

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И
ОПТИКИ**

Факультет Информационных технологий и программирования

Направление подготовки 01.03.02 Прикладная математика и информатика

ОТЧЕТ

по производственной практике

Тема задания: Реализация симуляции квантового компьютера и алгоритма Variational Quantum Eigensolver с использованием CMA-ES для языка Python

Студент Самарин Артур Олегович, группа № M3437

Руководитель практики от университета: Шалыто А. А., профессор ФИТиП

Ответственный за практику от университета: Нигматуллин Н.Г., тьютор ФИТиП

Практика пройдена с оценкой _____

Дата _____

**Санкт-Петербург
2019**

1. Цели и задачи практики

Цели практики: реализация необходимых компонентов для дальнейшей разработки алгоритмов с участием квантового компьютера.

Задачи практики: реализация наиболее производительной реализации симуляции квантового компьютера, реализация алгоритма VQE для языка Python 3 с использованием численного алгоритма оптимизации CMA-ES.

2. Сведения об организации

Университет ИТМО (полное наименование — «Федеральное государственное автономное образовательное учреждение высшего образования „Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики“») — высшее учебное заведение Санкт-Петербурга, один из национальных исследовательских университетов России.

3. Занимаемая должность

Занимаемая должность – программист.

4. Цели проекта

Цели проекта – разработка компонентов, реализующих алгоритмы с использованием квантового компьютера (или его симуляции), результаты проекта будут использоваться для исследования более сложных алгоритмов с участием квантового компьютера.

5. Используемые технологии

В ходе практики использовались современные технологии:

1. Язык программирования Python 3.
2. Язык программирования C++.
3. Библиотека Qutip (работа с матрицами, соответствующими волновым функциям и операторам).

4. Библиотека Qiskit (симуляция квантового компьютера для Python).
5. Библиотека QuEST (симуляция квантового компьютера для C/C++).
6. Библиотека Eigen 3 (работа с объектами линейной алгебры для C++).
7. Библиотека libcmaes (алгоритм численной оптимизации CMA-ES для языка C++)

6. Описание выполненного проекта

Было проведено сравнение производительности существующих решений для симуляции квантового компьютера. Производительность оценивалась по числу операций вычисления результирующей волновой функции некоторой фиксированной квантовой схемы в единицу времени. Результаты измерений приведены в таблицах 1 (для схемы на 4 кубита, рисунок 1) и 2 (для схемы на 8 кубит, рисунок 2).

Таблица 1: Производительность симуляции 4-кубитной квантовой схемы

Подход	Производительность симуляции (ед/с)
QuEST (C)	140 000
Биндинги QuEST для Python	1300
Qutip (Python)	61
Qiskit (Python)	35

Таблица 2: Производительность симуляции 8-кубитной квантовой схемы

Подход	Производительность симуляции (ед/с)
QuEST (C)	18 000
Биндинги QuEST для Python	500
Qutip (Python)	60
Qiskit (Python)	33

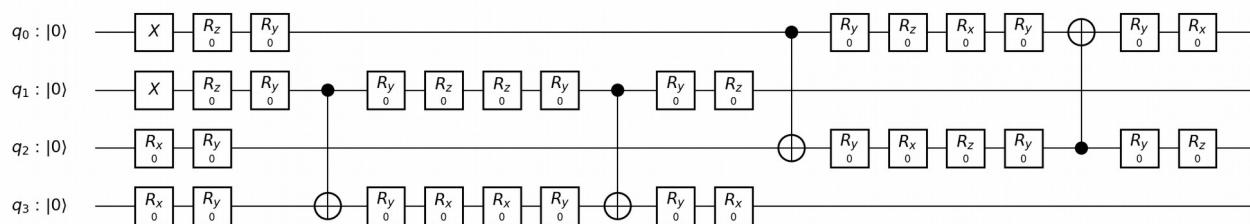


Рисунок 1: Схема для 4 кубит, на которой исследуется производительность симуляции

