

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

Институт Кибернетики
Направление подготовки 09.04.01 Информатика и вычислительная техника
Кафедра Информационных систем и технологий

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
Проектирование инфраструктуры и реализация распределенного информационно-вычислительного кластера на базе персональных компьютеров

УДК 004.75.032.24:004.382.7

Студент

Группа	ФИО	Подпись	Дата
8ВМ5Б	Скопченко А.		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Ботыгин И.А	К.Т.Н.		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Конопский В.Ю.	К.Э.Н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Акулов П.А.			

ДОПУСТИТЬ К ЗАЩИТЕ:

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

Томск – 2017 г.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП

Код результатов	Результат обучения (выпускник должен быть готов)
Общепрофессиональные компетенции	
P1	Воспринимать и самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте.
P2	Владеть и применять методы и средства получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе в глобальных компьютерных сетях.
P3	Демонстрировать культуру мышления, способность выстраивать логику рассуждений и высказываний, основанных на интерпретации данных, интегрированных из разных областей науки и техники, выносить суждения на основании неполных данных, анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями.
P4	Анализировать и оценивать уровни своих компетенций в сочетании со способностью и готовностью к саморегулированию дальнейшего образования и профессиональной мобильности. Владеть, по крайней мере, одним из иностранных языков на уровне социального и профессионального общения, применять специальную лексику и профессиональную терминологию языка.
Профессиональные компетенции	
P5	Выполнять инновационные инженерные проекты по разработке аппаратных и программных средств автоматизированных систем различного назначения с использованием современных методов проектирования, систем автоматизированного проектирования, передового опыта разработки конкурентно способных изделий.
P6	Планировать и проводить теоретические и экспериментальные исследования в области проектирования аппаратных и программных средств автоматизированных систем с использованием новейших достижений науки и техники, передового отечественного и зарубежного опыта. Критически оценивать полученные данные и делать выводы.
P7	Осуществлять авторское сопровождение процессов проектирования, внедрения и эксплуатации аппаратных и программных средств автоматизированных систем различного назначения.
Общекультурные компетенции	
P8	Использовать на практике умения и навыки в организации исследовательских, проектных работ и профессиональной эксплуатации современного оборудования и приборов, в управлении коллективом.
P9	Осуществлять коммуникации в профессиональной среде и в обществе в целом, активно владеть иностранным языком, разрабатывать документацию, презентовать и защищать результаты инновационной инженерной деятельности, в том числе на иностранном языке.
P10	Совершенствовать и развивать свой интеллектуальный и общекультурный уровень. Проявлять инициативу, в том числе в ситуациях риска, брать на себя всю полноту ответственности.
P11	Демонстрировать способность к самостоятельному обучению новым методам исследования, к изменению научного и научно-производственного

Код результатов	Результат обучения (выпускник должен быть готов)
	профиля своей профессиональной деятельности, способность самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности, способность к педагогической деятельности.

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

Институт кибернетики
Направление подготовки 09.04.01 Информатика и вычислительная техника
Кафедра Информационных систем и технологий

УТВЕРЖДАЮ:
Зав. Кафедрой

(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Магистерской диссертации

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

Студенту:

Группа	ФИО
8ВМ5Б	Скопченко Антон

Тема работы:

Проектирование инфраструктуры и реализация распределенного информационно-вычислительного кластера на базе персональных компьютеров.

Утверждена приказом директора (дата, номер)

Срок сдачи студентом выполненной работы:

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе

(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).

Язык программирования: Java и комплект разработчика: Java Development Kit 8. Среда разработки: NetBeans IDE 8.2.

<p>Перечень подлежащих исследованию, проектированию и разработке вопросов</p> <p><i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p>	<ol style="list-style-type: none"> 1. Аналитический обзор программного обеспечения для организации распределенных вычислительных систем. 2. Тестирование производительности кластера. 3. Разработка функциональной структуры распределенной вычислительной системы. 4. Разработка модулей и алгоритмов распределенной вычислительной системы. 5. Организация распределенного файлового хранилища данных на кластере из персональных компьютеров. 6. Технологическая схема реализации и использования информационно-вычислительного кластера на базе персональных компьютеров. 7. Обсуждение результатов выполненной работы. 8. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение. 9. Социальная ответственность.
<p>Перечень графического материала</p> <p><i>(с точным указанием обязательных чертежей)</i></p>	<ol style="list-style-type: none"> 1. Схема передачи файла в кластер. 2. Схема возврата файла агенту. 3. Архитектура информационно-вычислительного кластера 4. Схема коммуникаций в распределенной вычислительной системе. 5. Архитектура BOINC.
<p>Консультанты по разделам выпускной квалификационной работы</p>	
<p>Раздел</p>	<p>Консультант</p>
<p>Финансовый менеджмент, ресурсоэффективность и ресурсосбережение</p>	<p>Доцент Конотопский В.Ю.</p>
<p>Социальная ответственность</p>	<p>Ассистент Акулов П.А.</p>
<p>Раздел на иностранном языке</p>	<p>Старший преподаватель Кудряшова А.В.</p>
<p>Названия разделов, которые должны быть написаны на русском и иностранном языках:</p>	
<p>Исследование эффективности информационно-вычислительного кластера на базе персональных компьютеров. Разработка архитектуры распределенной вычислительной</p>	

системы на базе персональных компьютеров.

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

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Ботыгин И.А.	к.т.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ВМ5Б	Скопченко Антон		

Реферат

Выпускная квалификационная работа содержит 111 с., 49 рис., 28 табл.

Ключевые слова: Распределенные вычислительные системы, GRID-системы, параллельные вычисления, кластерные вычислительные системы, сетевое администрирование, распределенное хранение данных.

Объект исследования: распределенные вычислительные системы и технологии.

Цель работы: разработка программного инструментария для формирования информационно-вычислительных кластеров на базе персональных компьютеров.

В процессе исследования проводились работы по анализу программного обеспечения для организации распределенных вычислительных систем, тестированию производительности кластера, разработки функциональной структуры распределенной вычислительной системы и технологической схема реализации и использования информационно-вычислительного кластера на базе персональных компьютеров.

В результате исследования была сформирована концепция распределенного вычислительного кластера на базе персональных компьютеров, спроектирована архитектура гетерогенной распределенной информационно-вычислительной системы на базе персональных компьютеров, разработано программное обеспечение всех основных компонентов вычислительного кластера. Для иллюстрации функциональных возможностей спроектированного информационно-вычислительного кластера на его инфраструктуре было развернуто распределенное хранилище данных.

Основные конструктивные, технологические и технико-эксплуатационные характеристики: платформонезависимость, горизонтальная масштабируемость.

Степень внедрения: находится в опытной эксплуатации.

Область применения: Корпорации с целью динамического создания конфигурации распределенных информационно-вычислительных кластеров для решения различных вычислительно-емких задач, не требующих работы в режиме реального времени или распределенного хранения больших данных.

Экономическая эффективность проекта обусловлена необходимостью ускорить расчеты вычислительно емких задач, не требующих обработки в режиме реального времени. Конечная система позволит на основе имеющихся персональных компьютерах, развернуть распределенный информационно-вычислительный кластер, что позволит без особых затрат получить необходимую вычислительную мощность, не арендуя при этом суперкомпьютер. Использование подобной системы в научной сфере, позволит повысить эффективность исследований, так как пользователю будет требоваться меньше времени на ожидание результата. Подводя итог вышесказанному, отметим, что экономический эффект от реализации проекта, может быть выражен в отказе от затрат на аренду или покупку мощных суперкомпьютеров, повышении оперативности вычислений.

В будущем планируется улучшить протокол взаимодействия основных компонентов информационно-вычислительного кластера за счет использования коммуникационного сервера с целью обеспечения подключения любых компьютеров, имеющих выход в интернет.

Определения

Распределенные вычислительные системы – это системы, состоящие из физических компьютеров или программных средств, реализующих параллельную обработку на множестве вычислительных узлов.

GRID системы – это вид распределенных вычислений, в котором суперкомпьютер представлен в виде кластера. В данной системе все узлы соединены с помощью сети слабосвязных гетерогенных связей, узлы которого работают вместе над выполнением различных заданий.

Кластер – это некоторое количество компьютеров, которые объединены высокоскоростными каналами связи.

Ethernet – семейство технологий пакетной передачи данных для компьютерных сетей.

TCP/IP – набор сетевых протоколов передачи данных, используемые в сетях.

Поток данных – абстракция, используемая для чтения или записи файлов, сокетов и др.

Параллельные вычисления – организация компьютерных вычислений, в которых при разработке программ учитывается, что программы должны работать как набор взаимодействующих вычислительных процессов работающих параллельно.

Узел – аппаратное устройство в РВС.

P2P – (англ. Peer-to-peer) – одноранговая сеть, основанная на равноправии участников.

MapReduce – это модель распределенных вычислений от компании Google. Используется для параллельных вычислений при работе над очень большими данными в компьютерных кластерах.

Суперкомпьютер - вычислительная машина, которая значительно превосходит по своим техническим параметрам большинство существующих компьютеров [15].

Обозначения и сокращения

ОС – Операционная система.

ЦП - Центральный процессор.

ПО – Программное обеспечение.

БД – База данных.

ООП – Объектно-ориентированное программирование

ТСР (англ. Transmission control protocol) – Протокол управления передачей.

IP (англ. Internet Protocol) – Межсетевой протокол.

Поток – Поток данных.

РВС – Распределенная вычислительная система.

НИИ – Научно исследовательский институт.

P2P (англ. Peer-to-peer) – одноранговая сеть.

ЦОД – Центр Обработки Данных.

