

УДК 004.4

Полигон для квантовых вычислений на гетерогенной платформе HybridIT

Д. В. Беляков¹, А. А. Боголюбская¹, М. И. Зуев¹, Ю. Г. Палий^{1,2}, Д. В. Подгайный¹,
О. И. Стрельцова¹, Д. А. Янович¹

¹ Лаборатория информационных технологий,
Объединенный институт ядерных исследований,
ул. Жолио-Кюри 6, Дубна, Московская область, Россия, 141980

² Институт Прикладной Физики,
Молдавский государственный университет,
ул. Академическая 5, Кишинев, Республика Молдова, MD 2028

Email: zuevmax@jinr.ru

На ресурсах экосистемы ML/DL/HPC гетерогенной платформы HybridIT был развернут полигон для квантовых вычислений. Характеристики сервера позволяют проводить расчеты с использованием библиотек, поддерживающих параллельные вычисления как на центральных процессорах, так и на графических ускорителях. В качестве основы вычислительной среды выбран JupyterLab, что предоставляет возможность пользователям наглядно работать с квантовыми схемами и проводить расчеты в веб-браузере. На настоящий момент установлены симуляторы квантовых вычислений Cirq, Qiskit, PennyLane, Qutip. В качестве примера в работе приведены результаты расчетов задачи по поиску состояния с наименьшей энергией в модели Изинга с продольным магнитным полем с использованием квантового аппроксимационного оптимизационного алгоритма (QAOA). Показана зависимость эффективности проведения расчетов от конфигурации используемых вычислительных ресурсов.

Ключевые слова: квантовые симуляторы, информационные технологии, высокопроизводительные вычисления, Python, JupyterLab

1. Введение

Гетерогенная платформа HybridIT [1] развивается как платформа для обеспечения высокопроизводительных и параллельных вычислений, хранения и обработки данных и использования пакетов прикладных программ. Для этого была создана единая программно-информационная среда, объединяющая учебно-тестовый полигон и суперкомпьютер «Говорун». Для проведения расчетов в batch-режиме был реализован механизм запуска задач через планировщик SLURM. Однако, для решения задач в области искусственного интеллекта, машинного и глубокого обучения возникла потребность в упрощении процесса разработки и отладки алгоритмов, в частности, с использованием веб-интерфейсов. Наиболее широко используемым решением для подобных задач является использование среды JupyterLab. Она позволяет создавать модели, использовать различные библиотеки и проводить расчеты в интерактивном режиме, сразу же просматривая результаты выполнения кода. Для обеспечения таких возможностей на ресурсах платформы HybridIT была создана экосистема ML/DL/HPC [2].

2. Полигон для квантовых вычислений

Со временем пользователи гетерогенной платформы все чаще стали решать задачи, связанные с разработкой квантовых алгоритмов и применением симуляторов квантовых вычислений. Для этих целей был организован полигон для квантовых вычислений. Для работы на полигоне для квантовых вычислений доступны следующие варианты.

- Работа через batch-систему (Рис. 1, слева).
Симуляторы квантовых вычислений и все необходимые библиотеки установлены в сетевую файловую систему CVMFS, для определения необходимых переменных окружения в рабочей сессии пользователя используется утилита Modules. Запуск задач осуществляется через планировщик SLURM. Из преимуществ этой реализации можно отметить возможность проведения многоузловых расчетов с применением технологии MPI и использования ресурсов всей гетерогенной платформы, в том числе суперкомпьютера «Говорун».
- Работа через веб-интерфейс (Рис. 1, справа).

В последнее время все больше квантовых симуляторов получает реализацию на языке Python. Для упрощения работы с ними было реализовано расширение полигона для квантовых вычислений, позволяющее проводить исследования с использованием квантовых симуляторов в среде Jupyter. Это предоставило пользователям возможность наглядно работать в браузере с квантовыми схемами и проводить расчеты. Полигон для квантовых вычислений развернут на базе сервера, позволяющего проводить расчеты как на центральных процессорах, так и графических ускорителях. Веб-интерфейсом был выбран JupyterLab – многопользовательская версия Jupyter, на котором установлены в виде отдельных модулей симуляторы квантовых вычислений. При выборе квантового симулятора открывается Jupyter Notebook с библиотеками и настроенными переменными окружения. Значимым преимуществом работы в Jupyter Notebook является возможность наглядно вести разработку алгоритмов, визуализировать квантовые схемы, а обширные материалы в открытом доступе на Python позволяют существенно ускорить исследования.

