

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

---

Институт кибернетики  
Направление подготовки 09.03.02 Информационные системы и технологии  
Кафедра Информационных систем и технологий

**БАКАЛАВРСКАЯ РАБОТА**

Тема работы
<b>Разработка программной системы поддержки развёртывания распределённых приложений</b>

УДК 004.415:004.75

Студент

Группа	ФИО	Подпись	Дата
8ИЗБ	Журавлев Сергей Сергеевич		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент кафедры ИСТ ИК ТПУ	Мирошниченко Евгений Александрович	к.т.н., доцент		

**КОНСУЛЬТАНТЫ:**

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент кафедры Менеджмента. ИСТ. ТПУ	Рахимов Тимур Рустамович	к.э.н., доцент		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент кафедры ЭБЖ	Акулов Петр Анатольевич	-		

**ДОПУСТИТЬ К ЗАЩИТЕ:**

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
Доцент кафедры ИСТ ИК ТПУ	Мальчуков Андрей Николаевич	к.т.н., доцент		

Томск 2017 г.

**ЗАПЛАНИРОВАННЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ОСНОВНОЙ  
ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЕ ПОДГОТОВКИ БАКАЛАВРОВ 09.03.02  
«ИНФОРМАЦИОННЫЕ  
СИСТЕМЫ И ТЕХНОЛОГИИ», ИК ТПУ, ПРОФИЛЬ «ГЕОИНФОРМАЦИОННЫЕ  
СИСТЕМЫ»**

Код результата	Результат обучения (выпускник должен быть готов)
P1	Применять базовые и специальные естественно научные и математические знания для комплексной инженерной деятельности по созданию, внедрению и эксплуатации геоинформационных систем и технологий, а также информационных систем и технологий в бизнесе.
P2	Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач.
P3	Ставить и решать задачи комплексного анализа, связанные с созданием геоинформационных систем и технологий, информационных систем в бизнесе, с использованием базовых и специальных знаний, современных аналитических методов и моделей.
P4	Выполнять комплексные инженерные проекты по созданию информационных систем и технологий, а также средств их реализации (информационных, методических, математических, алгоритмических, технических и программных).
P5	Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретация полученных данных, в области создания геоинформационных систем и технологий, а также информационных систем и технологий в бизнесе.
P6	Внедрять, эксплуатировать и обслуживать современные геоинформационные системы и технологии, информационные системы и технологии в бизнесе, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.
<i>Универсальные компетенции</i>	
P7	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности.
P8	Осуществлять коммуникации в профессиональной среде и в обществе в целом. Владеть иностранным языком (углублённый английский язык), позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности.
P9	Эффективно работать индивидуально и в качестве члена команды, состоящей из специалистов различных направлений и квалификаций.

P10	Демонстрировать личную ответственность за результаты работы и готовность следовать профессиональной этике и нормам ведения комплексной инженерной деятельности.
P11	Демонстрировать знания правовых, социальных, экологических и культурных аспектов комплексной инженерной деятельности, а также готовность к достижению должного уровня физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности.

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

---

Институт кибернетики  
Направление подготовки 09.03.02 «Информационные системы и технологии»  
Кафедра информационных систем и технологий

УТВЕРЖДАЮ:  
Зав. кафедрой  
\_\_\_\_\_ Мальчуков А.Н.  
(Подпись) (Дата) (Ф.И.О.)

**ЗАДАНИЕ**  
**на выполнение выпускной квалификационной работы**

В форме:

Бакалаврской работы
---------------------

Студенту:

Группа	ФИО
8ИЗБ	Журавлеву Сергею Сергеевичу

Тема работы:

<b>Разработка программной системы поддержки развёртывания распределённых приложений</b>	
Утверждена приказом директора (номер, дата)	№ 664/с 03.02.2017

Срок сдачи студентом выполненной работы:

29.05.2017

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ:**

<b>Исходные данные к работе</b>	Спроектировать и реализовать систему поддержки развёртывания распределённых приложений; система должна обеспечивать функционал, необходимый для развёртывания, управления и поддержания работоспособности распределённого приложения.
<b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b>	<ul style="list-style-type: none"><li>– Выявление цели и задач исследования</li><li>– Анализ предметной области</li><li>– Проектирование информационной системы</li><li>– Выбор программных компонент и технологий реализации</li><li>– Реализация запланированного функционала</li><li>– Финансовый менеджмент, ресурсоэффективность и ресурсосбережение</li><li>– Социальная ответственность</li><li>– Заключение по работе</li></ul>
<b>Перечень графического материала</b>	
<b>Консультанты по разделам выпускной квалификационной работы</b>	

<b>Раздел</b>	<b>Консультант</b>
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Рахимов Тимур Рустамович
Социальная ответственность	Акулов Пётр Анатольевич

<b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b>	05.09.2016
---	------------

**Задание выдал руководитель:**

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
Доцент кафедры ИСТ	Мирошниченко Е. А.	к. т. н.		05.09.2016

**Задание принял к исполнению студент:**

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
8ИЗБ	Журавлев Сергей Сергеевич		05.09.2016

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

---

Институт кибернетики  
Направление подготовки 09.03.02 «Информационные системы и технологии»  
Уровень образования бакалавриат  
Кафедра информационных систем и технологий  
Период выполнения осенний / весенний 2016/2017 учебного года

Форма представления работы:

Бакалаврская работа
---------------------

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН**  
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	29.05.2017
--	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
20.05.17	Основная часть	75
22.05.17	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
22.05.17	Социальная ответственность	10

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент кафедры ИСТ	Мирошниченко Е. А.	К. Т. Н.		

**СОГЛАСОВАНО:**

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
ИСТ	Мальчуков А.Н.	-		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И  
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
8ИЗБ	Журавлев Сергей Сергеевич

Институт	ИК	Кафедра	ИСТ
Уровень образования	Бакалавр	Направление/специальность	Информационные системы и технологии

**Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:**

<ol style="list-style-type: none"> <li>1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</li> <li>2. Нормы и нормативы расходования ресурсов</li> <li>3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</li> </ol>	<p>Человеческие ресурсы: 1 чел.</p>
---	-------------------------------------

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

<ol style="list-style-type: none"> <li>1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения</li> </ol>	<p>Проведена оценка коммерческого потенциала:</p> <ol style="list-style-type: none"> <li>1. Потенциальные потребители результатов исследования.</li> <li>2. Анализ конкурентных технических решений</li> <li>3. QUAD-анализ</li> <li>4. SWOT-анализ</li> </ol>
<ol style="list-style-type: none"> <li>2. Планирование и формирование бюджета научных исследований</li> </ol>	<p>Произведен расчет бюджета научных исследований.</p>
<ol style="list-style-type: none"> <li>3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования</li> </ol>	<p>Определена ресурсная, финансовая, бюджетная эффективность исследования посредством расчета интегрального финансового показателя, интегрального показателя ресурсоэффективности и эффективности.</p>

**Перечень графического материала (с точным указанием обязательных чертежей):**

<ol style="list-style-type: none"> <li>1. Оценка конкурентоспособности технических решений</li> <li>2. Матрица QUAD</li> <li>3. Матрица SWOT</li> <li>4. Альтернативы проведения НИ</li> <li>5. График проведения и бюджет НИ</li> <li>6. Оценка ресурсной, финансовой и экономической эффективности НИ</li> </ol>
--

<b>Дата выдачи задания для раздела по линейному графику</b>	
---	--

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Рахимов Тимур Рустамович	К.э.н., доцент		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8ИЗБ	Журавлев Сергей Сергеевич		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»**

Студенту:

<b>Группа</b>	<b>ФИО</b>
8ИЗБ	Журавлев Сергей Сергеевич

<b>Институт</b>	Кибернетики	<b>Кафедра</b>	Информационных систем и технологий
<b>Уровень образования</b>	Бакалавр	<b>Направление/специальность</b>	09.03.02 «Информационные системы и технологии»

**Исходные данные к разделу «Социальная ответственность»:**

1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Разработка программной системы поддержки развёртывания распределённых приложений Разработанная программная система позволит развертывать распределенные приложения на произвольные сетевые инфраструктуры. Разработка планируется к внедрению в АО «Атомик Софт».
--	--

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

<b>1. Производственная безопасность</b> 1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения 1.2. Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения	1.1 Анализ вредных факторов: • Нарушение параметров микроклимата; • Недостаточная освещенность рабочей зоны/ 1.2 Анализ опасных факторов: • Повышенный уровень статического электричества; • Повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека.
<b>2. Экологическая безопасность:</b>	2.1 Проведен анализ отходов, образующихся в процессе работы 2.2 Рассмотрена защита литосферы
<b>3. Безопасность в чрезвычайных ситуациях:</b>	3.1 Рассмотрена наиболее вероятная чрезвычайная ситуация – пожар • Исключение условий образования горючей среды; • Исключение источников зажигания. 3.2 Рассмотрен пожар в вычислительных центрах, так как там выполняются тяжелые вычисления

<p><b>4. Правовые и организационные вопросы обеспечения безопасности</b></p>	<p>4.1 Приведен анализ специальных правовых норм, рабочего места и в особенности при работе с ПЭВМ.</p> <ul style="list-style-type: none"> <li>• Трудовой кодекс Российской Федерации" от 30.12.2001 N 197-ФЗ (ред. от 30.12.2015).</li> </ul> <p>4.2 Влияние разработанной системы на работу пользователя.</p>
--	---

<p><b>Дата выдачи задания для раздела по линейному графику</b></p>	
--	--

-

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата

## РЕФЕРАТ

Выпускная квалификационная работа 72 с., 19 рис., 16 табл., 27 источников, 3 прил.

Ключевые слова: информационная система, распределенные приложения, система поддержки распределенных приложений, библиотека сетевого взаимодействия, protocol buffer, Qt Creator.

Объектом исследования является разработка программной системы поддержки развёртывания распределённых приложений.

В процессе исследования проводились изучение предметной области, проектирование и разработка сетевой части системы поддержки развёртывания распределенных приложений.

В результате исследования была спроектирована и разработана система поддержки развёртывания распределенных приложений.

Основные конструктивные, технологические и технико-эксплуатационные характеристики: поддержка платформ Linux, Windows.

Степень внедрения: планируется к внедрению на предприятии АО «Атомик Софт».

Область применения: развертывание, управление поддержание работоспособности распределенных приложений.

В будущем планируется добавление вычислительных компонентов, в зависимости от отрасли применения приложения, добавить.

## ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ

**Программный интерфейс приложения (API)** – набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) или операционной системой для использования во внешних программных продуктах.

**Интегрированная среда разработки (IDE)** – система программных средств, используемая программистами для разработки программного обеспечения.

**Протокол управления передачей (TCP)** – протокол передачи данных, предназначенный для управления передачей данных.

**Унифицированный язык моделирования (UML)** – язык графического описания для объектного моделирования в области разработки программного обеспечения.

**Protocol Buffers (protobuf)** – бинарный протокол сериализации данных.

## Содержание

ТЕХНИЧЕСКОЕ ЗАДАНИЕ .....	13
Наименование продукта.....	13
Краткая характеристика области применения .....	13
Назначение разработки .....	13
Требования к программной документации .....	15
Стадии и этапы разработки .....	15
Порядок контроля и приемки .....	15
ВВЕДЕНИЕ.....	16
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	18
Структура работ по реализации .....	20
2. ВЫБОР СРЕДСТВ РАЗРАБОТКИ .....	21
3. РЕАЛИЗАЦИЯ.....	24
Типы разработанных исполняемых файлов .....	24
Работа с разработанной системой.....	24
Конфигурационные файлы .....	25
Описание схемы .....	25
Схема развертывания .....	27
Работа с клиентским приложением терминал .....	28
Работа с приложением сетевого узла.....	31
Компоненты системы.....	32
1. Реализация библиотеки сетевого взаимодействия link .....	32
2. Реализация библиотеки сетевого взаимодействия protolink.....	36
3. Реализация маршрутизации.....	37
4. Реализация сетевой часть клиентского приложения.....	39
ЗАКЛЮЧЕНИЕ .....	65
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....	66

## **ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

### **Наименование продукта**

Система поддержки распределенных приложений.

### **Краткая характеристика области применения**

Информационная система предназначена для использования сотрудниками предприятий, нуждающихся в распределенных приложениях.

### **Документ, на основании которого ведется разработка**

Выпускная квалификационная работа бакалавра.

### **Организация, утвердившая документ**

Государственное образовательное учреждение высшего профессионального образования «Национальный исследовательский Томский политехнический университет».

### **Назначение разработки**

Система позволит разворачивать распределённые приложения, состоящие из нескольких компонентов и физически расположенных на разных сетевых узлах.

Пользователю предоставляется возможность разворачивать распределенное приложение в соответствии с конфигурацией, стартовать работу приложения, останавливать работу, реконфигурировать, просматривать статистику на узлах.

Основной целью разрабатываемой системы является создание единой информационной среды и автоматизация технологических процессов предприятия.

### **Требования к функциональным характеристикам**

Сетевое взаимодействие должно происходить по специальному бинарному протоколу поверх TCP/IP.

Узлы должны иметь возможность посылать друг другу 3 вида сообщений:

- просто сообщение
- запрос
- ответ

Возможность описания схемы развертывания, согласно которой распределенное приложение может быть развернуто на различные сетевые инфраструктуры без изменения самого описания распределенного приложения.

Для клиентского приложения должны быть функции:

1. Выбор существующей схемы описания приложения
2. Выбор существующей схемы развертывания
3. Создание схемы описания приложения
4. Создание схемы описания развертывания
5. Просмотр и редактирование схем описания и развертывания приложений
6. Развертка приложения в соответствие с конфигурацией, то есть на основе выбранных схем
7. Старт работы распределенного приложения
8. Остановка работы приложения
9. Просмотр статистики
10. Просмотр логов

#### **Требования к надежности**

Пакеты, посылаемые от одного узла к другому не должны теряться.

#### **Требования к составу и параметрам технических средств**

К компьютеру, на котором будет использоваться приложение сетевого узла либо клиентское приложение предъявляются следующие минимальные системные требования:

- Процессор с тактовой частотой 1.6 ГГц;
- Не менее 512 МБ оперативной памяти;
- Не менее 50 МБ свободного места на жестком диске;

- Клавиатура, мышь;
- Видеокарта и монитор.

### **Требования к исходным кодам и языкам программирования**

Исходный код программы должен быть написан на языке программирования C++.

### **Требования к информационной и программной совместимости**

Система должна запускаться на платформах Windows и Linux.

