



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(национальный исследовательский университет)»

Институт №8 «Информационные технологии и прикладная математика» Кафедра 810Б
Направление подготовки 02.04.02 ФИИТ Группа М80-210М-19
Квалификация (степень) магистр

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА МАГИСТРА (МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ)

На тему: Программный и пользовательский интерфейсы для сервиса квантово-механического расчета с помощью пакета Quantum Espresso

Автор квалификационной работы Стафцев Егор Алексеевич (_____)
(Фамилия, имя, отчество)

Научный руководитель Абгарян Каринэ Карленовна (_____)
(Фамилия, имя, отчество)

Рецензент Харченко Вячеслав Александрович (_____)
(Фамилия, имя, отчество)

К защите допустить

Зав. кафедрой Абгарян Каринэ Карленовна (_____)
(Фамилия, имя, отчество)

“24” мая 2021г.

Москва 2021

РЕФЕРАТ

Магистерская диссертация содержит 31 страниц, 13 рисунков. Список использованных источников содержит 7 позиций.

ПЕРВОПРИНЦИПНЫЕ МЕТОДЫ, ПАКЕТЫ КВАНТОВО-МЕХАНИЧЕСКОГО РАСЧЕТА, МИКРОСЕРВИСЫ, ДОКУМЕНТНЫЕ БАЗЫ ДАННЫХ, ПРОГРАММНЫЕ ИНТЕРФЕЙСЫ, ПОЛЬЗОВАТЕЛЬСКИЕ ИНТЕРФЕЙСЫ

Магистерская диссертация посвящена анализу существующих пакетов для квантово-механических расчетов, графических интерфейсов, и этапов разработки программных комплексов, а также созданию программного и пользовательского интерфейсов.

Для создания сервиса были рассмотрены особенности разработки Rest API интерфейсов с помощью языка программирования Kotlin в паре с библиотекой для создания клиентских и серверных приложений, взаимодействие интерфейсов с документоориентированной базой данных MongoDB, управление состоянием клиентского приложения благодаря языку TypeScript и двух библиотек React и Redux.

Реализован программный интерфейс, который является контролером модуля вычислительного сервиса и позволяет введенные данные пользователя через web-интерфейс передать на вычислительный кластер с программным пакетом квантово-механического расчета Quantum Espresso, и получить данные квантового расчета с последующим сохранением их в базе данных для дальнейшего использования.

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	3
ВВЕДЕНИЕ	4
ОСНОВНАЯ ЧАСТЬ.....	7
1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ	8
1.1. Первопринципные методы квантовой химии	8
1.1.1. Программные пакеты	9
1.1.2. Quantum Espresso	10
1.1.3. Существующие графические интерфейсы.....	12
1.1.4. Пример выполнения расчета	12
1.2. Разработка сервиса квантово-механического расчета	15
1.2.1. Требования к реализуемому программному сервису	15
1.2.2. Программный интерфейс.....	17
1.2.3. Пользовательский интерфейс.....	17
1.2.4. Хранение данных.....	19
2. ПРАКТИЧЕСКАЯ ЧАСТЬ.....	22
2.1. Архитектура платформы и предполагаемого сервиса	22
2.2. Серверное приложение	24
2.2.1. Бизнес логика.....	25
2.2.2. Способ хранения данных	26
2.2.3. Взаимодействие с модулем Quantum Espresso	26
2.2.4. Программный интерфейс	27
2.3. Клиентское приложение.....	28
2.4. Взаимодействие серверного и клиентского приложений.....	28
ЗАКЛЮЧЕНИЕ	30
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	31

ВВЕДЕНИЕ

В наше время существует огромное количество научных публикаций, разработок и открытий, касающихся области квантовой физики. Эта та часть теоретической физики, которая изучает квантово-механические и квантово-полевые системы, а также законы их движения. Несомненно, квантовая физика является самым сложным разделом для понимания, но несмотря на это, она открывает огромные возможности, большинство из которых для человеческого сознания пока еще не постижимы.

Одной из этих возможностей является создание квантового компьютера. Над ним уже давно трудятся такие компании как IBM, Microsoft и Google. Особенностью такого вычислителя является его технология, основанная на использовании кубитов. Кубиты представляют из себя подобие обычных битов, только в отличие от вторых имеют способность находиться в двух состояниях одновременно, так называемая “суперпозиция”. Именно поэтому квантовому компьютеру нет необходимости перебирать все существующие варианты состояний системы, как это делают современные компьютеры. Решение сложных задач ему дается за считанные секунды, что дает право называться “супервычислитель”.

Независимо от того, какая решается задача, связанная с квантовой физикой, от человека-специалиста требуется обладать уверенными знаниями в этой области и умениями проводить определенные квантово-механические расчеты. К счастью для этих людей существует уже большое количество программных пакетов, которые производят автоматическое вычисление, от пользователя требуется только задавать параметры системы, такие как: размер решетки, типы атомов, значения псевдопотенциалов и др. Но как бы не облегчал работу пакет для квантово-механических расчетов, он вынуждает специалиста глубоко погружаться в работу самой программы. Чтобы не терять на это время, возникла идея создать сервис для квантово-механического расчета с удобным пользовательским и программным интерфейсом.

Специалист-пользователь получил бы возможность без какой-либо сложности запустить квантово-механический расчет на одном из двух наиболее мощных пакетов по квантовой механике и при этом ему не требовалось бы изучать методику установки программы на свое вычислительное устройство и не пришлось бы погружаться с головой в особенности работы программного пакета. Единственное, что требуется от специалиста - это поставить корректно задачу, задать входные параметры, получить нужный ему ответ. Такой стиль работы позволит экономить время на изучении работы программного пакета и тратить его на генерацию новых идей и гипотез.