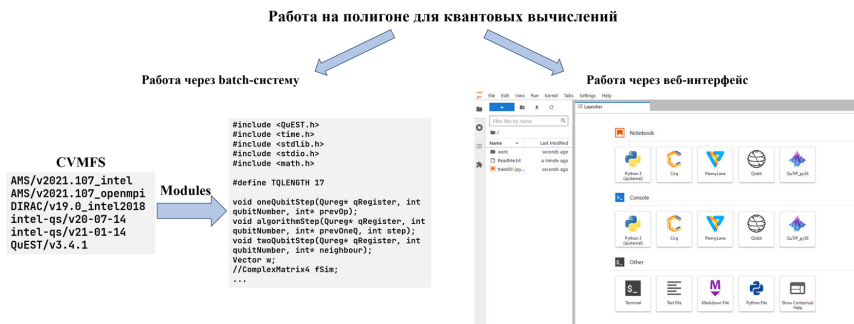


Рис. 1. Полигон для квантовых вычислений

3. Тестовая задача

Для тестирования был выбран квантовый симулятор QuEST [3]. С его помощью проверялась эффективность проведения расчетов как на системах с общей и распределенной памятью, так и на графических ускорителях с использованием технологии CUDA. Результаты тестирования представлены на Рис. 2 и Рис. 3.

На Рис. 2 представлен график зависимости времени работы квантового симулятора от количества кубитов для разного числа потоков m . Из графика видно, что удалось достичь размеров квантовой системы в 33 кубита. На Рис. 3 представлен график

времени работы квантового симулятора с различным числом кубитов на различных графических ускорителях.

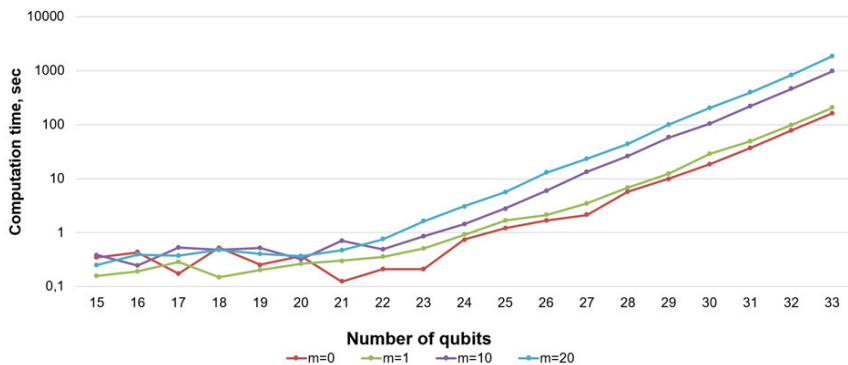


Рис. 2. Зависимость времени работы квантового симулятора от числа кубитов при разном числе потоков

