Математика и её приложения: ЖИМО. 2011. Вып. 1 (8). С. 113 – 120.

УДК 512.54

А. А. Толстопятов[1]

Квантовый алгоритм разбиения файла на буферы

Ключевые слова: кубит, гейт, булево сжатие.

Изложена общая схема квантовых вычислений. Показано, что регистр квантового компьютера должен содержать $N-1$ кубитов, где N – число кортежей в файле. Установлено соответствие между отдельным разбиением файла и квантовым содержанием кубитов регистра. Показано, что разбиение файла на буферы можно представить гейтом, действующим на кубитах регистра. Показано, что такое изменение разбиения файла представляется эрмитовым оператором.

Keywords: e-bit, gate, boolean compress.

We state the general scheme of quantum calculations. We show that a quantum computer register must contain $N-1$ qubits, where N is a number of tuples in the file. We receive the correspondence between file splitting and quantum state of qubits of register. We show, that it is possible to represent the file splitting on buffers by gate, acting on qubits of register. We show, that such change of the file splitting is represented by Hermitian operator.

Задача о разбиении файла на буферы при булевом сжатии является экспоненциально сложной. Одним из возможных подходов к решению таких задач является построение квантовых алгоритмов [1, 2, 3]. Например, сложность задачи о вычислении дискретного логарифма или задачи о разложении натурального числа в произведение степеней простых делителей в известном алгоритме П. Шора удалось путем построения квантового алгоритма снизить с 2^n до n^3 [7]. Известный квантовый алгоритм Л. Гровера о поиске в неупорядоченной базе данных снижает сложность алгоритма с 2^n до $2^{n/2}$ [6]. Сравнение квантовых алгоритмов с алгоритмами, сводящимися к решению булева уравнения, рассмотрены в [4]. Идея об использовании квантовых алгоритмов обсуждалась в [5]. В настоящей работе эта идея рассматривается более подробно.

[1]Ивановский государственный университет; E-mail: khash2@mail.ru. Работа выполнена при финансовой поддержке РФФИ (проект 10-07-00350а).

1. Квантовые вычисления

В квантовой информатике ячейка памяти, содержащая бит, заменяется на кубит — двумерное векторное пространство над полем $\mathbb{C}$. Если ввести базис ($|0>$, $|1>$), то вектор, задающий состояние кубита $|\psi>$ есть

$$|\psi>=\alpha|0>+\beta|1>, \tag{1}$$

где $\alpha, \beta \in \mathbb{C}$, причем

$$|\alpha|^2+|\beta|^2=1. \tag{2}$$

Условие (2) означает, что вектор $|\psi>$ имеет единичную длину. Вероятность того, что при выведении информации из кубита в классический компьютер (КЛАК) мы получим 0, есть $W_0=|\alpha|^2$, а 1 — $W_1=|\beta|^2$. Условие (2) тогда дает, что

$$W_0+W_1=1. \tag{3}$$

Условие (3) — следствие единичной длины $|\psi>$. На вектор $|\psi>$ можно действовать оператором $\hat{H}$, который меняет коэффициенты α и β в (1), а значит меняет и вероятности W_0 и W_1:

$$\hat{H}|\psi>=\alpha'|0>+\beta'|1>=|\psi'>. \tag{4}$$

Для того, чтобы выполнялось соотношение

$$|\alpha'|^2+|\beta'|^2=1, \tag{5}$$

а, следовательно, и

$$W_0'+W_1'=1, \tag{6}$$

где $W_0'=|\alpha|^2$, $W_1'=|\beta'|^2$, оператор $\hat{H}$ должен быть унитарным, т. е. должно выполняться условие

$$\hat{H}\hat{H}^T=I. \tag{7}$$

Из кубитов строится регистр квантового компьютера, содержащий n независимых кубитов; его состояние задается n векторами $|\psi_i>$:

$$|\psi_i>=\alpha_i|0>+\beta_i|1>. \tag{8}$$

В квантовой механике состояние системы, содержащей n независимых подсистем, задается вектором $|\psi>$:

$$|\psi>=\prod_{i=1}^{n}|\psi_i>. \tag{9}$$

Состояние (9) можно менять, действуя на $|\psi>$ унитарным оператором $\hat{H}$ следующим образом

$$\hat{H}|\psi>=|\psi'>=\prod_{i=1}^{n}\hat{H}'_i|\psi_i>=\prod_{i=1}^{n}|\psi'_i>, \tag{10}$$

где $\hat{H}_i$ — унитарный оператор, действующий на i-ом кубите.

