

Министерство образования Российской Федерации
Пензенский государственный университет
Кафедра «Вычислительная техника»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовой работе
по курсу «Распределённые системы»
на тему «Разработка параллельной программы
для кластера облачного типа»

Вариант - 8

Выполнила:

студент группы 16BB2

Жидкова К.О.

Принял:

Федюнин Р.Н.

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет Вычислительной техники
Кафедра "Вычислительная техника"

"УТВЕРЖДАЮ"

Зав. кафедрой ВТ
профессор М.А. Митрохин

« 10 » марта 2020г

ЗАДАНИЕ

на курсовое проектирование по курсу

Распределённые системы

Студенту Мигровой Карине Олеговне Группа 16ВВ2
Тема проекта Разработка параллельной программы для кластера облачного
типа _____

Исходные данные (технические требования) на проектирование

1) Клиент-серверное приложение для управления вычислительным процессом должно состоять из двух частей: клиента и сервера, которые должны образовывать единую распределенную вычислительную систему облачного типа. 2) Серверное приложение должно обеспечивать одновременное подключение нескольких клиентов. 3) Клиентское приложение должно обеспечивать подключение к серверу (порт и IP сервера вводятся в окне клиента) и иметь возможность осуществить отключение от сервера в любой момент. 4) В окне серверного приложения должен отображаться прогресс решения вычислительной задачи и примерное время, оставшееся до ее решения с учетом текущей скорости вычислительного процесса, а также указываться список подключившихся клиентов и шаг интегрирования. 5) Для каждого клиента должна быть предусмотрена возможность указания предельно допустимого уровня использования вычислительных ресурсов (процессорного времени) в процентах. Соответствующее значение вводится на клиентской. 6) Кластер должен решать задачу вычисления определенного интеграла $\int \sin(x^2)$. 7) Для интегрирования должен использоваться методом правых прямоугольников. 8) Приложения (клиент и сервер) должны обладать интуитивно понятным графическим интерфейсом. 9) Для обмена данными использовать протокол TCP/IP и библиотека WinSock. 10) Язык программирования C++, среда разработки Microsoft Visual Studio.

1. Графическая часть

- 1) диаграмма классов; _____
- 2) диаграмма деятельности; _____
- 3) диаграмма вариантов использования; _____
- 4) диаграмма развёртывания; _____
- 5) диаграмма последовательности _____

2. Экспериментальная часть

Тестирование распределенной вычислительной системы _____

Срок выполнения проекта по разделам

В соответствии с графиком _____

Дата выдачи задания “10” марта 2020г.

Дата защиты проекта “12” мая 2020г.

Руководитель Редюкин Р.Н. Фир

Задание получил “10” марта 2020г.

Студент Ниркова К.О. Кош

Содержание

Введение.....	7
1 Постановка задачи.....	8
2 Выбор решения.....	9
3 Описание разработки программы.....	12
4 Отладка и тестирование.....	Ошибка! Закладка не определена.
5 Описание программы.....	Ошибка! Закладка не определена.
6 Руководство пользователя.....	Ошибка! Закладка не определена.
Заключение	15
Список используемых источников.....	16
Приложение А1 Листинг программы «ServerApp»	Ошибка! Закладка не определена.
Приложение А2 Листинг программы «ClientApp»	Ошибка! Закладка не определена.
Приложение В Результаты работы программы	Ошибка! Закладка не определена.

Введение

Многие задачи требуют вычислений с большим количеством операций, которые занимают значительные ресурсы даже современной техники, более того, можно с уверенностью считать, что, каких бы скоростей ни достигла вычислительная техника, всегда найдутся задачи, на решение которых потребовалось значительное время. Многие из таких сложных задач требуют, чтобы результат был получен за как можно меньшее время или даже строго ограниченное. К таким задачам, например, относится прогнозирование погоды, обработка изображений и распознавание образов при управлении техникой. С другой стороны, представляет большую техническую проблему уменьшение времени исполнения каждой операции в микропроцессоре.