Оглавление

Введение.....	18
1. Обзор и анализ программного обеспечения для организации распределенных вычислительных систем.....	21
1.1. Фреймворк Apache Hadoop.....	23
1.2. Программная платформа VOINC.....	30
1.3. Программное обеспечение HTCCondor.....	33
1.4. Инструментарий Globus Toolkit.....	35
1.5. Платформа Unicore.....	36
2. Исследование эффективности информационно-вычислительного кластера на базе персональных компьютеров.....	40
3. Разработка архитектуры распределенной вычислительной системы на базе персональных компьютеров.....	45
3.1. Основные компоненты распределенной вычислительной системы на базе ПК.....	45
3.2. Модули распределенной вычислительной системы.....	46
3.2.1. Модуль сбора информации.....	47
3.2.2. Модуль сохранения собранной информации.....	47
3.2.3. Модуль разбиения и сборки файлов.....	47
3.2.4. Модуль тестирования.....	48
3.2.5. Модуль получения и отправки данных.....	48
3.2.6. Модуль взаимодействия.....	49
3.3. Схема основных взаимодействий в распределённом вычислительном кластере.....	50
3.4. Организация распределенного файлового хранилища данных.....	52
4. Технологическая схема реализации и использования информационно- вычислительного кластера на базе персональных компьютеров.....	56
5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение....	65
5.1. Организация и планирование работ.....	65
5.1.1. Продолжительность этапов работ.....	66

5.1.2. Расчет нарастания технической готовности работ	69
5.2. Расчет сметы затрат на разработку ПО и проведение эксперимента	71
5.2.1. Расчет затрат на материалы	71
5.2.2. Расчет основной заработной платы.....	71
5.2.3. Расчет отчислений от заработной платы	72
5.2.4. Расчет затрат на электроэнергию	73
5.2.5. Расчет амортизационных расходов.....	73
5.2.6. Расчет прочих расходов.....	74
5.2.7. Расчет общей себестоимости разработки	74
5.2.8. Прибыль	75
5.2.9. НДС.....	75
5.2.10. Цена разработки НИР	75
5.3. Оценка экономической эффективности проекта	75
5.4. Оценка научно-технического уровня НИР.....	76
6. Социальная ответственность.....	79
6.1. Введение.....	79
6.2. Производственная безопасность	79
6.2.1. Вредные производственные факторы	80
6.2.1.1. Микроклимат рабочей среды.....	80
6.2.1.2. Недостаточная освещенность рабочей зоны.....	82
6.2.2. Опасные производственные факторы.....	83
6.2.2.1. Опасность поражения электрическим током	83
6.2.2.2. Опасность возникновения пожара	84
6.2.3. Мероприятия и рекомендации по устранению и минимизации	84
6.3. Экологическая безопасность.....	86
6.4. Безопасность в чрезвычайных ситуациях.....	87
6.5. Правовые и организационные вопросы обеспечения безопасности	89
6.5.1. Правовые нормы трудового законодательства для рабочей зоны пользователя персонального компьютера	89
6.5.2. Организационные мероприятия при компоновке рабочей зоны	91

7. Заключение.....	92
Список используемых источников.....	93
ПРИЛОЖЕНИЕ А.....	97

Введение

На сегодняшний день существует масса задач, которые требуют высокопроизводительные вычислений, и с каждым днем их становится больше. Зачастую такие задачи выполняются на одном высокопроизводительном компьютере, за исключением тех случаев, когда институт, лаборатория, предприятие и др., обладают суперкомпьютером. Одной из таких задач является набирающая популярность тема обработки больших данных, в связи с тем, что о человеке, собирается информация повсюду, будь то мобильное устройство, ноутбук или персональный компьютер в компании. Помимо того, что информация собирается, ее нужно, где-то хранить. Отсюда следует, что одной из проблем является хранение больших объемов данных. Сразу стоит отметить, что в данном исследовании нас не интересует обработка в реальном времени, так как реальные персональные компьютеры, даже объединенные в один кластер, через сеть, не могут идти ни в какое сравнение с высокопроизводительным кластером, состоящим из 1000 ядер и соединённых быстрыми шинами передачи данных. В данном исследовании, одним из вариантов реального использования разработанного программного обеспечения - это хранение и размещение больших данных, а здесь режим реального времени не всегда обязателен, так как это не критично, если программа проработает лишней час. Хотелось бы сразу отметить, что, конечно, такие крупные гиганты как Google умеют решать такие проблемы, путем постройки для себя огромных высокопроизводительных ЦОДов, но, к сожалению, не каждый может построить по такому ЦОДу. В связи с этим, данное исследование будет полезно для тех, кто не может построить себе ЦОД, зато имеет некоторое количество персональных компьютеров.

Целью выпускной квалификационной работы является разработка программного инструментария для формирования информационно-вычислительных кластеров на базе персональных компьютеров. Для достижения цели необходимо было решить следующие задачи:

- формирование информации об инфраструктуре проектируемого кластера на базе персональных компьютеров;
- разработка методов и алгоритмов организации взаимодействия узлов информационно-вычислительного кластера;
- разработка обобщенной функциональной структуры информационно-вычислительного кластера;
- исследование эффективности работы информационно-вычислительного кластера;
- организации распределенного хранилища данных.

Объектом исследования в данной работе выступают способы организации распределенных вычислительных систем.

Предметом исследования является задача разработки программного обеспечения реализующего инструментарий для формирования информационно-вычислительных кластеров на базе персональных компьютеров.

Научная новизной является организация комплексного подхода к проектированию и реализации информационно-вычислительного кластера корпорации на базе персональных компьютеров.

Краткое изложение **основного содержания работы**

В первой главе проведен обзор и анализ программного обеспечения для организации распределенных вычислительных систем, а именно: фреймворка Apache Hadoop, программной платформы BOINC, программного обеспечения HTCondor, инструментария Globus Toolkit и платформы Unicore.

Во второй главе было проведено исследование эффективности информационно-вычислительного кластера на базе персональных компьютеров, а именно были произведены тесты производительности с участием одного, двух, трех узлов вычислений.

В третьей главе приводится разработка архитектуры распределенной вычислительной системы на базе персональных компьютеров. Показаны основные компоненты распределенной вычислительной системы на базе ПК,

модули, входящие в состав кластера, схемы взаимодействий в кластере и организация распределенного файлового хранилища данных.

В четвертой главе приведена технологическая схема реализации и использования информационно-вычислительного кластера на базе ПК. Проиллюстрирована работа разработанного программного обеспечения.

В пятой главе приведен анализ финансово-экономических аспектов научно-исследовательской работы с целью оценки трудовых и денежных затрат, направленных на реализацию данного исследования.

В шестой главе приведен анализ различных вредных и опасных факторов производственной среды для пользователей персональных компьютеров при работе с разработанным программным продуктом.

1. Обзор и анализ программного обеспечения для организации распределенных вычислительных систем.

Для начала необходимо отметить, что распределенные вычислительные системы, основанные на добровольных вычислениях, строятся на базе обычных компьютеров и выгодно отличаются от суперкомпьютеров тем, что для организации связи между узлами может быть использована технология Ethernet [11]. В связи с этим, обычные персональные компьютеры, могут быть объединены в распределенную вычислительную систему. Ресурсы, полученные после объединения, могут быть использованы для решения последующих задач.

Для распределенных вычислительных систем характерны особенности:

- масштабируемость;
- отсутствие синхронности в работе, так как нет единого времени работы компонентов входящих в узлы системы;
- отсутствие общей памяти, в связи с этим необходима организация взаимодействия между узлами системы;
- узлы системы могут быть территориально распределены;
- узлы системы независимы и гетерогенны. Узлы, входящие в распределенную вычислительную систему, могут иметь различные параметры, такие как производительность, количество оперативной памяти и других характеристик. Узлы могут предоставлять необходимые ресурсы для пользы кластера.

Распределенная система может обладать следующими свойствами:

- прозрачность;
- открытость;
- безопасность;
- масштабируемость.

Прозрачность распределенной системы – это способность системы скрывать свою распределенную природу, такую как распределение ресурсов, процессов и др., от пользователя и представляться для него единой системой.

Открытая распределенная система – это способность системы предоставлять службы, вызов которых предполагает стандартный синтаксис и семантику. Службы в распределенных системах определяются через интерфейсы. Распределенная система должна обладать способностью к взаимодействию, то есть предоставлять необходимый интерфейс по просьбе другого процесса. Так же компоненты разных производителей должны иметь способность совместно работать. Для этого необходимо, чтобы они соответствовали общему стандарту. Для открытой системы характерна переносимость, то есть способность системы, одинаково исполняться на каждом из узлов кластера, не завися от его параметров. Гибкость открытой системы, это способность простого конфигурирования системы.

Масштабируемость – это способность распределенной вычислительной системы, эффективно работать при увеличении числа узлов. Систему считают масштабируемой, если она способна увеличивать свою производительность при включение нового узла. Масштабируемость делят на следующие направления:

- вертикальная и горизонтальная масштабируемость. Горизонтальная масштабируемость – это включение в кластер новых узлов. Для вертикальной характерна способность системы увеличивать свою производительность, например заменой комплектующих;
- территориальная масштабируемость – это возможность системы сохранять свою производительность при географическом разнесении компонентов системы;
- административная масштабируемость предоставляет простоту обслуживания различными организациями одной распределенной системой.

Общедоступные ресурсы [12] входящие в распределенную вычислительную систему включают:

- Ресурсы хранения – это способность системы, реализовать универсальный интерфейс по передаче и управлению данными. Основной характеристикой является объем.

- Вычислительные ресурсы – это способность системы, предоставлять пользователям процессорные мощности для вычислений. Ресурсами являются как узлы кластера, так и в целом кластер. Основной характеристикой является производительность.

- Информационные ресурсы каталоги – один из видов ресурсов хранения, характеризующиеся хранением метаданных о других узлах системы.

- сетевой ресурс - это связь между узлами кластера. Основой характеристикой является скорость передачи данных.

1.1. Фреймворк Apache Hadoop

Проект фонда Apache Software Foundation, свободно распространяемый набор утилит, библиотек и фреймворк, для разработки и выполнения распределенных вычислительных программ, которые состоят из кластеров, в которые входят сотни и тысячи узлов [1]. Используется для реализации поисковых и контекстных механизмов высоконагруженных веб-сайтов, например: Yahoo и Facebook. Проект написан на Java в рамках модели MapReduce [2].

Данное программное обеспечение включает в себя четыре модуля:

- Hadoop Common – это набор библиотек и утилит для управления файловыми системами и распределенной обработкой;

- HDFS – распределенная файловая система, предназначение которой, хранение файлов больших размеров;

- YARN – система планирования заданий и управления кластером;

- Hadoop MapReduce – платформа программирования и выполнения распределенных MapReduce-вычислений.

Hadoop Common

Состоит из библиотек управления файловыми системами, которые поддерживает Hadoop, а так же сценарии создания инфраструктуры и управления РВС, специально для удобства был сделан интерпретатор строки.

Большая часть команд сделана по аналогии с Unix – системами для удобства пользователей.

HDFS

Hadoop Distributed File System – распределенная файловая система, предназначение которой, хранить файлы больших объемов, которые в свою очередь поблочно распределены между узлами кластера. Все блоки HDFS имеют одинаковый размер, и каждый блок размещен в нескольких узлах кластера. Репликация делает кластеры отказоустойчивыми. При развертывании HDFS существует необходимость в центральном узле, который хранит метаданные файловой системы и др. информацию.

YARN

Yet Another Resource Negotiator – данный модуль отвечает за планирование заданий и управление ресурсами кластера. Предоставляет возможность работы нескольких задач параллельно в рамках кластера, при этом обеспечивая их изоляцию.

Hadoop MapReduce

Представляет собой каркас программирования РВС. Особенность данного модуля в том, что разработчику необходимо реализовать обработчик, который на каждом узле кластера обеспечит преобразования пар «key-value» в промежуточный набор, а так же обработчик, который сводит промежуточный набор в окончательный [3].

Архитектура Apache Hadoop

Взаимодействие модулей представлено на рисунке 1.