### **Требования к программной документации**

В комплект программной документации должно входить руководство пользователя.

### **Стадии и этапы разработки**

- Составление технического задания
- Выбор средств разработки
- Реализация
- Тестирование

### **Порядок контроля и приемки**

Контроль и приемка должны осуществляться в соответствии с документом «Техническое задание».

## ВВЕДЕНИЕ

Промышленное производство в различных отраслях использует множество датчиков, также площади производства часто бывают очень большими, в результате чего возникает потребность в распределенных приложениях, так как необходимо собирать данные с датчиков, распределенных в пространстве на большой территории, обрабатывать данные в специальных дата-центрах.

Для решения данной проблемы целесообразно использовать приложение, работающее как единое целое, но распределенное в пространстве. Так как задачи, которые ставятся перед таким приложением зависят от отрасли промышленности, в которой оно используется, имеет смысл создать единый каркас для таких приложений, систему которая будет использоваться для развертывания и управлением работой таких приложений.

Система позволит развертывать распределенные приложения на произвольные сетевые инфраструктуры. Основное отличие от обычных приложений состоит в том, что описание приложения должно быть не связано с той инфраструктурой, на которой оно развертывается, так как, требуется возможность развертывать одно и то же приложение на разные инфраструктуры. Разработка планируется к внедрению в АО «Атомик Софт».

**Цель:** Разработать и протестировать систему поддержки развёртывания распределённых приложений.

**Задачи:**

- Выбрать программные средства для разработки;
- Спроектировать архитектуру;
- Разработать приложение.

### **Личный вклад:**

Так как проект выполнялся в команде, необходимо отметить личный вклад. Личный вклад состоял из следующих модулей: Link, ProtoLink, реализующих сетевое взаимодействие в приложении, роутинга, т.е. реализация отправки, доставки и маршрутизации данных между логическими портами экземпляров компонентов, реализация развертывания, старта и остановки работы распределенного приложения в соответствии со схемой конфигурации на клиентском приложении, проведение тестирования системы на ОС Windows.

## 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Область информатики, изучающая распределенные системы, то есть системы, в которой компоненты, расположенные на сетевых компьютерах, обмениваются и координируют свои действия путем передачи сообщений.

Приложение, работающее в распределенной системе, называется распределенным приложением. [1]

Распределенное приложение состоит из одного или нескольких локальных или удаленных клиентов, которые обмениваются данными с одним или несколькими серверами на нескольких машинах, связанных через сеть. [2]

Система поддержки распределенных приложений – такая система, задача которой состоит в управлении работой распределенного приложения, а также поддержке работоспособности распределенного приложения.

Основное отличие от обычных приложений состоит в том, что описание приложения не должно быть связано с той инфраструктурой, на которой оно разворачивается, так как, требуется возможность разворачивать одно и то же приложение на разные инфраструктуры.

Распределенное приложение отличается от сосредоточенного следующими характеристиками:

- За счет распределения вычислительных задач достигается большее быстродействие;
- При отказе одного из узлов, можно перераспределить компоненты на другие рабочие, что гарантирует повышенную надежность и устойчивость к сбоям;
- Масштабируемость т.е. простое наращивание функционала приложения;
- Простая реконфигурация приложения;
- Возможность распределения вычислительной нагрузки между узлами.

По некоторым причинам невозможно выполнить все вычислительные процессы на одной машине. Рассмотрим возможные причины:

- Географическая удаленность (например, датчики);

- Тяжелые вычисления и невозможность выполнить все вычисления на одной машине;
- Требования безопасности (например, имеется требование, чтобы база данных располагалась на отдельной машине);
- Пониженные требования к операционным системам, установленным на узлах, так как на каждом узле может стоять своя операционная система.

Распределенные приложения начали привлекать интерес в результате увеличения количества датчиков, расширения площадей производства, повышения потребности в автоматизации технологических процессов. Компоненты распределенного приложения разнесены в пространстве.

Распределенные приложения описывают решения по автоматизации в таких отраслях промышленности: металлургия, электроснабжения, нефтедобычи и нефтепереработки, химическое производство, системы управления дорожно-транспортным движением.

Рассмотрим на примере процессов добычи и транспорта нефти и газа, для которых характерна рассредоточенность объектов в пространстве: нагнетательные и добывающие скважины, замерные установки, насосные станции, участки нефте-газо-проводов.

Поиск систем поддержки распределенных приложений не дал результатов, так как обычно распределённые приложения представляют ограниченный набор компонентов и не имеют API для разработки собственных.

Основная концепция работы системы поддержки развёртывания распределённых приложений:

1. Приложение разбивается на компоненты
2. Описываются связи между компонентами
3. Приложение развертывается на определенную инфраструктуру

Компонент должен иметь входные и выходные порты.

Разрабатываемое приложение должно содержать две схемы:

1. Схема описания приложения – схема должна описывать компоненты и связи между компонентами

2. Схема развертывания приложения – схема должна описывать сетевые узлы и отображение компонентов на узлы.

Задачи, которые необходимо решить:

- Создание компонентов и протокола их коммуникации;
- Реализация средств для работы и взаимодействия сетевых узлов
  - Сетевое взаимодействие;
  - проработка протоколов передачи данных низкого уровня;
  - поддержка соединения, детектирование обрывов связи и автоматическое восстановление связи;
  - организация механизмов запрос-ответ;
- Разработка средств поддержки и мониторинга приложения.

### **Структура работ по реализации**

- Разработать средства для описания распределенного приложения
  - Разработать формат описания распределенного приложения
  - Разработать приложение для визуального редактирования
- Разработать средства описания схемы развертывания приложения на реальную сетевую инфраструктуру и средства собственно развертывания
  - Разработать формат описания схемы развертывания
  - Разработать средства для создания схем развертывания и собственно развертывания
- Разработать компоненты для запуска на сетевых узлах, которые бы позволили им осуществлять сетевое взаимодействие в событийном виде
  - Разработать библиотеку для сетевого взаимодействия между узлами по ТСР/IP
  - Разработать эффективный протокол взаимодействия
  - Разработать серверное приложение для узлов сети

## 2. ВЫБОР СРЕДСТВ РАЗРАБОТКИ

При выборе среды разработки рассматривались следующие популярные среды разработки (IDE):

- Microsoft Visual Studio[16] – линейка продуктов от компании Microsoft, включающая интегрированную среду разработки программного обеспечения.
- Qt Creator[8] – кроссплатформенная IDE для разработки на языках программирования C, C++.

Таблица 1 – Сравнительная таблица IDE

Критерии сравнения	Microsoft Visual Studio	Qt Creator
Разработка для ОС	Windows	Linux, Windows, Mac OS
Поддерживаемые языки программирования	Visual Basic, C, C++, C#, CSS, HTML, HTML5, JavaScript, Python, Ruby	C, C++
Функционал	Редактор исходного кода, отладчик	Редактор исходного кода, отладчик
Требования к ОС	Windows	Linux, Windows, Mac OS
Лицензия	Наличие студенческой лицензии	бесплатно
Наличие инструментов проектирования пользовательского интерфейса	есть	есть
Возможность командной разработки	есть	есть

Критерии сравнения	Microsoft Visual Studio	Qt Creator
Наличие обучающей литературы	есть[9]	есть[8]

В таблице 1 приведены некоторые критерии сравнения рассматриваемых сред разработки. Можно сказать, что для кроссплатформенной разработки стоит использовать Qt Creator, так как Microsoft Visual Studio позволяет вести разработку только на операционной системе Windows. Кроме того исходя из личного впечатления, документация для Qt[8] намного удобнее чем для Visual Studio[9]. Поэтому, в качестве среды разработки был выбран Qt Creator.

При выборе библиотек для реализации модулей сетевого взаимодействия рассматривались следующие наиболее популярные библиотеки:

- POCO [11]
- Boost::Asio [12]

Таблица 2 – Сравнительная таблица библиотек работы с сетью

Критерии сравнения	POCO	Boost::Asio
Протоколы транспортного уровня	TCP, UDP	TCP, UDP
Поддержка SSL	есть	есть
Поддерживаемые версии протокола ip	IPv4, IPv6	IPv4, IPv6
Поддержка асинхронных операций	есть	есть
Поддерживаемые платформы	Windows, Mac OS, Linux	Windows, Mac OS, Linux
Отложенные операции	есть	есть
Стоимость	бесплатно	бесплатно

В таблице 2 приведены некоторые критерии сравнения библиотек работы с сетью, как видно обе библиотеки обладают близким функционалом, однако Boost::Asio является лишь одной из библиотек собрания Boost [14], которая предоставляет множество других полезных библиотек, например Boost::Signals2 [15], реализующие концепцию сигналов и слотов, которую удобно использовать при разработке библиотеки сетевого взаимодействия. Помимо перечисленного выше по библиотеке Boost::Asio проще найти документацию, и на таких популярных сервисах как stackoverflow, она является более обсуждаемой, чем POCO. Кроме того библиотека Boost::Asio предоставляет API на более низком уровне, а низкоуровневый код предоставляет больше возможностей для реализации.

В результате анализа для разработки библиотеки сетевого взаимодействия была выбрана библиотека Boost::Asio.

В качестве протокола сериализации данных использовался Protocol Buffer [4] также именуемый как protobuf от компании Google.

Данный протокол используется в таких крупных компаниях как:

- Google [3]
- Twitter Inc.[5]
- Apache Software Foundation[6]
- OpenStreetMap Foundation[7]

Помимо protobuf рассматривались также форматы XML и JSON. Преимущества protobuf в сравнении с XML [4]:

- Проще;
- Получаемые пакеты от 3 до 10 раз меньше;
- Сериализация от 20 до 100 раз быстрее;
- Позволяет генерировать классы доступа к данным, которые проще использовать программно.

### **3. РЕАЛИЗАЦИЯ**

#### **Типы разработанных исполняемых файлов**

Система поддержки распределенных приложений состоит из двух типов исполняемых файлов:

- Приложение сетевого узла – консольное приложение, без пользовательского интерфейса, при запуске которого необходимо указывать конфигурационный порт.
- Клиентское приложение «Терминал» – используется для развертки приложения в соответствие с конфигурационными схемами на сетевые узлы, общается с приложениями сетевого узла через конфигурационные порты.

#### **Работа с разработанной системой**

Для работы распределенного приложения, необходимо:

1. Разбить распределенное приложение на компоненты, так чтобы оно могло быть представлено в виде графа компонентов;
2. Распределить компоненты по сетевым узлам;
3. Составить схемы описания приложения и схемы развертывания приложения, представляющие конфигурацию, то есть результат 1 и 2 пунктов;
4. Запустить приложения сетевого узла на машинах с указанием конфигурационного порта;
5. Загрузить составленные схемы в приложение терминала;
6. Развернуть подготовленную конфигурацию на узлы ;
7. Запустить работу распределенного приложения;
8. По окончании работы, можно остановить работу распределенного приложения.

## Конфигурационные файлы

Далее рассмотрим конфигурационные файлы распределенного приложения. Они представляются двумя XML-файлами:

- Описание схемы – xml файл, который описывает для каждого компонента тип и связь с другими компонентами через логические порты.
- Схема развёртывания – xml файл, описывающий то, в пределах какого сетевого узла работает тот или иной экземпляр компонента.

Рассмотрим обе схемы подробнее:

### Описание схемы

1. Контейнер, содержащий описание схемы приложения

Синтаксис

```
<app>  
  <instance name="X" kind="Y"/>  
  <connection source="X" source_out="index" dest="Y" dest_in="index"/>  
</app>
```

2. Описание экземпляра компонента, тег:

```
<instance>
```

Синтаксис

```
<instance name="X" kind="Y"/>
```

Атрибуты

- name - идентификатор компонента
- kind - тип компонента

3. Описание связи между компонентами, тег:

```
<connection>
```

Синтаксис

```
<connection source="X" source_out="index" dest="Y" dest_in="index"/>
```

Атрибуты

- source - идентификатор компонента, отправляющего сообщение

- source\_out - идентификатор выхода у отправляющего компонента
- dest - идентификатор компонента, получающего сообщение
- dest\_in - идентификатор входа у получающего компонента

Пример описания схемы приложения. Было разработано несколько вычислительных компонентов, в дальнейшем можно будет разрабатывать новые компоненты, зависящие от отрасли, деятельности компании и т.д. Пусть требуется описать такую логическую цепочку экземпляров компонентов:

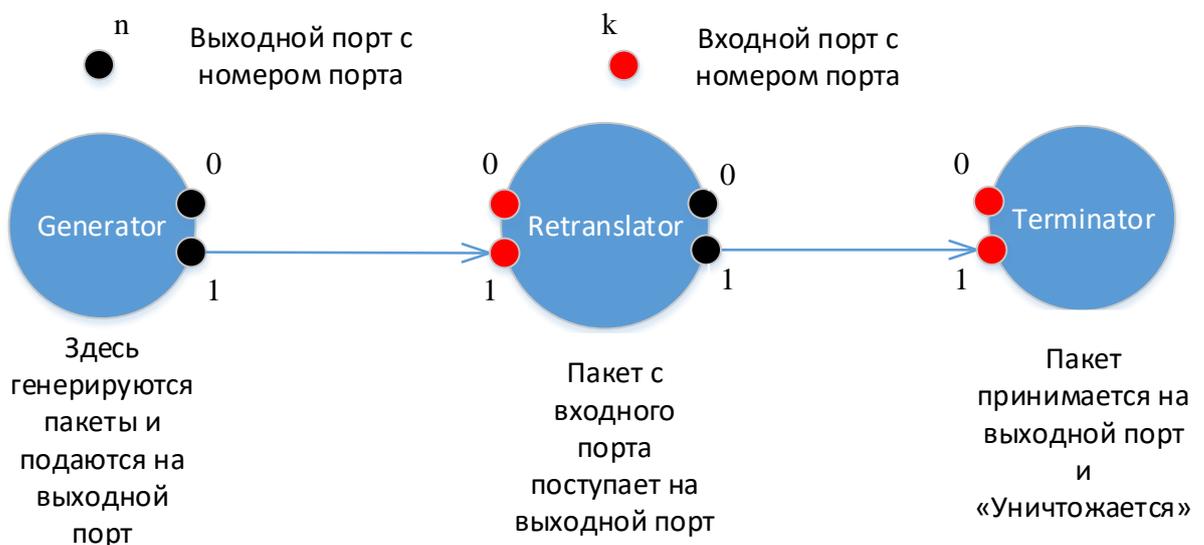


Рисунок 1 – Граф вычислительных компонентов

Тогда представление описания приложения будет выглядеть так:

```

<app>
  <instance name="A" kind="timed_generator"/>
  <instance name="B" kind="retranslator"/>
  <instance name="C" kind="terminator"/>
  <connection source="A" source_out="1" dest="B" dest_in="1"/>
  <connection source="B" source_out="1" dest="C" dest_in="1"/>
</app>

```

Рисунок 2 – Описание приложения

## Схема развертывания

### 1. Контейнер, содержащий схему развертывания приложения

Синтаксис

```
<deploy>
```

```
  <node name="X" address="y" port="0-65535">
```

```
    <map instance="X" node="Y"/>
```

```
</deploy>
```

### 2. Описание узла

Синтаксис

```
<node name="X" address="y" port="0-65535">
```

Атрибуты

- name: Имя узла;
- address: IP-адрес узла;
- port: порт, через который осуществляется взаимодействие с другими узлами в ходе работы распределенного приложения;
- config\_port: порт, через который будет осуществляться управление работой распределенного приложения.