Данный сервис должен будет стать частью платформы многомасштабного моделирования, разрабатываемого отделом “Математическое моделирование гетерогенных систем” вычислительного центра им. А.А.Дородницына Федерального исследовательского центра “Информатика и управление” Российской академии наук.

В магистерской диссертации ставятся задачи, связанные с поиском, выбором и тестированием работы одного из существующих программных комплексов для квантово-механического расчета, который удовлетворял бы таким качествам как:

- свободный доступ
- высокие функциональные возможности пакета

После анализа работы программного пакета, пробных запусков расчетов и изучения формата входных и выходных данных, встал вопрос о выборе инструментов для создания сервиса, а именно:

- какой язык программирования необходимо выбрать в качестве основного
- каким образом будет производиться взаимодействие компонентов в сети
- с помощью какой системы управлять базой данных
- какой веб фреймворк использовать для создания пользовательского интерфейса

Финальной частью магистерской диссертации является реализация программного интерфейса, который позволяет запускать через сервис квантово-механические расчеты с помощью Quantum Espresso на удаленном вычислительном кластере, а также макет пользовательского интерфейса для удобного взаимодействия специалиста в области квантовой физики с сервисом.

ОСНОВНАЯ ЧАСТЬ

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1. Первопринципные методы квантовой химии

Первопринципные методы – это методы, которые основаны на квантовой химии и определяются термином *ab initio*. Впервые этот термин начали использовать в квантовой химии Робертом Парром и его сотрудниками в исследовании возбужденных состояний бензола. *Ab initio* - «от первых принципов» или «с самого начала», подразумевает, что единственными входными данными в расчет *ab initio* являются физические константы. *Ab initio* методы пытаются решить электронное уравнение Шредингера с учетом положений ядер и количества электронов, это дает возможность получить полезную информацию, такую как плотность электронов, энергии и другие свойства системы. Возможность проводить эти расчеты позволила химикам-теоретикам решить ряд проблем, важность которых подчеркивается присуждением Нобелевской премии Джону Поплу и Уолтеру Кону.

Ab initio методы направлены на то, чтобы вычислить многоэлектронную функцию, которая является решением нерелятивистского решения электронного уравнения Шредингера (в приближении Борна – Оппенгеймера). Многоэлектронная функция представляет из себя линейную комбинацию многих простых электронных функций, но доминирующей функцией является функция Хартри-Фока. Далее каждая из этих простых функций аппроксимируется с использованием только одноэлектронных функций. Затем одноэлектронные функции расширяются, как линейная комбинация конечного набора базисных функций. Этот подход имеет одно преимущество, его можно привести к точному решению, когда базовый набор стремится к пределу полного набора и где включены все возможные конфигурации («Полный CI»). Однако эта сходимость к пределу требует больших вычислительных затрат, и большинство вычислений далеки от предела. При выборе подхода для решения данной проблемы стоит учитывать вычислительные затраты методы *ab initio*. По сравнению с менее точными подходами, например, молекулярная механика,

методы *ab initio* часто требуют большего количества компьютерного времени, памяти и дискового пространства, но благодаря достижениям в области компьютерных наук, это проблема скоро исчезнет.

1.1.1. Программные пакеты

Существует не малое количество программных пакетов для первых принципов расчетов электронной структуры и моделирования материалов, которые позволяют ученым выполнять сложные расчеты и быстро получать результат. Одни являются инструментами для энергетических расчетов многоэлектронных систем, другие дают возможность провести расчет электронной структуры молекул и твердых тел и решить задачу молекулярной динамики, третьи обладают всеми названными возможностями. На Рис.1.1 представлены одни из таких пакетов.



Рис. 1.1. Примеры программных пакетов для квантово-механических расчетов

- VASP – это комплексный пакет для выполнения моделирования *ab-initio* квантово-механической молекулярной динамики с использованием псевдопотенциалов или метода проекционно-дополненных волн и базисного набора плоских волн. Подход,

реализованный в VASP, основан на аппроксимации локальной плотности (конечной температуры) со свободной энергией в качестве вариационной величины и точной оценке мгновенного электронного основного состояния на каждом временном шаге

- CASTEP – это академический и коммерческий программный пакет с общим исходным кодом, использующий теорию функционала плотности с базисным набором плоских волн вычислить электронные свойства кристаллических твердых тел, поверхностей, молекул, жидкостей и аморфных материалов из первых принципов. CASTEP позволяет оптимизировать геометрию и конечную температурную молекулярную динамику с неявными ограничениями симметрии и геометрии, а также вычислять широкий спектр производных свойств электронной конфигурации
- SCIGRESS – это набор программного обеспечения для молекулярного моделирования, вычислительной химии, дизайна лекарств и материаловедения, преемник программного обеспечения для компьютерной химии (CASChe)
- Quantum Espresso – набор для расчетов электронной структуры первых принципов и моделирования материалов
- Siesta – это компьютерная программная реализация для выполнения эффективных расчетов электронной структуры и *ab initio* молекулярно-динамического моделирования молекул и твердых веществ
- Plato - это набор программ для расчета электронной структуры. Он получил свое название от выбора базисного набора (числовых атомных орбиталей), используемого для расширения электронных волновых функций

1.1.2. Quantum Espresso

Quantum Espresso – это набор для расчетов электронной структуры первых принципов и моделирования материалов, распространяемый бесплатно и в виде свободного программного обеспечения под лицензией GNU General Public License. Он основан на теории функционала плотности, базисных множествах плоских волн и псевдопотенциалах, как сохраняющих форму, так и сверхмягких.

