

Раздел VI. Вычислительные комплексы нового поколения и нейрокompьютеры

О.К. Евсеев, С.М. Гушанский, В.Ф. Гузик

РАЗРАБОТКА МОДЕЛИ КВАНТОВОГО ВЫЧИСЛИТЕЛЯ

Все компьютеры, существующие на данный момент, строятся на принципах последовательного выполнения операций в рамках задачи. Конечно, производительность современных компьютеров высока, но существуют классы задач, принципиально не решаемых при данном подходе – вычислительные задачи экспоненциальной сложности (вычислительные и временные затраты на решение растут экспоненциально сложности задачи). К таким задачам относятся: факторизация больших чисел [2,3], поиск в неупорядоченных массивах данных, взлом криптографических кодов и некоторые другие.

Квантовый вычислитель (компьютер) [1] – вычислительное устройство, работающее по законам квантовой физики и имеющее экспоненциальную производительность за счёт использования эффектов запутанности состояний (entanglement) и квантового параллелизма вычислений.

Вычислитель может быть физически реализован на основе спинов электронов и атомных ядер и их вероятностных характеристик [4]. Результат решения задачи также имеет вероятностный характер, а состояние квантовой системы разрушается непосредственно после его считывания. Под запутанностью состояний подразумевается свойство квантовых частиц сохранять “информационное” взаимодействие друг с другом, даже, если сами частицы физически полностью изолированы. Так управление состоянием одной частицы при определённых начальных условиях вызывает изменение состояния всех других частиц, находящихся с нею в состоянии квантовой связи. Данное явление и приводит к квантовому параллелизму вычислений. Благодаря этому эффекту, при надлежащей алгоритмической реализации задачи, квантовый компьютер может иметь экспоненциальную производительность.

Из-за принципиальной разницы в организации вычислительного процесса, квантовый алгоритм отличен от классического. Он задаёт конфигурацию квантовой вычислительной системы, начальные состояния её элементов и набор операторов, определяющих её эволюцию. Алгоритм обычно представляется в виде квантовой схемы, задающей набор и последовательность преобразований (порядок следования операторов слева–направо определяет последовательность выполнения). В качестве входной информации используются пси-функции, определяющие пространство начальных состояний элементов системы и их вероятности. На выходе имеем результат эволюции системы в виде выходной пси-функции, соответствующей вероятностному результату вычисления.

Квантовый вычислитель даёт принципиальный выигрыш в быстродействии при решении далеко не всех типов задач, поэтому приоритетной целью стало создание как можно большего числа квантовых алгоритмов, расширяющих область его применения. Наличие модели квантовой вычислительной (не физической) системы существенно ускоряет разработку и проверку работоспособности квантовых

алгоритмов, т.к. избавляет от необходимости каждый раз просчитывать поведение системы.

Предложен ряд способов реализации моделирующих систем [5]: написание закрытых моделей с фиксированным набором возможностей и операций на языках программирования высокого уровня (Delphi, C++); создание открытых библиотек для сред программирования [6] (QDD: A C++ Quantum Computer Emulation Library); разработка “кантовых” языков программного описания систем (QASM [7], Qgol [8]); дополнение стандартных программ моделирования типа Maple “квантовыми” библиотеками [9] (проект OpenQUACS); использование возможностей языков аппаратного описания (*HDL) [10] и т.д.

Так или иначе, подавляющее большинство существующих моделирующих систем базируется на матричном методе, т.к. он практически полностью копирует общепринятое математическое представление квантовых вычислений [2].

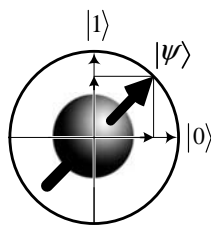


Рис.1.

Суперпозиция состояний квантового бита

Вероятностное состояние квантового бита (рис.1) представляется как вектор-столбец состояния одиночного кубита Ψ (одна из форм записи Ψ -функции квантовой системы, характеризующей её возможные состояния):

$$\Psi = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \quad (1)$$

где α и β – амплитуды кубита такие, что $|\alpha|^2 + |\beta|^2 = 1$.

