

УДК 512.6, 517.9, 519.6

## ЭФФЕКТИВНОЕ МОДЕЛИРОВАНИЕ КВАНТОВЫХ АЛГОРИТМОВ НА СИМУЛЯТОРАХ КЛАССИЧЕСКОЙ АРХИТЕКТУРЫ

**Зрелов Петр Валентинович<sup>1</sup>, Иванцова Ольга Владимировна<sup>2</sup>, Кореньков Владимир Васильевич<sup>3</sup>, Рябов Никита Владимирович<sup>4</sup>, Ульянов Сергей Викторович<sup>5</sup>**

<sup>1</sup>Кандидат физико-математических наук, начальник отдела;  
Объединенный институт ядерных исследований,  
141980, Московская обл., г. Дубна, ул. Жолио-Кюри, д. 6;  
Ведущий научный сотрудник научной лаборатории «Облачные технологии и аналитика Больших данных»;  
Российский экономический университет имени Г.В. Плеханова;  
117997, Россия, г. Москва, Стремянный пер., 36;  
e-mail: zrelov@jinr.ru.

<sup>2</sup>Научный сотрудник;  
Объединенный институт ядерных исследований;  
141980, Московская обл., Дубна, Жолио-Кюри, 6;  
Старший преподаватель;  
Государственный университет «Дубна»;  
141980, Россия, Московская обл., г. Дубна, ул. Университетская, 19;  
e-mail: ivancova@jinr.ru.

<sup>3</sup>Директор лаборатории;  
Объединенный институт ядерных исследований;  
141980, Московская обл., г. Дубна, ул. Жолио-Кюри, д. 6;  
Доктор технических наук, заведующий кафедрой;  
Государственный университет «Дубна»;  
141980, Московская обл., г. Дубна, ул. Университетская, 19;  
Заведующий научной лабораторией «Облачные технологии и аналитика Больших данных»;  
Российский экономический университет имени Г.В. Плеханова;  
117997, Россия, г. Москва, Стремянный пер., 36;  
e-mail: korenkov@cv.jinr.ru.

<sup>4</sup>Аспирант;  
Государственный университет «Дубна»;  
141980, Московская обл., г. Дубна, ул. Университетская, 19;  
e-mail: ryabov\_nv95@mail.ru.

<sup>5</sup>Доктор физико-математических наук, профессор;  
Государственный университет «Дубна»;  
141980, Московская обл., г. Дубна, ул. Университетская, 19;  
Ведущий научный сотрудник;  
Объединенный институт ядерных исследований;  
141980, Московская обл., г. Дубна, ул. Жолио-Кюри, д. 6;  
e-mail: uyanovsv46\_46@mail.ru.

В представленной статье описывается методология применения симуляторов квантовых алгоритмов и реализация тестового и поискового квантовых алгоритмов на симуляторах с использованием классической архитектуры.

**Ключевые слова:** Квантовые алгоритмы поиска, квантовые схемы, квантовые симуляторы.

### Для цитирования:

Эффективное моделирование квантовых алгоритмов на симуляторах классической архитектуры / П. В. Зрелов [ и др.] // Системный анализ в науке и образовании: сетевое научное издание. 2022. № 1. С. 42–54. URL : <http://sanse.ru/download/459>.

## EFFECTIVE SIMULATION OF QUANTUM ALGORITHMS ON SIMULATORS OF CLASSICAL ARCHITECTURE

Zrelov Petr V.<sup>1</sup>, Ivancova Olga V.<sup>2</sup>, Korenkov Vladimir V.<sup>3</sup>, Ryabov Nikita V.<sup>4</sup>, Ulyanov Sergey V.<sup>5</sup>

<sup>1</sup>PhD in Physico-mathematical sciences, Department head;  
Joint Institute for Nuclear Research,  
6 Joliot-Curie Str., Dubna, Moscow region, 141980, Russia;  
Leading Researcher at the Scientific Laboratory «Cloud Technologies and Big Data Analytics»;  
Plekhanov Russian University of Economics;  
36 Stremyanny lane, Moscow, 117997, Russia;  
e-mail: zrelov@jinr.ru.

<sup>2</sup>Senior researcher;  
Joint Institute for Nuclear Research;  
6 Joliot-Curie Str., Dubna, Moscow region, 141980, Russia;  
Senior teacher;  
Dubna State University;  
19 Universitetskaya str., Dubna, Moscow region, 141980, Russia;  
e-mail: o\_ivancova@jinr.ru.

<sup>3</sup>Laboratory Director;  
Joint Institute for Nuclear Research;  
6 Joliot-Curie Str., Dubna, Moscow region, 141980, Russia;  
Grand PhD in engineering, head of the Department;  
Dubna State University;  
19 Universitetskaya Str., Dubna, Moscow region, 141980, Russia;  
Head of the Scientific Laboratory «Cloud Technologies and Big Data Analytics»;  
Plekhanov Russian University of Economics;  
36 Stremyanny lane, Moscow, 117997, Russia;  
e-mail: korenkov@cv.jinr.ru.

<sup>4</sup>PhD Student;  
Dubna State University;  
19 Universitetskaya str., Dubna, Moscow region, 141980, Russia;  
e-mail: ryabov\_nv95@mail.ru.

<sup>5</sup>Grand PhD in Physical and Mathematical Sciences, professor;  
Dubna State University;  
19 Universitetskaya Str., Dubna, Moscow region, 141980, Russia;  
Leading Researcher;  
Joint Institute for Nuclear Research;  
6 Joliot-Curie Str., Dubna, Moscow region, 141980, Russia;  
e-mail: ulyanovsv46\_46@mail.ru.

*The article describes the methodology of applying simulators of quantum algorithms and the implementation of test and search quantum algorithms on simulators using classical architecture*

**Keywords:** Quantum search algorithms, quantum circuits, quantum simulators.

### **For citation:**

Zrelov Petr V. Effective simulation of quantum algorithms on simulators of classical architecture. System Analysis in Science and Education, 2021;(1): 42–54(In Russ). Available from: <http://sanse.ru/download/459>.