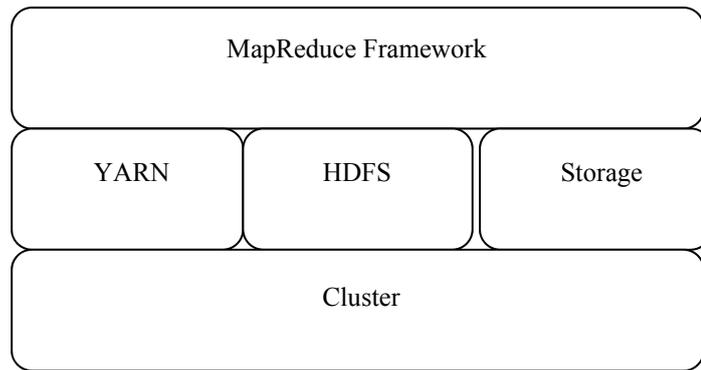


Рисунок 1 - Взаимодействие модулей Hadoop

- **Cluster (Кластер)** – состоит из набора узлов (компьютеров), является аппаратной частью инфраструктуры, может быть разделен на стойки.
- **YARN** – фреймворк, отвечающий за предоставление вычислительных ресурсов (память, процессор и др.). Содержит два важных элемента:

1. Менеджер ресурсов один на каждый кластер, располагается на главном узле в кластере. Данный менеджер позволяет знать, где располагаются дочерние узлы и сколько ресурсов имеется в наличии. Сюда же входит планировщик ресурсов. Подробнее о данном модуле можно увидеть на рисунке 2.



Рисунок 2 - Архитектура Менеджера ресурсов

2. Менеджер узлов располагается на каждом кластере и является дочерним узлом инфраструктуры. При запуске узла, он оповещает о себе менеджеру ресурсов, а так же периодически отправляет сигнал, о том, что он «жив». Каждый менеджер узла предлагает некоторое количество ресурсов для кластера. Во время работы планировщик ресурсов будет решать, каким образом использовать предлагаемые ему ресурсы. Подробнее о работе можно увидеть на рисунке 3.



Рисунок 3 - Архитектура менеджера узлов

Контейнер представляет собой часть емкости менеджера узла и используется клиентом для выполнения программ.

- **HDFS** – как и описывалось ранее это фреймворк, который отвечает за обеспечение постоянного, надежного и распределенного хранения данных.
- **Storage** – Альтернативное хранилище данных, например: Amazon и его S3.
- **MapReduce** – представляет программный уровень реализации модели MapReduce.

Важно отметить, что YARN и HDFS полностью независимы друг от друга, первый предоставляет ресурсы для запуска приложения, второй обеспечивает хранение. Ниже мы рассмотрим, как запускается приложения с помощью YARN (рисунок 4).

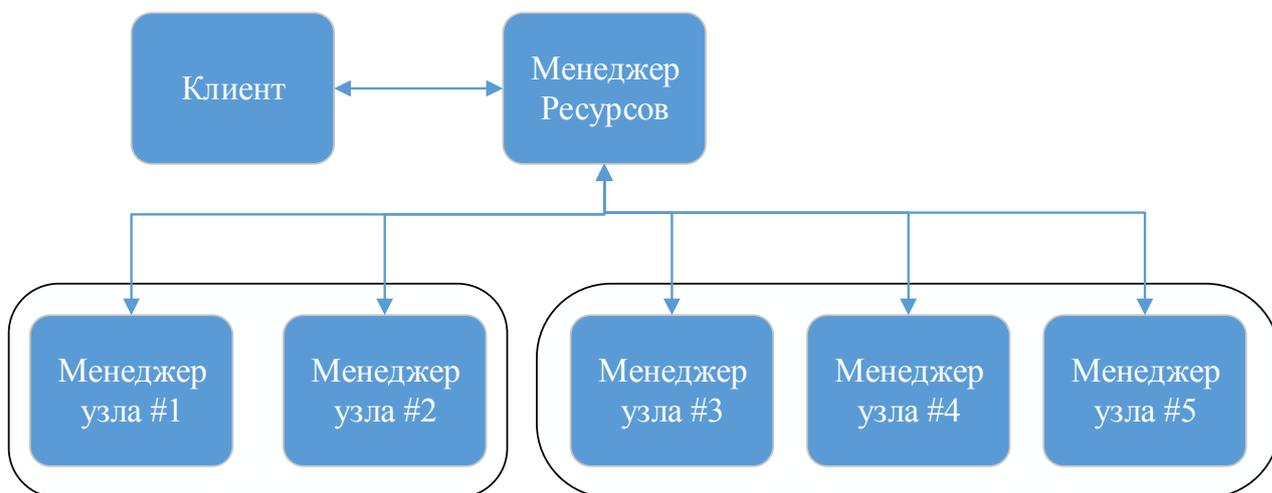


Рисунок 4 - Запуск приложения YARN

В данном модуле учувствуют как минимум три актера:

- Клиент – создает и отправляет задания.
- Менеджер ресурсов – он же мастер.
- Менеджер узлов.

Алгоритм работы представляется в следующем и отображен на рисунке

5.

1. Клиент подает заявку менеджеру ресурсов.
2. Менеджер ресурсов связывает с планировщиком ресурсов и назначает контейнер.
3. Менеджер ресурсов связывает менеджера узла с контейнером.
4. Менеджер узлов запускает контейнер.
5. Контейнер выполняет ведущие приложение.

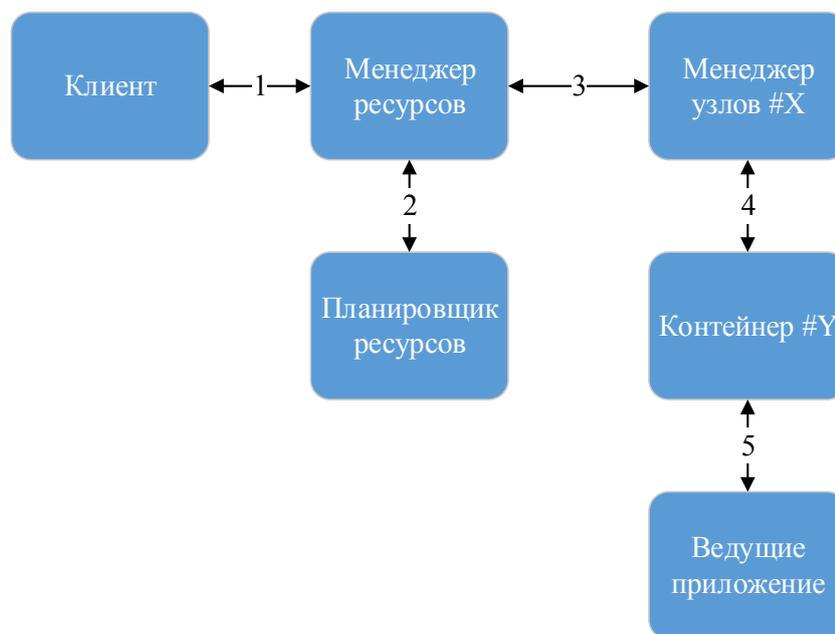


Рисунок 5 - Алгоритм работы

Ведущее приложение отвечает за выполнение только одного приложения. В случае необходимости он просит контейнеры у планировщика ресурсов (менеджер ресурсов), а так же выполняет специальные команды.

Единственной точкой отказа в YARN, является Менеджер ресурсов, но YARN распространяет метаданные связанные с запущенными приложениями между кластерами. Это позволяет снизить нагрузку на Менеджер ресурсов и быстрее восстановить его в случае необходимости.

Архитектура HDFS

Описание что такое HDFS было описано выше, здесь будет рассмотрено, как происходит запись в Hadoop и алгоритм действий, а так же опишем все ключевые моменты.

Состоит из узлов трех типов:

Главный или ведущий узел (NameNode) – обычно 1 на кластер, в нем хранятся все метаданные системы.

Второстепенный узел (SecondaryNode) – предназначается для быстрого восстановления главного узла, в случае выхода его из строя. Не является копией главного узла, а периодически проводит слияние состояния файловой системы и логов.

Узел данных (DataNode) – хранит непосредственно блоки файлов, таких узлов множество в кластере. Данный узел постоянно отправляет свой статус главному узлу, а так же периодически отправляет отчет обо всех хранимых на этом узле блоках, необходим для поддержания нужного уровня репликации.

Алгоритм записи данных в HDFS:

1. Клиент формирует из файла, цепочку блоков нужного размера.
2. Клиент соединяется с главным узлом и производит запрос на запись. В запрос входит кол-во блоков и нужный уровень репликации.
3. Главный узел отвечает клиенту цепочкой из узлов данных.
4. Клиент соединяется с данной цепочкой и производит запись первого блока своего нарезанного файла, на первый узел, этот узел данных делает запись на второй узел и т.д. В случае если клиенту не удалось подключиться к первому узлу, он делает попытку подключения ко второму и т.д. Если не получилось не с одним, то происходит откат.
5. По завершению, клиенту присылается ответ об успешной записи.
6. Сразу после того как клиент получил подтверждение об успешной записи, он оповещает об этом главный узел, и тот в свою очередь отправляет цепочку узлов данных для второго блока и повторяются шаги 4-6.

Пример работы алгоритма представлен на рисунке 6.

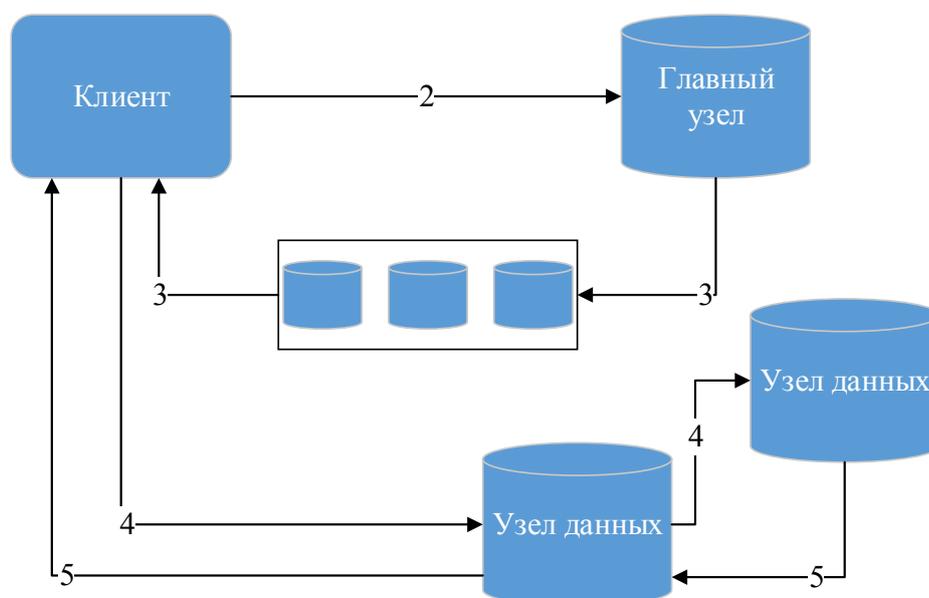


Рисунок 6 - Алгоритм работы HDFS

1.2. Программная платформа BOINC

Открытая инфраструктура Беркли для РВС (анг. Berkeley Open Infrastructure for Network Computing) – это открытая программная платформа для GRID-вычислений [4]. Используется, чтобы в кратчайшие сроки организовать распределенные вычисления. На сегодняшний день является универсальной платформой для проектов в различных областях: физика, математика и др. Дает возможность исследователю использовать вычислительный мощность ПК со всего мира. На данной платформе существует множество различных проектов.