$|\alpha|^2$ соответствует вероятности, что измерение состояния даст “0”, и $|\beta|^2$ – что “1”. Пример действия оператора на систему:

$$\sigma_x \times \Psi = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \times \begin{bmatrix} \alpha|0\rangle \\ \beta|1\rangle \end{bmatrix} = \begin{bmatrix} \alpha|1\rangle \\ \beta|0\rangle \end{bmatrix}, \quad (2)$$

где σ_x – оператор X-преобразования.

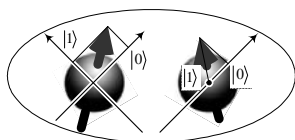


Рис.2.

Суперпозиция состояний системы из 2-х кубит

Системы из 2-х и более частиц (рис.2) характеризуются квантовыми связями, для описания которых не достаточно пси-функций исходных подсистем, которые больше не являются обособленными. Пси-функции кубитов системы:

$$\begin{cases} \psi_1 = \alpha_1|0\rangle + \beta_1|1\rangle \\ \psi_2 = \alpha_2|0\rangle + \beta_2|1\rangle \end{cases}. \quad (3)$$

Пси-функция системы:

$$\Psi = \psi_1 \otimes \psi_2 = \begin{bmatrix} \alpha_1 \\ \beta_1 \end{bmatrix} \otimes \begin{bmatrix} \alpha_2 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} \alpha_1\alpha_2 \\ \alpha_1\beta_2 \\ \beta_1\alpha_2 \\ \beta_1\beta_2 \end{bmatrix}. \quad (4)$$

Пример действия оператора (рис.3) на систему:

$$CN = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} \alpha_1\alpha_2|10\rangle \\ \alpha_1\beta_2|11\rangle \\ \beta_1\alpha_2|00\rangle \\ \beta_1\beta_2|01\rangle \end{bmatrix} = \begin{bmatrix} \alpha_1\alpha_2|10\rangle \\ \alpha_1\beta_2|11\rangle \\ \beta_1\alpha_2|01\rangle \\ \beta_1\beta_2|00\rangle \end{bmatrix}. \quad (5)$$

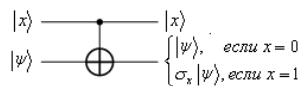


Рис.3. Действие оператора “Управляемое НЕ”

Основной проблемой при моделировании больших систем является увеличение размерности матриц пропорционально экспоненте сложности системы, что при больших размерах моделируемой вычислительной системы делает работу эмулятора экспоненциально сложной.

Университетом Мичигана был предложен способ экономии вычислительных ресурсов, получивший название QuIDD (Quantum Information Decision Diagram) [3] и заключающийся в построении модели, оперирующей более оптимальными в смысле скорости обработки списками и правилами их сжатия, а не громоздкими массивами.

Сжатие предусматривает метод “упаковки” матриц, позволяющий хранить только часть матрицы без необходимости её восстанавливать для проведения преобразований. Этот принцип основан на повторяющейся структуре матриц большинства квантовых операторов. В рамках метода матрицы операторов и векторы состояний представляются в виде графов.

Для ускоренного построения QuIDD-графов был разработан Метод блочного построения QuIDD-графов, суть которого заключается в том, что граф строится не целиком, а частями, т.к. при больших размерностях матриц это – запредельно

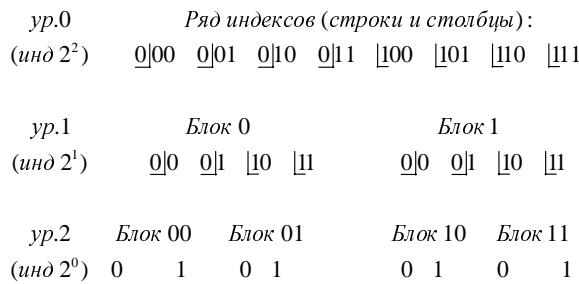


Рис.4.

Метод построения QuIDD-графов путём блочного разбиения матриц

размерностью $\frac{n}{2} \times \frac{n}{2}$, на 2-ом – $16 \times \frac{n}{4} \times \frac{n}{4}$, на последнем (k -ом) – $k \cdot n$ блоков 2×2 .