## Введение

В настоящее время квантовое моделирование на классическом компьютере представляет особый интерес, поскольку вычисления на современных квантовых компьютерах имеют ограничения в виде небольшого числа кубитов, наличия ошибок и отсутствия эффективного способа коррекции этих ошибок. Вследствие этих ограничений использование универсальных квантовых компьютеров для решения практически значимых задач вероятно станет эффективным не ранее, чем через 10 или даже 20 лет. Однако квантовые симуляторы на основе различных архитектур (атомных, молекулярных, оптических) могут быть полезны уже лет через 5 [1]. И на данный момент, судя по всему, это направление является наиболее важным. Существуют реализации от *Google*, *IBM*, *Rigetti*, *Intel*, *D-Wave* [2-6]. Одновременно развивается направление, связанное с программными квантовыми симуляторами для вычислений на компьютерах классической архитектуры с использованием *CPU* и *GPU* [7].

В настоящей работе исследуется эффективность использования квантовых программных симуляторов для квантовых алгоритмов (КА). Для решения этой задачи были выбраны симуляторы *QuEST* [8] и *Qiskit* [9], позволяющие использовать как *CPU* (например, с использованием библиотеки для параллельного программирования *OpenMP*), так и графические вычислители *GPU*. Эксперименты проводились на суперкомпьютере «Говорун» [10] в Лаборатории информационных технологий им. М. Г. Мещерякова Объединенного института ядерных исследований (ЛИТ ОИЯИ). Реализация квантовых алгоритмов на симуляторах с использованием классической архитектуры требует экспоненциального масштабирования памяти и времени вычислений [11].

Проведенные вычисления показали, что при моделировании на *GPU* можно достичь большего ускорения по сравнению с расчетами на *CPU*.

Следует отметить, что наибольшего ускорения расчетов, по утверждению компании *NVIDIA*, можно достичь с использованием нескольких графических процессоров [12-14], что было продемонстрировано компанией с помощью нового симулятора *cuQuantum* – было смитировано 3375 кубитов на платформе *NVIDIA DGX SuperPod*™ [15, 16].

Настоящее исследование не включает результаты, полученные с использованием симулятора *cuQuantum*.

## Квантовые алгоритмы и их проектирование

Квантовые вычисления являются одним из наиболее бурно развивающихся направлений квантовых сквозных технологий. Квантовое моделирование обещает найти применение при решении целого ряда проблем физики конденсированных сред, физики высоких энергий, атомной физики, квантовой химии, космологии, машинного обучения, искусственного интеллекта [17, 18].

Квантовые компьютеры, доступные в настоящее время, называют «зашумленными» квантовыми компьютерами промежуточного масштаба<sup>1</sup>. Они реализуют, как правило, от 50 до 100 кубитов, вследствие присутствия шумов при вычислениях возникают ошибки, а возможность их коррекции отсутствует.

Наряду с созданием квантовых компьютеров, одним из важных направлений является направление, связанное с разработкой алгоритмов для этих компьютеров, основанных на принципах, отличных от принципов классических алгоритмов.

Существуют различные научные подходы к формализации преобразований входных данных для вычисления выходных данных с использованием квантовых ресурсов. Модель квантовых вычислений определяется базовыми элементами, на которые декомпозируются вычисления [19].

Основной моделью, имеющей практическое значение, является гейтовая модель [20, 21]. В этой модели вычисления разбиваются на последовательность из нескольких кубитных квантовых гейтов (вентилей). Это модель вычислений, в которой биты заменяются кубитами, а логические преобразования – конечным набором унитарных гейтов, которые могут аппроксимировать любую произвольную унитарную операцию.

---

<sup>1</sup> Noise intermediate-scale quantum devices (NISQ).

Алгоритм, выполненный в одной из моделей квантовых вычислений называется квантовым алгоритмом (КА). Разработка КА основана на принципах, отличных от принципов классических алгоритмов. Даже классические алгоритмы должны быть приведены в специальную (обратимую) форму, прежде чем их можно будет запустить на квантовом компьютере.

Визуализация типичного рабочего процесса квантового алгоритма на квантовом компьютере с гейтовой моделью показана на рисунке 1. Особенностью этого процесса является конструирование алгоритма с помощью так называемых квантовых схем.

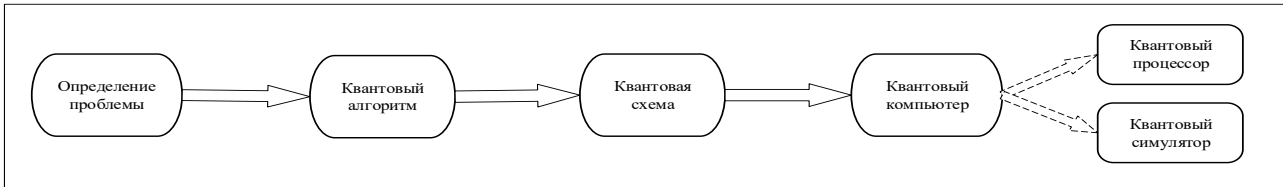


Рис. 1. Визуализация типичного рабочего процесса квантового алгоритма на квантовом компьютере с гейтовой моделью

Выбор КА<sup>2</sup> определяется характером задачи. На следующем шаге КА представляется в виде квантовой схемы, представляемой в виде набора квантовых гейтов<sup>3</sup>. Скомпилированная квантовая схема может быть выполнена либо на квантовом процессоре, либо реализована с помощью симулятора квантового компьютера.

Среди всего множества КА есть алгоритмы, имеющие явные преимущества перед известными классическими алгоритмами. Основной группой таких алгоритмов являются алгоритмы квантового поиска (АКП) [24, 25, 26]. Особенностью данных алгоритмов является достаточность 75% вероятности для выбранного решения (в отличие от классического метода достоверного поиска решения с вероятностью 1).