Очевидным способом увеличить скорость вычислений было бы применение не одного вычислительного устройства, а нескольких, работающих совместно над решением одной задачи. Такой подход носит название параллельных вычислений. Несмотря на кажущуюся простоту решения оно является подчас весьма нетривиальной задачей по проектированию вычислительной техники и разработки алгоритмов. Первая проблема кроется в том, что для того, чтобы задачу можно было решить с помощью параллельных вычислений алгоритм её решения должен допускать распараллеливание, мало того, далеко не каждая задача может быть решена параллельным алгоритмом. Другой же, не менее важной проблемой является построение системы, на которой бы возможна была реализация параллельных вычислений.

Программа, разрабатываемая в рамках данной курсовой работы, предназначена для параллельного вычисления определенного интеграла. Система состоит из клиента и сервера, обмен данными между которыми реализован по протоколу TCP/IP.

1 Постановка задачи

Необходимо разработать клиент-серверное приложение для управления вычислительным процессом, имеющее интуитивно понятный графический интерфейс. Для этого нужно определить зоны ввода необходимых для работы программы данных и область вывода информации о соединении, о ходе выполнения вычислений и других требуемых данных.

Программа должна выполнять вычисление определенного интеграла $\int \sin(x^2)$ методом правых прямоугольников. Для выполнения данного требования необходимо реализовать алгоритм вычислений по соответствующей формуле.

2 Выбор решения

Для реализации программы был выбран событийный механизм управления. Разработанная программа состоит из двух отдельных приложений: клиентское приложение «ClientApp» и серверное приложение «ServerApp».

В качестве языка программирования был выбран язык C++.

В качестве графической библиотеки была выбрана Qt5, предоставляющая большой набор готовых элементов управления для построения эффективных GUI. Qt5 поддерживает событийную модель управления, позволяющая привязать изменения программной системы с взаимодействием пользователя и интерфейса программы.

В качестве сетевой библиотеки была выбрана библиотека WinSock 2 от Microsoft. Библиотека поддерживает построение сетевого взаимодействия по широкому набору сетевых протоколов (TCP, UDP и др).

В качестве IDE была выбрана MS Visual Studio 2010, предоставляющая возможности как для разработки ПО, так и для его всестороннего тестирования.

На этапе проектирования программы было принято решения реализовать программу в объектно-ориентированном стиле: деление ПО на совокупность классов помогает инкапсулировать логику программы, программные параметры библиотек Qt5 и WinSock 2 вместе, что значительно упрощает разработку подобного приложения по сравнению с функциональным стилем.

Пользователю будет предоставлена возможность управления вычислениями на серверной стороне, задавать параметры задач (диапазон) и подключать/отключать клиентское приложение посредством элементов ввода «кнопка» и «текстовое поле». Для отладки будет введён лог сообщений, собирающий информацию от программы при посылке соответствующего события.

Каждому объекту будет выделена отдельная нить для ускорения работы программы и обеспечения асинхронной реакции на различные события, возникающие в процессе функционирования программы. Спроектированная

диаграмма классов для серверного (Рисунок 1) и клиентского (Рисунок 2) приложения представлена ниже.