### 3. Описание отображения экземпляра компонента на сетевой узел

Синтаксис

```
<map instance="X" node="Y"/>
```

Атрибуты

- instance: идентификатор экземпляра компонента
- node: идентификатор узла

Предположим, что экземпляры требуется развернуть на узлы следующим образом:

Таблица 3 – распределение компонентов по сетевым узлам

Компонент	Название сетевого узла	IP-адрес и порт узла
А	node1	192.168.1.1:8080
В	node2	192.168.1.2:31337

Компонент	Название сетевого узла	IP-адрес и порт узла
С	node3	192.168.1.3:31338

То есть каждый сетевой узел будет обладать одним вычислительным компонентом.

Тогда схема развертывания будет выглядеть так:

```
<deploy>
<node name="node1" address="127.0.0.1" port="8080" config_port="100"/>
<node name="node2" address="127.0.0.1" port="31337" config_port="101"/>
<node name="node3" address="127.0.0.1" port="31338" config_port="102"/>
<map instance="A" node="node1"/>
<map instance="B" node="node2"/>
<map instance="C" node="node3"/>
</deploy>
```

Рисунок 3 – Схема развертывания

### Работа с клиентским приложением терминал

Клиентское приложение (рисунок 4 ) позволяет создавать, сохранять и подгружать созданные ранее XML-файлы. Содержит области просмотра и редактирования xml-файлов; журнал развертывания; статистику узлов.

При загрузке приложением схемы, анализируется тег контейнера, и имя xml-файла появляется в том выпадающем списке (рисунок 5), к классу которого оно относится. После выбора схем конфигурации и нажатии кнопки «Загрузить», приложение терминала создает соединение с приложениями сетевых узлов, в таблице статуса узлов отображаются все подключенные сетевые узлы.

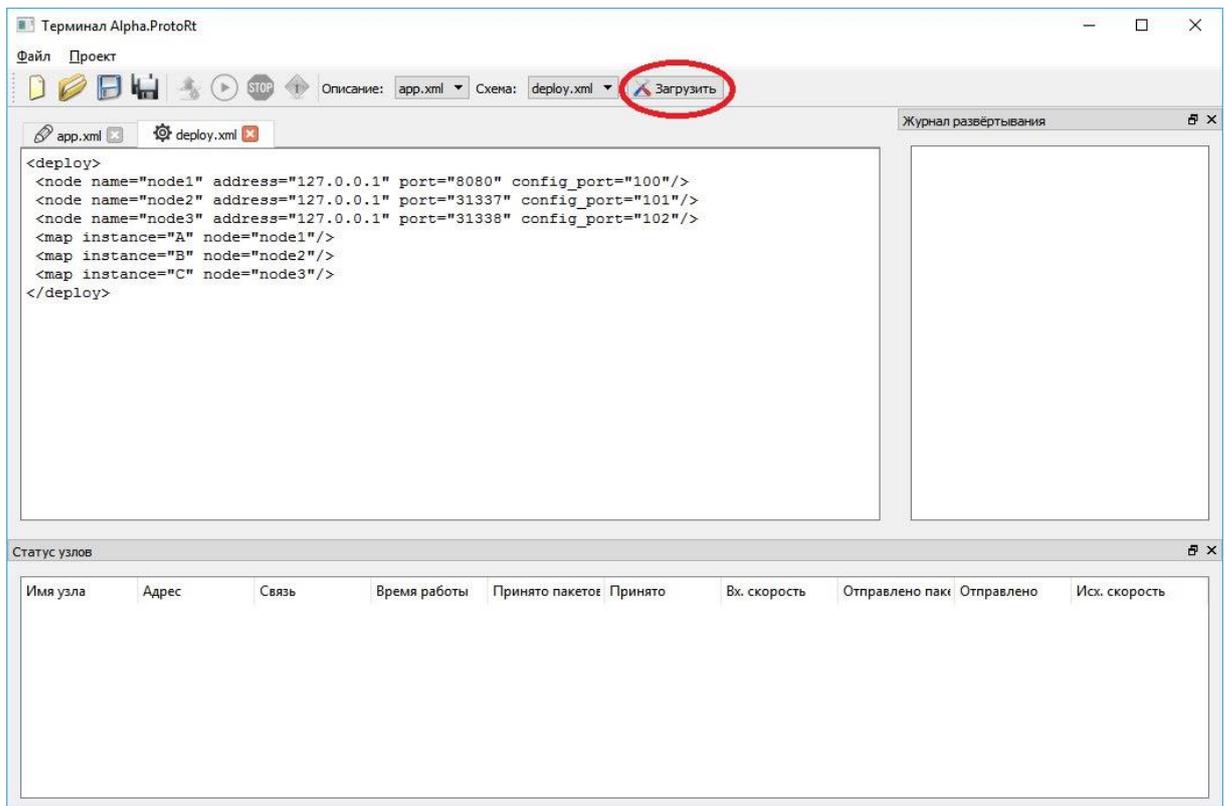


Рисунок 4 – Загрузка конфигурации



Рисунок 5 – Выпадающие списки

После загрузки схем, терминал начинает работать с данной конфигурацией, следующим шагом является развертывание конфигурации на сетевые узлы. После создания соединения становится доступной кнопка развертывания конфигурации, после нажатия на которую, будет развернута текущая конфигурация. После развертывания, в таблице статуса узлов с каждым узлом начинают отображаться списки компонентов, развернутых на нем.

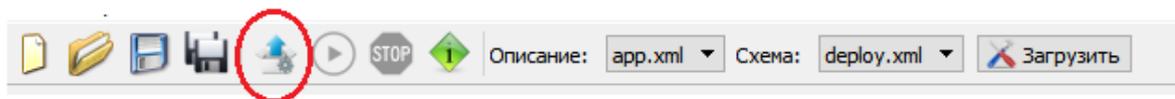


Рисунок 6 – Развертывание конфигурации

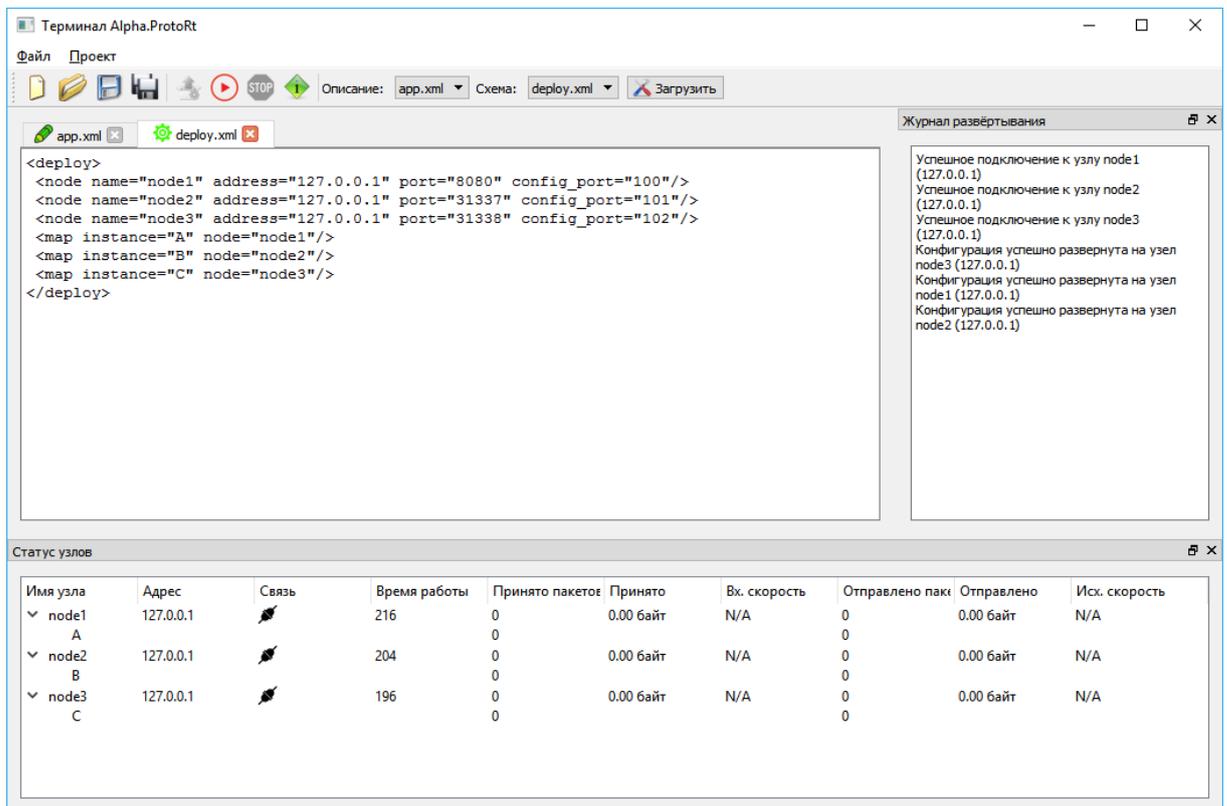


Рисунок 7 – Развертывание конфигурации

После развертывания кнопка старта работы распределенного приложения становится активной.

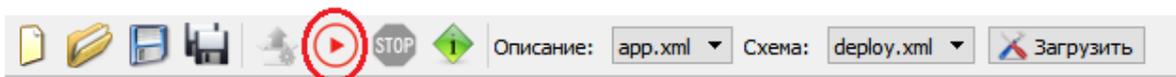


Рисунок 8 – Кнопка старта работы приложения

После старта работы приложения, таблица начинает динамически обновляться, отображая статистику узлов и компонентов.

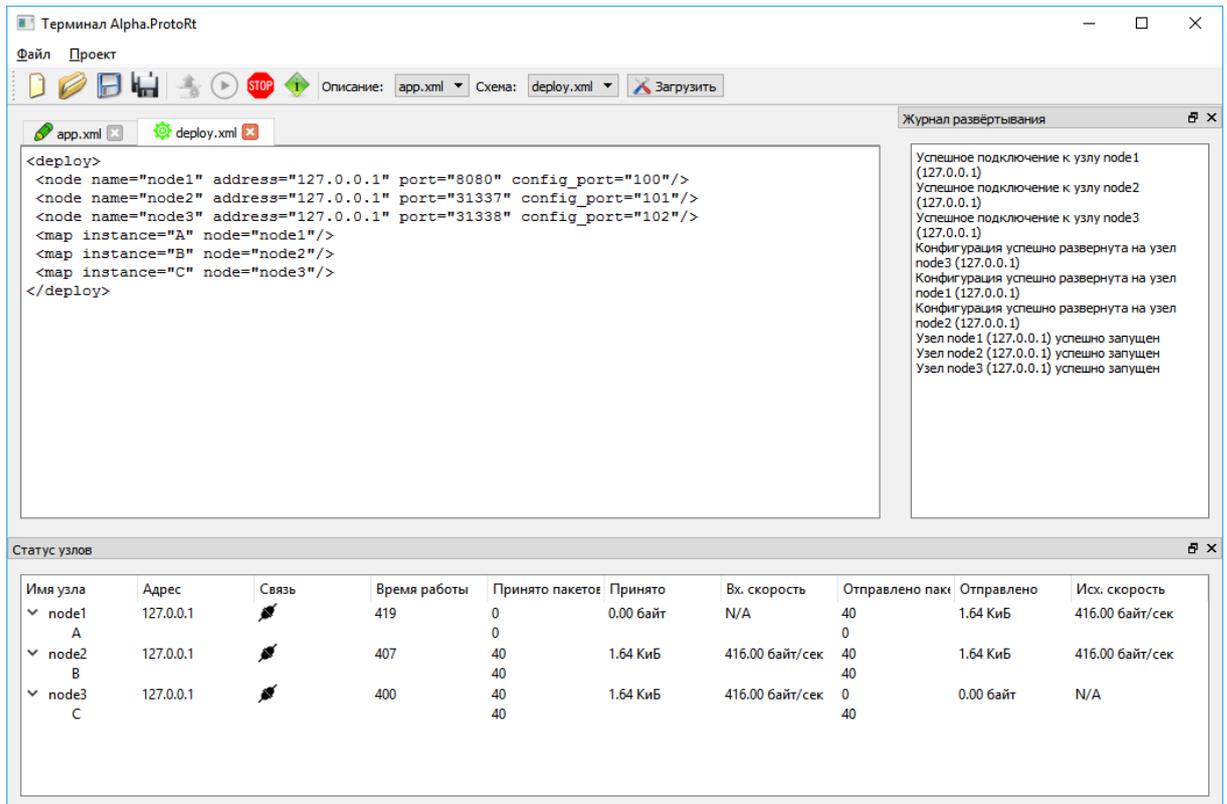


Рисунок 9 – Старт работы распределенного приложения

Затем работу распределенного приложения можно остановить.

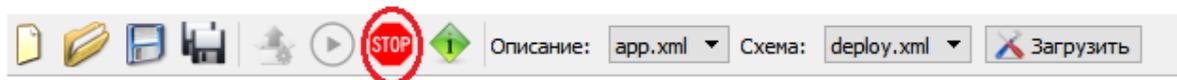


Рисунок 10 – Остановка работы приложения

### Работа с приложением сетевого узла

Приложение сетевого узла представляет собой консольное приложение и запускается на отдельном сетевом узле. Запуск осуществляется через консоль с указанием параметра – конфигурационного порта, т.е. порта через который будет осуществляться управление работой распределенного приложения.