## Общая структура КА

На рисунке 2 показана общая структура квантовой схемы.

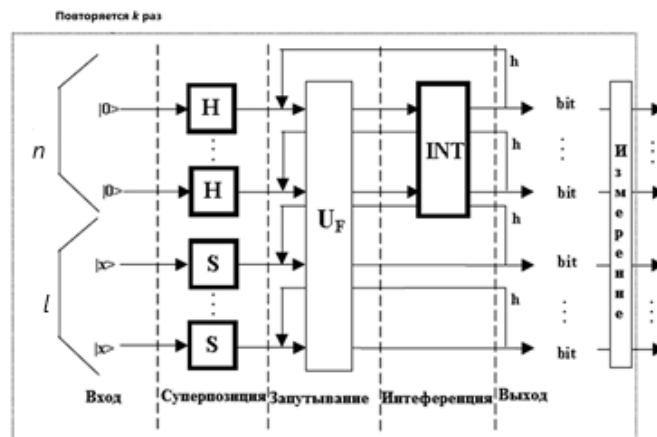


Рис. 2. Общая структура КА

Процесс проектирования КА включает матричную форму трех квантовых операторов: суперпозиции, квантовой запутанности (или квантового оракула) и интерференции. Оператором суперпозиции

<sup>2</sup> Веб-сайт Algorithm Zoo [22] предоставляет наиболее полный список существующих в настоящий момент квантовых алгоритмов.

<sup>3</sup> Квантовый гейт (или квантовый вентиль) представляет собой базовую квантовую схему, работающую на небольшом количестве кубитов. Гейты являются строительными блоками квантовых схем точно также, как классические логические вентили для обычных цифровых схем. Подразделяются на однокубитовые и многокубитовые. К первой группе относятся квантовые гейты тождественного преобразования (I), Адамара (H), Паули (X, Y, Z) и др. (см., например, [23]).

является тензорное произведение  $n$  операторов Адамара и  $l$  операторов тождественного преобразования. Полученный таким образом оператор действует на первые  $n$  кубитов (первый регистр), создавая их суперпозицию, и действует тождественно на последние  $l$  кубитов (второй регистр), оставляя его без изменений. Вид оператора запутывания  $U_F$  зависит от свойств исходной функции  $f$ . Оператор интерференции зависит от рассматриваемого алгоритма. Оператором интерференции могут выступать, например, квантовое преобразование Фурье, оператор Адамара и другие.

КА воздействует на исходный базисный вектор для генерации суперпозиции базисных векторов в качестве выходных данных. После создания суперпозиции выполняется измерение для извлечения информации об ответе. В квантовой механике измерение — это недетерминированная операция, которая выдает на выходе один из базисных векторов во входной суперпозиции. Вероятность того, что каждый базисный вектор будет результатом измерения, зависит от его амплитуды вероятности во входящей линейной комбинации.

Квантовые гейты и операция измерения составляют квантовый блок, который повторяется  $k$  раз, чтобы создать набор из  $n$  базисных векторов. Поскольку измерение является недетерминированной операцией, эти базисные векторы не обязательно будут идентичными, и каждый базисный вектор кодирует часть информации, необходимой для решения задачи. Последняя часть алгоритма включает в себя интерпретацию собранных базисных векторов, чтобы получить окончательный ответ для исходной задачи с некоторой вероятностью.

На рис. 3 показана типовая структура КА.

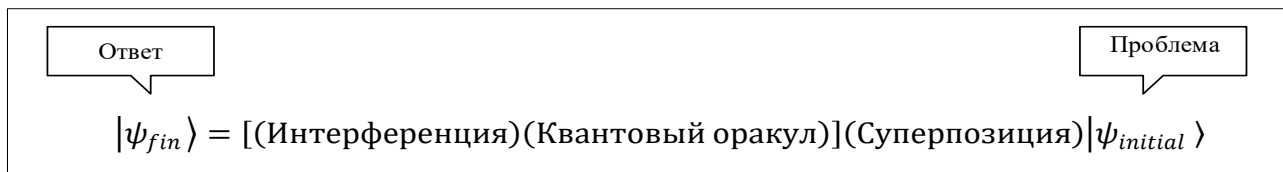


Рис. 3. Структура КА

Сформировать разные модели квантовых вычислений можно, выбирая различные виды оператора  $U_F$ .

В общем виде модель квантовых вычислений состоит из следующих этапов:

- приготовление начального (классического или квантового) состояния  $|\psi_{initial}\rangle$ ;
- выполнение преобразования Адамара ( $H$ ) для начального состояния с целью формирования состояния суперпозиции;
- применение оператора запутанных состояний или оператора квантовой корреляции (квантового оракула) к состоянию суперпозиции;
- применение оператора интерференции;
- использование оператора измерения для результата квантовых вычислений  $|\psi_{fin}\rangle$ .

Процесс проектирования квантового алгоритма, как следует из рис. 2 и 3, состоит из матричной формы представления трех операторов [27]:  $Sup$  – суперпозиции,  $U_F$  – квантовой корреляции (запутывания) или квантовый оракул и  $Int$  – интерференции.

В общем виде структуру квантового алгоритма, представленного на рис. 3, можно описать следующим образом:

$$C = [(Int \otimes {}^n I) \cdot \{U_F\}]^{h+1} \cdot [{}^n H \otimes {}^m S], \quad (1)$$

где  $I$  – тождественный оператор, символ  $\otimes$  обозначает тензорное произведение,  $H$  – оператор Адамара,  $S$  обозначает  $I$  или  $H$  в зависимости от задачи. Основой квантового блока является квантовый гейт  $U_F$ , который зависит от свойств матрицы и применяется в алгоритме  $h+1$  раз. Первая часть процесса проектирования – это выбор типа оператора в формуле (1), зависящего от проблемы запутывания. Оператор суперпозиции большинства АКП может быть выражен следующим образом:

$$Sup = \left( \bigotimes_{i=1}^n H \right) \otimes \left( \bigotimes_{i=1}^m S \right),$$

где  $n$  и  $m$  – количество входов и выходов соответственно. Оператором может быть или оператор Адамара, или тождественный оператор в зависимости от алгоритма (см. таблицу 1).

