., Suchter S., Effective Multi-Tenant Distributed Systems. – USA: O’Reilly Media, Inc., 2016. – 102 p.

15. What is ETL? | SAS – URL: https://www.sas.com/en_sg/insights/data-management/what-is-etl.html

Приложение Б

custom_readline.py

```
def custom_readline(f, newline):
    """Custom readline() function.
    Custom_readline() separates content from a text file 'f'
    by delimiter 'newline' to distinct messages.
    The last line can be incomplete, if the input data flow is interrupted
    in the middle of data writing.
    Keyword arguments:
    f -- file/stream to read
    newline -- custom delimiter
    """
    poller = select.poll()
    poller.register(f, select.POLLIN)
    flags = fcntl.fcntl(f.fileno(), fcntl.F_GETFL)
    fcntl.fcntl(f.fileno(), fcntl.F_SETFL, flags | os.O_NONBLOCK)
    buf = ""
    while True:
        if poller.poll(500):
            chunk = f.read()
            if not chunk:
                yield buf
                break
            buf += chunk
        while newline in buf:
            pos = buf.index(newline)
            yield buf[:pos]
            buf = buf[pos + len(newline):]
```

AbstractStage.py (фрагменты кода)

```
class AbstractStage(object):
    """
    Class/instance variable description:
    * Argument parser (argparse.ArgumentParser)
        __parser
    * Parsed arguments (argparse.Namespace)
        ARGS
    """

    def __init__(self, description="DKB Dataflow stage"):
        """ Initialize the stage
        * set description
        * define common arguments (--mode, ...)
        * ...
        """
        self.ARGS = None
        self.__parser = argparse.ArgumentParser(
            formatter_class=argparse.RawDescriptionHelpFormatter,
            description=description,
            epilog=textwrap.dedent(
                '''\
                NOTES                    ''')
        )
        self.defaultArguments()

    def defaultArguments(self):
        """ Config argument parser with parameters common for all stages. """
        self.add_argument('-e', '--end-of-message', action='store', type=str,
```

```
                help=u'Custom end of message marker.',
                nargs='?',
                default=None,
                dest='eom'
                )
    self.add_argument('-E', '--end-of-process', action='store', type=str,
                help=u'Custom end of process marker.',
                nargs='?',
                default=None,
                dest='eop'
                )

def add_argument(self, *args, **kwargs):
    """ Add specific (not common) arguments. """
    self.__parser.add_argument(*args, **kwargs)

def parse_args(self, args):
    """ Parse arguments and set dependant arguments if needed. """
    self.ARGS = self.__parser.parse_args(args)
    if not self.ARGS.mode:
        raise ValueError(
            "Parameter -m|--mode must be used with value: -m MODE.")

    if self.ARGS.eom is None:
        self.ARGS.eom = '\n'
    elif self.ARGS.eom == '':
        raise ValueError("(ERROR) Empty EOM marker specified.")
    else:
        try:
            self.ARGS.eom = self.ARGS.eom.decode('string_escape')
```

```
        except (ValueError), err:
            sys.stderr.write("(ERROR) Failed to read arguments.\n"
                             "(ERROR) Case: %s\n" % (err))
            sys.exit(1)

    if self.ARGS.eop is None:
        if self.ARGS.mode == 's':
            self.ARGS.eop = '\0'
        else:
            self.ARGS.eop = ''
    else:
        try:
            self.ARGS.eop = self.ARGS.eop.decode('string_escape')
        except (ValueError), err:
            sys.stderr.write("(ERROR) Failed to read arguments.\n"
                             "(ERROR) Case: %s\n" % (err))
            sys.exit(1)

def print_usage(self, fd=sys.stderr):
    """ Print usage message. """
    self.__parser.print_usage(fd)

def run(self):
    """ Run the stage. """
    raise NotImplementedError("Stage method run() is not implemented")
```

Stage 010: papersFromGLANCE.sh (фрагменты кода)

```
#!/usr/bin/env bash

usage () {
  echo "
Usage info
"
}

EOMessage='\n'

EOP_set="N"
EOM_set="N"

while [[ $# > 0 ]]
do
  key="$1"
  case $key in
    -h|--help)
      usage
      exit
      ;;
    -f|--file)
      FILE="$2"
      shift
      ;;
    -p|--pipe)
      PIPE="$2"
      shift
      ;;
```

```
  -r|--retry)
    RETRY="YES"
    ;;
  -u|--user)
    USR="$2"
    shift
    ;;
  -t|--tmp)
    CLEAN="NO"
    ;;
  -e|--eom)
    EOM="$2"
    EOM_set="Y"
    shift
    ;;
  -E|--eop)
    EOP="$2"
    EOP_set="Y"
    shift
    ;;
  -*)
    echo "Unknown option: $key" >&2
    usage >&2
    exit 1
    ;;
  *)
    break
    ;;
esac
shift
```

```
done

[ -z "$CLEAN" ] && CLEAN="YES"
if [ -z "$USR" ] ; then
  echo "Username is not specified." >&2
  usage >&2
  exit 1
fi

if [ -n "$PIPE" ]; then
  if ! ( [ -p "$PIPE" ] || mkfifo "$PIPE" 2>&1 > /dev/null ) ; then
    echo "Can not create a FIFO named pipe: $PIPE. Exiting." >&2
    exit 2
  fi
  FILE="$PIPE"
fi

if [ -n "$FILE" ] || [ -n "$PIPE" ]
then
  EOProcess=""
else
  EOProcess="\0"
fi

[ "$EOM_set" == "Y" ] && EOMessage="$EOM"
[ "$EOP_set" == "Y" ] && EOProcess="$EOP"

if [ -z "$EOMessage" ] ; then
  echo "EOM marker is not specified. Exiting." >&2
  exit 1
```

```
fi

#Glance request
# Check Kerberos ticket
# Lxplus addressing. Options description. SSH command executing.

tmp=/tmp/list_of_papers.json
#scp command

echo -ne "$EOMessage" >> $tmp

if [ -n "$FILE" ] ; then
  cat $tmp > $FILE
else
  cat $tmp
fi

echo -ne "$EOProcess"

#clear and execute a remote command
```