Основные плосковолновые функции Quantum Espresso обеспечиваются компонентом PWscf (Plane-Wave Self-Consistent Field) - это набор программ для расчета электронных структур в рамках теории функционала плотности и теории возмущений функционала плотности с использованием базисных наборов плоских волн и псевдопотенциалов.

Программа написана в основном на fortran-90 с некоторыми частями на языке программирования C или на Fortran-77, была построена из слияния и реинжиниринга различных независимо разработанных основных пакетов плюс набор пакетов, предназначенных для взаимодействия с основными компонентами, которые позволяют выполнять более сложные задачи.

Основные пакеты включают Pwscf, который решает самосогласованные уравнения Кона-Шема.

Задачи, которые можно выполнить с помощью данного пакета:

- расчеты основного состояния
- структурная оптимизация
- переходные состояния и минимальные энергетические пути
- свойства отклика, такие как фононные частоты, электрон-фононные взаимодействия и химические сдвиги
- спектроскопические свойства
- квантовый импорт
- генерация псевдопотенциалов

Пакет Quantum Espresso предназначен для работы в операционной системе Linux. Все расчеты проводятся в нем в базисе плоских волн с использованием псевдопотенциалов, ультрамягких или сепарабельных сохраняющих норму.

1.1.3. Существующие графические интерфейсы

Существует всего 2 графических интерфейса, которые каким-то образом делают квантово-механические расчеты наглядными (Рис.1.2.)

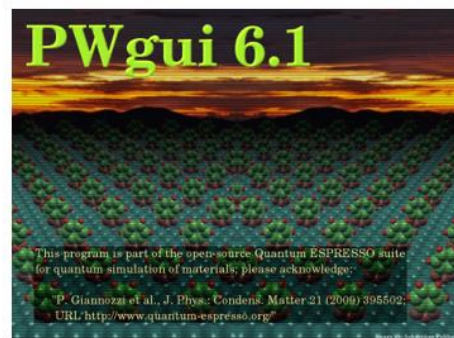


Рис. 1.2. Примеры графических интерфейсов

SCM (Software for Chemistry & Materials) – платное программное обеспечение, которое позволяет строить структуры, запуск и визуализацию геометрической оптимизации.

PWgui – доступный графический интерфейс разработанный создателями программного пакета Quantum Espresso (QE). PWgui вынуждает пользователя загрузить сам пакет QE, что идет в разрез идее создания простого сервиса.

1.1.4. Пример выполнения расчета

Для проведения расчета электронной структуры в пакете Quantum Espresso используется программа `pw.x`. На (Рис. 1.2.) приведен пример входного файла с необходимыми параметрами для `pw.x`. Это простой абстрактный конфигурационный файл для Quantum Espresso, который не задает какой-либо реальный материал.

```

&control
  prefix = 'simple',
  pseudo_dir = '../pseudo',
  outdir = './outdir',
/
&system
 ibrav = 1,
celldm(1) = 10,
ntyp = 1,
nat = 1,
ecutwfc = 50,
ecutrho = 100,
/
&electrons
/
ATOMIC_SPECIES
  Sn 118.71 Sn.pbe-dn-kjpaw_psl.1.1.0.0.UPF
ATOMIC_POSITIONS alat
  Sn 0 0 0
K_POINTS automatic
  4 4 4 1 1 1

```

Рис. 1.3. Пример входного файла

Как видно из (Рис. 1.3.) входные данные сгруппированы в отдельные блоки. В данном примере входной файл состоит из `&CONTROL`, `&SYSTEM`, `&ELECTRONS`, `ATOMIC_SPECIES`, `ATOMIC_POSITIONS`.

Ниже приведено подробное описание каждого блока:

&CONTROL – описывает общие характеристики расчета

- **calculation** – тип расчета
- **tstress** – логический параметр, который сообщает программе требуется ли проводить вычисление тензора напряжений или нет
- **tprnfor** – если `.true.`, то будут выполнены расчеты действующих на атомы сил
- **pseudo_dir** – задает путь до каталога, в котором будут храниться временные файлы, создаваемые в процессе расчета
- **outdir** - задает путь до каталога, в котором будут храниться временные файлы, создаваемые в процессе расчета

- **prefix** – сообщает программе, какой набор символов (префикс) использовать в имени файлов

&SYSTEM – описывает расположение атомов в системе и параметры сходимости процесса вычислений

- **ibrav** – тип решетки Браве
- **celldm(1)** – параметр решетки элементарной ячейки кристалла
- **nat** – число атомов
- **ntyp** – число сортов атомов
- **ecutwfc** – энергия обрезки плоских волн, используемых в разложении электронной волновой функции
- **ecutrho** – энергия обрезки плоских волн для представления зарядовой плотности и потенциала (Ry)

&ELECTRONS

- **conv_thr** – порог сходимости расчета
- **mixing_beta** – параметр смешивания

ATOMIC_SPECIES – характеристики атома и псевдопотенциал

ATOMIC_POSITIONS – координаты всех атомов

K_POINTS – координаты и весовые коэффициенты k-точек

После запуска программы проводится самосогласованный расчет. После расчета в выходном потоке будет вся интересующая информация, на данный момент интересует только полная энергия системы, которую можно найти в файле по восклицательному знаку в начале строки.

! total energy = -441.85937387 Ry

1.2. Разработка сервиса квантово-механического расчета

1.2.1. Требования к реализуемому программному сервису

В настоящее время не существует web-интерфейса, который позволил бы работать с программным пакетом Quantum Espresso в интерактивном режиме, не устанавливая никакого дополнительного программного обеспечения на свое устройство. Имеющиеся графические интерфейсы либо являются платными, либо не в полной мере упрощают работу конечного пользователя. Необходимо же создать такой интерфейс, который не вынуждает специалиста «копаться» в программной реализации и формировать входные данные в виде текстового файла.