Табл. 1. Параметры операторов суперпозиции и интерференции основных АКП

Алгоритм	Суперпозиция	$m$	Интерференция
Гровера	${}^n H \otimes H$	1	$D_n \otimes I$
Шора	${}^n H \otimes {}^n I$	$n$	$QFT_n \otimes {}^n I$

$QFT_n$  – квантовое преобразование Фурье,  $D_n$  – диффузия (инверсия относительно среднего).

## Вычислительные модели алгоритмов квантового поиска

Основная сложность моделирования КА на классическом компьютере состоит в том, что ядром КА является унитарная матрица, размерность которой возрастает экспоненциально при линейном увеличении размерности квантовой системы.

На текущий момент существуют различные подходы к разработке АКП [28, 29, 30, 31]. Авторы работы [32] выделяют среди них следующие:

1) Матричный подход, основанный на матричном представлении квантовых операторов. Этот подход является более стабильным и точным, но требует много памяти компьютера, что осложняет моделирование алгоритмов с большим числом кубитов.

2) Подход, основанный на алгоритмах, в которых элементы матрицы вычисляются по «требованию». Этот подход не требует выделения памяти компьютера для квантовых операторов, поскольку вычисляет каждый компонент только тогда, когда он требуется. Подход позволяет использовать немного больший размер входных данных по сравнению с матричным подходом. К недостаткам можно отнести необходимость хранения в памяти компьютера вектора состояний и необходимость дополнительного изучения структуры операторов.

3) Проблемно-ориентированный подход. Количество переменных, используемых для представления переменной состояния в этом подходе, является постоянным. Фиксированное число переменных для представления вектора состояния позволяет пересмотреть традиционную схему моделирования квантового поиска. Операции матричного произведения заменяются арифметическими операциями с фиксированным числом параметров независимо от числа кубитов.

В данной статье рассматриваются квантовые симуляторы *Qiskit* и *QuEST*, которые реализуют первый подход в этом списке, т.е. тот, что основан на представлении квантовых матричных операторов.

## Квантовый поиск Гровера

Задачу квантового поиска Гровера можно сформулировать следующим образом: из списка  $N$  предметов требуется определить один, удовлетворяющий какому-либо специфическому свойству.

На рисунке 4 приведена принципиальная схема алгоритма Гровера. Подробное описание алгоритма Гровера и его модификации приведены в [33-37].

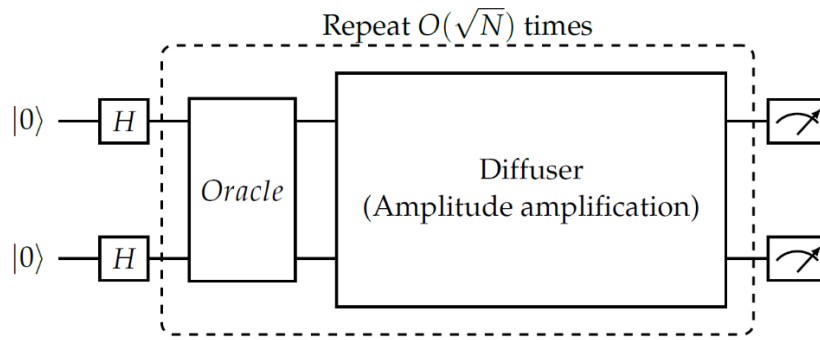


Рис. 4. Принципиальная схема алгоритма Гровера

## Выбор симуляторов

Среди большого количества квантовых программных симуляторов [38-40] одними из наиболее популярных на текущий момент являются:

- платформа *Qiskit* для квантовых вычислений с открытым исходным кодом. Разработана с учетом возможности расширения и поддержки исследований на квантовых системах *IBM*, основанных на сверхпроводящих кубитах, квантовых системах с ионной ловушкой *IonQ* и квантовых вычислительных устройствах *AQT*.
- Конструктор квантовых схем *Cirq*, который можно использовать с квантовыми устройствами *Google* и с другими квантовыми симуляторами.
- *PyQuil*, который можно использовать с квантовыми системами *Rigetti*.
- *ProjectQ* для поддержки исследований на квантовых системах *IBM*, *AQT*, *AWS Braket*, *IonQ*.

Использование перечисленных выше квантовых устройств либо достаточно дорого, либо ограничено небольшим количеством доступных кубитов.

В силу описанных ограничений запуск симуляции на классических компьютерах представляет особый интерес.

## Симуляторы на классической архитектуре с использованием CPU и GPU

Из внушительного списка квантовых симуляторов на классической архитектуре были выбраны те, которые позволяют использовать *OpenMP* и *GPU*.

В настоящее время на *CPU* для квантовых симуляторов доступны возможности запуска на многопроцессорных системах, причем некоторые из симуляторов способны работать и в распределённой вычислительной среде. Это открывает возможности симулировать системы с достаточно большим количеством кубитов.

Масштабируемость процесса симуляции на *GPU* развита меньше. Недавно *NVIDIA* представила собственный квантовый симулятор *cuQuantum*, который может работать на нескольких графических процессорах, связанных высокоскоростным интерфейсом *NVLink*. Один *GPU* с 32 Гб оперативной памяти может выполнить симуляцию 31 кубита. Вектор состояния для системы с большим количеством кубитов уже не может поместиться в память одного *GPU*, и для дальнейшего масштабирования на каждый новый кубит требуется удваивать количество используемых *GPU*. В таком случае, каждый *GPU* содержит часть вектора состояния и может применять гейты независимо. Если часть вектора состояния хранится на другом *GPU*, то применение гейта к полному вектору может потребовать передачи значительного объема данных, измеряемого гигабайтами, а это возможно только при наличии быстрого соединения между *GPU*. *NVLink* позволяет производить очень быстрый обмен данными, что невозможно при использовании нескольких *GPU*, подключенных к разным сокетам, или при использовании распределённых систем.