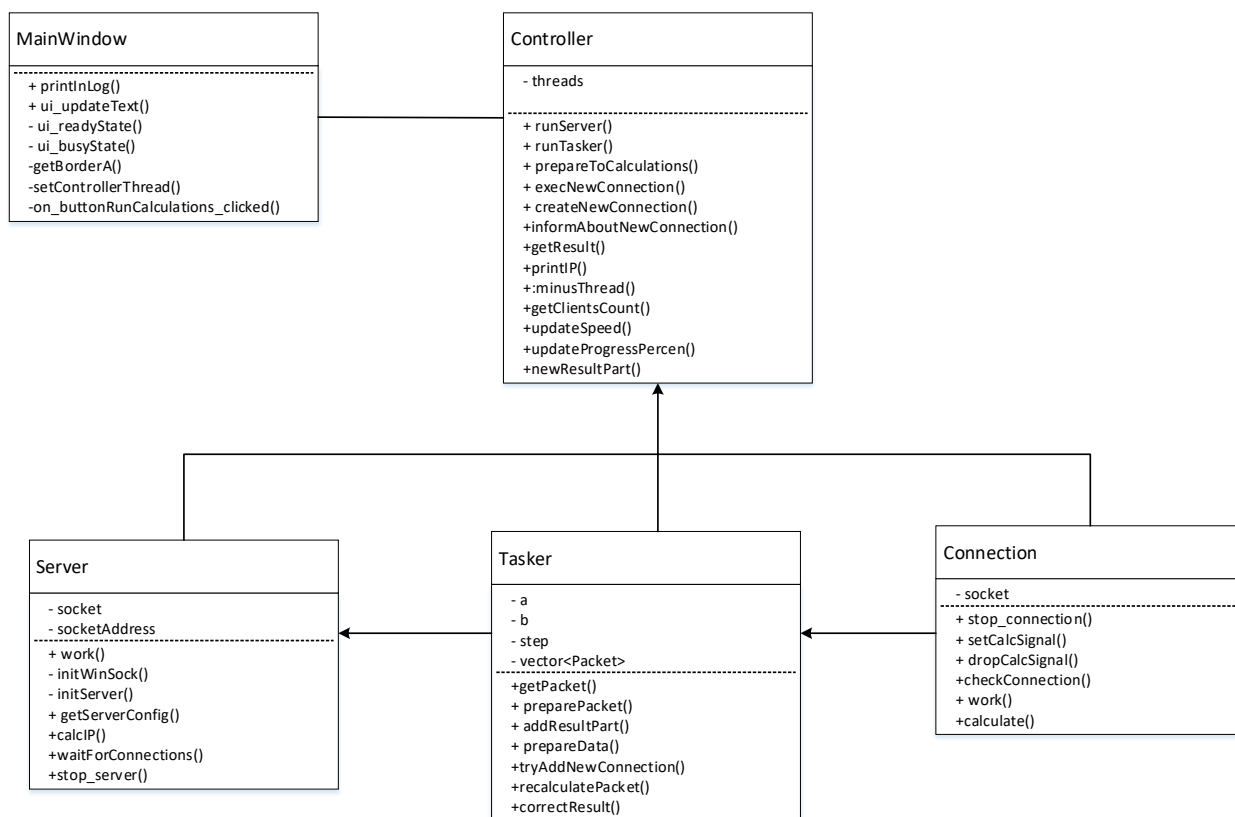


Рисунок 1. Диаграмма классов серверного приложения

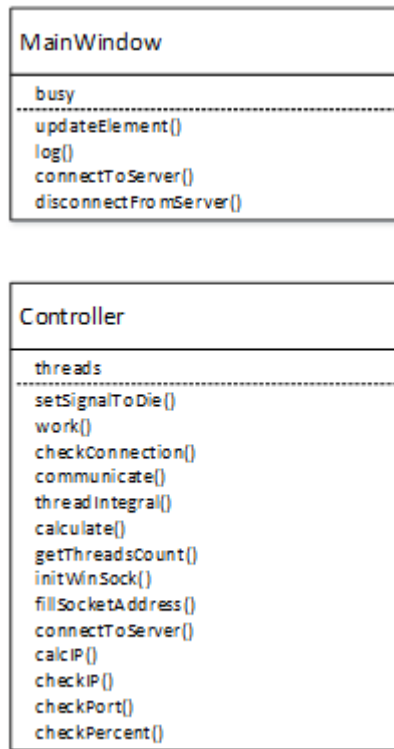


Рисунок 2. Диаграмма классов клиентского приложения

3 Описание разработки программы

Разработка программы началась с разработки графического интерфейса пользователя. Это позволило к моменту написания рабочей логики приложения управлять состояниями программы без дополнительных средств отладки, параметров командной строки и др. Диаграмма активностей для серверного (Рисунок 3) и клиентского (Рисунок 4) приложения представлена ниже.