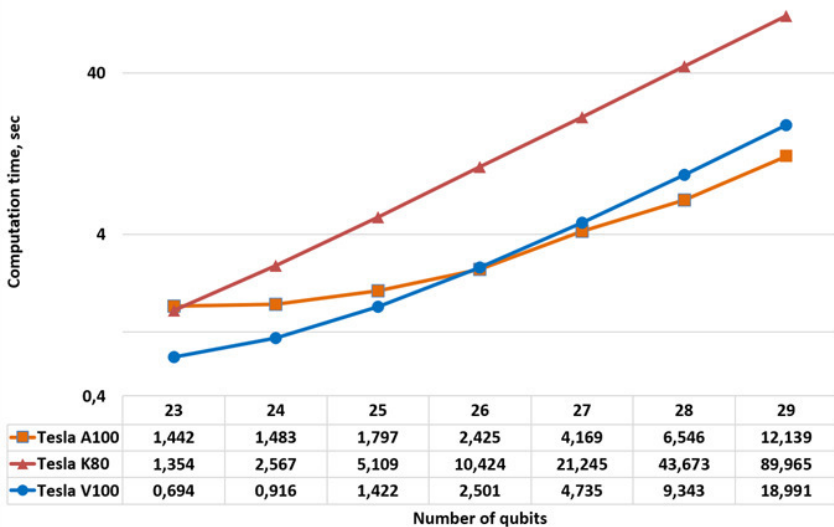


Рис. 3. Зависимость времени работы квантового симулятора от числа кубитов на различных графических ускорителях

4. Пример пользовательской задачи

Задача поиска состояния с наименьшей энергией в модели Изинга в продольном магнитном поле с использованием квантового аппроксимационного оптимизационного алгоритма (QAOA) [5] решалась на квантовом полигоне сначала с помощью симулятора Cirq [4], а на последующих этапах – симулятора qsim [8], в среде JupyterLab. Cirq – это библиотека на языке Python; она представляет собой среду для квантовых вычислений – позволяет создавать, преобразовывать, оптимизировать и эффективно моделировать схемы до 30 кубитов на квантовых симуляторах и квантовых компьютерах. Интегрированный с Cirq оптимизированный симулятор qsim, написанный на языке C++, использует инструкции SIMD для векторизации, OpenMP для многопоточности в расчетах на CPU, а также позволяет проводить вычисления на GPU с использованием библиотеки cuStateVec [9]. Это даёт возможность моделировать квантовые схемы до 40 кубитов.

Отметим, что алгоритм QAOA признан наиболее перспективным среди гибридных алгоритмов [6]. В нем роль квантового компьютера заключается только в построении вариационного анзаца $|x(\alpha)\rangle$ волновой функции с параметрами $\alpha = (\alpha_1, \dots, \alpha_n)$ и измерении квантовых средних, например, энергии E как среднего для гамильтониана \mathcal{H} :

$$E(\alpha) = \langle x(\alpha) | \mathcal{H} | x(\alpha) \rangle.$$

На классическом компьютере происходит процесс оптимизации параметров для достижения минимального значения среднего $E(\alpha)$.

В случае взаимодействия ближайших соседей, то есть спин-спинового взаимодействия с константой J , и взаимодействия спинов с внешним магнитным полем h , гамильтониан \mathcal{H} для рассматриваемой модели Изинга имеет вид

$$\mathcal{H}(Z) = -J \sum_{(i,j)} Z^{(i)} Z^{(j)} - h \sum_i Z^{(i)},$$

где (i, j) – это множество пар соседних спинов, а во втором слагаемом сумма идет по всем узлам решетки. Многокубитные операторы $Z^{(i)}$ представляют собой тензорные произведения

$$Z^{(i)} = \mathbb{1} \otimes \dots \otimes Z \otimes \dots \otimes \mathbb{1},$$

где оператор Паули Z стоит на i -м месте, то есть действует на i -й кубит. Представление модели Изинга на квантовом компьютере рассмотрено в работе [7].

Вариационный анзац $|x(\alpha)\rangle$ волновой функции в алгоритме QAOA, включающий p слоев, имеет вид:

$$|\psi(\gamma, \beta)\rangle = \underbrace{U(\beta_p, B)U(\gamma_p, \mathcal{H})}_{p} \dots \underbrace{U(\beta_1, B)U(\gamma_1, \mathcal{H})}_1 H^{\otimes n} |0\rangle^{\otimes n},$$

где H обозначает оператор Адамара, а операторы $U_k(\gamma_k, \mathcal{H}) = e^{i\pi\gamma_k\mathcal{H}}$ и $U_k(\beta_k, B) = e^{i\pi\beta_k B}$, $B = \sum_{j=1}^n X^{(j)}$ зависят от вариационных параметров $\gamma_k, \beta_k, k = 1, \dots, p$. С ростом числа слоев минимальное среднее $E_p(\gamma, \beta)$ стремится к минимальному значению $\min_{\mathcal{Z}} \mathcal{H}(Z)$ среди всех возможных спиновых конфигураций \mathcal{Z} решетки:

$$\lim_{p \rightarrow \infty} \min_{\gamma, \beta} E_p(\gamma, \beta) = \min_{\mathcal{Z}} \mathcal{H}(Z), \quad E_p(\gamma, \beta) \equiv \langle \psi(\gamma, \beta) | \mathcal{H} | \psi(\gamma, \beta) \rangle.$$