Предполагаемый сервис для квантово-механического расчета должен состоять из:

- программного интерфейса, который будет производить удаленное управление через SSH (Secure Shell – протокол удаленного управления компьютером) пакетом Quantum Espresso, «вытаскивать» параметры псевдопотенциалов из баз данных и взаимодействовать с пользовательским интерфейсом
- пользовательского интерфейса, для взаимодействия конечного пользователя с сервисом

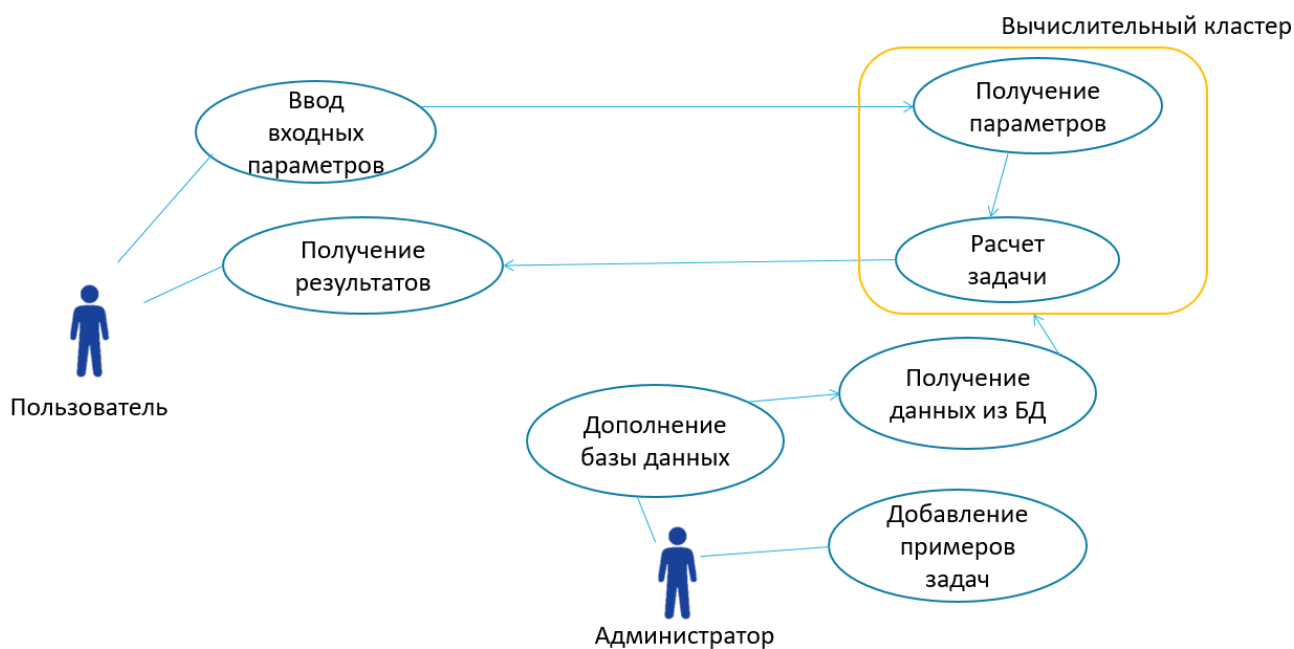


Рис. 1.4. Диаграмма прецедентов

Диаграмма прецедентов (Рис.1.4.) доступно и понятно показывает, как должно производиться взаимодействие пользователя с сервисом и какие сценарии будут реализованы:

- ввод пользователем входных параметров структуры системы
- получение вычислительным кластером данных и запуск расчета
- окончание квантово-механического расчета и предоставление пользователю конечного ответа

Исходя из этого можно выделить следующие технологии которые будут необходимы для создания сервиса:

- набор связанных сервисов – микросервисная архитектура
- Rest API – способ взаимодействия клиентского и серверного приложений
- Swagger интерфейс – позволяет в интерактивном режиме производить HTTP запросы
- Использование нереляционных баз данных – упрощает работу с конвертацией данных и снижает затраты в плане создания новых миграций

1.2.2. Программный интерфейс

Сервис для квантово-механического расчета должен являться частью одной большой платформы многомасштабного моделирования. Соответственно большинство инструментов для разработки, на которых создана платформа, будет применимо и для предполагаемого сервиса. Связано это с тем, что в дальнейшем данный модуль будет проще встроить в общую архитектуру системы, а также в будущем поддержка программного кода не окажется проблемой для разработчика.

В качестве основного языка программирования использовался статически типизированный, объектно-ориентированный язык Kotlin, который работает поверх Java Virtual Machine и разрабатывается компанией JetBrains. Благодаря статической типизации языка, разрабатываемые приложения являются более стабильными. В паре с библиотекой Ktor, фреймворка для построения асинхронных серверов и клиентов в подключенных системах, Kotlin облегчает разработку автономного приложения с встроенными серверами.

Ниже представлены иные варианты наборов инструментов для разработки API:

- Python + Flask - простой синтаксис языка с библиотекой, которая предоставляет простоту, аккуратность и гибкость в работе, позволяя пользователю выбирать как реализовать определенные вещи
- Python + Django - простой синтаксис языка с пакетом «все включено»
- Java/Kotlin + SpringBoot — статически типизированный язык с фреймворком, в котором реализованы все инструменты для создания серверного приложения.