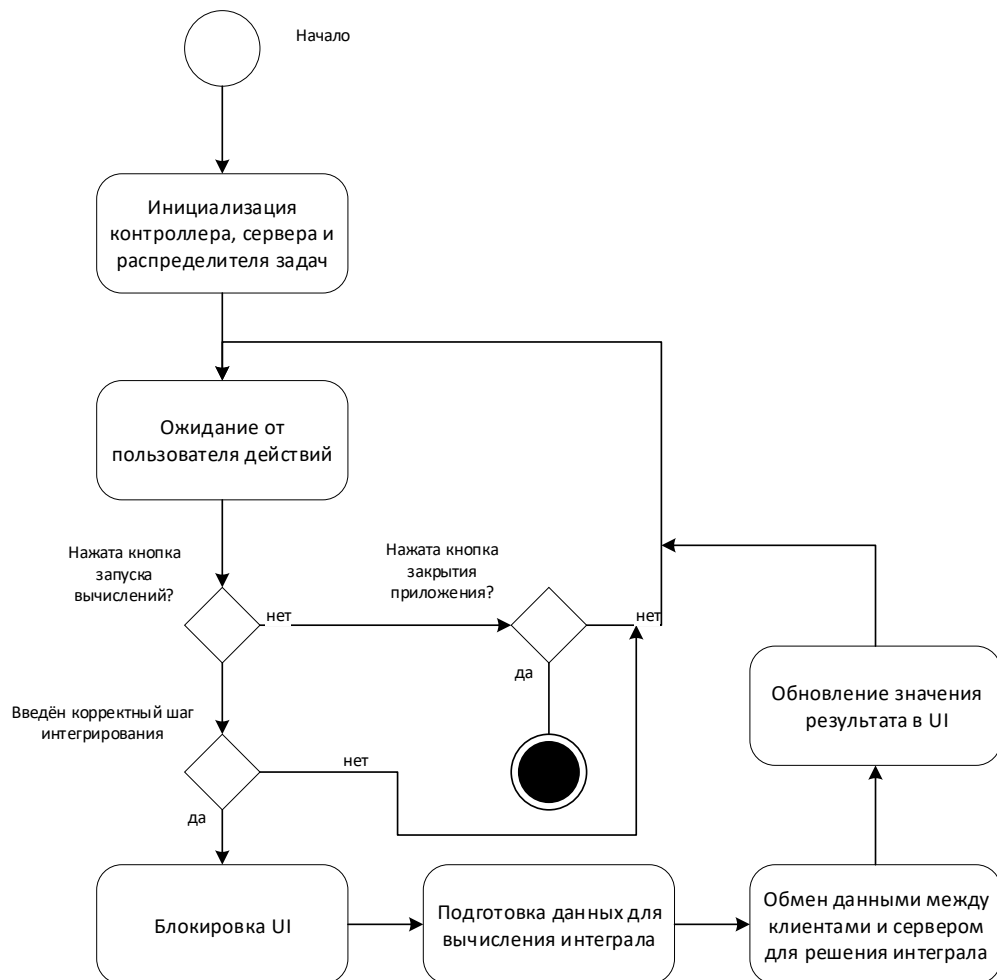


Рисунок 3. Диаграмма активностей для серверного приложения

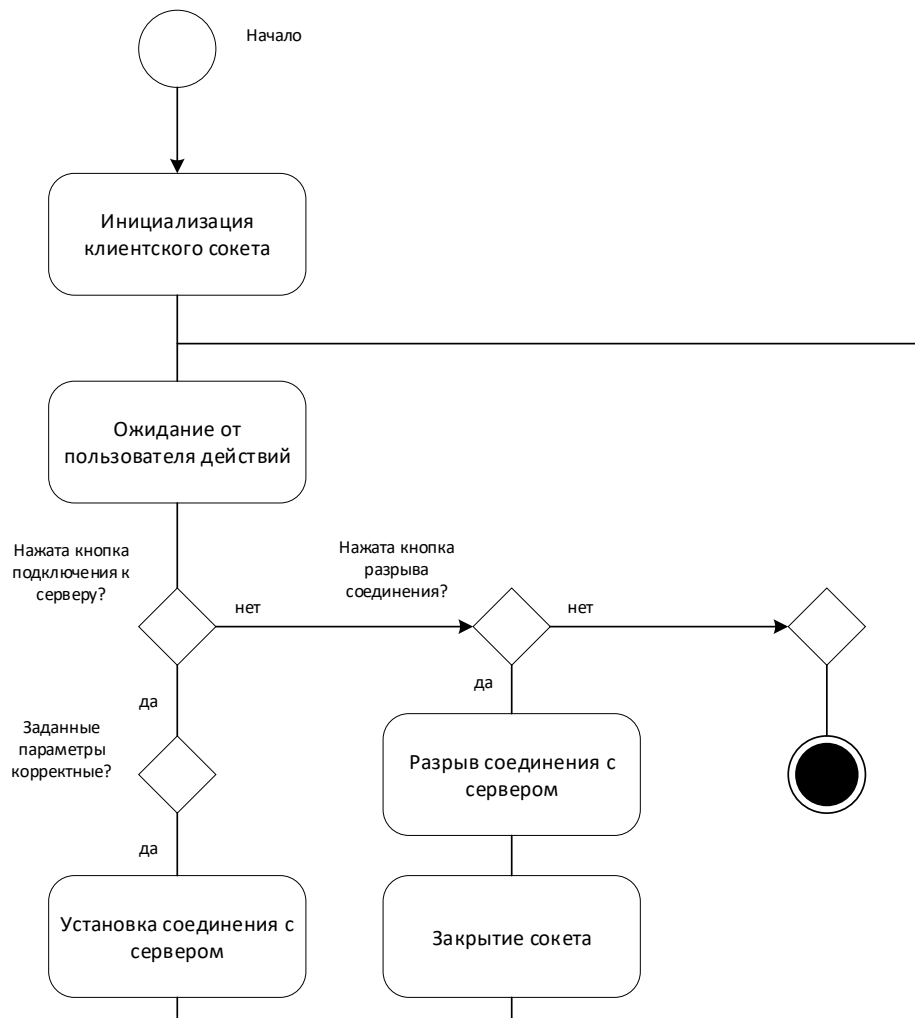


Рисунок 4. Диаграмма активностей для клиентского приложения

С использованием библиотеки WinSock был реализован клиент-серверный код с поддержкой соединения по протоколу TCP/IP. Программный сокет создан с использованием функции `socket()` и структуры `SOCK_ADDR`, содержащей настройки серверного сокета.

Рабочая логика программа была написана в виде совокупности классов, обменивающихся информацией через сигналы и слоты. Был написан класс для общения с UI-частью программы (Controller), классы для обеспечения связи с клиентами (Server и Connection), класс распределения задач (Tasker). На клиентской стороне также был создан класс Controller для связи с сервером.

Далее был реализован алгоритм вычисления определенного интеграла $\int \sin(x^2)$ методом правых прямоугольников с учетом требования многопоточности. Затем был добавлен механизм ограничения предельно

допустимого уровня использования процессорного времени. Реализация вычисления определённого интеграла была выполнена с использованием QFuture – отдельных потоков исполнения программы. Многопоточность вычислений позволила эффективно использовать имеющиеся вычислительные ресурсы ЭВМ пользователя и ускорить время выполнения программы.

Динамическое отключение/подключение клиентов реализовано на событийной основе signal-slot. Ожидание новых соединений было выделено в отдельную нить, сигнализация о начале вычислений – асинхронная.

Полученная диаграмма последовательностей программы представлена ниже (Рисунок 5).