В зависимости от размерности решетки анзацу $|\psi(\gamma, \beta)\rangle$ соответствуют квантовые схемы различной степени сложности. На Рис. 4 приведена квантовая схема для однослойного ($p = 1$) анзаца на решетке 2×2 . График рассчитанной по этой схеме

энергии $E_1(\gamma, \beta)$ от параметров γ и β представлен на Рис. 5. При значениях параметров $\gamma = 1.0, \beta = 0.5$ анзац $|\psi(\gamma, \beta)\rangle$ соответствует выходному состоянию регистра $|q_{00}q_{01}q_{10}q_{11}\rangle = |0000\rangle$, то есть битовой строке $Z = 1111$. Такая ориентация спинов на решетке действительно обеспечивает минимум энергии для гамильтониана $\mathcal{H}(Z)$. Минимизация энергии $E_1(\gamma, \beta)$ при помощи метода градиентного спуска, проводившаяся уже на классическом компьютере приводит к тем же параметрам. Траектория спуска из некоторой начальной точки показана на графике (Рис. 5) красной стрелкой.

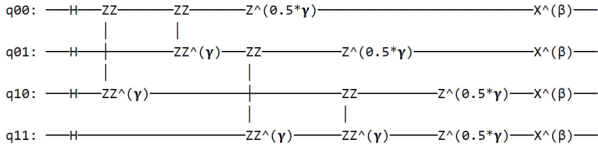


Рис. 4. Квантовая цепь для анзаца QAOA с одним слоем ($p = 1$), созданная в среде Cirq

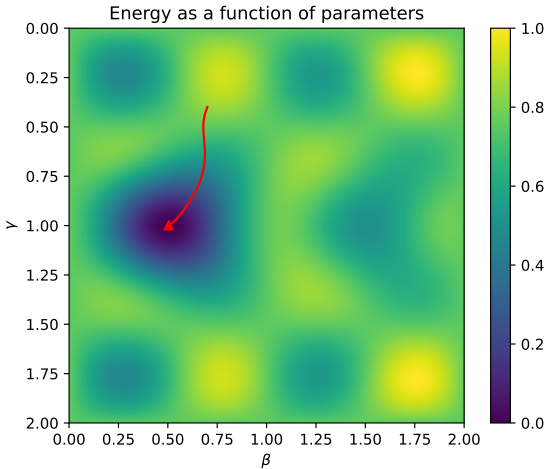


Рис. 5. Зависимость энергии $E_1(\gamma, \beta)$ от значений параметров, найденная по схеме, представленной на Рис. 4. Траектория градиентного спуска обозначена на графике красной линией со стрелкой

При увеличении размерности решетки сложность данной задачи возрастает экспоненциально. Так, размер регистра квантового компьютера растет как 2^n , где n – число узлов решетки, а сложность оператора, соответствующего квантовой схеме QAOA, растет как 2^{2n} . Соответственно, растут требования и к вычислительным мощностям – объему оперативной памяти, количеству и быстродействию ядер CPU и GPU. Уже для задачи с размерностью решетки $3 \times 3 \times 3$ расчет тестовой программы на

персональном компьютере (Intel i5-13490F@3.5GHz, 32Gb DDR5 RAM, GPU NVIDIA RTX 4070) занимает более двух суток.

Для ускорения вычислений использовался симулятор `qsim`. На одном из серверов квантового полигона с двумя процессорами AMD EPYC 7763 (64 ядра @ 2.45 ГГц) на 128 потоках было достигнуто время работы в 3 часа 20 минут, на графическом ускорителе NVIDIA A100 с подключением библиотеки `cuStateVec` удалось сократить время до 14.5 минут.