1.2.3. Пользовательский интерфейс

На сегодняшний момент подавляющее большинство библиотек – JavaScript решения. Это обусловлено тем, что веб-технологии доминирует на рынке. Практически все приложения и новые решения работают в браузерах. Поэтому разумнее разрабатывать приложение, которое будет хорошо работать на любых устройствах, вместо того, чтобы создавать целую партию приложений отдельно для рабочего стола, отдельно для телефона и отдельно для планшета. По этим причинам язык программирования JavaScript стал очень популярным.

Ниже представлены 5 самых популярных JavaScript-библиотек:

- **Webix UI Library** - представляет собой набор инструментов пользовательского интерфейса JavaScript/HTML5/CSS3 для разработки сложных и динамичных кроссплатформенных веб-приложений. Фреймворк разработан ИТ-аутсорсинговой компанией XB Software со штаб-квартирой в Минске, Беларусь
- **DHTMLX** - предлагает библиотеку компонентов Ajax, которая позволяет разработчикам создавать многофункциональные и интерактивные веб-интерфейсы. Он включает в себя набор готовых виджетов, написанных на чистом JavaScript и CSS
- **Ext JS** - это чистая платформа приложений JavaScript для создания интерактивных кросс-платформенных веб-приложений с использованием таких методов, как Ajax, DHTML и DOM-сценарии. Его можно использовать как простую компонентную структуру (например, для создания динамических сеток на других статических страницах), но также и как полную структуру для создания одностраничных приложений
- **React** - это библиотека JavaScript с открытым исходным кодом для создания пользовательских интерфейсов или компонентов пользовательского интерфейса. Он поддерживается Facebook и сообществом отдельных разработчиков и компаний. React можно

использовать в качестве основы при разработке одностраничных или мобильных приложений

- JQuery UI - представляет собой набор виджетов графического интерфейса, анимированных визуальных эффектов и тем, реализованных с помощью jQuery (библиотека JavaScript), каскадных таблиц стилей и HTML. По данным службы аналитики JavaScript, Libscore, пользовательский интерфейс jQuery использовался более чем на 197 000 из миллиона лучших веб-сайтов, что когда-то сделало его второй по популярности библиотекой JavaScript. Известными пользователями были Pinterest, PayPal, IMDb

Для создания пользовательского интерфейса была выбрана библиотека React в паре с языком программирования TypeScript. Это обусловлено тем, что React обладает такой особенностью, как использование виртуального DOM (Virtual DOM). Virtual DOM — это объект, в котором хранится информация о состоянии интерфейса. Другие ее особенности — это декларативность, компонентный подход, универсальность, использование виртуального DOM и JSX. TypeScript в свою очередь не меняет способ написания кода и расширяет возможности языка JavaScript. Он только добавляет дополнительную структуру, чтобы создать уровень проверки.

Эта пара позволит создать удобный интерфейс, который будет понятен конечному пользователю, а тот в свою очередь сможет с легкостью проводить расчеты.

1.2.4. Хранение данных

Модели баз данных или семейства баз данных являются шаблонами и структурами, которые используются для организации данных в системе управления базами данных (СУБД). От выбора той или иной модели зависит то, какие операции сможет выполнять приложение, в каком формате будут представлены данные и на функции СУБД.

Всего разделяют около 11 моделей:

- простые структуры данных – текстовые файлы являются ярким примером таких структур. Такой метод применяют сегодня для работы с небольшими объемами
- иерархические – в данной модели уже появляются связи между объектами. Каждая запись имеет «родителя».
- сетевые – расширяют функциональность иерархических – имеют больше одного родителя.
- реляционные базы данных – данные и связи между данными организованы с помощью таблиц. Каждый столбец обладает именем и типом.
- ключ-значение – для хранения информации предоставляется ключ и объект данных, который требуется сохранить
- документная БД – использует базовую семантику доступа и хранилищ ключей и значений. Документоориентированные в отличие от хранилищ «ключ-значение» хранят данные не в blob объектах, а в структурированных форматах – JSON, BSON, XML.
- графовые БД – устанавливают связи с помощью узлов, ребер и свойств
- колоночные БД – принадлежат к семейству NoSQL БД. Как и реляционные БД хранят данные, используя строки и столбцы, но с иной связью между элементами
- БД временных рядов – созданы для сбора и управления элементами, меняющимися с течением времени. Большинство организовано в структуры, которые записывают значения для одного элемента. Внутри каждое значение состоит из временной метки и определенного показателя
- NewSQL – наследуют реляционную структуру и семантику, но построены с использованием более временных, масштабируемых конструкций

- Многомодельные БД – объединяют функциональные возможности нескольких видов БД. Преимущества в том, что одна и та же системы может использовать различные представления для разных типов данных

В пакете Quantum Espresso реализован удобный вариант сохранения выходных данных в XML файл, поэтому выбор пал на документный тип БД, так как он работает с такими форматами данных.

По причине того, что большая часть сервисов платформы многомасштабного моделирования используют продукт MongoDB, то и для реализуемого сервиса эта база данных станет основной. Она будет отвечать за хранение результатов расчетов, файлов псевдопотенциалов и справочников.

MongoDB – документноориентированная система баз данных. Является продуктом с открытым исходным кодом. Не требует описания схемы таблиц.

2. ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1. Архитектура платформы и предполагаемого сервиса

Как уже было ранее сказано, сервис, для которого необходимо было создать программный и пользовательский интерфейсы, должен стать одной из частей интеграционной платформы многомасштабного моделирования, разрабатываемого отделом «Математическое моделирование гетерогенных систем» вычислительного центра им А.А.Дородницына Федерального исследовательского центра «Информатика и управление» Российской академии наук. Данная платформа представляет из себя web-приложение, которое позволяет проводить эффективное параллельное решение задач разных масштабов с учетом постоянного обмена данными [1]. Она состоит из модулей, которые в свою очередь отвечают за конкретный этап расчета и моделирования системы на одном из масштабных уровней. Схема общей архитектуры платформы представлена на (Рис. 2.1.)