Состоит из Серверной и клиентской части:

Сервер

Сам сервер состоит из веб-сайта проекта расположенного на Http-сервере, базы данных MySQL и набора управляющих демонов (планировщики, валидатор, генератор задание и др.). Сам сервер располагается только на Linux. Сервер представляет собой набор Php-скриптов которые необходимы для общего управления проектом. В базе данных хранятся пароли, записи заданий и др.

Клиент

Это универсальный клиент, который позволяет пользователю участвовать в проектах на BOINC платформе, пользователь может участвовать в нескольких проектах. В качестве интерфейса, клиенту устанавливается отдельное приложение, работа которого и заключается в мониторинге и управлении клиентом, так как сам клиент работает по технологии TCP/IP и работает в фоне. Для работы использовать менеджер не обязательно.

Архитектура BOINC

Архитектура представлена на рисунке 7, подробнее о каждом элементе будет рассказано после рисунка.

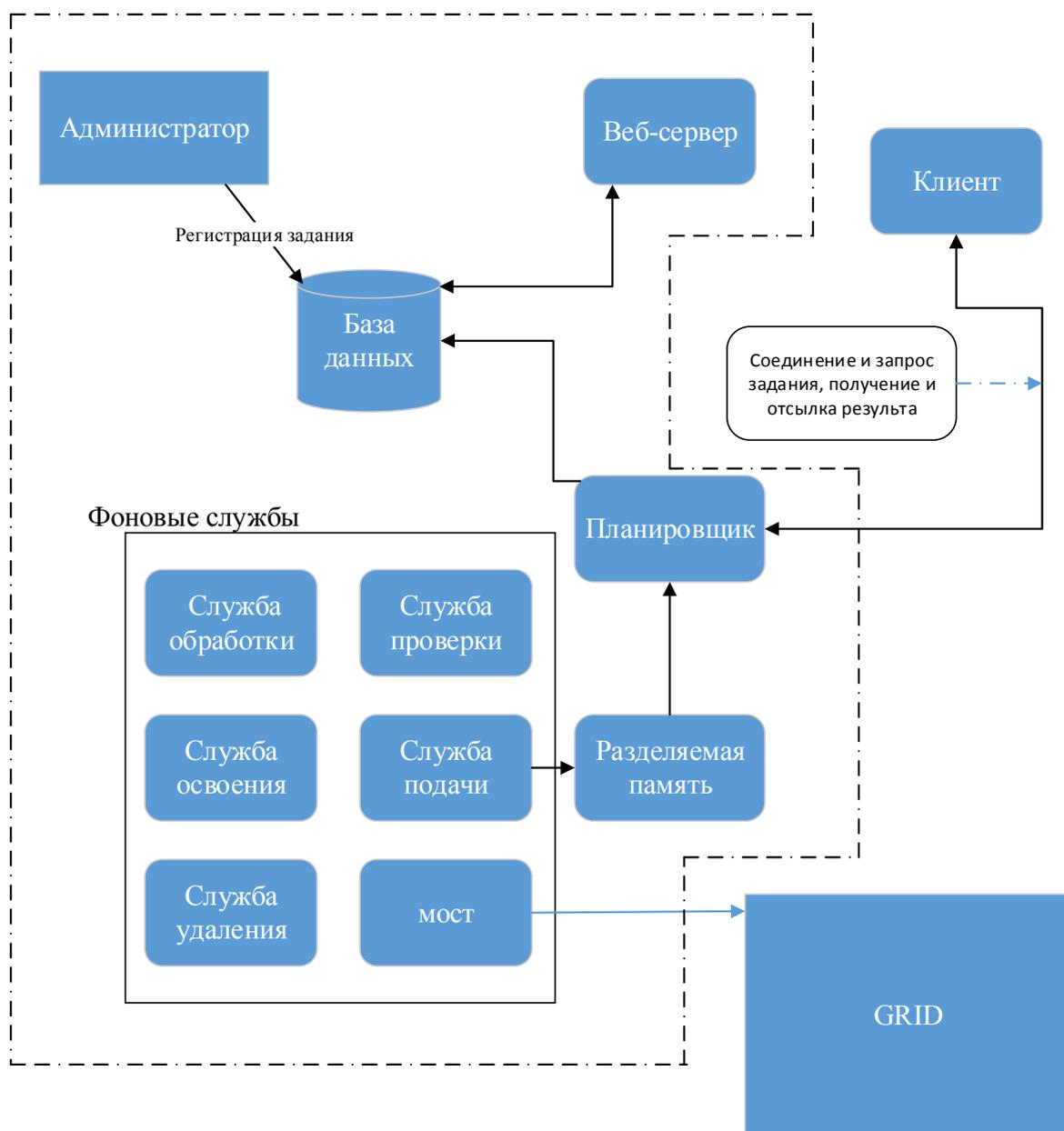


Рисунок 7 - Архитектура BOINC

Как можно видеть из рисунка выше, сервер состоит из множества модулей, которые выполняют определенные задачи. Также к серверу в случае необходимости могут быть присоединены Grid-сети, построенные с использованием инструментария Globus Toolkit, подробнее об этом поговорим, в конце данного раздела. Перейдем к детальному рассмотрению модулей архитектуры:

- Веб-сервер – предназначен в первую очередь для привлечения новых участников. Обычно отображает статистику и др. информацию. Пример: сайт проекта ClimatePrediction.Net.

- База данных – это сердце проекта. Здесь хранится информация о пользователях и связанные с этим пользователем хосты, о приложениях, результатах вычислений и др. Производительность сервера сильно зависит именно от базы данных, так как основная нагрузка по обмену информации связана с ней.

- Служба обработки состояния подзадач – фоновая служба, предназначение которой обработка состояний вычислительных подзадач и результатов их решения. Служба проверяет состояния подзадач и результатов вычислений в БД и обновляет их. Служба ресурсоемкая и может сильно нагружать процессор, поэтому ее можно разбить на некоторое кол-во процессов, каждый из которых может работать на разных серверах.

- Служба проверки – организует проверку полученных результатов, так как для достоверности подзадачи считаются на разных клиентах, то служба занимается сверкой этих результатов для получения более точного результата.

- Служба освоения – отвечает за проверку наличия завершенных задач, а дальше работает в зависимости от функций прописанных администратором. Это может быть запуск новой задачи или отправка результатов и т.д.

- Служба удаления – занимается очисткой системы от завершенных и освоенных подзадач.

- Служба подачи – вспомогательная служба, которая передает необработанные задачи в разделяемую память. Эта служба призвана разгрузить нагрузку на БД, ограничив к ней число запросов.

- Планировщик – служба, являющаяся связующим звеном в работе с клиентом. Она отвечает за назначение подзадач клиентам, так как знает информацию об их ПК и настройках. Так же принимает отчет о работе от клиентов.

- Мост – эта служба позволяет производить совместную работу проекта с инфраструктурой BOINC и стандартной GRID.

Алгоритм работы над заданиями

Алгоритм работы можно представить следующим образом:

1. Сервер создает подзадание с помощью генератора заданий необходимых для проекта.
2. BOINC при необходимости создает копию задания.
3. Планировщик назначает подзадание клиенту.
4. Клиент принимает задание и загружает необходимые данные для работы.
5. Клиент производит вычисления.
6. Клиент отправляет результаты вычислений и отчет о выполнении.
7. Служба проверки результатов проверяет полученные данные.
8. Если служба проверки одобряет решение, то в работу вступает служба освоения.
9. После запускается служба удаления для очистки системы от ненужных файлов.
10. Когда все подзадачи завершены, служба удаления очищает БД.

1.3. Программное обеспечение HTCondor

PBS с открытым исходным кодом, обладающая высокой пропускной способностью и возможностью распараллеливания ресурсоемких задач [5].

Ранее проект был известен под именем Condor. Продукт доступен бесплатно для исследователей UW-Madison и спонсорам. Обеспечивает политику планирования, мониторинг ресурсов и все то же, что и в других ранее описанных PBC. Одной из особенностью является то, что HTCondor можно настроить, чтобы агенты использовались в период своей неактивности (мышь и клавиатура в покое). Для HTCondor не обязательно использовать файловое хранилище, так как он может общаться напрямую. Так же поддерживает возможность переноса задания на другой узел, в случае его отключения.

Архитектура HTCondor

HTCondor представляет собой развернутый кластер узлов, с одним менеджером узлом, который выступает в роли менеджера ресурсов, где пользователи могут отправлять задания через планировщик заданий, а тот в свою очередь составляет очередь заданий, и вычислительные узлы работают согласно очереди заданий. Пример Архитектуры можно увидеть на рисунке 8.