После того, как получена иерархия блоков, можно начинать построение списковой структуры матрицы. Вначале строится граф, представляющий 0-ой (верхний) уровень иерархии, т.е. корневая вершина представляет матрицу, промежуточные вершины – индексы порядка $2^{\sqrt{n}-1}$, терминальные вершины – блоки 1-го уровня. Аналогично строятся блоки остальных уровней иерархии. В результате будет получена последовательность блоков матрицы, которую останется только составлять в единый граф.

Уточним общие правила построения QuIDD-графов [3]:

1. 1 двоичному разряду размерности матрицы соответствует 1 вершина.
2. Вершины в графе стыкуются посредством связей “=0” и “=1”.
3. Корневой узел связан с вершиной старшего разряда №а строки матрицы.
4. Число терминальных вершин = числу различных элементов матрицы.
5. Терминальными вершинами графа являются указатели на элементы массива, хранящего неповторяющиеся значения элементов матрицы.

Рассмотрим применение метода на примере матрицы духкубитового преобразования Адамара [3] (рис.5,6).

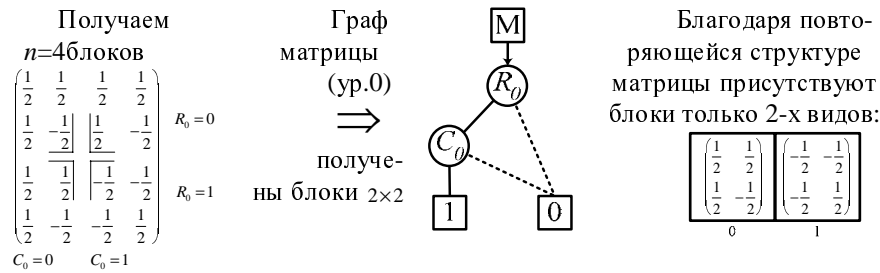


Рис.5. Построение QuIDD-графа уровня 0 матрицы оператора

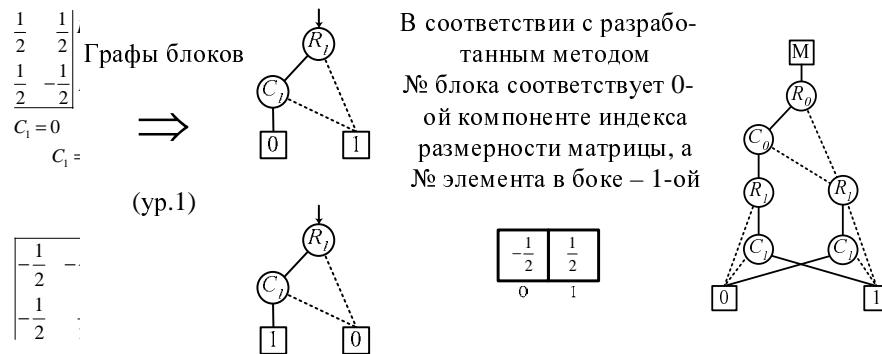


Рис.6. Построение QuIDD-графа уровня 1 и сборка графа матрицы оператора

В итоге, получена матрица из 2-х элементов вместо 16 и QuIDD-граф, хранящий структуру исходной матрицы.

Представление векторов состояний имеет некоторую специфику, но в целом проще. Вначале надо подготовить матрицы (т.е. векторы), далее графы векторов формируются по правилам, применяемым к графам матриц (рис.7).

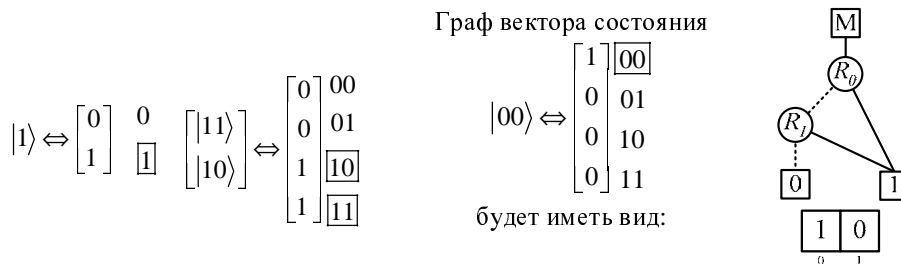


Рис.7. Построение QuIDD-графов векторов состояний

Отличительная особенность данного метода заключается в том, что при моделировании работы квантовой схемы, представленной QuIDD-графами, нет необходимости “распаковывать” графы, что увеличит быстродействие модели. Все квантовые преобразования реализуются через операцию преобразования графов.