*QuEST* – один из первых симуляторов, использующих многопоточность, распределенные вычисления и GPU-ускорители [41]. *QuEST* – это симулятор, представляющий чистые квантовые состояния с помощью векторов состояний и смешанные состояния с помощью матриц плотности. Количество

памяти необходимое для выполнения вычисления для вектора состояния можно посчитать по формуле  $b \cdot 2^{\text{(qubits-29)}} (GiB)$ , где  $b = 8$  с двойной точностью. Следует отметить, что одновременное использование нескольких серверов или даже нескольких графических процессоров на одном сервере невозможно.

*Qiskit* – самый популярный квантовый симулятор на данный момент. Он позволяет проводить моделирование на *CPU* и *GPU*, а недавно появилась возможность использовать *OpenMP* и *CUDA* [42]. Он также предоставляет возможность использовать различные методы моделирования. Одним из них является *MPS*<sup>4</sup>. Квантовое состояние в этом подходе описывается следующим образом:

$$|\Psi\rangle = \sum_{i=0}^1 \dots \sum_{i_n=0}^1 c_{i_1 \dots i_n} |i_1\rangle \otimes \dots \otimes |i_n\rangle$$

где состояния  $|0\rangle$ ,  $|1\rangle$  принадлежат двумерному Гильбертовому пространству. Этот подход [43] дает преимущество, когда в квантовой цепи низкая степень запутанности. *MPS* может увеличить скорость и уменьшить количество требуемой памяти, однако, это возможно при условии, что не требуется выводить вектор состояния, потому что эта операция экспоненциально зависит от времени. Использование этого метода на простых квантовых схемах позволяет выполнять моделирование более чем на пятистах кубитах.

*QSim* [44] содержит полный симулятор вектора состояния Шрёдингера и гибридный симулятор Шрёдингера-Фейнмана. Позволяет запускать симуляцию на процессоре с использованием *OpenMP* и *GPU*. На момент публикации *cuQuantum* от *NVIDIA*, имеющий большой потенциал для моделирования, интегрировался в *QSim*. Квантовый симулятор *QSim* поддерживается *Google*, в работе [45] приведены результаты его сравнения с квантовым устройством-

## Аппаратное обеспечение

Как было упомянуто выше, эксперименты проводились на суперкомпьютере Говорун в ЛИТ ОИЯИ.

Эксперименты с графическим процессором проводились на *DGX-1* [46] с *8X NVIDIA Tesla® V100 16 GB/GPU, 512 GB DDR4 RDIMM, 4X 20-Core Intel® Xeon® E5-2698 v4 2.2 GHz*.

Моделирование с использованием центрального процессора проводилось на следующих конфигурациях:

- 8 серверов с *4X 72-Core Intel® Xeon Phi™ 7290 1.50 GHz, 96 GB DDR4*;
- 4 сервера с *4X 24-Core Intel® Xeon® Platinum 8268 2.90 GHz, 192 GB DDR4*.

## Моделирование квантовых алгоритмов на компьютерах классической архитектуры

На этапе тестирования симуляторов *QuEST* и *Qiskit* были произведены эксперименты с использованием от 2 до 31 кубита, чтобы получить надлежащий масштаб для наблюдения за результатами. Для анализа результатов рассмотрим диапазон от 22 до 30 кубитов. Для меньшего количества кубитов симуляция занимает незначительное время. Симуляции для большего количества кубитов ограничены памятью.

При разработке схемы тестового алгоритма мы воспользовались работой *Google* [47], произведя некоторые упрощения (см. рис. 5), и используя 100 повторений, а это значит, что к каждому кубиту в алгоритме применяется около 800 гейтов.

<sup>4</sup> MPS - Matrix product state



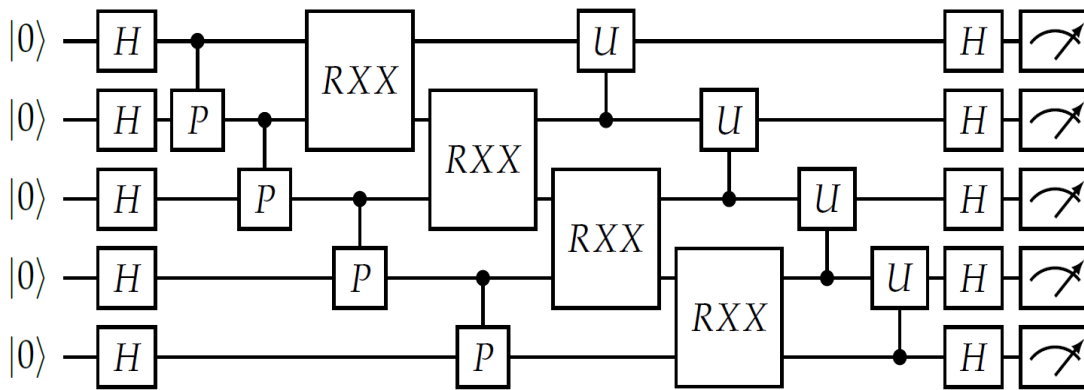


Рис. 5. Квантовая схема тестового алгоритма

Схема сконструирована таким образом, чтобы можно было корректно сравнить два симулятора, потому что не все квантовые гейты для более сложных задач в рассматриваемых симуляторах реализованы одинаково. Например, алгоритм Гровера может быть реализован на различных симуляторах по-разному, что не позволяет произвести объективное сравнение работы данного алгоритма на разных симуляторах.

На рисунке 6 представлены усредненные значения времени выполнения тестового алгоритма на обоих рассматриваемых симуляторах как на *CPU*, так и на *GPU*.