Квантовые вычисления начинаются с инсталяции регистра, когда все $\alpha_i=\frac{1}{\sqrt{2}}$, $\beta_i=\frac{1}{\sqrt{2}}$, $i=1,\ldots,n$. Тогда вместо (9) будем иметь:

$$|\psi>=\frac{1}{2^{\frac{n}{2}}}\prod_{i=1}^{n}(|0>+|1>)=\frac{1}{2^{\frac{n}{2}}}(|0>+|1>)^n. \tag{11}$$

Раскрывая степень в (11), получим:

$$|\psi>=\frac{1}{2^{\frac{n}{2}}}(|00\ldots00>+|00\ldots00>|00\ldots01>+\cdots+|11\ldots11>). \tag{12}$$

Каждая скобка в (12) содержит n нулей и единиц, а значит является двоичной записью натурального числа k с n разрядами; поэтому

$$|\psi>=\frac{1}{2^{\frac{n}{2}}}\sum_{k=0}^{2^n-1}|k>. \tag{13}$$

Таким образом, в начальном состоянии регистра вероятности W_i того, что в КЛАК будет выведен любой из 2^n файлов длиной в n битов, равны $W_i=2^{-n}$, $i=1,2,\ldots,2^n$. Действие оператора $\hat{H}$, который называется гейтом, приведет к тому, что эти вероятности перестанут быть равными. Построение квантового алгоритма заключается в том, чтобы найти такую последовательность гейтов, которая сделает вероятность, т. е. квадрат модуля коэффициента при одном из 2^n файлов, являющегося правильным результатом вычисления, намного большей вероятности при остальных 2^n-1 файлов. Таким образом, экспоненциальное увеличение эффективности квантовых алгоритмов по сравнению с классическими связано с тем, что квантовый компьютер (КВАК) работает сразу со всеми 2^n файлами длины n бит, в то время как КЛАК работает с одним файлом.

2. Постановка задачи

Пусть файл разбит на N кортежей длиной в n бит каждый. Пусть эти N кортежей объединены в L буферов, причем m_l, $l=1,\ldots,L$ — число кортежей в l-м буфере. Это объединение и есть разбиение файла на буферы. Всего существует 2^N-1 разных разбиений с $L=1,2,\ldots,N$. Каждому

буферу ставится в соответствие булев полином $f_l(x_i),\ i = 1, \ldots, n$, такой, что уравнение

$$f_l(x_i) = 0 \tag{14}$$

имеет s_l решений, где $s_l \leq m_l$ — число разных кортежей в l-м буфере. Булево сжатие основано на существовании решения кодирующего уравнения (см. [5])

$$F(e_k^l) = f_l, \tag{15}$$

где $F(e_k^l)$ — кодирующий полином, а e_k^l — элементы из множества порождающих булевых полиномов $\varphi_p,\ p = 1, \ldots, P,\ k = 1, \ldots, I$, где I — число булевых переменных, от которых зависит F. Параметрами, задающими разбиение файла на буферы являются $n,\ I,\ P,\ L$. Коэффициент сжатия файла k зависит от этих параметров следующим образом (см. [5])

$$k = \frac{n \sum\limits_{l=1}^{L} m_l}{2^I + 2^n P + LI \log_2 P + \log_2 \prod\limits_{l=1}^{L} \dfrac{C_{m_l-1}^{s_l-1} m_l!}{\prod\limits_{k=1}^{s} n_k^l!}}, \tag{16}$$

где $n_k^l,\ k = 1, \ldots, s_l$ — числа повтором k-го кортежа в l-ом буфере.

Задача о разбиении файла на буферы заключается в том, чтобы для заданного файла, разбитого на кортежи, найти такие параметры разбиения и числа $m_l,\ l = 1, \ldots, L$, для которых выполнено два условия:

$$\begin{aligned} &1)\ k > 1, \\ &2)\ \text{уравнение (15) имеет решение.} \end{aligned} \tag{17}$$

Поскольку существует 2^{N-1} разных разбиений файла, то построив регистр КВАКа из $N-1$ кубитов и установив соответствие между разбиением файла и 2^{N-1} последовательностями, содержащими $N-1$ нулей и единиц, мы получаем возможность работать сразу со всеми возможными разбиениями файла. Проверка условий (17) может быть осуществлена с помощью оракула, рассматриваемого в [6]. Переход от одного разбиения файла к другому будет индуцировать действие гейта $\hat{H}$ на $N-1$ кубитах регистра КВАКа. Это действие будет менять вероятности того, что в КЛАК будет выведен тот или иной номер разбиения файла. Это открывает возможность построения такой последовательности гейтов, что вероятность номеров разбиений, удовлетворяющих (17), будет больше, чем вероятность номеров не удовлетворяющих (17). В этой работе ставится задача

о построении указанного выше соответствия и установление соответствия между гейтами, действующими на регистре КВАКа, и прохождением таблицы всех возможных разбиений файла.