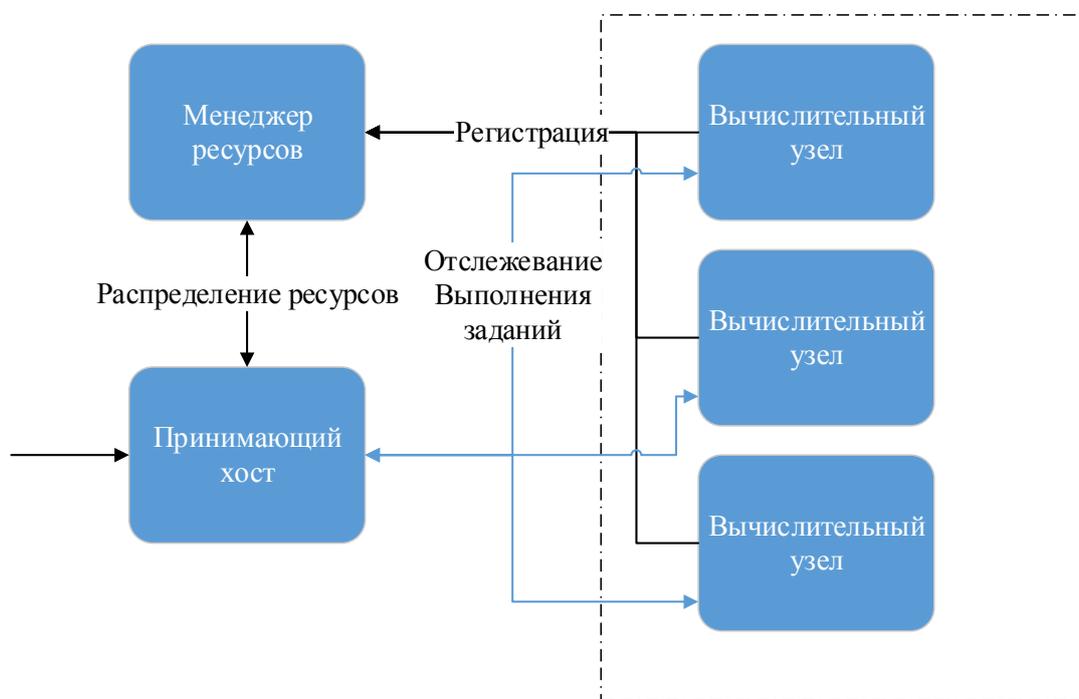


Рисунок 8 - Архитектура HTCondor

Менеджер ресурсов содержит БД вычислительных узлов наряду с их характеристиками, при появлении новых вычислительных узлов, они

обязательно регистрируются в менеджере ресурсов, и ожидают дальнейших указаний. Каждый хост запускает планировщик задач и предоставляет средства пользователю, сформированные задачи поступают в очередь заданий планировщика, и планировщик распределяет задания на основании очереди, рабочей нагрузки или приоритета. Планировщик может запросить ресурсы у менеджера ресурсов в случае необходимости. Менеджер ресурсов, получив запрос, предоставляет планировщику ресурсы, которые удовлетворяют его запросу. Получив необходимые ресурсы, планировщик передает на них задание. Планировщик в течение всего времени отслеживает выполнение задания, и если задание выполнено, то находит другую работу, чтобы машины не простаивали.

Требования планировщика могут быть нарушены только в нескольких случаях [7]:

- менеджер ресурсов, решил забрать узел и выделить его на другой проект с более высоким приоритетом пользователя или планировщика;
- планировщик сам отказывается от работы с узлом;
- узел нарушает требования работы с планировщиком, например, выключился.

Так как Менеджер ресурсов и планировщик разделены, это предоставляет более высокую степень масштабируемости и гибкости системы. Также стоит отметить, что архитектура не предполагает каких-либо идентификаторов или общей файловой системы.

1.4. Инструментарий Globus Toolkit

Globus Toolkit в отличие от Hadoop, BOINC и HTCondor не предоставляет готовую платформу для работы. Данный продукт предоставляет лишь инструментарий для разработки GRID – систем. Сам по себе Globus toolkit состоит из набора модулей, необходимого для организации распределенных вычислений. Каждый модуль имеет реализации под различные среды выполнения.

Модули внутри Globus Toolkit можно разделить на следующие группы:

- модули поиска и выделения ресурсов;
- информационные модули;
- аутентификационные модули;
- модули коммуникации;
- модули доступа к данным;
- модули создания процессов;

Рассмотрим некоторые из групп модулей:

- коммуникационный модуль – обеспечивает однородность доступа к ресурсам, например TCP/IP;
 - доступ к данным – так как данные в кластере, в большинстве случаев, распределены между некоторым количеством узлов, то для доступа в Globus используется специальный интерфейс удаленного доступа RIO.

В процессе работы необходима информация обо всех вычислительных ресурсах, сети и системы в целом. Сюда могут включаться различные параметры (пропускная способность каналов и т.п.). Часть этой информации нужна серверу, другая приложениям и т.д. Globus позволяет объединить все эти данные и предоставить к ним доступ с помощью сервиса «метакомпьютерной директории (анг. Metacomputing Directory Service (MDS))». Информация в данном сервисе хранится в виде списка пар «ключ-значение».

С помощью модулей создают высокоуровневые механизмы, примером может являться службы безопасности, которые могут обеспечивать как глобальную идентификацию, так и идентификацию, основанную на сертификатах, а так же централизованный доступ для локальных пользователей.

1.5. Платформа Unicore

Uniform Interface to Computing Resources (рус. Унифицированный интерфейс доступа к вычислительным ресурсам) – предлагает готовую к запуску систему, которая включает в себя клиент и серверное программное

обеспечение [8]. Перейдем сразу к рассмотрению архитектуры данной системы (рисунок 9) [9].

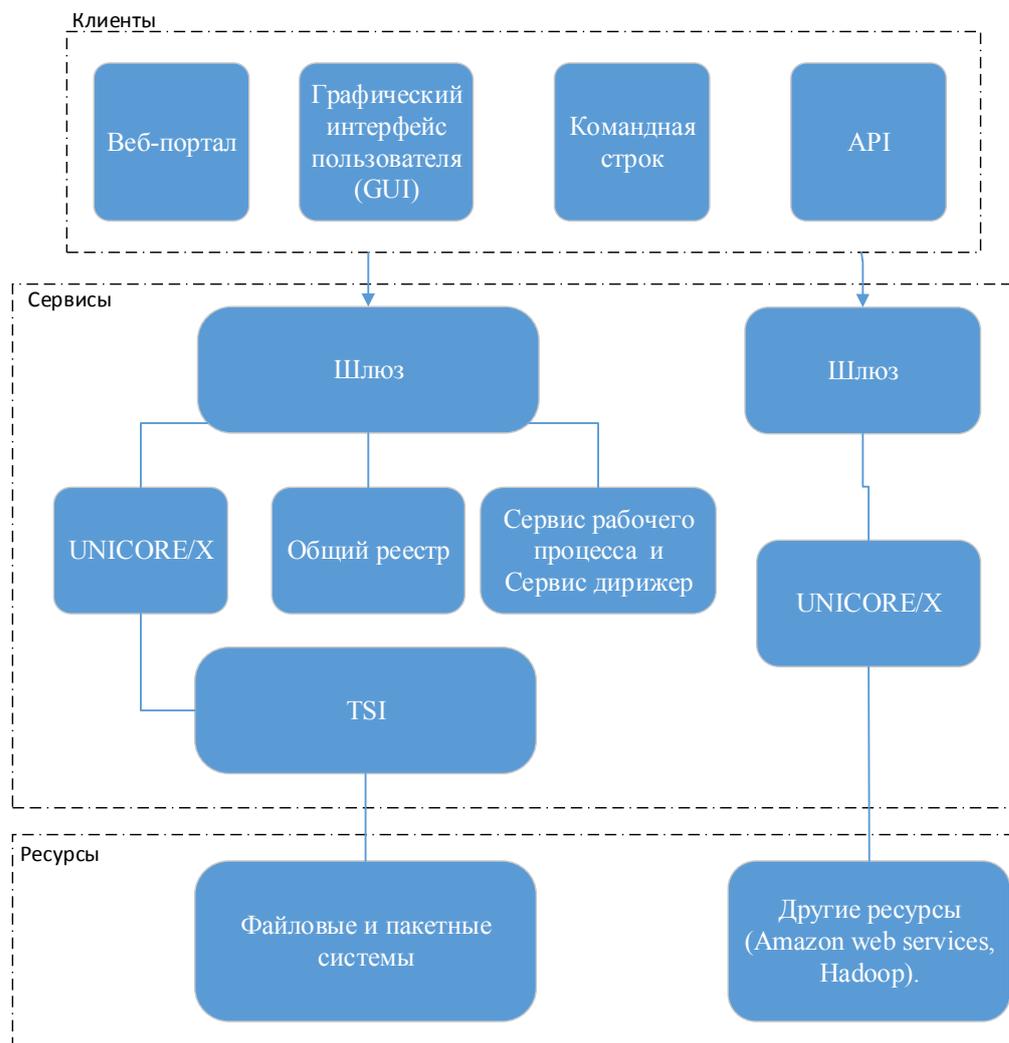


Рисунок 9 - Архитектура UNICORE

Рассмотрим работу системы более подробно.

Слой клиенты.

Unicore предоставляет ряд клиентских приложений, таких как веб-портал, графический клиент, клиент командной строки и API для построения пользовательских приложений.

Веб-портал – позволяет пользователям получать доступ и подсчитывать данные через веб-браузер, без необходимости устанавливать дополнительное ПО. Так же имеются функции передачи файлов между хранилищами, создавать загружать и редактировать файлы и др.

Графический интерфейс – это клиент Unicore Rich, является настольным десктопным приложением и обладает полным набором функциональных возможностей. Соответственно предоставляет всю информацию в окне приложения. Так же имеется возможность строить блоки, циклы с помощью мыши и др.

Клиент командной строки – это универсальный инструмент, который позволяет получить доступ ко всем функциям Unicore, однако требует от пользователя определенных компетенций. В функции клиента командной строки входят управление данными. То есть можно проводить различные манипуляции с данными.

Сервисный слой

Данный слой включает в себя все услуги и компоненты Unicore.

Шлюз – это обратный HTTPS прокси-сервер, служит в первую очередь брандмауэром, для избежания необходимости настройки открытых портов.

UNICORE/X – центральный компонент системы, который обеспечивает прием запросов от клиентов, передающихся через шлюз, проверяет подлинность запроса, проверяет авторизацию и вызывает соответствующие службы. Предоставляет набор сервисных интерфейсов, принимает задания и направляет их через системы TSI, позволяет загружать и скачивать данные с помощью различных протоколов. Таким образом, обеспечиваю большую часть функциональности UNICORE, так же данный компонент позволяет использовать другие ресурсы, такие как Apache Hadoop HDFS, Amazon S3 и др.

Общий реестр – содержит информацию о доступных сервисах, размещенных на нескольких UNICORE. Единый реестр позволяет наращивать и управлять UNICORE инфраструктурой.

Сервис рабочего процесса – поддержка рабочего процесса реализована в виде двухслойной архитектуры, состоящий из сервиса рабочего процесса и планировщика. Сервис рабочего процесса предлагает широкий спектр управляющих конструкций и функций. Данный сервис можно использовать на любом из вариантов клиентского слоя.

Сервис дирижера отвечает за выполнение отдельных задач в потоке операции, а так же занимается обработкой заданий и мониторинге.

TSI (Target System Layer)

Состоит из интерфейса к локальной файловой, операционной системам и управлением ресурсами пакетной системы. Данный интерфейс используется для отправки заданий, выполнение операций ввода и вывода, проверки состояния задач. TSI работает в фоне с правами администратора, а так же с привилегиями на ресурсы.