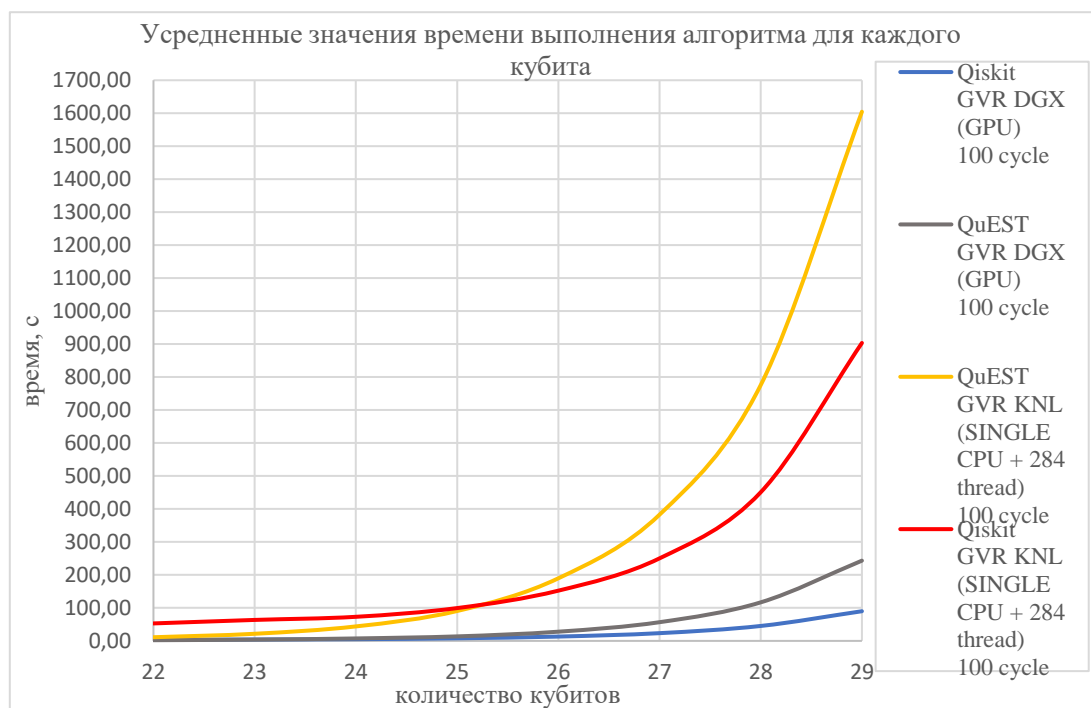


Рис. 6. Время выполнения тестового алгоритма на *QuEST* и *Qiskit* (*CPU* и *GPU*)

На обоих симуляторах на *GPU* по сравнению с *CPU* скорость выполнения алгоритма выше. Однако на *QuEST* нельзя произвести симуляцию более 29 кубит на *GPU* в силу ограничения памяти суперкомпьютера. Также стоит отметить, что *Qiskit* заметно быстрее *QuEST* на *CPU* со всей возможной параллелизацией.

Алгоритм Гровера (см. рис.4) также был реализован на обоих симуляторах (см. рис. 7).

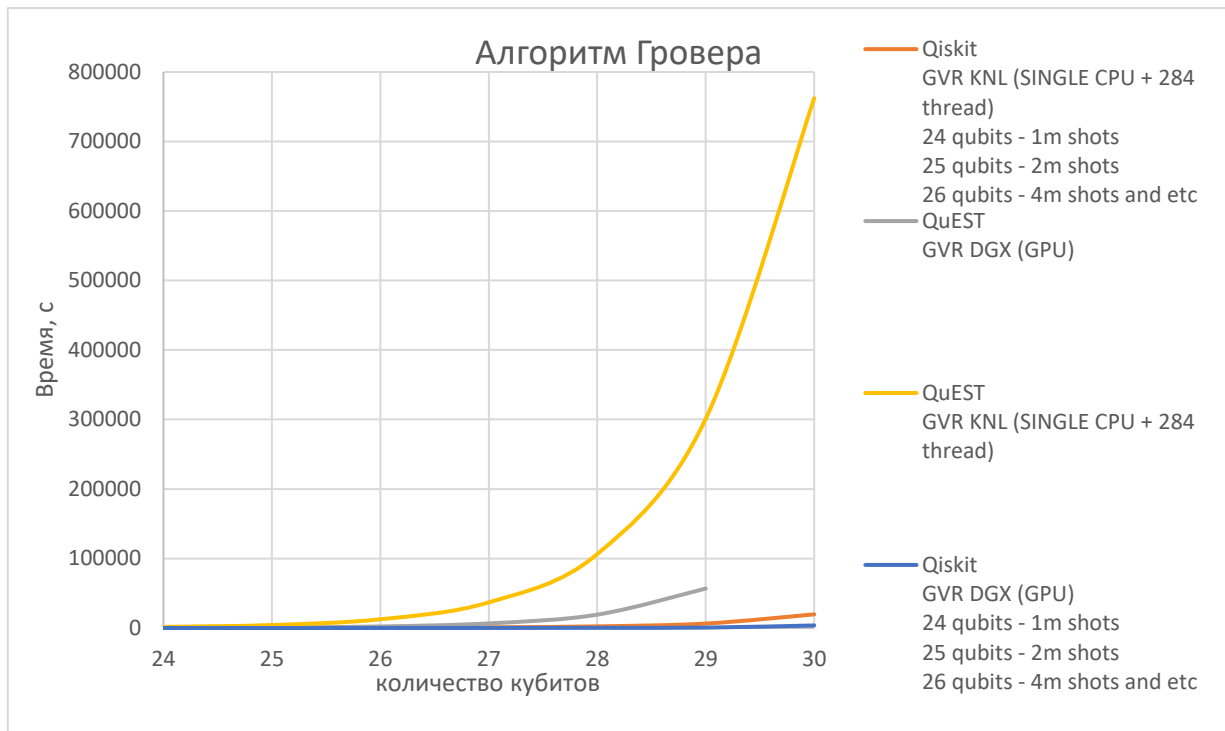


Рис. 7. Время выполнения алгоритма Гровера на QuEST и Qiskit (CPU и GPU)

На рисунке 7 представлены усредненные значения времени выполнения алгоритма Гровера на обоих рассматриваемых симуляторах как на CPU, так и на GPU. Для 24 кубит – 1 shots, для 25 – 2 shots, для 26 – 4 shots и т.д., shots – количество запусков алгоритма.