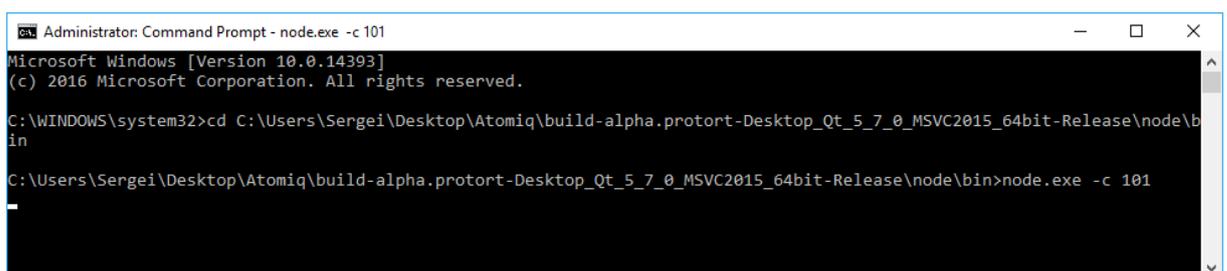


Рисунок 11 – Запущенное приложение сетевого узла

## Компоненты системы

Компоненты системы представляют собой основные подпроекты разрабатываемой системы:

- «node» – приложение сетевого узла, которое обеспечивает сетевое взаимодействие с другими узлами, работающее на одном сетевом узле и в пределах которого работают вычислительные компоненты.
- «terminal» – приложение, предназначенное для удалённой конфигурации сетевых узлов в соответствии со схемой развёртывания и схемой приложения.
- «protolink» – библиотека сетевого взаимодействия, реализующая примитивы сетевого взаимодействия двумя классами: client и server. Используется в приложении сетевого узла («node») и клиентском приложении («terminal») для организации сетевого взаимодействия.
- «link» – библиотека сетевого взаимодействия, реализующая примитивы сетевого взаимодействия двумя классами: client и server. Прототип библиотеки «protolink», без привязки к формату тела пересылаемых пакетов.
- «protocol» – бинарный протокол, соответствует протоколу уровня приложений модели OSI, описывает сообщения (пакеты) для сериализации для всех типов сообщений, пересылаемых между узлами, написанные с помощью protocol buffer language [13].

Так как работа выполнялась в команде, далее рассмотрим личный вклад в работу.

### 1. Реализация библиотеки сетевого взаимодействия link

Для реализации библиотеки сетевого взаимодействия использовалась сторонняя библиотека Boost::Asio [12], в качестве протокола транспортного уровня модели OSI использовался протокол TCP (Transmission Control Protocol), так как данный протокол в отличие от другого транспортного протокола UDP гарантирует доставку отправляемого пакета.

На рисунке изображена UML-диаграмма классов рассматриваемой библиотеки.

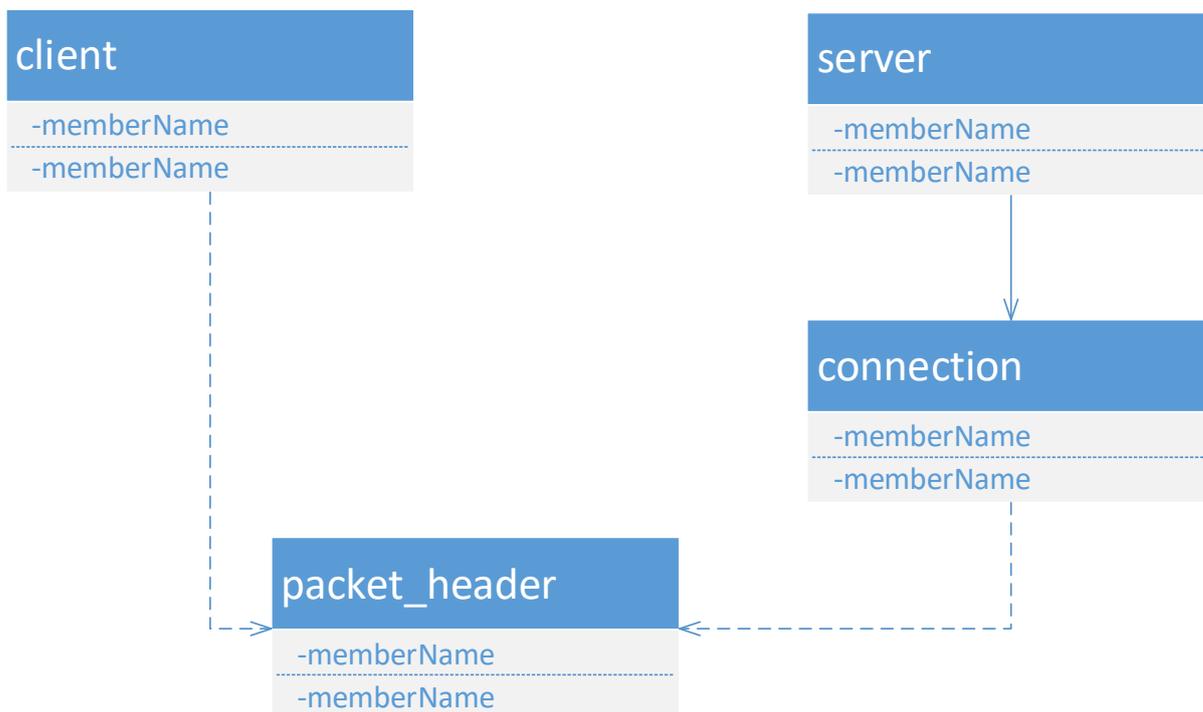


Рисунок 12 - UML-диаграмма классов библиотеки link

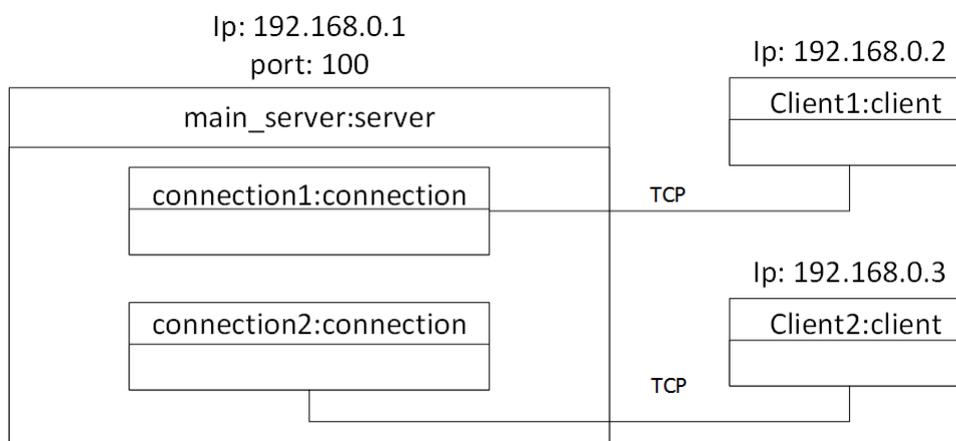


Рисунок 13 – Пример соединения сервера с двумя клиентами

В рамках реализации были созданы три шаблонных класса: *client*, *server*, *connection*. Класс *client* реализует концепцию сетевого клиента, класс *server* реализует концепцию сетевого сервера, класс *connection* используется классом *server* для создания соединения с клиентом и при каждом новом соединении создается отдельный экземпляр *connection*, экземпляр создается динамически с использованием «умного указателя» `boost::shared_ptr`, благодаря чему не

стоит волноваться об освобождении памяти. Пример соединения клиентов с сервером можно увидеть на рисунке 13.

Классы были созданы шаблонными, чтобы в качестве параметра передавать классы сетевого узла или клиентского приложения и использовать внутри *link* callback-методы. Для класса сетевого клиента (*client*) callback-методы используются для получения использующих его классов информации о подключении клиента либо информации о состоянии отправленного пакета, для класса сетевого сервера (*server*) данные методы используются для получения информации о новом соединении, для класса входящего соединения (*connection*) для получения нового пакета.

Основная задача класса клиента состоит в установлении соединения и отправке пакета. При неудачной попытке установления соединения, клиент будет пытаться подключиться через каждую секунду, а при разрыве соединения, клиент будет пытаться восстановить разорванное соединение.

Основная задача класса сервера заключается в прослушивании определенного порта, который указывается при создании экземпляра класса, принятии нового соединения, и с помощью экземпляра класса *connection* принимаются входящие пакеты.

Отправляемый клиентом и принимаемый сервером пакет состоит из двух частей: заголовка пакета, фиксированного размера и тела пакета. Заголовок определяется с помощью класса *packet\_header*.

После реализации данной библиотеки, был написан прототип приложения сетевого узла – *benchmarks*, реализованный в виде консольного приложения. Впоследствии на основе данного прототипа другим разработчиком реализовывалось приложение сетевого узла, однако основной целью данного тестового приложения было проверка работоспособности реализованной библиотеки *link* и тестирование пропускной способности и быстродействия.

Был реализован класс *test\_node*, в ходе работы приложения создается единственный экземпляр данного класса. На рисунке 14 изображена

диаграмма классов, где изображены основные классы тестового приложения сетевого узла.

Тестовое приложение поддерживает только два компонента, один из которых генерирует пакеты, а другой принимает и уничтожает.

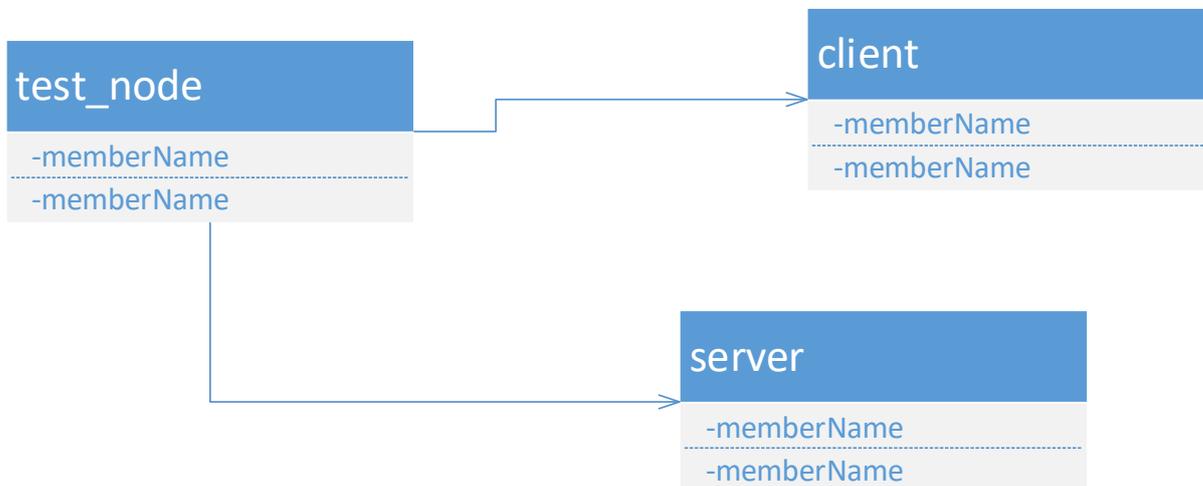


Рисунок 14 – UML-диаграмма классов тестового приложения сетевого узла

Во всех тестах использовалась схема компонентов, изображенная на рисунке 15 15 15 .

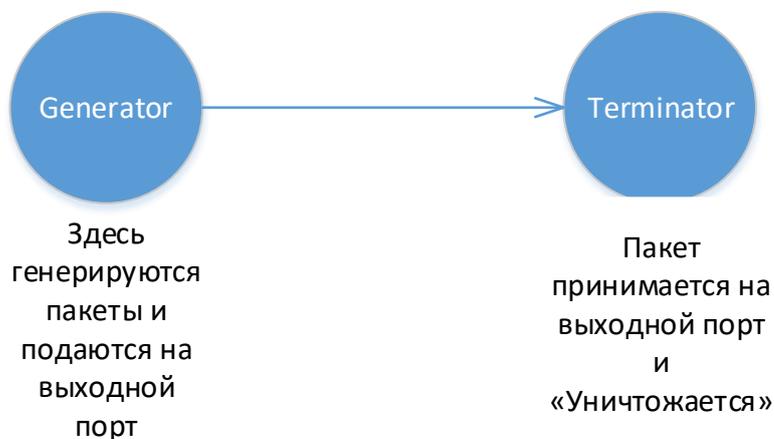


Рисунок 15 – Граф вычислительных компонентов для тестов

Во всех тестах на одной машине использовался размер пакета 200 KB в количестве 10000.

Intel Celeron CPU B830 @ 1.80 GHz Windows 10 - дает около 2.22 GBit/s.

```
size of packet: 200000
Run time: 6.71489
Speed: 0.432049
Speed (GBit/s): 2.21912
```

## 2. Реализация библиотеки сетевого взаимодействия *protolink*

На рисунке изображена диаграмма классов рассматриваемой библиотеки.

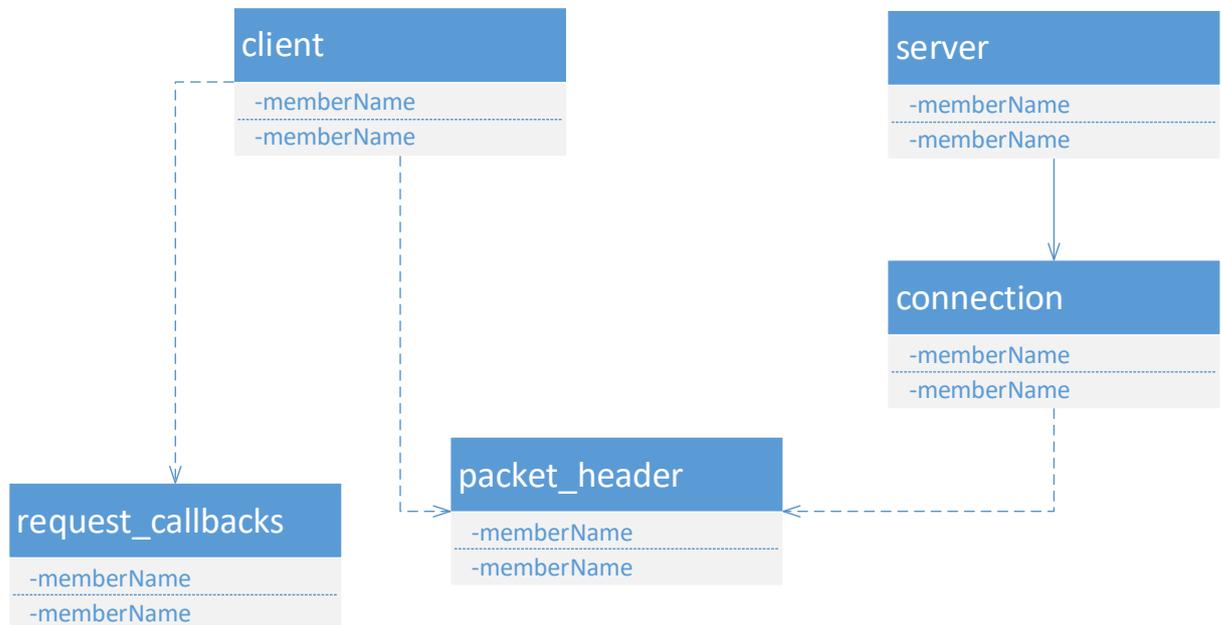


Рисунок 16 – UML-диаграмма классов библиотеки *protolink*

При реализации библиотеки *protolink* за основу бралась библиотека *link*, в результате работы для классов *client* и *connection* изменились методы приема и отправки пакетов, теперь они стали привязаны к структуре тела пакета. Структура тела пакета описывается с помощью *protocol buffer language*, который компилируется в C++ код, содержащий классы доступа к данным.