5. Заключение

На ресурсах гетерогенной платформы HybriLIT организован полигон для квантовых вычислений, предоставляющий возможность работы с симуляторами квантовых вычислений в двух режимах доступа: batch-режим и веб-интерфейс.

Гетерогенная структура платформ позволяет оперативно изменять характеристики полигона, подстраивая его под требования задач пользователей, добавляя сервера с необходимыми вычислительными компонентами, как в отношении центральных процессоров, так и графических ускорителей.

Запуск тестовой задачи показал возможность моделирования систем до 33 кубитов. На реальной задаче было получено ускорение работы квантового симулятора в более, чем 200 раз, по сравнению с вышеупомянутым персональным компьютером. Высокопроизводительный квантовый симулятор `qsim` позволяет более эффективно моделировать квантовые схемы на классических и графических процессорах квантового полигона.

Благодарности

Работа выполнена при поддержке гранта Министерства науки и высшего образования Российской Федерации № 075-10-2020-117.

Литература

1. Гетерогенная платформа HybriLIT ЛИТ ОИЯИ. URL: <http://hlit.jinr.ru> (Дата обращения: 20.02.2024).
2. Экосистема для задач машинного обучения, глубокого обучения и анализа данных. URL: http://hlit.jinr.ru/access-to-resources/ecosystem-for-ml_dl_bigdataanalysis-tasks (Дата обращения: 20.02.2024).
3. Jones T., Brown A., Bush I. et al. QuEST and High Performance Simulation of Quantum Computers. *Sci Rep* **9**, 10736 (2019). doi:10.1038/s41598-019-47174-9.
4. Cirq Developers. Cirq. – 2023. doi:10.5281/zenodo.10247207.
5. Farhi E., Goldstone J., Gutmann S. A Quantum Approximate Optimization Algorithm. doi:10.48550/arXiv.1412.6062.
6. Bharti K. et al. Noisy intermediate-scale quantum (NISQ) algorithms. <https://arxiv.org/abs/2101.08448>
7. Paliĭ Yu., Bogolubskaya A., Yanovich D. Quantum approximation optimization algorithm for the Ising model in an external magnetic field. URL: https://indico.jinr.ru/event/3505/contributions/21552/attachments/16335/27894/Paliĭ_GRID-23.pdf
8. Quantum AI team and collaborators. `qsim`. – 2020. doi:10.5281/zenodo.4023103.
9. `cuStateVec`: A High-Performance Library for State Vector Quantum Simulators. URL: <https://docs.nvidia.com/cuda/cuquantum/latest/custatevec> (Дата обращения: 20.02.2024).

UDC 004.4

Testbed for quantum computing on the heterogeneous platform HybriLIT

D. V. Belyakov¹, A. A. Bogolubskaya¹, Yu. G. Pali^{1,2}, D. V. Podgrainy¹, O. I. Streltsova¹, D. A. Yanovich¹, M. I. Zuev¹

¹*Laboratory of Information Technologies*

Joint Institute for Nuclear Research

Joliot-Curie 6, Dubna, Moscow region, 141980, Russia

²*Institute of Applied Physics,*

Moldova State University

5 Academiei Str. Chişinău, MD-2028, Republic of Moldova

Email: zuevmax@jinr.ru

A testbed for quantum computing was deployed on the resources of the ML/DL/HPC ecosystem of the heterogeneous HybriLIT platform. The characteristics of the server allow calculations to be carried out using libraries that support parallel computing on both central processors and graphics accelerators. JupyterLab was chosen as the basis of the computing environment, which allows users to visually work with quantum circuits and carry out calculations in a web browser. Currently installed quantum computing simulators Cirq, Qiskit, PennyLane, QuTiP. We exemplify the work of the testbed by searching for the state with the lowest energy in the Ising model with a longitudinal magnetic field using the quantum approximation optimization algorithm (QAOA). The dependence of the efficiency of calculations on the configuration of the used computing resources is shown.

Key words and phrases: quantum simulators, information technology, HPC, Python, JupyterLab