Основной результат: Наименьшие временные затраты при выполнении данного алгоритма показывает Qiskit, а скорость выполнения на GPU меньше, чем на CPU.

## Заключение

Моделирование квантовых алгоритмов на симуляторах с использованием классической архитектуры в эпоху NISQ представляется перспективным еще на достаточно долгий период, вероятно, до момента появления квантовых процессоров с количеством кубитов более  $10^5$  и корректируемыми ошибками. До этого момента разработка и тестирование КА на симуляторах с использованием классической архитектуры хотя и ограничено памятью компьютера или суперкомпьютера, но позволяет применять КА для решения ряда задач в самых различных областях: квантовой химии, развития нанотехнологий, сбора и обработки экспериментальных результатов для интеллектуального управления различными системами ускорителя в мега-сайнс проекте НИКА [48], квантового глубокого машинного обучения в робототехнике и др.

Выбор квантовых симуляторов для реализации вычислений на суперкомпьютере Говорун, позволяет приступить к формированию библиотеки КА для решения инженерных задач. Особое место в данной библиотеке отводится поисковому алгоритму Гровера, матричная форма которого была реализована на обоих симуляторах. В дальнейших планах – реализация быстрых модификаций алгоритма Гровера и других КА, формирующих ядро библиотеки-для решения инженерных задач.

Данная работа была проведена в рамках гранта Минобрнауки РФ №075-10-2020-117 «Сверхтяжелые ядра и атомы: пределы масс ядер и границы Периодической Таблицы Д. И. Менделеева».

## Список источников

1. Quantum Simulators: Architectures and Opportunities/ E. Altman [et al.] // PRX Quantum. 2021. vol. 2. No. 1. DOI: 10.1103/prxquantum.2.017003.
2. Google Research. Google Quantum Computing. URL: <https://resarch.google/teams/applied-science/>.

3. IBM Research. IBM Quantum Computing. URL: <https://www.ibm.com/quantum-computing/>.
4. Quantum Computing // Rigetti Computing. Rigetti & Co, Inc., 2020–2022. URL: <https://www.rigetti.com/>
5. Intel Quantum Computing // Intel Corporation. URL: <https://www.intel.com/content/www/us/en/research/quantum-computing.html>
6. D-Wave Quantum Computing // D-Wave Systems. D-Wave Systems Inc., 2022. URL: <https://www.dwavesys.com/>
7. List of QC simulators // Quantiki : Quantum Information Portal and Wiki. URL : <https://quantiki.org/wiki/list-qc-simulators/>
8. QuEST and High Performance Simulation of Quantum Computers / T. Jones, A. Brown, I. Bush, S. C. Benjamin. Scientific Reports. 2019. № 9. Article number: 10736. DOI: <https://doi.org/10.1038/s41598-019-47174-9>
9. Open-Source Quantum Development // Qiskit. URL: <https://qiskit.org/>
10. Суперкомпьютер «Говорун» // Платформа "Hybrilit". HybriLIT, 2022. URL: [http://hlit.jinr.ru/supercomputer\\_govorun/](http://hlit.jinr.ru/supercomputer_govorun/)
11. Quantum Computer Systems for Scientific Discovery / Y. Alexeev [et al.] // PRX Quantum. 2021. vol. 2. Article number: 017001. DOI: <https://doi.org/10.1103/PRXQuantum.2.017001>
12. Carrazza S., Cruz-Martinez J. VegasFlow: Accelerating Monte Carlo simulation across multiple hardware platforms // Computer Physics Communications. 2020. Vol. 254. P. 107376. DOI: 10.1016/j.cpc.2020.107376.
13. LaRose R. Distributed Memory Techniques for Classical Simulation of Quantum Circuits. 2018. arXiv: <https://arxiv.org/abs/1801.01037>.
14. Smelyanskiy M., Sawaya N. P. D., Aspuru-Guzik A. qHiPSTER: The Quantum High Performance Software Testing Environment. 2016. arXiv:1601.07195 [quant-ph].
15. cuQuantum. Accelerate Quantum Information Science // NVIDIA.Developer. NVIDIA Corporation, 2022. URL : <https://developer.nvidia.com/cuquantum-sdk>
16. cuQuantum SDK: A High-Performance Library for Accelerating Quantum Information Science / NVIDIA.Developer. NVIDIA Corporation, 2022. URL: <https://docs.nvidia.com/cuda/cuquantum/index.html>
17. Quantum Computing and Simulations for Energy Applications: Review and Perspective / Hari P. Paudel [et al.] // ACS Engineering. 2020. № 2(3). Pp. 151-196. DOI: <https://doi.org/10.1021/acsenengineering.1c00033>
18. Georgescu I. M., Ashhab S., Nori F. Quantum simulation. Rev. Mod. Phys. 2014 .Vol. 86, Iss. 1 Pp. 153–85. DOI : <https://doi.org/10.1103/RevModPhys.86.153> .
19. What is Quantum Computing? Top 18 Quantum Computing Companies // PAT RESEARCH : [веб-сайт]. PAT RESEARCH, 2013-2021. URL: <https://www.predictiveanalyticstoday.com/what-is-quantum-computing/>
20. Preskill J. Quantum Computing in the NISQ era and beyond // Quantum.2018. vol. 2. p. 79. DOI: 10.22331/q-2018-08-06-79.
21. Gyongyosi L., Imre S. Circuit Depth Reduction for Gate-Model Quantum Computers // Scientific Reports. 2020. Article number: 11229 (2020). DOI: <https://doi.org/10.1038/s41598-020-67014-5>
22. Jordan S. Quantum Algorithm Zoo : [веб-сайт]. URL: <https://quantumalgorithmzoo.org/>
23. Crooks G. E. Gates, States, and Circuits. Notes on the circuit model of quantum computation. Berkeley Institute for Theoretical Sciences, 2022. URL: [https://threeplusone.com/pubs/on\\_gates.pdf](https://threeplusone.com/pubs/on_gates.pdf)
24. Shor P. Algorithms for Quantum Computation: Discrete Logarithms and Factoring // Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on – IEEE, 1994. P. 124–134.