Добавился класс *request\_callbacks*, содержащий в себе сигналы. Здесь реализуется концепция сигналов и слотов. Сигналы могут хранить в себе указатели на несколько *callback*-функций, а слоты как раз являются этими самыми функциями. Так было сделано, потому что в клиентском приложении имеется класс, связанный с пользовательским интерфейсом, и методы этого класса привязываются к сигналам класса *request\_callbacks*, то есть к сигналу завершения транзакции можно привязать различные методы зависящие от типа запроса, то есть при выполнении запроса на развертывание, после прихода ответа от удаленного узла будут вызываться одни методы, а при

выполнении запроса на старт работы распределенного приложения другие методы. При отправке запроса создается новый объект класса *request\_callbacks*. При реализации данного класса использовалась библиотека Boost::Signals2 [15].

Для класса сетевого клиента были добавлены методы для реализации концепции «Запрос - Ответ», т.е. после отправки запроса клиент начинает ожидать прихода пакета от сервера. В результате расширения функционала клиент может посылать просто сообщение и запрос, а затем получать ответ. При отправке запроса в тело пакета добавляется информация для корректного получения ответа на запрос, а именно номер транзакции, после получения ответа, номер транзакции от ответного пакета ищется в списке текущих транзакций, затем вызывается callback-метод объекта класса *request\_callbacks* для данной транзакции и в качестве параметра передается полученный пакет.

Для класса входящего соединения (*connection*) были добавлены методы отправки ответа на запрос от клиента.

### **3. Реализация маршрутизации**

С помощью маршрутизации реализуется отправка, доставка и маршрутизация данных между логическими портами экземпляров компонентов, в том числе и через сетевой слой, в соответствии с описанием приложения и схемой развёртывания.

Каждый сетевой узел (рисунок 17) содержит граф локальных компонентов с созданными экземплярами компонентов и информацию об удаленных компонентах, связанных с локальными, а именно ip-адрес и порт узла, на котором расположен удаленный компонент и идентификатор компонента.

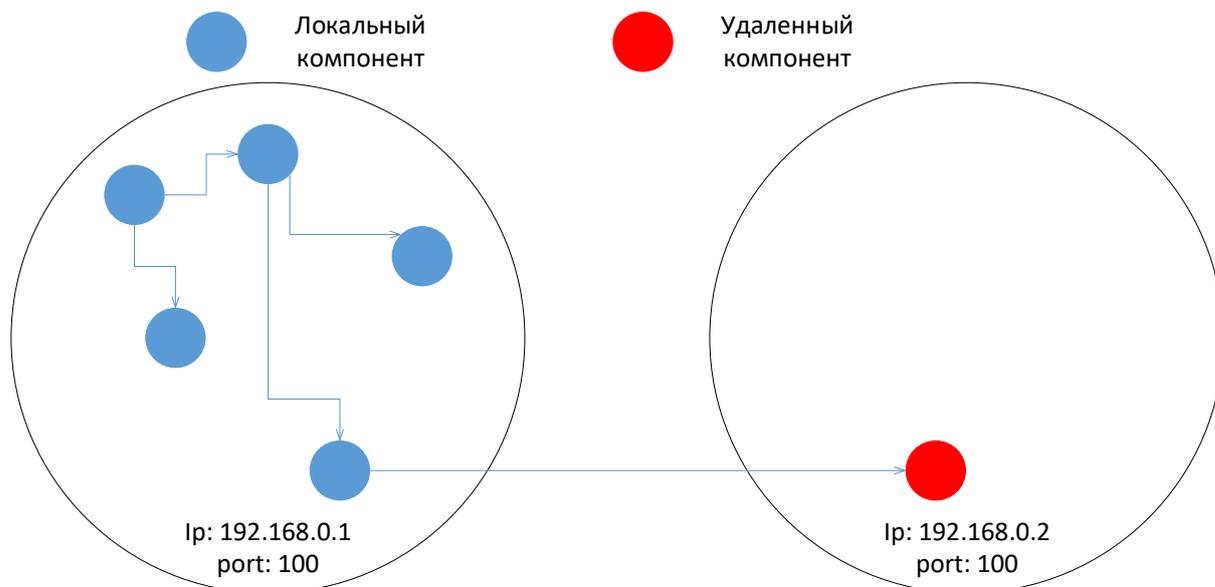


Рисунок 17 – Граф компонентов для сетевого узла

На рисунке изображена диаграмма классов реализации маршрутизации.

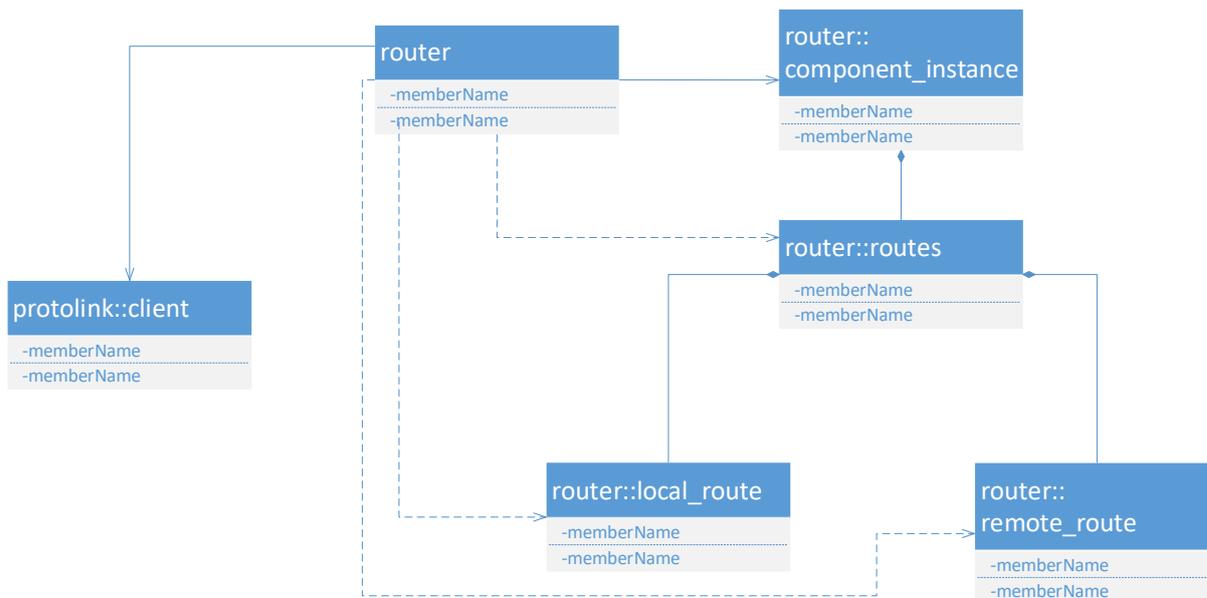


Рисунок 18 – UML-диаграмма классов для маршрутизации

Маршрутизация реализуется с помощью класса *router* и используется в приложении сетевого узла.

Класс *router* содержит следующие вложенные классы:

- *local\_route* - Используется для адресации локального компонента, то есть компонента, расположенного на данном узле;
- *remote\_route* - Используется для адресации удаленного компонента, то есть компонента, расположенного на удаленном узле;

- *routes* - Класс маршрутов, содержит списки локальных и удаленных компонентов;
- *component\_instance* - используется для маршрутизации пакетов между компонентами в рамках графа, описывающего структуру приложения. Содержит необходимую информацию о компоненте для его идентификации внутри графа.

После поступления пакета на компонент, он выполняет вычисления, зависящие уже от самого компонента, и затем передает результат связанным с ним компонентам с помощью вызова метода API класса *router*, то есть классы маршрутизации предоставляют API компонентам для передачи пакета.

Так как компоненты графа могут располагаться как на одном узле так и на других, поэтому для каждого удаленного компонента создается экземпляр класса *protolink::client*, с помощью методов которого осуществляется отправка пакета на другой связанный с ним компонент. Пакет отправляется с помощью «простого сообщения» на другой узел, при отправке, в пакет добавляется идентификатор компонента на удаленном узле, то есть после прихода пакета на удаленный узел и его десериализации содержимое пакета отправляется на нужный компонент с помощью метода API предоставляемого приложению сетевого узла.

#### **4. Реализация сетевой часть клиентского приложения**

Были реализованы классы, реализующие сетевое взаимодействие клиентского приложения с узлами для развертывания, старта и остановки работы распределенного приложения.

Рассмотрим диаграмму классов, где изображены основные созданные классы клиентского приложения.

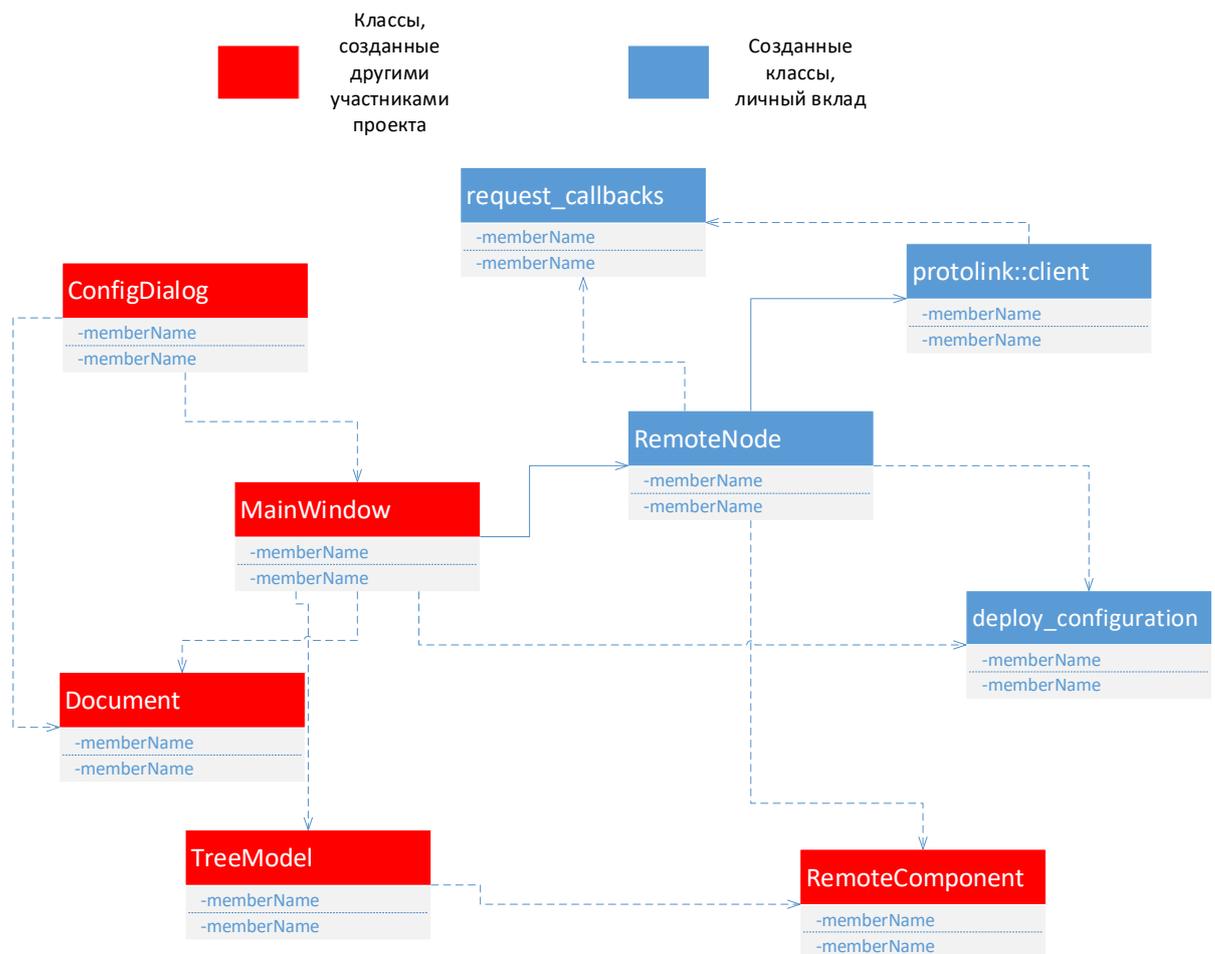


Рисунок 19 – UML-диаграмма классов для клиентского приложения, где синим цветом отмечены классы, созданные автором, а красным созданные другими участниками проекта

Были созданы классы *RemoteNode*, реализующий методы развертывания, старта и остановки работы распределенного приложения для одного сетевого узла, используемые в классе *MainWindow*; *deploy\_configuration*, обладающий структурами данных, содержащих информацию о компонентах, узлах, связях в конфигурации, используется для удобного и оптимального получения данных классом *RemoteNode*. При загрузки терминалом файлов конфигурации, т.е. при нажатии на кнопку «Загрузить» (рисунок 4) создаются экземпляры класса *RemoteNode* и клиентское приложение создает соединение с сетевыми узлами.

Для одной конфигурации создается один экземпляр класса *deploy\_configuration*, хранящийся в классе *MainWindow*.

Развертывание, старт и остановка реализованы с помощью запросов к узлам. В классе *RemoteNode* реализована концепция сигналов и слотов, описанная ранее, то есть перед отправкой запроса создается экземпляр класса *request\_callbacks* и к нему привязываются callback-методы, а затем экземпляр передается в класс *protolink::client*, таким образом после прихода ответа на запрос, для разных типов запросов будут вызваны разные callback-методы, в качестве параметра которым будет передан пришедший пакет.