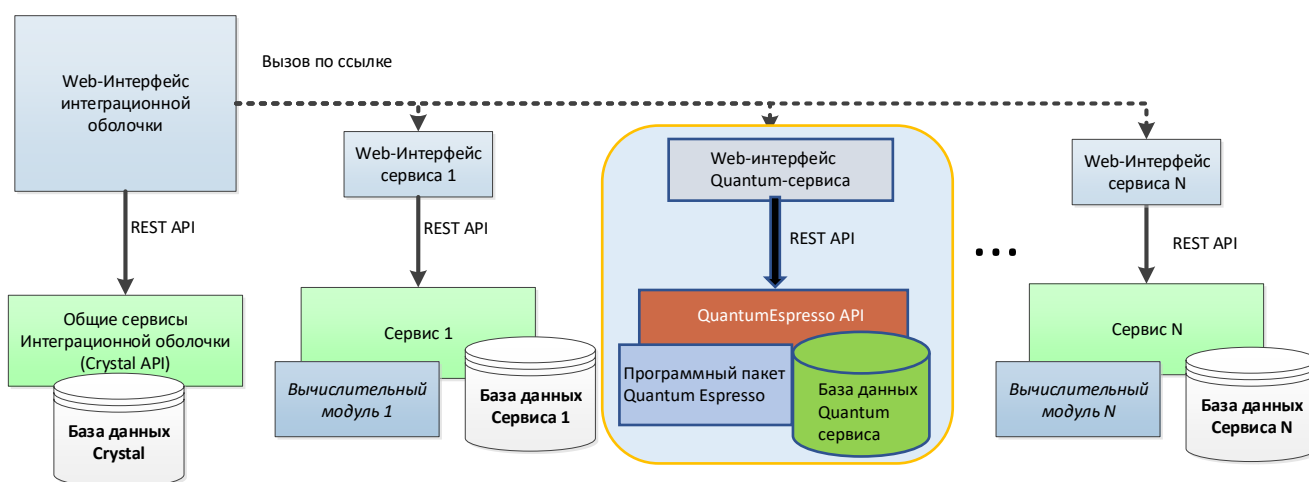


Рис. 2.1. Архитектура платформы

Как видно из архитектуры, сервис, к которому разрабатываются программный и пользовательский интерфейс в рамках магистерской диссертации, является одной из деталей большой платформы. Более подробная его структура и схема взаимодействия компонентов изображена на (Рис. 2.2.)

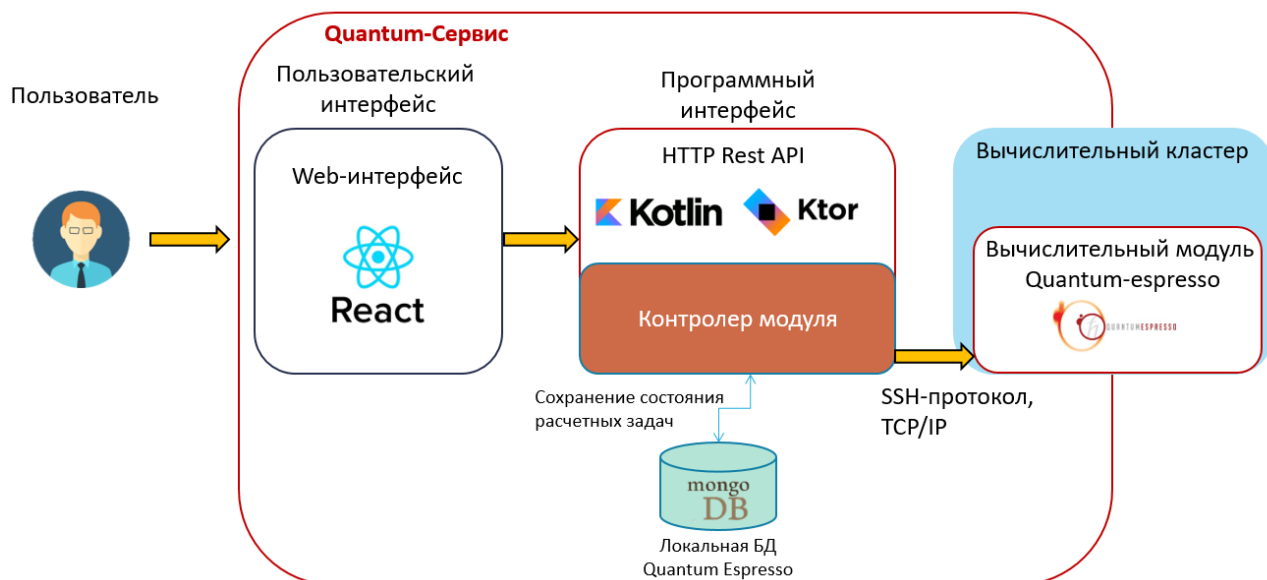


Рис. 2.2. Архитектура Quantum-сервиса

Компоненты, использованные в ходе разработки:

- Пользовательский интерфейс – web интерфейс, который взаимодействует с программным интерфейсом посредством серверного протокола передачи данных HTTP. Реализован с помощью JavaScript библиотеки (React)
- Программный интерфейс – приложение, которое выполняет всю основную работу и является центральной частью разрабатываемого сервиса. Взаимодействует с базой данных, куда сохраняет промежуточные результаты, производит передачу состояния представлений с пользовательским интерфейсом, для того, чтобы пользователь мог сам задавать параметры системы и получать выходные результаты расчетов. Написан с помощью статически типизированного языка программирования Kotlin совместно с библиотекой Ktor
- База данных – использовалась система управления базами данных MongoDB. Она хранит в себе все результаты вычислений

предыдущих расчетов, а также свою необходимую базу стандартных решений и набор псевдопотенциалов

- Quantum-модуль – представляет из себя программный пакет для квантово-механических расчетов Quantum Espresso, который будет размещен на вычислительном кластере для большей скорости выполнения задач

2.2. Серверное приложение

Приложение реализовано по схеме – View, Controller, Service, Repository.