25. Grover L. K. A fast quantum mechanical algorithm for database search // Proceedings, 28th Annual ACM Symposium on the Theory of Computing, 1996.
26. Giri P. R., Korepin V. E. A review on quantum search algorithms // Quantum Information Processing. 2017. Vol. 16. № 12. Pp. 1-36. DOI: <https://doi.org/10.1007/s11128-017-1768-7>
27. Генетические и квантовые алгоритмы. Ч. 1. Инновационные модели в обучении / С. В. Ульянов [и др.] // Системный анализ в науке и образовании: сетевое научное издание. 2010. № 3. С. 226–253.
28. A Modified Quantum Search Algorithm / H. Mehri-Dehnavi [et al.] // International Journal of Theoretical Physics. 2018. Vol. 57. Pp. 3668–3681. DOI: <https://doi.org/10.1007/s10773-018-3880-6>
29. Ulyanov S. V., Litvintseva L. V., Ulyanov S. S. Quantum information and quantum computational intelligence: Design & classical simulation of quantum algorithm gates. Universita degli Studi di Milano: Polo Didattico e di Ricerca di Crema Publ. 2005. Vol. 80.
30. Quantum information and quantum computational intelligence: Classically efficient simulation of fast quantum algorithms (SW / HW Implementations) / S.V.Ulyanov [et al.] // [M] Note del Polo, Milan Univ, 2005, 79.
31. Tang X., Xu J., Duan B. A memory-efficient simulation method of grover's search algorithm // Computers, Materials & Continua. 2018. vol. 57, no.2. pp. 307–319. DOI: [doi:10.32604/cmc.2018.03693](https://doi.org/10.32604/cmc.2018.03693)
32. Quantum Software Engineering Toolkit. Quantum fast search algorithms. Quantum simulators on classical computers. Quantum control information models. Part I / O. V. Ivancova, V. V. Korenkov, S. V. Ulyanov, P. V. Zrellov.— M.: KURS, 2022. — 463 p.
33. Zalka, C.: Grover's Quantum Searching Algorithm Is Optimal. Phys. Rev. A 60(4), 2746 (1999).
34. Jones J.A., Mosca M., Hansen R. H. Implementation of a quantum search algorithm on a quantum computer. Nature 393, 344 (1998).
35. Implementation of grover's quantum search algorithm in a scalable system / K.A. Brickman [et al.] // Phys. Rev. 2005. A 72, 050306.
36. Quantum amplitude amplification and estimation/ G. Brassard, M. Boyer, P. Hoyer, A. Tapp // Contemp. Math. 2002. Vol. 305. Pp. 53.
37. Wang Y., Krstic P. S. Prospect of using Grover's search in the noisy-intermediate-scale quantum-computer era // Phys. Rev. A 102, 042609. Published 20 October 2020
38. Fingerhuth M., Babej T., Wittek P. Open source software in quantum computing. URL: <https://arxiv.org/pdf/1812.09167.pdf>
39. Lloyd S. Universal Quantum Simulators // Science. 1996. vol. 273, no. 5278, pp. 1073–1078. DOI: [10.1126/science.273.5278.1073](https://doi.org/10.1126/science.273.5278.1073).
40. Vidal G. Efficient Classical Simulation of Slightly Entangled Quantum Computations // Physical Review Letters. 2003. vol. 91, no. 14. DOI: [10.1103/physrevlett.91.147902](https://doi.org/10.1103/physrevlett.91.147902).
41. QuEST and High Performance Simulation of Quantum Computers / T. Jones, A. Brown, I. Bush, S. C. Benjamin // Scientific Reports. 2019. vol. 9, no. 1. DOI: [10.1038/s41598-019-47174-9](https://doi.org/10.1038/s41598-019-47174-9).
42. Running with Threadpool and DASK // Qiskit. URL : <https://qiskit.org/documentation/apidoc/parallel.html>
43. Schollwöck U. The density-matrix renormalization group in the age of matrix product states // Annals of Physics. 2011. Vol. 326, No. 1, Pp. 96–192. DOI: [10.1016/j.aop.2010.09.012](https://doi.org/10.1016/j.aop.2010.09.012).
44. qsim / Quantum AI team and collaborators // Zenodo, Sep. 2020. DOI: [10.5281/zenodo.4023103](https://doi.org/10.5281/zenodo.4023103).
45. Quantum Supremacy Is Both Closer and Farther than It Appears / I. L. Markov, A. Fatima, S. V. Isakov, S. Boixo. 2018.
46. NVIDIA DGX-1 // NVIDIA. NVIDIA Corporation, 2022. URL: <https://www.nvidia.com/en-us/data-center/dgx-1/>

47. Quantum Supremacy using a Programmable Superconducting Processor / F. Arute [et al.] // Nature. 2019. Vol. 574. Pp. 505–510. DOI: <https://doi.org/10.1038/s41586-019-1666-5>
48. NICA : Nuclotron-based Ion Collider fAcility / Объединенный институт ядерных исследований, 2016 - 2022 . URL: <https://nica.jinr.ru/>