Так как конфигурация содержит информацию по всем узлам и компонентам, для одного конкретного сетевого узла эта информация избыточна, поэтому перед посылкой запроса на развертывание конфигурации на узле, для него выделяются только те данные, которые будут использовать данный узел.

## **4. ФИНАНСОВЫЙ МЕНЕДЖМЕНТ**

### **4.1. Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения**

Так как конфигурация содержит информацию по всем узлам и компонентам, для одного конкретного сетевого узла эта информация избыточна, поэтому перед посылкой запроса на развертывание конфигурации на узле, для него выделяются только те данные, которые будут использовать данный узел.

#### **4.1.1. Техническая актуальность**

Промышленное производство в различных отраслях использует множество датчиков, часто площади производства очень большие, в результате чего возникает потребность в распределенных приложениях, так как необходимо собирать данные с датчиков, распределенных в пространстве на большой территории, обрабатывать данные в специальных дата-центрах.

Распределенные приложения описывают решения по автоматизации в таких отраслях промышленности: металлургия, электроснабжения, нефтедобычи и нефтепереработки, химическое производство, системы управления дорожно-транспортным движением.

Рассмотрим необходимость в разрабатываемой системе. По некоторым причинам невозможно выполнить все вычислительные процессы на одной машине. Рассмотрим возможные причины:

- Географическая удаленность (например, датчики);
- Тяжелые вычисления и невозможность выполнить все вычисления на одной машине;
- Требования безопасности (например, имеется требование, чтобы база данных располагалась на отдельной машине);
- Пониженные требования к операционным системам, установленным на узлах, так как на каждом узле может стоять своя операционная система.

#### **4.1.2. Экономическая актуальность**

Распределенные приложения позволят уменьшить затраты на оборудование, например из-за географической удаленности различного оборудование, необходимо создавать центры обработки данных, однако за счет использования распределенных приложений можно разместить лишь один компонент в удаленном месте, а данные со всех таких мест отсылать в единый центр обработки данных, что значительно сократит расходы на оборудование, и помимо этого сократит трудоемкость, так как нет необходимости наличие большого количества человек для обслуживания удаленного компонента.

#### **4.1.3. Цели и задачи разработки**

**Цель:** Разработать и протестировать систему поддержки развёртывания распределённых приложений.

**Задачи:**

- Выбрать программные средства для разработки;
- Спроектировать архитектуру;
- Разработать приложение.

#### **4.1.4. Критерии эффективности**

##### **4.1.4.1. Техническая актуальность**

- Надежность – Обеспечение безопасности за счет разграничения мест хранения и обработки данных (пример: база данных располагается на отдельной машине);
- Время обработки данных – Уменьшение времени сбора и обработки данных.

##### **4.1.4.2. Экономическая**

- Трудоемкость – Уменьшение трудоемкости за счет автоматизации сбора данных и их обработки;

- Затраты – Уменьшение затрат за счет создание единых центров обработки данных.

#### 4.1.5. Сегментирование рынка

Данное ПО может использоваться всеми лицами, заинтересованными в использовании распределенных приложений. Распределенные приложения описывают решения по автоматизации в таких отраслях промышленности: металлургия, электроснабжения, нефтедобычи и нефтепереработки, химическое производство, системы управления дорожно-транспортным движением. На основании данной информации была составлена карта сегментирования рынка:

Таблица 4 – Карта сегментирования рынка

	Отрасль			
Площадь		нефтедобыча	металлургия	электроснабжения
	Большая			
	Средняя			
	Малая			

#### 4.1.6. Потенциальные потребители результатов разработки

Целевой рынок разработки – крупные нефтяные компании, такие как АО «Транснефть», ПАО «НК «Роснефть», ПАО «Газпром».

Таблица 5 – Оценка качества и перспективности разработки по технологии QuaD

Критерий оценки	Вес критерия	Баллы	Максимальный балл	Относительное значение	Средневзвешенное значение
Показатели оценки качества разработки					
Надежность	0,40	80	100	0,8	32
Функциональная пригодность	0,01	80	100	0,8	0,8
Уровень производительности	0,30	90	100	0,9	27
Качество интеллектуального	0,02	50	100	0,5	1

интерфейса					
Переносимость	0,05	90	100	0,9	4,5
Удобство использования	0,02	80	100	0,8	1,6
<b>Показатели оценки коммерческого потенциала разработки</b>					
Финансовая эффективность научной разработки	0,15	80	100	0,8	12
Уровень проникновения на рынок	0,03	80	100	0,8	2,4
Послепродажное обслуживание	0,01	85	100	0,85	0,85
Срок выхода на рынок	0,01	90	100	0,9	0,90
<b>Итого:</b>					<b>84,25</b>

Средневзвешенное значение показателя качества и перспективности научной разработки получилось равным 84.25, что говорит о перспективности разработки.

#### 4.1.7. SWOT-анализ

В данном разделе приведен SWOT-анализ научного проекта. SWOT – Strengths (сильные стороны), Weaknesses (слабые стороны), Opportunities (возможности) и Threats (угрозы). Данная матрица представлена в приложении А.

Таблица 6 – Зависимость возможностей и сильных сторон проекта

Сильные стороны проекта							
Возможности проекта		C1	C2	C3	C4	C5	C6
	B1	-	-	-	-	-	-
	B2	-	-	-	-	-	+
	B3	-	-	-	-	+	-
	B4	-	-	-	-	-	-

Таблица 7 – Зависимость возможностей и слабых сторон проекта

Слабые стороны проекта			
Возможности проекта		Сл1	Сл2
	B1	-	-
	B2	+	-
	B3	-	-
	B4	-	+

Таблица 8 – Зависимость угроз и сильных сторон проекта

Сильные стороны проекта							
Угрозы проекта		C1	C2	C3	C4	C5	C6
	У1	-	-	-	-	-	-
	У2	-	+	-	-	-	-

	У3	-	-	-	+	+	-
	У4	-	-	-	-	-	-
	У5	-	-	-	-	-	-

Таблица 9 – Зависимость угроз и слабых сторон проекта

Слабые стороны проекта			
Угрозы проекта		C1	C2
	У1	-	-
	У2	-	-
	У3	+	-
	У4	-	-
	У5	-	-

#### 4.2. Определение возможных альтернатив проведения научных исследований

С помощью морфологического подхода были выделены возможные альтернативы проведения исследования. Определение возможных альтернатив разработки продукта является очень важным этапом в планировании разработки продукта, т. к. с помощью него можно выбрать самый оптимальный вариант.

Таблица 10 – Морфологический анализ

	1	2	3
А. Команда разработчиков	Сотрудники компании	Подрядчики	Стажеры
В. Количество разработчиков	3	4	5
С. Среда разработки	Visual Studio	Qt Creator	Code::Blocks
Д. Сериализация данных при пересылке	Protocol Buffers	XML	JSON
Е. Язык программирования	C++	C#	Java
Ф. Сроки выполнения	Установленные в ТЗ	Срок прохождения стажировки	Срок прохождения стажировки + включая переговоры с заказчиком
Г. Языковая локализация проекта	Более 2 языков	2 языка	Только русский язык

В качестве наиболее вероятных вариантов для реализации данной системы были выбраны:

- А3В3С2Д1Е1Ф3Г3
- А1В1С1Д1Е1Ф1Г3
- А2В2С3Д2Е2Ф1Г1

### 4.3. Планирование научно-исследовательских работ

#### 4.3.1. Структура работ в рамках научного исследования

Работу над проектом можно разделить на следующие этапы:

Таблица 11 – Этапы работы над проектом

Основные этапы	№ работы	Содержание работ	Исполнитель
Анализ существующих аналогов	1	Изучение существующих на данный момент российских и зарубежных программных продуктов.	Исполнитель
Определение функционала приложения	2	Определение функций, которые должно выполнять разрабатываемое приложение.	Руководитель, исполнитель
Проектирование архитектуры приложения	3	Определение архитектуры системы, определение компонентов	Руководитель, исполнитель
Выбор технологий для реализации	4	Выбор среды разработки, выбор сетевой библиотеки, выбор протокола сериализации	Руководитель, исполнитель
Реализация	5	Реализация библиотек сетевого взаимодействия	Исполнитель
	6	Разработка структуры пакетов для сериализации	Исполнитель
	7	Разработка пользовательского интерфейса	Исполнитель
	8	Реализация маршрутизации	Исполнитель
	9	Реализация приложения сетевого узла	Исполнитель
	10	Разработка демонстрационных компонентов	Исполнитель
	11	Реализация сетевой части в клиентском приложении	Исполнитель
	12	Реализация сбора и отображения статистики	Исполнитель
Тестирование	13	Выявление ошибок в работе приложения	Руководитель, исполнитель
Основные этапы	№ работы	Содержание работ	Исполнитель
Исправление ошибок	14	Исправление ошибок, выявленных в результате тестирования	Исполнитель

### **4.3.2. Определение трудоемкости выполнения работ**

Трудоемкость – это количество времени, затрачиваемое на единицу качественной продукции. Трудоемкость измеряется в человеко-днях. Расчет трудоемкости необходим для того, чтобы определить самый оптимальный вариант реализации.

Для построения календарного плана-графика необходимо рассчитать временные показатели проведения научного исследования. Данные расчеты представлены в приложении Б.

### **4.3.3. Разработка графика проведения научного исследования**

Для наглядного представления распределения работ участников проекта и затраченного времени была построена диаграмма Ганта – горизонтальный ленточный график, на котором работы по теме представляются протяженными во времени отрезками, характеризующимися датами начала и окончания выполнения данных работ. Построенная диаграмма Ганта представлена в приложении В.

### **4.3.4. Бюджет научно-технического исследования**

При планировании бюджета НТИ должно быть обеспечено полное и достоверное отражение всех видов расходов, связанных с его выполнением. В процессе формирования бюджета НТИ были рассмотрены следующие статьи:

- материальные затраты НТИ;
- затраты на специальное оборудование для научных работ;
- основная заработная плата исполнителей темы;
- дополнительная заработная плата исполнителей темы;
- отчисления во внебюджетные фонды (страховые отчисления);
- затраты научные и производственные командировки;
- контрагентные расходы;
- накладные расходы.

#### 4.3.4.1. Расчет материальных затрат НТИ

В варианте исполнения № 2 используется среда разработки Microsoft Visual Studio. Цена на данное ПО составляет 65 780 руб.

Таблица 12 – Материальные затраты НТИ

Наименование	Единица измерения	Количество			Цена за ед., руб.			Затраты на материалы, (Зм), руб.		
		Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3
Лицензия на IDE Microsoft Visual Studio Professional	Шт.	0	1	0	0	65780	0	0	65780	0
Итого								65780		

#### 4.3.4.2. Расчет затрат на специальное оборудование для научных работ

В данной статье расходов следует учесть затраты на амортизационные отчисления. Разработка выполнялась с использованием ноутбука стоимостью 12 000 руб. Возьмем срок службы данного ноутбука равным 5 годам. Следовательно, ежегодные отчисления будут составлять 2400 руб. Длительность разработки составляет 90 день. Следовательно, амортизационные отчисления за этот период составляют:  $(90/365)*2400 = 591.78$  руб.

#### 4.3.4.3. Основная заработная плата

Исходя из того, что разработке уделялось 2 часа в день, почасовая ставка для руководителя составляет 300 рублей, а для исполнителя 100 рублей, можно рассчитать основную заработную плату.

Расчет основной заработной платы представлен в таблице 13.

Таблица 13 – Расчет основной заработной платы

№ п/п	Наименование этапов	Исполнители по категориям	Трудоемкость, чел.-дн.			Всего заработная плата по тарифу (окладам), тыс. руб.		
			Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
1	Анализ существующих аналогов	Исполнитель	0.7	0.7	0.7	0.56	0.56	0.56
2	Определение функционала приложения	Руководитель, исполнитель	4	4	4	12.8	12.8	12.8
3	Проектирование архитектуры приложения	Руководитель, исполнитель	6.8	6.8	6.8	21.76	21.76	21.76
4	Выбор технологий для реализации	Руководитель, исполнитель	3.4	3.4	3.4	10.88	10.88	10.88
5	Реализация библиотек сетевого взаимодействия	Исполнитель	7	7	7	5.6	5.6	5.6
6	Разработка структуры пакетов для сериализации	Исполнитель	4.4	4.4	4.4	3.52	3.52	3.52
7	Реализация маршрутизации	Исполнитель	6.8	6.8	6.8	5.44	5.44	5.44
8	Реализация приложения сетевого узла	Исполнитель	7.4	7.4	7.4	5.92	5.92	5.92
9	Разработка демонстрационных компонентов	Исполнитель	4.5	4.5	4.5	3.6	3.6	3.6
10	Реализация сетевой части в клиентском приложении	Исполнитель	7.6	7.2	7.8	6.08	5.76	6.24
11	Реализация сбора и отображения статистики	Исполнитель	8	8	8	6.4	6.4	6.4
12	Тестирование	Исполнитель	4.8	4.8	4.8	3.84	3.84	3.84
13	Исправление ошибок	Исполнитель	9.2	11	13.5	7.36	8.8	10.8
<b>Итого (с учетом коэффициентов, <math>k_d = 0.2</math>):</b>						<b>93.76</b>	<b>95.4</b>	<b>98.08</b>

#### 4.3.4.4. Дополнительная заработная плата

Примем коэффициент  $k_{доп} = 0.12$ , исходя из этого рассчитаем дополнительную заработную плату для каждого из вариантов исполнения (тыс. руб.):

1. 11.25

2. 11.45

3. 11.79

Отсюда найдем полную заработную плату для каждого ВИ:

1. 105.01

2. 106.85

3. 109.87

#### 4.3.4.5. Отчисления во внебюджетные фонды

Для учреждений, осуществляющих образовательную и научную деятельность,  $k_{\text{внеб}}$  принимается равным 0.271. Отсюда находим отчисления во внебюджетные фонды для каждого варианта исполнения:

1. 28.45

2. 28.96

3. 29.77

Таблица 14 – Отчисления во внебюджетные фонды

Наименование статьи	Сумма, руб.			Примечание
	Исп.1	Исп.2	Исп.3	
1. Материальные затраты НТИ	0	65780	0	Пункт 3.4.1
2. Затраты на специальное оборудование для научных (экспериментальных) работ	591.78	591.78	591.78	Пункт 3.4.2
3. Затраты по основной заработной плате исполнителей темы	93760	95400	98080	Пункт 3.4.3
4. Затраты по дополнительной заработной плате исполнителей темы	4170	4820	5160	Пункт 3.4.4
5. Отчисления во внебюджетные фонды	28450	28960	29770	Пункт 3.4.5
6. Затраты на научные и производственные командировки	0	0	0	Пункт 3.4.6
7. Контрагентские расходы	0	0	0	Пункт 3.4.7
8. Накладные расходы	20315.48	31288.28	21376.28	16 % от суммы ст. 1-7
9. Бюджет затрат НТИ	147287.726	226840.06	154978.06	Сумма ст. 1- 8

#### 4.4. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

##### 4.4.1. Интегральный финансовый показатель

Для альтернативных вариантов исполнения были рассчитаны интегральные финансовые показатели по формуле:

$$I_{\text{финр}}^{\text{исп.}i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}}, \text{ где}$$

$\Phi_{pi}$  – стоимость  $i$ -го варианта исполнения;

$\Phi_{\text{max}}$  – максимальная стоимость исполнения.