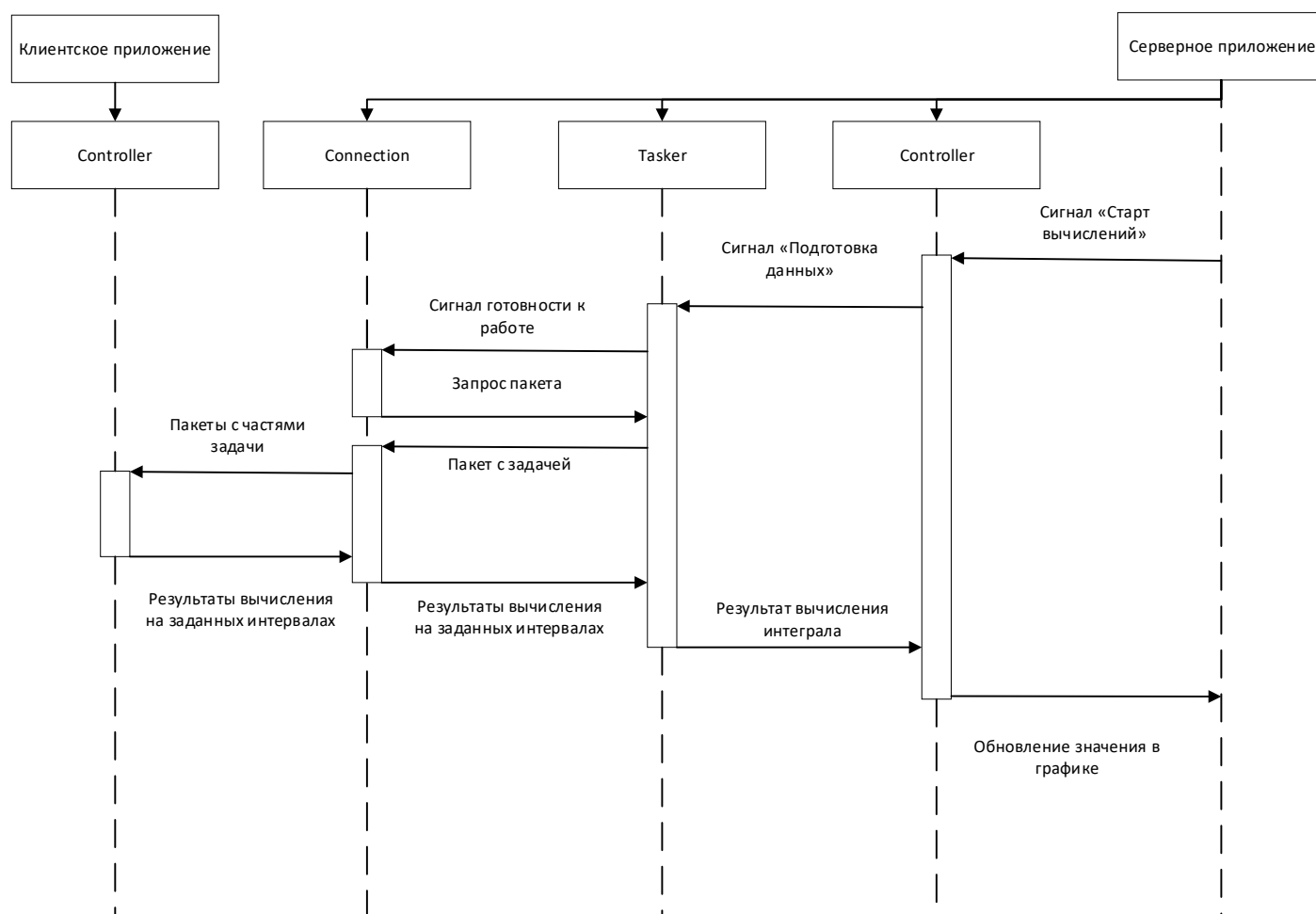


Рисунок 5. Диаграмма последовательностей клиент-серверной программы

Заключение

При выполнении данной курсовой работы были получены навыки разработки параллельной программы, усвоены механизмы синхронизации. Получены базовые навыки работы с библиотекой WinSock и организации обмена данных по протоколу TCP/IP.

В рамках выполнения курсового проекта было создано клиент-серверное приложение, позволяющее производить расчет значения определенного интеграла. Пользовательский интерфейс интуитивно понятен и прост. Имеется возможность ограничения потребления клиентами ресурсов ЦП.

В дальнейшем программу можно улучшить различными способами в соответствии с потребностями пользователя. Можно добавить возможность вводить другие подынтегральные функции, а также менять диапазон вычислений.

Список используемых источников

1. MSDN
2. Антонов А. С. Технологии параллельного программирования MPI и OpenMP – Изд. МГУ, 2012.
3. Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы. – СПб., Питер, 2007.
4. Хортон А. Visual C++ 2010. Полный курс, Издательство «Вильямс», 2011.
5. Определенный интеграл онлайн [Электронный ресурс]. – Электрон.дан. – URL: https://math24.biz/integral_def – 27.04.2018