Заключение

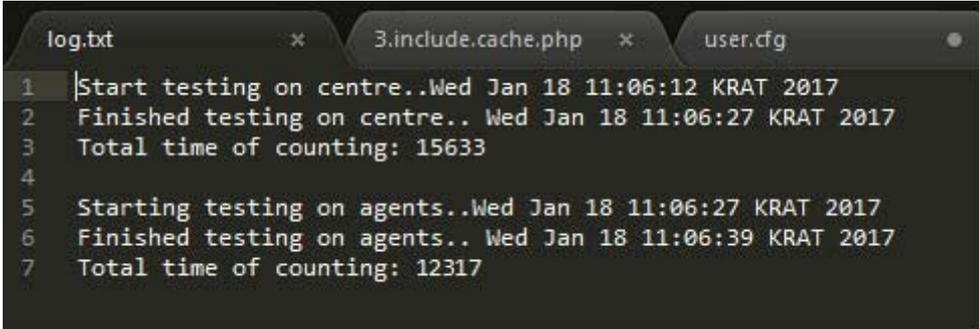
Рассмотрев платформы и инструментарий по созданию РВС, можно заметить, что они давно на рынке и хорошо проработаны. Но не всегда требуется настолько сложные платформы для работы, поэтому для этого была разработана архитектура, которую можно было максимально быстро развернуть в любом месте для работы.

2. Исследование эффективности информационно-вычислительного кластера на базе персональных компьютеров

В процессе разработки распределенного информационно-вычислительного кластера проводились различные эксперименты с целью установить насколько эффективно распределенные вычисления в сравнении с вычислениями на одном персональном компьютере. Для нагрузки и сравнения использовались различные операции по созданию, удалению массивов, словарей, очередей и др. При эксперименте центр выполнял весь объем тестов, а агенты части от него. Сравнение затраченного времени показано на таблице 1 и рисунке.

Таблица 1 – Сравнение затраченного времени

Начало теста		Конец теста		Затраченное время (сек.)	
Центр	Агенты	Центр	Агенты	Центр	Агенты
11:06:12	11:06:27	11:06:27	11:06:39	15,633	12,317



```
log.txt x 3.include.cache.php x user.cfg
1 Start testing on centre..Wed Jan 18 11:06:12 KRAT 2017
2 Finished testing on centre.. Wed Jan 18 11:06:27 KRAT 2017
3 Total time of counting: 15633
4
5 Starting testing on agents..Wed Jan 18 11:06:27 KRAT 2017
6 Finished testing on agents.. Wed Jan 18 11:06:39 KRAT 2017
7 Total time of counting: 12317
```

Рисунок 10- Log-файл с информацией о затраченном времени

Как можно заметить из таблицы, центр затратил на 4 теста 15,633 секунды, в отличие от агентов, которые потратили на это 12 317 секунд, что на 3,316 секунд быстрее, и это с учетом задержки времени на передачу команды и получения ответа.

Проведем еще несколько тестов с участием одного, двух и трех агентов. На рисунке 11 отображена загрузка ЦП во время ожидания подключения агентов.

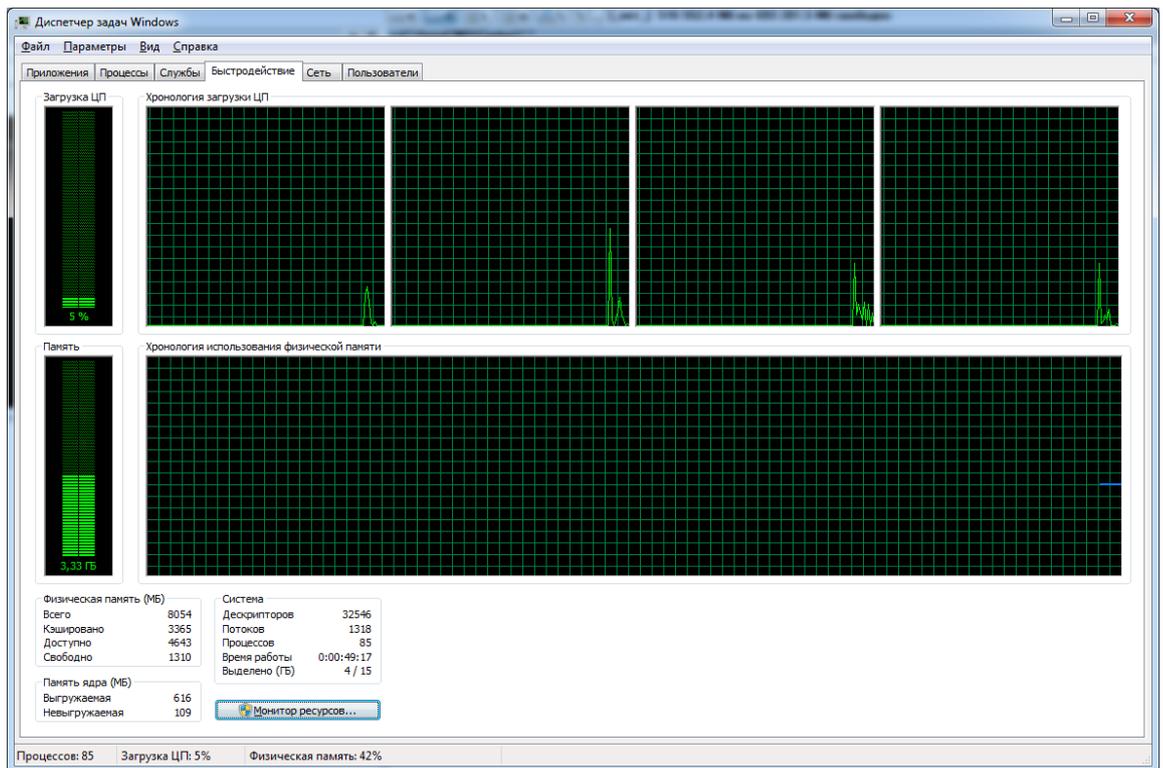


Рисунок 11 - загрузка ЦП во время ожидания подключения агентов
Подключаем одного агента и проводим тест из 5 итерации (рисунок 12).

```
test
Count of test:
5
Start testing on centre..Fri Jun 09 10:39:45 GMT+07:00 2017
Finished testing on centre.. Fri Jun 09 10:40:00 GMT+07:00 2017
Total time of counting: 15046
Starting testing on agents..Fri Jun 09 10:40:00 GMT+07:00 2017
From agent /109.123.146.10:30724 got I'm alive
From agent /109.123.146.10:30724 got 13094 milliseconds
Finished testing on agents.. Fri Jun 09 10:53:00 GMT+07:00 2017
```

Рисунок 12 - тестирование с одним агентом

Как видно, центр затратил на тест 15,046 секунд, тогда как агент справился за 13,094, что на 1,952 секунды быстрее. Это обусловлено отсутствием нагрузки у агента.

Подключаем двух агентов (рисунок 13) и проведем тест из 4 итерации (рисунок 14).

```
Waiting for a client connection... IP:/109.123.146.12:64036
Got a client!
From agent /109.123.146.10:30566 got 4
Got a client!
From agent /109.123.146.11:64037 got 4
```

Рисунок 13 - подключение агентов

```

test
Count of test:
4
Start testing on centre..Fri Jun 09 10:00:35 GMT+07:00 2017
Finished testing on centre.. Fri Jun 09 10:00:49 GMT+07:00 2017
Total time of counting: 14752
Starting testing on agents..Fri Jun 09 10:00:49 GMT+07:00 2017
From agent /109.123.146.10:30566 got I'm alive
From agent /109.123.146.11:64037 got I'm alive
From agent /109.123.146.11:64037 got 6229 milliseconds
From agent /109.123.146.10:30566 got 6850 milliseconds
Finished testing on agents.. Fri Jun 09 10:01:02 GMT+07:00 2017

```

Рисунок 14 – тестирование с двумя агентами

Сначала задачу выполняет центр. На это у него уходит 14,752 секунды.

Далее каждый из подключенных агентов выполняет ровно половину от общего числа тестов, то есть каждый агент выполняет по 2 цикла. Как видно, агенты справились за 13,079 секунды, что на 1,673 секунды быстрее.

Проведем тест на трех агентах. Для этого подключаем трех агентов (рисунок 15)

```

Waiting for a client connection... IP:/109.123.146.12:64044
Got a client!
From agent /109.123.146.10:30686 got 4
Got a client!
From agent /109.123.146.11:64047 got 4
Got a client!
From agent /109.123.146.14:64048 got 4

```

Рисунок 15 - подключение трех агентов

Запустим тест и выберем 3 итерации. По итерации на каждого агента.

Загрузка Центра при тесте отображена на рисунке 16.

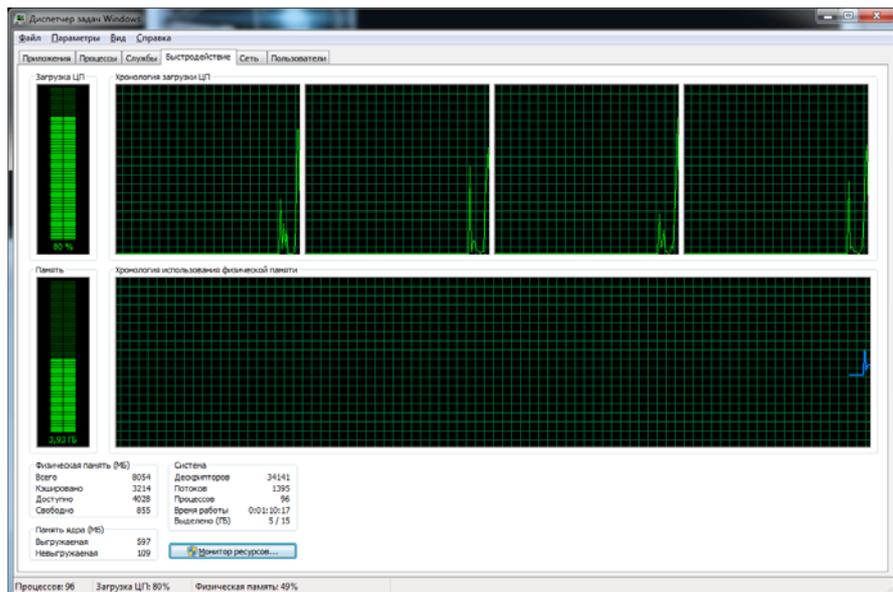


Рисунок 16 - загрузка ЦП центра во время теста

Результаты теста представлены на рисунке 17.

```

test
Count of test:
3
Start testing on centre..Fri Jun 09 10:12:32 GMT+07:00 2017
Finished testing on centre.. Fri Jun 09 10:12:44 GMT+07:00 2017
Total time of counting: 12182
Starting testing on agents..Fri Jun 09 10:12:44 GMT+07:00 2017
From agent /109.123.146.11:64047 got I'm alive
From agent /109.123.146.14:64048 got I'm alive
From agent /109.123.146.10:30686 got I'm alive
From agent /109.123.146.11:64047 got 3704 milliseconds
From agent /109.123.146.10:30686 got 3801 milliseconds
From agent /109.123.146.14:64048 got 3754 milliseconds
Finished testing on agents.. Fri Jun 09 10:12:55 GMT+07:00 2017

```

Рисунок 17 - тестирование с тремя агентами

Как можно заметить центр потратил 12,182 секунды на тест, а агенты справились за 11,259 секунд, что на 923 секунды быстрее. В данном тесте уже становится заметно, что агенты выполняют тест быстрее, но незначительно, вследствие того, что центр выполняет тесты подряд, а у агентов тратится время на передачу команд.