3. Структура множества разбиений файла на буферы

Если кортежи объединены в L буферов, причем $L = 0, 1, 2, \ldots, N-1$, то число различных разбиений $M(L)$ будет равно

$$M(L) = C_{N-1}^{L}, \tag{18}$$

т. к. оно совпадает с числом способов установить $L-1$ границу между N кортежами. Полное число разбиений файла M при всех возможных L есть

$$M = \sum_{L=0}^{N-1} M(L) = \sum_{L=0}^{N-1} C_{N-1}^{L} = 2^{N-1}. \tag{19}$$

Все различные разбиения можно упорядочить, разбив таблицу этих разбиений на блоки, содержащие одно и тоже число буферов L. Длина каждого блока будет $M(L)$ из (18). Всего таких блоков будет N. Тогда все возможные разбиения файла из N кортежей разместятся в следующей таблице.

$$\begin{aligned}
0.&\ \ m_1 = N;\ L = 1\\
1.&\ \ m_1 = 1,\ m_2 = N-1,\ L = 2\\
2.&\ \ m_1 = 2,\ m_2 = N-2,\ L = 2\\
&\ \ \ldots\\
C_{N-1}^{1}.&\ \ m_1 = N-1,\ m_2 = 1,\ L = 2\\
C_{N-1}^{1}+1.&\ \ m_1 = 1,\ m_2 = 1,\ m_3 = N-2,\ L = 3\\
&\ \ \ldots\\
C_{N-1}^{1}+C_{N-1}^{2}.&\ \ m_1 = N-2,\ m_2 = 1,\ m_3 = 1,\ L = 3\\
&\ \ \ldots\\
2^{N-1}.&\ \ m_1 = 1,\ m_2 = 1,\ \ldots,\ m_N = 1,\ L = N
\end{aligned} \tag{20}$$

Таким образом, каждому разбиению файла на буферы поставлен в соответствие его номер в таблице (20). По этому номеру можно однозначно восстановить последовательность m_l, $(l = 1, \ldots, L,\ L = 1, \ldots, N)$ и наоборот, по m_l вычислить номер в (20). Таблица (20) позволяет каждому разбиению файла поставить во взаимно однозначное соответствие заполнение регистра КВАКа.

4. Структура регистра КВАКа

В общем случае, когда i-й кубит содержит вектор $|\psi_i>$,

$$|\psi_i>=\alpha_i|0>+\beta_i|1>, \tag{21}$$

регистр содержит вектор $|\psi>$,

$$|\psi>=\prod_{i=1}^{N-1}|\psi_i>=\prod_{i=1}^{N-1}\alpha_i|0>+\beta_i|1>. \tag{22}$$

Раскрывая произведение в (22), получим:

$$\begin{aligned}|\psi>=\alpha_1\alpha_2\ldots\alpha_{N-2}\alpha_{N-1}|00\ldots00>+\\ +\alpha_1\alpha_2\ldots\alpha_{N-2}\beta_{N-1}|00\ldots01>+\cdots+\\ +\beta_1\beta_2\ldots\beta_{N-2}\beta_{N-1}|11\ldots11>.\end{aligned} \tag{23}$$

Всего в правой части стоит 2^{N-1} скобок $|\cdots>$, которые можно рассматривать как базис в 2^{N-1}-мерном векторном пространстве на полем $\mathbb{C}$. Квадраты модулей комплексных коэффициентов перед каждым из 2^{N-1} векторов базиса дают вероятности того, что в КЛАК будут выведен именно этот файл из $N-1$ бита:

$$\begin{aligned}W_{00\ldots00}&=|\alpha_1\alpha_2\ldots\alpha_{N-2}\alpha_{N-1}|^2,\\ W_{00\ldots01}&=|\alpha_1\alpha_2\ldots\alpha_{N-2}\beta_{N-1}|^2,\\ &\ldots\\ W_{11\ldots11}&=|\beta_1\beta_2\ldots\beta_{N-2}\beta_{N-1}|^2.\end{aligned} \tag{24}$$

Начальное инсталлированное состояние регистра КВАКа получается, если положить

$$\alpha_1=\cdots=\alpha_{N-1}=\beta_1=\cdots=\beta_{N-1}=\frac{1}{\sqrt{2}}. \tag{25}$$

Тогда из (24) будем иметь:

$$W_{00\ldots00}=W_{00\ldots01}=\ldots W_{11\ldots11}=\frac{1}{2^{N-1}}, \tag{26}$$

т. е. все номера от 0 до $2^{N-1}-1$ могут быть введены в КЛАК с равной вероятностью. Вероятности (26) будут меняться, когда будет установлено соответствие между базисными векторами регистра КВАКа и разбиениями файла, и эти разбиения будут посылаться в оракул для проверки выполнения (17).