В отличие от экспоненциального роста затрат на хранение и обработку числового массива (количество хранимых “узлов”), результаты, полученные в [1] (рис.8), указывают на всего лишь сверхполиномиальный рост списковых структур, что значительным образом отразится на производительности моделирующей системы.

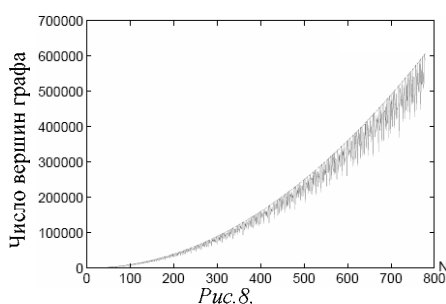


Рис. 8.
Увеличение размерности QuIDD-представления оператора обратного преобразования Фурье для $N=2^n$ квантовых бит.

терминах графов; результирующая пси-функция транслируется в матричный вид. Помимо результирующей пси-функции, на выходе моделирующей системы предусматривается получение диаграммы временных затрат на моделирование вычисления. Согласно результатам [1], использование методики QuIDD существенно сокращает временные затраты и значительно ускоряет получение решения, что позволит создать высокоэффективную систему моделирования работы квантового вычислителя.

На основании проведённой исследовательской работы производится построение модели квантового вычислителя по методике QuIDD. Начальное состояние моделируемой системы задаётся вводом пси-функций кубитов системы, далее преобразуемых в QuIDD-граф пси-функции системы. Эволюция системы определяется квантовой схемой, набираемой из примитивов в редакторе схем. Подразумевается использование шаблонных QuIDD-графов, где это возможно и их синтез по выведенной методике, в специфических частях схемы. Весь процесс преобразований протекает в

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Квантовый компьютер и квантовые вычисления // Журнал “Регулярная и хаотическая динамика”, под ред. Садовниченко В.А. – Ижевск., 1999. – 262 с.
2. Риффель Э., Полак В. Основы квантовых вычислений // Журнал “Квантовый компьютер и квантовые вычисления”. – М., 2000, т. 1, № 1.
3. Viamontes G. F., Markov I. L., Hayes J. P. High-Performance QuIDD-based Simulation of Quantum Circuits. / <http://arXiv:quant-ph/0309060>, v2, 29, Nov 2003.
4. Кокин А.А. Физические реализации квантового компьютера. М.: ФТИАН / <http://elanina.narod.ru/lanina/index.files/student/tehnology/text/kvant.htm#J18>
5. Гузик В.Ф., Гушанский С.М. Моделирование квантовых схем // Известия ТРТУ. Специальный выпуск. – Таганрог: Изд-во ТРТУ, №9(53), 2005. – С. 66.
6. Greve David. QDD: A C++ Quantum Computer Emulation Library. / <http://thegreves.com/david/index.html>
7. Chuang Isaac L. Quantum Computation and Quantum Information. / <http://ichuang@mit.edu>
8. Baker Gregory. Документация к пакету Q-gol / <http://www.ifost.org.au/~gregb/qgol/QgolThesis.pdf>
9. McCubbin Chris. OpenQUACS / <http://userpages.umbc.edu/~cmccub1/quacs/quacs.html>
10. Udrescu Mihai. Using Hardware Engineering in Quantum Computation: Efficient Circuit Simulation and Reliability Improvement / <http://www.sigda.org/daforum/abs/38.pdf>

С.Ю. Касаев, В.Е. Золотовский

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ РАНЖИРОВАНИЯ ПРИ ИХ АППАРАТНЫХ РЕАЛИЗАЦИЯХ

При проектировании различных устройств часто возникает проблема сортировки чисел. Это очень длительный и трудоемкий процесс. Существует большое количество различных методов, которые решают данную задачу. Каждый метод