Сводим данные тестов в таблицу 2.

Таблица 2- Данные тестирования

	Центр	Агент	Центр	2 Агента	Центр	3 Агента
Начало теста	10:39:45	10:40:00	10:00:35	10:00:49	10:12:32	10:12:44
Конец теста	10:40:00	10:53:00	10:00:49	10:01:02	10:12:44	10:12:55
Затраченное время (сек)	15,046	13,094	14,752	13,079	12,182	11,259

Как можно видеть из таблицы, эффективность информационно-вычислительного кластера зависит от задач, которые на нем выполняются. Из таблицы видно, что количество времени, затрачиваемое на тест, увеличивается при увеличении количества агентов. В связи с этим, можно сделать вывод, что агенты будут выполнять задачи быстрее только в том случае, когда затраты на передачу команды несоизмеримо меньше, чем выполнение команды. Таким образом, можно сделать вывод, что информационно-вычислительный кластер на базе персональных компьютеров эффективен только для задач, не требующих работы в режиме реального времени.

Заключение

В данной главе были проведены программные эксперименты, численно иллюстрирующие эффективность распределенных вычислений по сравнению с вычислениями на одном персональном компьютере.

5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

Целью данного раздела является проведение анализа финансово-экономических аспектов научно-исследовательской работы. Необходимо оценить трудовые и денежные затраты направленные на реализацию данного исследования, а так же уровня их научно-технической результативности.

5.1. Организация и планирование работ

При организации процесса реализации конкретного проекта необходимо оптимально планировать занятость каждого из его участников и сроки проведения отдельных работ. На данном этапе составляется полный перечень проводимых работ, и определяются их исполнители и оптимальная продолжительность. Результатом планирования работ является сетевой, либо реализации проекта. Наиболее удобным, простым и наглядным способом для этих целей является использование линейного графика. Для его построения составим перечень работ и соответствие работ своим исполнителям, продолжительность выполнения этих работ и сведем их в таблицу 10.

Таблица 10 - Перечень работ и продолжительность их выполнения

Этапы работы	Исполнители	Загрузка исполнителей
Постановка целей и задач, получение исходных данных	НР, И	НР – 45% И – 55%
Подбор и изучение материалов по тематике	НР, И	НР – 20% И – 100%
Разработка календарного плана	НР, И	НР – 50% И – 50%
Обсуждение литературы	НР, И	НР – 10% И – 100%
Выбор алгоритмов для решения поставленной задачи	НР, И	НР – 25% И – 100%
Планирование архитектуры программного обеспечения	НР, И	НР – 20% И – 100%
Разработка программного обеспечения	И	И – 100%
Тестирование и отладка программного обеспечения	И	И – 100%
Оформление расчетно-пояснительной записки	И	И – 100%
Подведение итогов	НР, И	НР – 80% И – 100%

НР – научный руководитель.

И – исполнитель

5.1.1. Продолжительность этапов работ

Расчет продолжительности этапов работ осуществляется двумя методами:

- технико-экономическим;
- опытно-статистическим.

В данном случае используется опытно-статистический метод, который реализуется двумя способами:

- аналоговый;
- экспертный (вероятностный).

Для определения ожидаемого значения продолжительности работ $t_{ож}$ применяется вероятностный метод – метод двух оценок t_{min} и t_{max} .

$$t_{ож} = \frac{3 \cdot t_{min} + 2 \cdot t_{max}}{5}, \quad (1)$$

где t_{min} – минимальная трудоемкость работ, чел/дн.;

t_{max} – максимальная трудоемкость работ, чел/дн.

Для выполнения перечисленных в таблице 10 работ требуются специалисты:

- инженер – в его роли действует исполнитель НИР (ВКР);
- научный руководитель.

Для построения линейного графика необходимо рассчитать длительность этапов в рабочих днях, а затем перевести в календарные дни.

Расчет продолжительности выполнения каждого этапа ведется по формуле:

$$T_{РД} = \frac{t_{ож}}{K_{ВН}} \cdot K_{Д}, \quad (2)$$

где $t_{ож}$ – трудоемкость работы, чел/дн.;

$K_{ВН}$ – коэффициент выполнения работ ($K_{ВН} = 1$);

$K_{Д}$ – коэффициент, учитывающий дополнительное время на компенсации и согласование работ ($K_{Д} = 1.2$).

Расчет продолжительности этапа в календарных днях ведется по формуле:

$$T_{КД} = T_{РД} \cdot T_{К}, \quad (3)$$

где $T_{РД}$ – продолжительность выполнения этапа в рабочих днях;

$T_{КД}$ – продолжительность выполнения этапа в календарных днях;

$T_{К}$ – коэффициент календарности.

Коэффициент календарности рассчитывается по формуле:

$$T_{К} = \frac{T_{КАЛ}}{T_{КАЛ} - T_{ВД} - T_{ПД}}, \quad (4)$$

где $T_{КАЛ}$ – календарные дни ($T_{КАЛ} = 365$);

$T_{ВД}$ – выходные дни ($T_{ВД} = 52$);

$T_{ПД}$ – праздничные дни ($T_{ПД} = 10$).

$$T_{К} = \frac{365}{365 - 52 - 10} = 1,205.$$

Таблица 11 - Трудозатраты на выполнение проекта

Этап	Исполнители	Продолжительность работ, дни			Длительность работ, чел/дн.			
					$T_{РД}$		$T_{КД}$	
		t_{min}	t_{max}	$t_{ож}$	НР	И	НР	И
Постановка целей и задач, получение данных	НР, И	1	3	1,8	0,972	1,188	1,17126	1,43154
Подбор и изучение материалов по тематике	НР, И	10	15	12	2,88	11,52	3,4704	13,8816
Разработка календарного плана	НР, И	2	4	2,8	1,68	1,68	0,2044	0,2044
Обсуждение литературы	НР, И	2	5	3,2	0,384	3,84	0,46272	4,6272
Выбор алгоритмов для решения поставленных задач	НР, И	10	17	12,8	3,84	15,36	4,6272	18,5088
Планирование архитектуры программного обеспечения	НР, И	15	22	17,8	4,272	21,36	5,14776	25,7388
Разработка программного обеспечения	И	30	50	38	0	45,6	0	54,948
Тестирование и отладка программного обеспечения	И	20	25	22	0	26,4	0	31,812
Оформление расчетно-пояснительной записки	И	5	7	5,8	0	6,96	0	8,3868
Подведение итогов	НР, И	2	4	2,8	2,688	3,36	3,23904	4,0488
Итого:				119	16,716	137,268	18,32278	163,58794

5.1.2. Расчет нарастания технической готовности работ

Величина нарастания технической готовности работы показывает процент выполнения работы на каждом этапе. Расчетная формула:

$$H_i = \frac{t_{Hi}}{t_0} \cdot 100\%, \quad (5)$$

где t_{Hi} – нарастающая трудоемкость с момента начала работы i -го этапа;

t_0 – общая трудоемкость.

Общая трудоемкость вычисляется по формуле:

$$t_0 = \sum_{i=1}^n t_{OЖi}, \quad (6)$$

где $t_{OЖi}$ – ожидаемая продолжительность i -го этапа.

Удельный вес каждого этапа Y_i определяется по формуле:

$$Y_i = \frac{t_{OЖi}}{t_0} \cdot 100\% \quad (7)$$

Результаты вычислений H_i и Y_i отражены в таблице 12.

Таблица 12 - Нарастание технической готовности и удельный вес этапов

Этап	H_i , %	Y_i , %
Постановка целей и задач, получение исходных данных	1,51	1,51
Подбор и изучение материалов по тематике	11,59	10,08
Разработка календарного плана	13,94	2,35
Обсуждение литературы	16,63	2,68
Выбор алгоритмов для решения поставленных задач	27,39	10,75
Планирование архитектуры программного обеспечения	42,35	14,95
Разработка программного обеспечения	74,28	31,93
Тестирование и отладка программного обеспечения	92,77	18,48
Оформление расчетно-пояснительной записки	97,647	4,87
Подведение итогов	100	2,35

Таблица 13 - Линейный график работ

Этап	Т _{кд}		январь			февраль			март			апрель			май		
	НР	И	10	20	30	10	20	30	10	20	30	10	20	30	10	20	30
	1	1,171	1,431														
2	3,470	13,881															
3	0,204	0,204															
4	0,462	4,627															
5	4,627	18,508															
6	5,147	25,738															
7	0	54,948															
8	0	31,812															
9	0	8,386															
10	3,239	4,0488															

■ НР ■ И

5.2. Расчет сметы затрат на разработку ПО и проведение эксперимента

В состав затрат на разработку ПО и проведение эксперимента включается стоимость всех расходов, необходимых для реализации комплекса работ, составляющих содержание данной разработки. Расчет сметной стоимости на выполнение данной разработки производится по следующим статьям затрат:

- материалы и покупные изделия;
- основная заработная плата;
- отчисления в социальные фонды;
- расходы на электроэнергию;
- амортизационные отчисления;
- работы, выполняемые сторонними организациями;
- прочие расходы.

5.2.1. Расчет затрат на материалы

К данной статье расходов относится стоимость материалов, покупных изделий, полуфабрикатов и других материальных ценностей, расходуемых непосредственно в процессе выполнения работ. Так как основным рабочим инструментом при разработке данного проекта выступал персональный компьютер, затраты на материалы и оборудование составляют 0 рублей в связи с отсутствием необходимости в расходных материалах.

5.2.2. Расчет основной заработной платы

Данная статья расходов включает заработную плату научного руководителя и инженера, а также премии, входящие в фонд заработной платы. Расчет основной заработной платы выполняется на основе трудоемкости выполнения каждого этапа и величины месячного оклада исполнителя.

Среднедневная заработная плата рассчитывается по формуле:

$$\text{Дневная з/плата} = \frac{\text{Месячный оклад}}{25,17 \text{ дней}} \quad (8)$$

Расчеты затрат на основную заработную плату приведены в таблице 6. При расчете учитывалось, что в году 302 рабочих дня и, следовательно, в месяце 25,17 рабочих дня. Затраты времени на выполнение работы по каждому исполнителю брались из таблицы 11. Также был принят во внимание коэффициент, учитывающий коэффициент по премиям $K_{\text{доп.ЗП}} = 1,188$, $K_{\text{ПР}} = 1,1$ и районный коэффициент $K_{\text{РК}} = 1,3$ ($K = 1 * K_{\text{ПР}} * K_{\text{РК}} * K_{\text{РК}} = 1 * 1,1 * 1,3 * 1,188 = 1,699$).