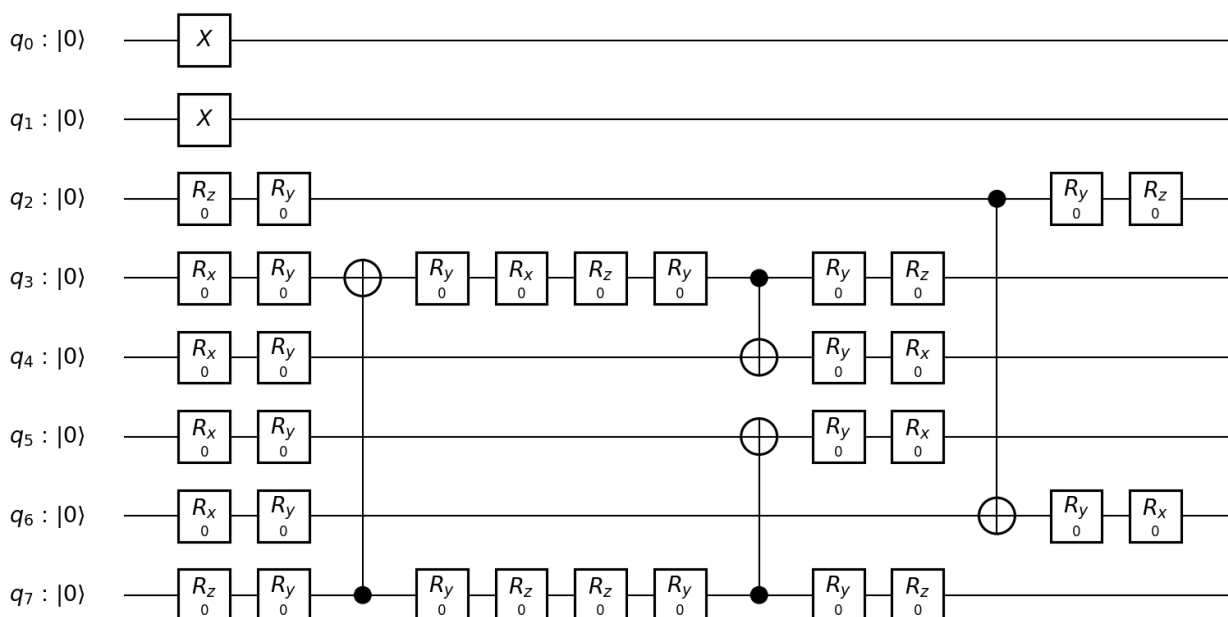


Рисунок 2: Схема для 8 кубит, на которой исследуется производительность симуляции

Разница в производительности между существующими решениями для языка Python и решением QuEST на языке C составляет более одного порядка. Кроме того, стоит заметить, что производительность Qutip и Qiskit слабо поменялась от изменения числа кубит. Вероятно, это связано с тем, что подходы имеют медленно работающую часть, время работы которой не зависит от числа кубит. Такой частью может быть, например, обработка схемы. Из этого следует, что для получения хорошей производительности алгоритма VQE следует реализовывать его на C++ с использованием QuEST напрямую, так как при реализации VQE на Python с использованием одной из библиотек для Python производительность упадет более, чем на порядок.

На языке C++ был реализован алгоритм VQE с использованием библиотеки `libctaes` для численной оптимизации. Так как требованием к проекту было использование разработанного компонента из языка Python, было разработано расширение Python, предоставляющее доступ к разработанной реализации VQE. Для разработки расширения использовалась библиотека `boost::python`, значительно упрощающая написание расширений Python на языке C++. Python-интерфейс разработанного компонента имеет следующие объекты:

1. Класс `Gates` с методами `gx`, `rz`, `cnot`, `block_a`, возвращающих экземпляры одиночек
2. Класс `QCircuit` с методами `num_parameters`, `set_parameters`, `add_gate`
3. Класс `VqeResult` с результатами работы VQE
4. Класс `Vqe` с методов `optimize`

Измерения производительности разработанной реализации, а также производительности альтернативных решений, приведены в таблице 4. Полученная производительность позволяет проводить эксперименты с методами, использующими VQE.

Таблица 3: Сравнение производительности реализаций VQE для 4-кубитной схемы (рисунок 1)

Подход		Производительность VQE (ед/с)
VQE	Симуляция	
СМА-ES на C++	QuEST (C)	31
СМА-ES на Python	Биндинги QuEST для Python	0,08
СМА-ES на Python	Qutip (Python)	0,03

Таблица 4: Сравнение производительности реализаций VQE для 8-кубитной схемы (рисунок 2)

Подход		Производительность VQE (ед/с)
VQE	Симуляция	
СМА-ES на C++	QuEST (C)	10
СМА-ES на Python	Биндинги QuEST для Python	0,09
СМА-ES на Python	Qutip (Python)	0,03

7. Выводы

Выполненная реализация имеет производительность, значительно превышающую производительность существующих решений. Это можно использовать для быстрого производства экспериментов при разработке новых алгоритмов с участием квантового компьютера.

8. Список источников

1. Cao Y. [и др.]. Quantum Chemistry in the Age of Quantum Computing // arXiv:1812.09976 [quant-ph]. 2018.
2. McClean J.R. [и др.]. The theory of variational hybrid quantum-classical algorithms // New Journal of Physics. 2016. № 2 (18). С. 023023.
3. Nielsen M.A., Chuang I.L. Quantum Computation and Quantum Information: 10th Anniversary Edition / M.A. Nielsen, I.L. Chuang, Anniversary edition, Cambridge ; New York: Cambridge University Press, 2011. 702 с.