5. Соответствие между разбиением файла на буферы и заполнением регистра КВАКа

Соответствие между разбиением файла на буферы и заполнением регистра КВАКа устанавливается следующим образом. Так как существуют 2^{N-1} разных разбиений, которые можно упорядочить с помощью таблицы (20) (причем каждому разбиению присваивается номер от 0 до 2^{N-1}), и существует 2^{N-1} базисных векторов регистра КВАКа

$$|00\ldots00>, \ldots, |11\ldots11>,$$

которые уже пронумерованы числами от 0 до $2^{N-1}-1$, (т. е. стоящие в скобках последовательности из $N-1$ нулей и единиц и есть такие числа в двоичном коде), то для установления указанного выше соответствия достаточно отождествить эти номера. Тогда исходные вероятности того, что будет выбрано то или иное разбиение файла, будут равны, согласно (26). Изменить эти вероятности может проверка условий (17). Эту проверку можно проводить в два этапа. Сначала проверять первое условие из (17), поскольку оно проверяется просто. После этого можно, используя конструкцию оракула в алгоритме Гровера [6], построить последовательность гейтов, такую, что она будет поворачивать вектор $|\psi>$ как раз к тем базисным векторам регистра КВАКа, для которых это условие выполнено. После этого в оракул запускаются оставшиеся разбиения файла для проверки второго условия из (17). И опять можно использовать алгоритм Гровера для поворота $|\psi>$ к тем базисным векторам, для которых второе условие из (17) выполнено. Вероятности $W_{00\ldots00}, \ldots, W_{11\ldots11}$ согласно (24) будут тем больше, чем ближе вектор $|\psi>$ к соответствующему вектору базиса регистра КВАКа, поскольку

$$\begin{aligned} <00\ldots00|\psi> &= \alpha_1\alpha_2\ldots\alpha_{N-2}\alpha_{N-1}, \\ <00\ldots01|\psi> &= \alpha_1\alpha_2\ldots\alpha_{N-2}\beta_{N-1}, \\ &\ldots \\ <11\ldots11|\psi> &= \beta_1\beta_2\ldots\beta_{N-2}\beta_{N-1}\,. \end{aligned} \tag{27}$$

Если обозначить через W_k вероятность того, что в КЛАК будет выведен файл с двоичным номером k, то

$$\sum_{k=0}^{2^{N-1}-1} W_k = 1\,. \tag{28}$$

Из (27) и (28) следует, что вероятность разбиений файла на буферы, для которых не выполнено одно из условий (17), будет меньше, чем (26),

после действия алгоритма Гровера [6]. Поскольку алгоритм Гровера, как указывалось выше, не меняет экспоненциальности сложности задачи (2^n), а лишь "извлекает из нее квадратный корень", то детали этого алгоритма не рассматривались в надежде получить более эффективный квантовый алгоритм на основе построенного соответствия между разбиениями файла на буферы и заполнением регистра КВАКа.

Список литературы

1. *Валиев К. А., Канин А. А.* Квантовые компьютеры: надежды и реальность. – Москва-Ижевск: НТЦ "Регулярная и хаотическая динамика", 2001. – 352 с.
2. *Китаев А., Шень А., Вялый М.* Классические и квантовые вычисления. – М.: МЦНМО, ЧеРо, 1999. – 192 с.
3. *Стин Э.* Квантовые вычисления. – Москва-Ижевск: R&C Dynamica, 2000. – 112 с.
4. *Толстопятов А. А.* Об алгоритмах решения задачи поиска в неупорядоченной базе данных на КЛАКах, КВАКах и БУЛЬКах // Вестник ИвГУ. – 2000. – Вып. 3. – С. 93–100.
5. *Толстопятов А. А.* Возможные подходы к разбиению файла на буферы при булевом сжатии // Математика и ее приложения: Журн. Иванов. матем. об-ва. – 2009. – Вып. 1 (6). – С. 129–138.
6. *Grover L.* A fast quantum mechanical algorithm for data base search // STOC'28. – 1996. – P. 212–219.
7. *Shor P. W.* Algorithm for Quantum Computation: Discrete log and Factorin // FOCS'35. – 1994. – P. 124.

Поступила в редакцию 15.12.2011