

Задание №1 Параллельная программа на OpenMP, которая реализует однокубитное квантовое преобразование.

Срок сдачи задания 25.09.2013

Задание

1. Реализовать параллельную программу на C++ с использованием OpenMP, которая выполняет однокубитное квантовое преобразование над вектором состояний длины 2^n , где n – количество кубитов, по указанному номеру кубита k . Описание однокубитного преобразования дано ниже в разделе методические рекомендации[1]. Для работы с комплексными числами возможно использование стандартной библиотеки шаблонов[2].
2. Определить максимальное количество кубитов, для которых возможна работа программы на системе Regatta. Выполнить теоретический расчет и проверить его экспериментально.
3. Протестировать программу на системе Regatta. В качестве теста использовать преобразование Адамара по номеру кубита:
 - а) который соответствует Вашему номеру в списке группы плюс 1.
 - б) 1
 - в) n

Начальное состояние вектора должно генерироваться случайным образом. Заполнить таблицу и построить график зависимости ускорения параллельной программы от числа процессоров для каждого из случаев а)-в):

Количество кубитов	Количество процессоров	Время работы программы (сек)	ускорение
20	1		
	2		
	4		
	8		
24	1		
	2		
	4		
	8		
28	1		
	2		
	4		
	8		
Максимально возможное количество кубитов	1		
	2		
	4		
	8		

4. Написать отчет, который будет содержать листинг программы, а так же результаты выполнения пунктов 2-3.

Методические рекомендации

[1] Имеется комплексный входной вектор (массив) размерности 2^n : $\{a_i\} = \{a_0, a_1, \dots, a_{2^n-1}\}$; n – параметр задачи (число кубитов). Над такими векторами нам необходимо производить так называемые однокубитные операции. Обе эти операции переводят вектор в новый вектор такой же размерности (длины массива).

Однокубитная операция задается двумя параметрами: комплексной матрицей размера 2×2 и числом от 1 до n (данный параметр обозначает номер кубита, по которому проводится операция). Итак, дана комплексная матрица:

$$U = \begin{pmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{pmatrix}$$

и k - номер индекса от 1 до n (номер кубита).

Такая операция преобразует вектор $\{a_{i_1 i_2 \dots i_n}\}$ в $\{b_{i_1 i_2 \dots i_n}\}$, где все 2^n элементов нового вектора вычисляются по следующей формуле:

$$b_{i_1 i_2 \dots i_k \dots i_n} = \sum_{j_k=0}^1 u_{i_k j_k} a_{i_1 i_2 \dots j_k \dots i_n} = u_{i_k 0} a_{i_1 i_2 \dots 0 \dots i_n} + u_{i_k 1} a_{i_1 i_2 \dots 1 \dots i_n}$$

Например, если мы производим такую операцию на первом кубите ($k = 1$) двухкубитного вектора ($n = 2$) то коэффициенты нового вектора b будут:

$$\begin{aligned} b_0 &= b_{00} = u_{00}a_{00} + u_{01}a_{10} \\ b_1 &= b_{01} = u_{00}a_{01} + u_{01}a_{11} \\ b_2 &= b_{10} = u_{10}a_{00} + u_{11}a_{10} \\ b_3 &= b_{11} = u_{10}a_{01} + u_{11}a_{11} \end{aligned}$$

Жирным шрифтом для наглядности выделен индекс, по которому производится операция. Как видно, для вычисления каждого элемента нового вектора используются лишь два элемента старого вектора, определяемые изменением k -го бинарного индекса. Преобразование Адамара задается следующей матрицей:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

[2] Шаблон для работы с комплексными числами:

```
#include <complex>
typedef std::complex<double> complexd;
```

Рекомендуемая литература

1. Антонов А.С. "Параллельное программирование с использованием технологии OpenMP: Учебное пособие".-М.: Изд-во МГУ, 2009. - 77 с.
2. Инструкция по работе с системой Regatta
<http://wiki.cs.msu.su/Main/RegattaInstructions>
3. Кронберг Д.А, Ожигов Ю.И., Чернявский А.Ю. «Квантовый компьютер и квантовая информатика». http://sqi.cs.msu.su/store/storage/th25kzj_quantum_computer.pdf