- View – выводит данные клиенту, визуализация состояния логики
- Controller – он отвечает за взаимодействие с пользователем посредством ввода параметров системы и выводов состояния работы программы
- Repository – слой, который отвечает за получение и использование данных из внешнего источника, в данном случае из базы данных.
- Service – отвечает за бизнес логику.

Можно разделить приложение на 4 блока Рис.2.3.

Первый - модуль квантово-механического расчета, он запускается ядром приложения посредством удаленного управления с помощью SSH протокола, третьим блоком или Service. Второй блок базы данных, который является хранилищем произведенных расчетов и существующих файлов псевдопотенциалов, выступает в роли Repository, с ним также взаимодействует третий блок, сохраняя в него промежуточные результаты и использует хранимые файлы псевдопотенциалов. Помимо этого ядро принимает HTTP запросы от четвертого, клиентского блока приложения, Controller и View, с введенными пользователем данными и возвращает ему обратно произведенные результаты расчета.

Клиентское приложение

Серверное приложение

Программный пакет квантово-

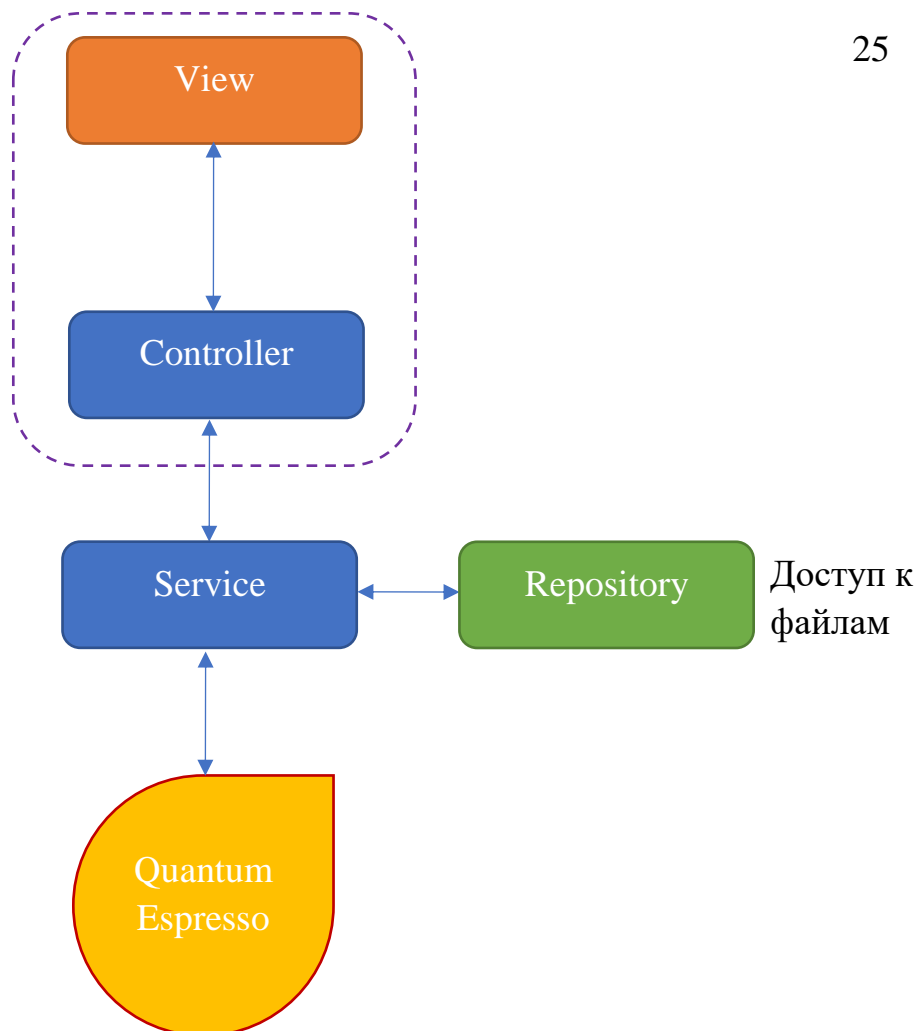


Рис 2.3. Структура приложения

Со структурой приложения можно ознакомиться на Рис.2.3.

2.2.1. Бизнес логика



Рис 2.4. Блок серверного приложения

Серверное приложение Рис 2.4. представляет из себя набор взаимодействий с модулем Quantum Espresso и базой данных. Бизнес логика ограничивается лишь ими двумя. Однако для управления всем приложением необходимо какое-то центральное ядро. Оно должно будет оперировать управлением запусков и остановки расчета, извлекать предыдущие результаты экспериментов и помещать только посчитанные в базу данных и при этом

предоставлять клиентскому приложению необходимые файлы и данные для взаимодействия пользователя с платформой

2.2.2. Способ хранения данных

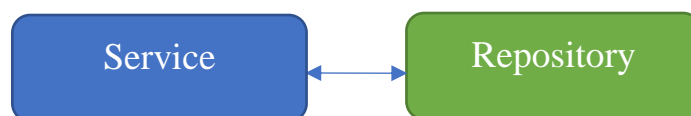


Рис 2.5. Блок связи с базой данных

В приложении организована возможность хранения промежуточных результатов и результатов предыдущих экспериментов. На Рис.2.3. за базу данных отвечает компонент Repository. Хитрой обработки массива данных не предусмотрено, необходим лишь ограниченный функционал записи и чтения, а также обновления. Функционал:

- Сохранение данных предыдущих расчетов
- Получение данных об эксперименте по Id
- Удаление данных об эксперименте по Id

2.2.3. Взаимодействие с модулем Quantum Espresso

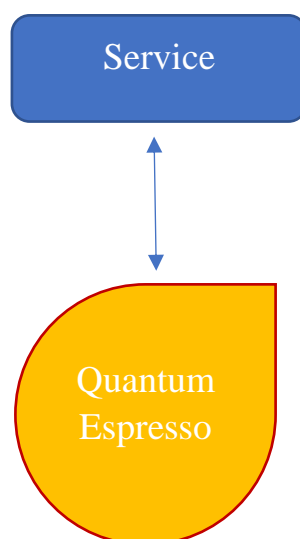


Рис 2.6. Блок связи ядра приложения с пакетом Quantum Espresso

Модуль Quantum Espresso представляет из себя программный пакет Quantum Espresso, который позволяет производить расчеты для вычисления полной и парциальной энергии, определять энергетические барьеры и пути реакции. Для быстрой скорости работы и быстрого проведения расчетов для сложных систем, этот пакет размещен на кластере, который в свою очередь обладает высокими показателями вычислительной мощности.

На вход программному пакету отправляются значения параметров предполагаемой системы, для которой будет произведен тот или иной расчет. Пример структуры входных данных приведен в главе 1.1.4. Пример выполнения расчета. При запуске приложения производится старт программы pw.x, происходит расчет и выходные данные сохраняются в файл и в дальнейшем выводятся пользователю на экран.

Взаимодействие с модулем Quantum Espresso предполагалось произвести с помощью HTTP протокола и Rest API напрямую с предполагаемым сервисом. Это уменьшает общее количество разворачиваемых приложений, а также исключает возможность ошибки и неправильно реализованной логики внутри.

Управление программным пакетом производится благодаря протоколу ssh, поэтому были созданы Shell скрипты для запуска/остановы расчета, конверторы

2.2.4. Программный интерфейс



Controller

Рис.2.7. Ядро приложения

В программном интерфейсе было предусмотрено использование endpoint'ов которые описывают опции коммуникаций:

- Address – где находится;
- Binding – как отправлять сообщения;
- Contract – что содержит сообщение.

В протоколе HTTP имеется несколько вариантов отправки запроса на один и тот же endpoint:

- POST /run – запуск расчета
- GET /run – информация обо всех расчетах
- GET /run/{id} – информация о расчете
- GET /run - пример данных для запусков расчетов
- DELETE /run/{id} – принудительное завершение расчета

2.3. Клиентское приложение

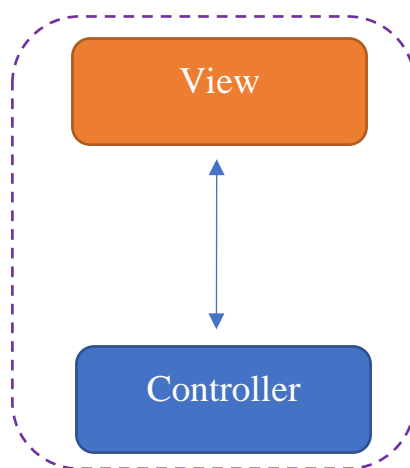


Рис. 2.8. Блок клиентского приложения

Клиентское приложение будет представлять из себя совокупность графического интерфейса View, где будут отрисовываться компоненты и взаимодействие пользователя с этим интерфейсом Controller Рис. 2.8.

2.4. Взаимодействие серверного и клиентского приложений

Взаимодействие серверного и клиентского приложений Рис.2.9. было решено реализовать только с помощью Rest API, так как он позволяет проводить связь с сервером не только через пользовательский интерфейс, но и через веб фреймворк Swagger, который дает возможность не только

интерактивно просматривать спецификацию RESTful API, но и отправлять запросы.

Swagger позволяет создать полностью функционирующий продукт, а сам API написать потом. С другими вариантами взаимодействия такое бы не получилось реализовать.



Рис. 2.9. Схема взаимодействия

ЗАКЛЮЧЕНИЕ

В рамках магистерской диссертации были учтены все лучшие технологии проектирования и разработки серверных и клиентских приложений. Реализован сервис для квантово-механического расчета с помощью Quantum Espresso. Аналогов данного сервиса на данный момент в мире не существует, что делает этот проект уникальным в своем роде.

Готовый программный сервис предполагается интегрировать в платформу многомасштабного моделирования и в дальнейшем производить его поддержку и дополнение новыми возможностями.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Абгарян К.К, Гаврилов Е.С. «Интеграционная платформа для многомасштабного моделирования нейроморфных систем», «Информатика и ее применения», том 14, выпуск 2, 2020 г. Стр. 104-110.
- [2] Jemerov, D., Isakova, S. Kotlin in Action. Manning, 2017 г.
- [3] Robert C. Martin, James Grenning, Simon Braun. 2018. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Pearson Education, Inc.
- [4] Marc Garreau, Will Faurot, Redux in action. 2018, Manning Publications Co.
- [5] Michael S. Mikowski, Josh C. Powell, Single Page Web Applications, 2014, Manning Publications Co.
- [6] Официальная документация Quantum Espresso [Электронный ресурс] URL: <https://www.quantum-espresso.org>
- [7] Официальная документация Kotlin [Электронный ресурс] URL: <https://kotlinlang.org/>