Далее приведен данный показатель для всех вариантов исполнения:

1. 0.64
2. 1
3. 0.68

##### 4.4.2. Интегральный показатель ресурсоэффективности

Были выделены следующие критерии для расчета интегрального показателя ресурсоэффективности:

Таблица 15 – Интегральный показатель ресурсоэффективности

Критерии	Весовой коэффициент	Исп. 1	Исп. 2	Исп. 3
1. Удобство в разработке	0.1	5	4	4
2. Удобство в использовании	0.25	5	3	5
3. Удобство поддержки продукта	0.25	5	4	3
4. Требовательность к ресурсам клиента	0.15	4	4	3
5. Надежность	0.25	5	4	3
<b>Итого</b>		4.85	3.75	3.6

##### 4.4.3. Интегральный показатель эффективности вариантов исполнения разработки

Определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{\text{финр}}^{\text{исп.}i} = \frac{I_{p-\text{исп}i}}{I_{\text{финр}}^{\text{исп.}i}},$$

Сравнительная эффективность проекта ( $\text{Эср}_i$ ):

$$\mathcal{E}_{\text{ср}_i} = \frac{I_{\text{исп}i}}{I_{\text{исп.}_\text{min}}},$$

Таблица 16 – Сравнительная эффективность разработки

№ п/п	Показатели	Исп. 1	Исп. 2	Исп. 3
1	Интегральный финансовый показатель разработки	0.64	1	0.68
2	Интегральный показатель ресурсоэффективности разработки	4.85	3.75	3.6
3	Интегральный показатель эффективности	7.57	3.75	5.29
4	Сравнительная эффективность вариантов исполнения	2.12	1	1.41

Исходя из таблицы можно сделать вывод, что выбранный вариант исполнения № 1 является наиболее эффективным как с финансовой стороны, а также обладает наибольшей ресурсной эффективностью. Кроме того, данный вариант является самым удобным для разработки и сопровождения, а также является наименее требовательным к ресурсам клиента. Совокупность указанных критериев делает данный вариант самым целесообразным для разработки.

## **5. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ**

Целью проекта ВКР состоит в разработке системы поддержки развертывания распределенных приложений, разработка планируется к внедрению на АО «Атомик Софт». Ввиду высокой специализации программного обеспечения для предприятия, оно планируется к использованию только на предприятиях, приобретающих продукты данной организации.

Разрабатываемой системой предполагается к использованию операторам (контроль и управление работой распределенного приложения), системные администраторы (ответственные за запуск приложений сетевых узлов на машинах).

Для анализа вопросов социальной ответственности необходимо рассмотреть рабочее место оператора персонального компьютера, опасности, которые может принести эксплуатация ЭВМ, и меры защиты от них.

### **5.1. Производственная безопасность**

#### **5.1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения**

##### **5.1.1.1. Нарушение параметров микроклимата**

В СанПиН 2.2.4-96 «Гигиенические требования к микроклимату производственных помещений», определяются параметры микроклимата рабочих мест:

- температура воздуха;
- температура поверхностей;
- относительная влажность воздуха;
- скорость движения воздуха;
- интенсивность теплового облучения.

Эти показатели возникают вследствие воздействия нескольких факторов на микроклимат: работа оборудования, устройство производственных помещений, времени года. Эти параметры микроклимата влияют на функциональное состояние, самочувствие, работоспособность и здоровье людей.

Проводимая работа относится к работе разработчика-программиста и относится к категории тяжести 1а. Согласно СанПиН 2.2.2/2.4.1340-03 для категории тяжести работ 1а температура воздуха должна быть в холодный период года не более 22-24С, в теплый период года 20-25С. Относительная влажность должна составлять 40-60%, скорость движения воздуха — 0,1 м/с. Для поддержания оптимальных значений микроклимата используется система отопления и кондиционирования воздуха. Для повышения влажности воздуха в помещении следует применять увлажнители воздуха с дистиллированной или кипяченой питьевой водой.

В целях профилактики могут быть использованы средства местного кондиционирования воздуха, воздушного душирования, спецодежда, помещения для отдыха и обогрева, регламентация времени работы, увлажнения воздуха. [17]

#### **5.1.1.2. Недостаточная освещенность рабочей зоны**

Освещённость рабочей зоны влияет на работу глаз. Неправильное задание освещенности пагубно влияет на глаза, обеспечивая излишнее напряжение на нервную систему в результате усилий на обработку менее четких сигналов. В результате периодическое напряжение приводит к ослаблению зрения. При этом установлено, что наилучшей освещённостью является естественный свет. Именно поэтому необходимо установить искусственный свет таким образом, чтобы он максимально имитировал естественный. Это достигается путем настройки местного и общего освещения.

Согласно СанПиН 2.2.2/2.4.1340-03 освещенность поверхности стола в зоне размещения документа должна быть 300-500 лк. Допускается установка светильников местного освещения для подсветки документов. Местное освещение не должно создавать бликов на поверхности экрана и увеличивать освещенность экрана более 300 лк. Прямую блескость от источников освещения следует ограничить. Яркость светящихся поверхностей (окна, светильники),

находящихся в поле зрения, должна быть не более 200 кд/м<sup>2</sup>. Яркость бликов на экране монитора не должна превышать 40 кд/м<sup>2</sup>. 71 Соотношение яркости между рабочими поверхностями не должно превышать 3:1 – 5:1, между рабочими поверхностями и поверхностями стен и оборудования 10:1. Стол следует поставить сбоку от окна так, чтобы свет падал слева. Наилучшее освещение для работы с компьютером – рассеянный не прямой свет, который не дает бликов на экране. В поле зрения пользователя не должно быть резких перепадов яркости, поэтому окна желательно закрывать шторами либо жалюзи. [19]

## **5.1.2. Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения**

### **5.1.2.1. Повышенный уровень статического электричества**

Статическое электричество является одним из опасных факторов. Оно возникает в результате накопления заряда на поверхности или объеме диэлектриков или на изолированных проводниках. Опасность возникает при разряде этого заряда, поскольку заряд может быть велик, что может вызвать поломку оборудования, угрозу для жизни оператора в результате прохождения заряда через тело человека. Накопление заряда может быть вызвано изоляцией оборудования или оператора.

Для устранения необходимо обеспечить заземление компьютера, чтобы заряд мог уходить на землю, не накапливаясь на корпусе компьютера. Правила заземления и другие мероприятия по устранению статического электричества перечислены в ГОСТ 12.1.030-81. [18]

Повышенное значение напряжения в электрической цепи. При прикосновении к электроустановкам, по которым протекает ток возможно прохождение тока через тело человека, так как земля имеет нулевой потенциал. Сила поражения зависит от времени воздействия, характера тока, состояния человека, места соприкосновения, пути прохождения тока. Поражающими факторами являются ожоги, сбои в работе сердечной мышцы, нервной системы.

Причиной прикосновения является как правило нарушение изоляции изделия и нарушение техники безопасности.

Для питания персональных компьютеров в России как правило используется сетевое напряжение 230В переменного тока при частоте 50 Гц. После прохождения через блок питания ток трансформируется в постоянный ток напряжения 5 и 12 В. Согласно ГОСТ 12.1.038-82 допустимо прохождение через тело человека переменного тока с напряжением 2В, силой 0.3мА, постоянного тока 8В, силой 1мА. Очевидно, прикосновение к проводке компьютера в любой 73 из его частей недопустимо, хотя после преобразования в блоке питания может не нести поражающих действий. [20]

Для защиты от прохождения тока по телу необходимо соблюдать технику безопасности использования персонального компьютера, использовать ГОСТ Р 50571.3-94 по защите от поражения электрического тока. При обнаружении нарушений изоляции необходимо в первую очередь отключить прибор от источника питания, использовать изоляционные материалы. [22]

## **5.2. Экологическая опасность**

В процессе работы персональные компьютеры оказывают вредное воздействие на окружающую среду. Воздействие заключается в создании вредного излучения, сжигании кислорода, создании вибраций, загрязнении вредными отходами деятельности. Для обеспечения экологической безопасности необходимо рассмотреть воздействие и обеспечить защиту селитебной зоны, атмосферы, гидросферы и литосферы.

### **5.2.1. Защита литосферы**

При выполнении работы на персональных компьютерах, некоторые из частей компьютера могут прийти в негодность или устареть, тогда необходимо их утилизировать. В качестве отходов в результате деятельности программирования может выступать бумага, отходы электроники, ртутьсодержащие лампы. Эти твердые бытовые отходы загрязняют

окружающую среду и их необходимо перерабатывать, чтобы максимально снизить их вредное воздействие. Для складирования мусора перед утилизацией предлагается вывоз на территории, предназначенные для этого. Утилизация электроники производится специальными компаниями и является обязательной, в особенности утилизация ртутьсодержащих ламп. Отработанные ртутьсодержащие лампы подлежат проведению демеркуризации.

### **5.3. Безопасность в чрезвычайных ситуациях**

В процессе работы на персональном компьютере единственной чрезвычайной ситуацией, которую создаёт компьютер может стать пожар. Пожар в течении работы персонального компьютера может возникнуть в результате замыкания электропроводки, приводящее к возгоранию аппаратуры. Далее огонь может распространиться на другие объекты рабочего помещения, создавая техногенную чрезвычайную ситуацию.

Федеральный закон №123 "Технический регламент о требованиях пожарной безопасности" устанавливает основные принципы установления пожарной безопасности. Для предупреждения возникновения пожаров необходимо прежде всего следовать технике безопасности пользования персональным компьютером. Приведенный технический регламент предусматривает предотвращение пожаров путем исключения условий образования горючей среды (статья 49) и исключений условий образования в горючей среде источников зажигания (статья 50). [23]

Исключение условий образования горючей среды в сложившихся рабочих условиях включает в себя: применение минимального количества горючих веществ и материалов и наиболее безопасного их размещения, изоляция источников зажигания от горючих веществ, понижение концентрации окислителя в горючей среде в защищаемом объеме, удаление из помещений пожароопасных отходов производства, отложений пыли и пуха. Исключение источников зажигания достигается путем применения оборудования,

исключающих образование статического электричества, устройство молниезащиты зданий, сооружений и оборудования, применение устройств, исключающих возможность распространения пламени из одного объема в смежный.

При возникновении пожара необходимо обеспечить технические и организационные мероприятия по оповещению, эвакуации людей, тушению 75 пожара. При обнаружении пожара необходимо оповестить пожарную службу посредством пожарной сигнализации и провести эвакуацию через эвакуационные пути и выходы. Ликвидация последствий пожара обеспечивается автоматическими установками пожаротушения и пожарными службами.

### **5.3.1. Пожар в вычислительных центрах**

Тяжелые вычисления часто выполняются в вычислительных центрах. Пожары в вычислительном центре представляют особую опасность, так как сопряжены с большими материальными потерями. Характерная особенность вычислительных центров — небольшие площади помещений. Как известно, пожар может возникнуть при взаимодействии горючих веществ, окислителя и источников зажигания. В помещениях вычислительных центров присутствуют все три основных фактора, необходимых для возникновения пожара.[26]

Горючими компонентами на вычислительных центрах являются: строительные материалы для акустической и эстетической отделки помещений, перегородки, двери, полы, изоляция кабелей и др.[26]

Источниками зажигания в вычислительных центрах могут быть электрические схемы компьютера, приборы, применяемые для технического обслуживания, устройства электропитания, кондиционирования воздуха, где в результате различных нарушений образуются перегретые элементы, электрические искры и дуги, способные вызвать загорания горючих материалов.[26]

В современных компьютерах очень высокая плотность размещения элементов электронных схем. В непосредственной близости друг от друга располагаются соединительные провода, кабели. При протекании по ним электрического тока выделяется значительное количество теплоты. При этом возможно оплавление изоляции. Для отвода избыточной теплоты от оборудования служат системы вентиляции и кондиционирования воздуха. При постоянном действии эти системы представляют собой дополнительную пожарную опасность.[26]

Для большинства помещений вычислительных центров установлена категория пожарной опасности В.[26]

Одна из наиболее важных задач пожарной защиты — защита строительных помещений от разрушений и обеспечение их достаточной прочности в условиях воздействия высоких температур при пожаре. Учитывая высокую стоимость электронного оборудования вычислительного центра, а также категорию его пожарной опасности, здания для вычислительного центра и части здания другого назначения, в которых предусмотрено размещение компьютеров, должны быть первой и второй степени огнестойкости. Для изготовления строительных конструкций используются, как правило, кирпич, железобетон, стекло, металл и другие негорючие материалы. Применение дерева должно быть ограничено, а в случае использования необходимо пропитывать его огнезащитными составами.[26]

#### **5.4. Правовые и организационные вопросы обеспечения безопасности**

Правовые и организационные вопросы задают специфичные требования для выбранного рода деятельности, а именно программирования. Для учёта всех требований законодательства по организации работы необходимо провести анализ специальных норм трудового законодательства и организационных мероприятий по компоновке рабочей зоны для обеспечения эффективности работы трудящихся.

Регулирование отношений между работником и работодателем, касающихся оплаты труда, трудового распорядка, особенности регулирования труда женщин, детей, людей с ограниченными способностями и проч., осуществляется законодательством РФ, а именно трудовым кодексом РФ [27].