Таблица 14 - Затраты на основную заработную плату

Исполнитель	Оклад, руб./мес.	Среднедневная ставка, руб./день	Затраты времени, дни	Коэффициент	Фонд з/платы, руб.
НР	23 264,86	924,30	16,72	1,699	26256,8
И	7864,11	312,43	137,27	1,62	69477,4
Итого:					95734,2

5.2.3. Расчет отчислений от заработной платы

Затраты по этой статье составляют отчисления в социальные фонды.

Отчисления по заработной плате определяются по следующей формуле:

$$C_{\text{соц}} = K_{\text{соц}} * C_{\text{зп}}, \quad (9)$$

где $K_{\text{соц}}$ – коэффициент, учитывающий размер отчислений из заработной платы. Данный коэффициент составляет 30,2% от затрат на заработную плату и включает в себя:

- отчисления в пенсионный фонд;
- на социальное страхование;
- на медицинское страхование.

Итак, отчисления из заработной платы составили:

$$C_{\text{соц}} = 0,302 * 95734,2 = \mathbf{28911,7 \text{ руб.}}$$

5.2.4. Расчет затрат на электроэнергию

Данный вид расходов включает в себя затраты на электроэнергию при работе оборудования. Затраты на электроэнергию при работе оборудования для технологических целей рассчитываются по формуле:

$$\text{Э}_{\text{ОБ}} = P_{\text{ОБ}} \cdot \text{Ц}_{\text{Э}} \cdot t_{\text{ОБ}}, \quad (10)$$

где $\text{Э}_{\text{ОБ}}$ – затраты на электроэнергию, потребляемую оборудованием, руб.;

$P_{\text{ОБ}}$ – мощность, потребляемая оборудованием, кВт;

$\text{Ц}_{\text{Э}}$ – тарифная цена за 1 кВт·час, $\text{Ц}_{\text{Э}} = 5,782$ руб/кВт·час;

$t_{\text{ОБ}}$ – время работы оборудования, час.

Время работы оборудования вычисляется на основе данных для $T_{\text{РД}}$ таблицы 11 для инженера из расчета, что продолжительность рабочего дня равна 8 часов.

Мощность, потребляемая оборудованием, определяется по формуле:

$$P_{\text{ОБ}} = P_{\text{УСТ.ОБ}} \cdot K_{\text{С}}, \quad (11)$$

где $P_{\text{УСТ.ОБ}}$ – установленная мощность оборудования, кВт;

$K_{\text{С}}$ – коэффициент загрузки, зависящий от средней степени использования номинальной мощности.

Для технологического оборудования малой мощности $K_{\text{С}} = 1$.

Затраты на электроэнергию для технологических целей приведены в таблице 15.

Таблица 15 - Затраты на электроэнергию для технологических целей

Наименование оборудования	Время работы оборудования $t_{\text{ОБ}}$, час	Потребляемая мощность $P_{\text{ОБ}}$, кВт	Затраты $\text{Э}_{\text{ОБ}}$, руб.
Персональный компьютер	975,4	0,3	1691,9
Итого:			1691,9

5.2.5. Расчет амортизационных расходов

В статье «Амортизационные отчисления» от используемого оборудования рассчитывается амортизация за время выполнения работы для оборудования, которое имеется в наличии.

Амортизационные отчисления рассчитываются на время использования ЭВМ по формуле:

$$C_{AM} = \frac{N_A \cdot C_{OB}}{F_D} \cdot t_{BT} \cdot n, \quad (12)$$

где N_A – годовая норма амортизации, $N_A = 33\%$;

C_{OB} – цена оборудования, $C_{OB} = 15000$ руб.;

F_D – действительный годовой фонд рабочего времени, $F_D = 2384$ часов;

t_{BT} – время работы вычислительной техники при создании программного продукта, $t_{BT} = 975,4$ часа;

n – число задействованных ПЭВМ, $n = 1$.

Итак, затраты на амортизационные отчисления составили:

$$C_{ам} = 2025,3 \text{руб}$$

5.2.6. Расчет прочих расходов

В статье «Прочие расходы» отражены расходы на разработку проекта, которые не учтены в предыдущих статьях.

Прочие расходы составляют 10 % от единовременных затрат на выполнение технического продукта и проводятся по формуле:

$$C_{проч} = (C_{зп} + C_{соц} + C_{эл.об.} + C_{ам}) * 0,1 \quad (13)$$

$$C_{проч} = (95734,2 + 28911,7 + 1691,9 + 2025,3) * 0,1 = 12\ 836,91$$

5.2.7. Расчет общей себестоимости разработки

Проведя расчет сметы затрат на разработку, можно определить общую стоимость разработки проекта «Проектирование инфраструктуры и реализация распределенного информационно-вычислительного кластера на базе персональных».

Таблица 16 - Смета затрат на разработку проекта

Статья затрат	Условное обозначение	Сумма, руб.
1 Материалы и покупные изделия	C_{MAT}	0
2 Основная заработная плата	$C_{ЗП}$	95734,2
3 Отчисления в социальные фонды	$C_{СОЦ}$	28911,7
4 Расходы на электроэнергию	Э	1691,9

5 Амортизационные отчисления	C_{AM}	2025,3
6 Работы, выполняемые сторонними организациями	$C_{СТОР}$	0
7 Прочие расходы	$C_{ПРОЧ}$	12 836,91
Итого:		141 200,01

Таким образом, расходы на разработку составили $C = 141\ 200,01$ руб.

5.2.8. Прибыль

Прибыль составляет **28 240** рублей (20 %) от расходов на разработку.

5.2.9. НДС

НДС составляет 18% от суммы затрат на разработку. Сумма НДС составила **25 416** руб.

5.2.10. Цена разработки НИР

Цена равна сумме полной себестоимости, прибыли и НДС:

$$C_{НИР(КР)} = 141\ 200,01 + 28\ 240 + 25\ 416 = 194\ 856,01 \text{ руб.}$$

5.3. Оценка экономической эффективности проекта

Экономическая эффективность проекта обусловлена необходимостью ускорить расчеты вычислительно-емких задач, не требующих обработки в режиме реального времени. Конечная система позволит на основе имеющихся персональных компьютерах, развернуть распределенный информационно-вычислительный кластер, что позволит без особых затрат получить необходимую вычислительную мощность, не арендуя при этом суперкомпьютер. Использование подобной системы в научной сфере, позволит повысить эффективность исследований, так как пользователю будет требоваться меньше времени на ожидание результата.

Подводя итог вышесказанному, отметим, что экономический эффект от реализации проекта, может быть выражен в отказе от затрат на аренду или покупку мощных суперкомпьютеров, повышении оперативности вычислений.

Проведение количественных расчетов показателей эффективности в рамках данной работы невозможно, так как отсутствует необходимая для этого информация.

5.4. Оценка научно-технического уровня НИР

Научно-технический уровень характеризует, в какой мере выполнены работы и обеспечивается научно-технический прогресс в данной области. Для оценки научной ценности, технической значимости и эффективности, планируемых и выполняемых НИР, используется метод бальных оценок. Бальная оценка заключается в том, что каждому фактору по принятой шкале присваивается определенное количество баллов. Обобщенную оценку проводят по сумме баллов по всем показателям или рассчитывают по формуле. На этой основе делается вывод о целесообразности НИР.

Сущность метода заключается в том, что на основе оценок признаков работы определяется коэффициент ее научно-технического уровня по формуле:

$$K_{НТУ} = \sum_{i=1}^3 R_i \cdot n_i, \quad (14)$$

где $K_{НТУ}$ – коэффициент научно-технического уровня;

R_i – весовой коэффициент i -го признака научно-технического эффекта;

n_i – количественная оценка i -го признака научно-технического эффекта, в баллах.

Таблица 17 - Весовые коэффициенты признаков НТУ

Признак НТУ	Примерное значение весового коэф-та n_i	
1 Уровень новизны	Систематизируются и обобщаются сведения, определяются пути дальнейших исследований	0,4
2 Теоретический уровень	Разработка способа (алгоритм, программа мероприятий, устройство, вещество и т.п.)	0,1
3 Возможность реализации	Время реализации в течение первых лет	0,5

Таблица 18 - Баллы для оценки уровня новизны

Уровень новизны	Характеристика уровня новизны	Баллы
Принципиально новая	Новое направление в науке и технике, новые факты и закономерности, новая теория, вещество, способ	8 – 10
Новая	По-новому объясняются те же факты, закономерности, новые понятия дополняют ранее полученные результаты	5 – 7
Относительно новая	Систематизируются, обобщаются имеющиеся сведения, новые связи между известными факторами	2 – 4
Не обладает новизной	Результат, который ранее был известен	0

Таблица 19 - Баллы значимости теоретических уровней

Теоретический уровень полученных результатов	Баллы
1 Установка закона, разработка новой теории	10
2 Глубокая разработка проблемы, многоспектральный анализ, взаимодействия между факторами с наличием объяснений	8
3 Разработка способа (алгоритм, программа и т. д.)	6
4 Элементарный анализ связей между фактами (наличие гипотезы, объяснения версии, практических рекомендаций)	2
5 Описание отдельных элементарных факторов, изложение наблюдений, опыта, результатов измерений	0,5

Таблица 20 - Возможность реализации научных, теоретических результатов по времени и масштабам

Время реализации	Баллы
В течение первых лет	10
От 5 до 10 лет	4
Свыше 10 лет	2

Результаты оценок признаков научно-технического уровня приведены в таблице 21.

Таблица 21 - Количественная оценка признаков НИОКР

Признак научно-технического эффекта НИР	Характеристика признака НИОКР	R_i
Уровень новизны	Систематизируются, обобщаются имеющиеся сведения, новые связи между известными факторами	00,4
Теоретический уровень	Разработка способа (алгоритм, программа мероприятий, устройство, вещество и т.п.)	00,1
Возможность реализации	Время реализации 5-10 лет	00,5

Обоснование оценки признаков НИОКР приводится в таблице 22.

Таблица 22 - Оценки научно-технического уровня НИР

Фактор НТУ	Значимость	Уровень фактора	Выбранный балл	Обоснование выбранного балла
Уровень новизны	0,4	Относительно новая	4	Ускорит эффективность работы в научной сфере.
Теоретический уровень	0,1	Разработка способа	6	Спроектирована и реализована инфраструктура распределенного информационно-вычислительного кластера на базе персональных компьютеров
Возможность реализации	0,5	В течение первых лет	7	Полученный продукт находится на стадии разработки и тестирования

Исходя из оценки признаков НИОКР, показатель научно-технического уровня для данного проекта составил:

$$K_{\text{НТУ}} = 0,4*4 + 0,1*6 + 0,5*7 = 1,6 + 0,6 + 3,5 = 5,7$$

Таким образом, исходя из данных в таблице 23, проект «Проектирование инфраструктуры и реализация распределенного информационно-вычислительного кластера на базе персональных компьютеров» имеет средний уровень научно-технического эффекта.

Таблица 23 - Оценка уровня научно-технического эффекта

Уровень НТЭ	Показатель НТЭ
Низкий	1-4
Средний	4-7
Высокий	8-10