Нормальная продолжительность рабочего времени не может превышать 40 часов в неделю. Порядок исчисления нормы рабочего времени на определенные календарные периоды (месяц, квартал, год) в зависимости от установленной продолжительности рабочего времени в неделю определяется федеральным органом исполнительной власти, осуществляющим функции по выработке государственной политики и нормативно-правовому регулированию в сфере труда. Продолжительность ежедневной работы (смены) не может превышать:

- Для работников в возрасте от 15 до 16 лет – 5 часов, в возрасте от 16 до 18 лет – 7 часов;
- Для учащихся общеобразовательных учреждений, образовательных учреждений начального и среднего профессионального образования, совмещающих в течение учебного года учебу с работой, в возрасте от 14 до 16 лет – 2,5 часа, в возрасте от 16 до 18 лет – 4 часов;
- Для инвалидов – в соответствии с медицинским заключением, выданным в порядке, установленном федеральными законами и иными нормативными правовыми актами российской федерации.

Для работников, занятых на работах с вредными и (или) опасными условиями труда, где установлена сокращенная продолжительность рабочего времени, максимально допустимая продолжительность ежедневной работы (смены) не может превышать:

- При 36-часовой рабочей неделе - 8 часов;
- При 30-часовой рабочей неделе и менее - 6 часов.

Продолжительность работы (смены) в ночное время сокращается на один час без последующей отработки. К работе в ночное время не допускаются:

- беременные женщины;

- работники, не достигшие возраста 18 лет, за исключением лиц, участвующих в создании и (или) исполнении художественных произведений,
- и других категорий работников в соответствии с настоящим Кодексом и иными федеральными законами.

#### **5.4.1. Специальные правовые нормы трудового законодательства**

Специальные ограничения работы специалиста – программиста практически ограничиваются общими нормами трудового законодательства и СанПиН 2.2.2/2.4.1340-03.

Согласно 12 пункту СанПиН 2.2.2/2.4.1340-03 женщины со времени установления беременности допускаются к работе на ПЭВМ до 3 часов за рабочую смену. Лица работающие с ПЭВМ более 50% рабочего времени должны проходить обязательные предварительные медицинские осмотры. [21]

Согласно постановлению главного государственного санитарного врача РФ от 03.06.2003 № 118 продолжительность непрерывной работы без регламентированного перерыва должна быть меньше 1 часа. В зависимости от категории работы с ПЭВМ уровень нагрузки за рабочую смену может достигать до 6 часов с суммарным временем регламентированным временем перерывов от 50 мин. до 140 мин. В зависимости от продолжительности смены. Рекомендуется организовывать перерывы на 10-15 мин. через каждые 45-50 мин. работы. [24]

#### **5.4.2. Организационные мероприятия по компоновке рабочей зоны**

Организационные мероприятия по компоновке рабочей зоны заключаются в правильном расположении персонального компьютера относительно оператора и правильном положении самого оператора во время работы. Рабочая зона, в том числе и рабочее место должна обеспечивать наивысшую эффективность работы оператора с минимальными нагрузками на здоровье человека. При несоблюдении условий работник может получить производственные травмы и хронические заболевания.

Большинство мероприятий регулируются 10 пунктом СанПиН 2.2.2/2.4.1340-03 “Требования к организации и оборудованию рабочих мест с ПЭВМ для взрослых пользователей”. В этом документе предусматриваются правила расположения высоты рабочей поверхности, рабочего стола, параметры конструкции рабочего стула по ширине, глубине, высоте, краях, углах наклона, подлокотниках. Данные параметры должны регулироваться для обеспечения комфорта работника. Помимо этого, рабочее место необходимо оборудовать подставкой для ног определенных параметров, клавиатуру как главный инструмент работника располагать определенным образом. [21]

Более подробно сидячее рабочее место определяется в ГОСТ 12.2.032-78. Этот документ описывает размерные характеристики рабочего места, требования к размещению органов управления, требования к размещению средств отображения информации. [25]

При следовании этим документам необходимо также учитывать размеры программиста, которые могут влиять на некоторые из заданных величин, которые были выведены по среднестатистическим показателям.

Следование установленным правилам и дополнительным медицинским рекомендациям, выполнением специальных физических упражнений, снижающих нагрузку, позволит избежать травм и болезней, увеличить производительность работников, предупреждая раннее утомление.

#### **5.4.3. Влияние разработанной системы на работу пользователя.**

Система поддержки развертывания распределенных приложений позволит управлять и следить за работой распределенного приложения. Данная система рассчитана на применение на производствах, распределенных в пространстве. Так тяжелые вычисления будут выполняться в вычислительных центрах, данные будут поступать с удаленных датчиков.

Система может использоваться в целом ряде отраслей: нефтедобывающая, металлургия, химическое производство, электроснабжение, нефтепереработка.

Для каждой отрасли необходимы свои вычислительные компоненты, для этого система предоставляет API для их создания и документы, спецификацию, то есть система является каркасом для работы распределенных приложений.

Для запуска распределенного приложения необходимо его настроить по сети, для этого администратор приложения должен работать с сетевым администратором. В результате работы создаются конфигурационные файлы приложения. В случае отказа какого-либо из сетевых узлов, администратор должен реконфигурировать.

## ЗАКЛЮЧЕНИЕ

В ходе работы была достигнута главная цель выпускной квалификационной работы – была разработана система поддержки развертывания распределенных приложений.

Для достижения данной цели были решены следующие задачи:

- Были выбраны средства реализации;
- Была спроектирована архитектура;
- Была реализована система.

Система разработана на языке C++ в среде разработки Qt Creator с использованием сетевой библиотеки Boost::Asio. Соответственно были улучшены навыки программирования на C++, был получен опыт работы с библиотекой Boost.

Разработанная система соответствует функциональным требованиям:

- Поддерживает развертку распределенного приложения согласно конфигурации;
- Позволяет управлять работой приложения;
- Позволяет поддерживать работу распределенного приложения.

Разработка была командной, в качестве личного вклада была разработана и протестирована библиотека сетевого взаимодействия, реализованы классы маршрутизации пакетов и было добавлено сетевое взаимодействие в клиентское приложение. То есть была проделана работа по реализации сетевой части системы.

Разработанный продукт планируется использовать в составе организации АО «Атомик Софт».

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Distributed computing [Электронный ресурс] URL: [https://en.wikipedia.org/wiki/Distributed\\_computing](https://en.wikipedia.org/wiki/Distributed_computing) (дата обращения: 12.05.17);
2. What Is a Distributed Application [Электронный ресурс] URL: [https://docs.oracle.com/cd/E13203\\_01/tuxedo/tux71/html/addist2.htm](https://docs.oracle.com/cd/E13203_01/tuxedo/tux71/html/addist2.htm) (дата обращения: 12.05.17);
3. Protocol Buffers [Электронный ресурс] URL: <https://developers.google.com/protocol-buffers> (дата обращения: 12.05.17);
4. Developer Guide [Электронный ресурс] URL: <https://developers.google.com/protocol-buffers/docs/overview> (дата обращения: 12.05.17);
5. twitter/elephant-bird [Электронный ресурс] URL: <https://github.com/twitter/elephant-bird> (дата обращения: 12.05.17);
6. Protobuf [Электронный ресурс] URL: <https://cwiki.apache.org/confluence/display/CAMEL/Protobuf> (дата обращения: 12.05.17);
7. PBF Format [Электронный ресурс] URL: [http://wiki.openstreetmap.org/wiki/PBF\\_Format](http://wiki.openstreetmap.org/wiki/PBF_Format) (дата обращения: 12.05.17);
8. Qt Creator Manual [Электронный ресурс] URL: <http://doc.qt.io/qtcreator/> (дата обращения: 29.05.17);
9. Developer Network [Электронный ресурс] URL: <https://msdn.microsoft.com/ru-ru/> (дата обращения: 12.05.17);
10. Adaptive Communication Environment [Электронный ресурс] URL: <http://www.cs.wustl.edu/~schmidt/ACE.html> (дата обращения: 12.12.16);
11. РОСО Libraries [Электронный ресурс] URL: <https://росoproject.org> (дата обращения: 13.05.17);

12. Boost.Asio [Электронный ресурс] URL:  
[http://www.boost.org/doc/libs/1\\_64\\_0/doc/html/boost\\_asio.html](http://www.boost.org/doc/libs/1_64_0/doc/html/boost_asio.html) (дата обращения: 26.05.17);
13. Language Guide [Электронный ресурс] URL:  
<https://developers.google.com/protocol-buffers/docs/proto> (дата обращения: 22.05.17);
14. Boost Library Documentation [Электронный ресурс] URL:  
<http://www.boost.org/doc/libs/> (дата обращения: 30.05.17);
15. Boost.Signals2 [Электронный ресурс] URL:  
[http://www.boost.org/doc/libs/1\\_64\\_0/doc/html/signals2.html](http://www.boost.org/doc/libs/1_64_0/doc/html/signals2.html) (дата обращения: 18.05.17);
16. Microsoft Visual Studio [Электронный ресурс] URL:  
<https://www.visualstudio.com> (дата обращения: 12.05.17);
17. СанПиН 2.2.4-96 «Гигиенические требования к микроклимату производственных помещений»
18. ГОСТ 12.1.030-81 «Электробезопасность. Защитное заземление, зануление»
19. СанПиН 2.2.4.1294-03 «Гигиенические требования к аэроионному составу воздуха производственных и общественных помещений»
20. ГОСТ 12.1.038-82 «Система стандартов безопасности труда. Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов»
21. СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы»
22. ГОСТ Р 50571.3-94 «Требования по обеспечению безопасности. Защита от поражений электрическим током.»
23. Федеральный закон №123 «Технический регламент о требованиях пожарной безопасности» [Электронный ресурс]. URL:

[http://www.consultant.ru/document/cons\\_doc\\_LAW\\_78699/](http://www.consultant.ru/document/cons_doc_LAW_78699/). – Яз. рус. Дата обращения: 05.06.2017 г.

24. Постановление Главного государственного санитарного врача РФ от 03.06.2003 № 118 [Электронный ресурс]. URL: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_42836/](http://www.consultant.ru/document/cons_doc_LAW_42836/), свободный. – Яз. рус. Дата обращения: 05.06.2017г.
25. ГОСТ 12.2.032-78 «Рабочее место при выполнении работ сидя. Общие эргономические требования.»
26. Пособие по БЖД. [Электронный ресурс] URI: <http://www.studfiles.ru/preview/434015>, свободный. – Загл. с экрана. – Яз. рус. Дата обращения: 03.06.2017
27. ТК РФ КонсультантПлюс – надежная правовая служба. Трудовой Кодекс Российской Федерации [Электронный ресурс]. Режим доступа – [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_34683/](http://www.consultant.ru/document/cons_doc_LAW_34683/) (Дата обращения 22.05.2017).

## ПРИЛОЖЕНИЕ А

### Матрица SWOT

	<p>С1. Повышенная надежность пересылаемых между узлами пакетов;          С2. Высокая скорость передачи пакетов между узлами и компонентами;          С3. Кроссплатформенность;          С4. Простое конфигурирование и развертывание;          С5. Наличие графических инструментов;          С6. Простота реконфигурирования.</p>	<p>Сл1. Нет графических средств редактирования описаний и схем развертывания, конфигурация описывается в xml-файлах;          Сл2. Нет поддержки динамических библиотек компонентов, вместо этого компоненты встраиваются в исполнительный файл.</p>
<p>В1. Создание новых вычислительных компонентов;          В2. Создание графических средств описания конфигурации;          В3. Создание графического интерфейса для приложения сетевого узла.          В4. Добавление поддержки динамических библиотек компонентов;</p>	<p>Есть возможность создания новых вычислительных компонентов.          Есть возможность добавления поддержки динамических библиотек компонентов.          Если возможность создания графического интерфейса приложения сетевого узла.</p>	<p>Необходимо будет создавать новые вычислительные компоненты для использования их в различных отраслях.</p>
<p>У1. Разработка конкурентами более дешевой аналогичной системы;          У2. Разработка конкурентами более быстрой системы;          У3. Разработка конкурентами с более удобным интерфейсом;          У4. Обнаружение ошибок работы;          У5. Выход из строя машин, участвующих в работе развернутой конфигурации.</p>	<p>Для предотвращения возможности падения интереса к приложению из-за появления на российском рынке продуктов конкурентов необходимо улучшить функционал приложения.</p>	<p>Для того, чтобы приложение оставалось конкурентоспособным, необходимо улучшать его скорость работы и создавать новые вычислительные компоненты.</p>

## ПРИЛОЖЕНИЕ Б

### Временные показатели научного исследования

Название работы	Трудоёмкость работ									Исполнители			Длительность работ в рабочих днях $T_{pi}$			Длительность работ в календарных днях $T_{ki}$		
	tmin, чел-дни			tmax, чел-дни			toжi, чел-дни											
	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Анализ существующих аналогов	0.5	0.5	0.5	1	1	1	0.7	0.7	0.7	1	1	1	0.7	0.7	0.7	1.1	1.1	1.1
Определение функционала приложения	3	3	3	5	5	5	4	4	4	2	2	2	2.4	2.4	2.4	3.7	3.7	3.7
Проектирование архитектуры приложения	6	6	6	8	8	8	6.8	6.8	6.8	2	2	2	3.4	3.4	3.4	5.3	5.3	5.3
Выбор технологий для реализации	3	3	3	4	4	4	3.4	3.4	3.4	2	2	2	1.2	1.2	1.2	1.9	1.9	1.9
Реализация библиотек сетевого взаимодействия	5	5	5	7	7	7	7	7	7	2	2	2	0.6	0.6	0.6	0.9	0.9	0.9
Разработка структуры пакетов для сериализации	4	4	4	5	5	5	4.4	4.4	4.4	1	1	1	4.4	4.4	4.4	6.9	6.9	6.9
Реализация маршрутизации	5	5	5	8	8	8	6.8	6.8	6.8	1	1	1	1.4	1.4	2.8	2.2	2.2	4.4
Реализация приложения	7	7	7	8	8	8	7.4	7.4	7.4	1	1	1	1.4	1.4	1.4	2.2	2.2	2.2

сетевого узла																		
Разработка демонстрационных компонентов	4	4	4	5	5	5	4.5	4.5	4.5	1	1	1	12	19.2	14.4	18.7	29.9	22.5
Реализация сетевой части в клиентском приложении	7	7	7	8	8	8	7.8	7.2	7.6	1	1	1	5.8	8.2	16.4	9	12.8	25.6
Реализация сбора и отображения статистики	7	7	7	9	9	9	8	8	8	1	1	1	11.2	11.2	11.2	17.5	17.5	17.5
Тестирование	4	4	4	6	6	6	4.8	4.8	4.8	2	2	2	2.4	2.4	2.4	3.7	3.7	3.7
Исправление ошибок	8	10	12	10	12	15	9.2	11	13.5	1	1	1	8.8	10.8	13.2	13.7	16.8	20.6

## ПРИЛОЖЕНИЕ В

### Календарный план-график

