

Московский Государственный Университет им. М.В. Ломоносова  
Факультет Вычислительной Математики и Кибернетики Кафедра  
Суперкомпьютеров и вантовой Информатики

**Курс: практикум на ЭВМ.**

**Отчет №1.**

**Реализация однокубитного квантового преобразования с  
использованием OpenMP**

Работу выполнила

**Килина М. Л.**

**Москва, 2020**



## Постановка задачи и формат данных.

**Задача.** 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  $k$ .

**Параметры, передаваемые командной строке:**

- Количество выделяемых нитей программы
- Количество кубитов  $n$
- Номер кубита  $k$
- Имя файла, в который записывается время работы программы

## Описание алгоритма.

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  $2 \times 2$  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:

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

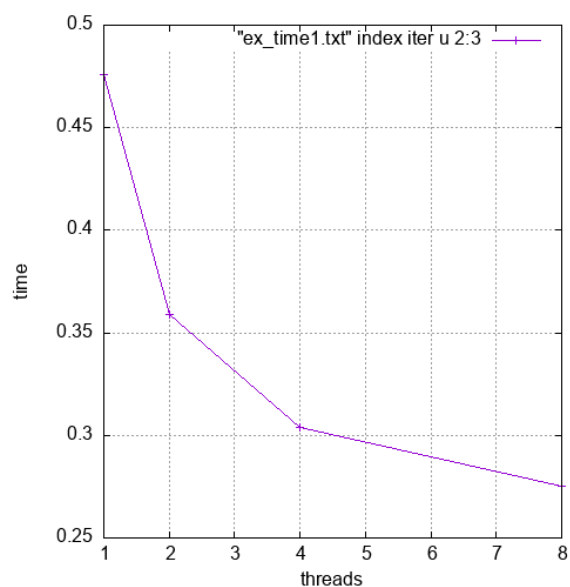
В программе используется преобразование Адамара.

## Результаты выполнения.

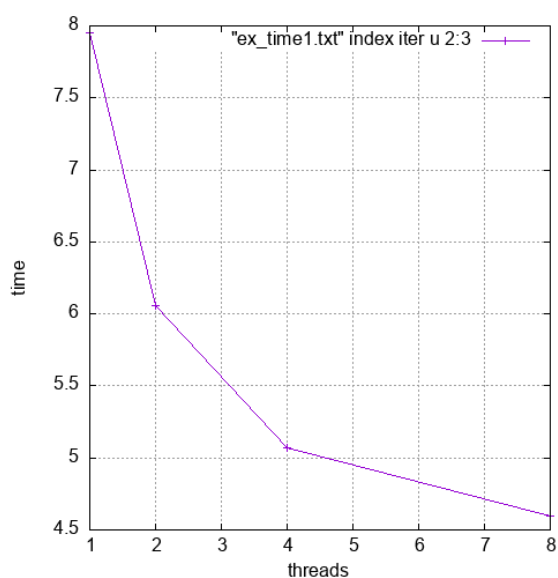
1) Результат для  $k = 1$

Количество кубитов	Количество процессов	Время работы (сек)	Ускорение
20	1	0,475981	1
20	2	0,359147	1,325309692
20	4	0,30417	1,564851892
20	8	0,275047	1,730544234
24	1	7,95436	1
24	2	6,06994	1,312614
24	4	5,06994	1,570118
24	8	4,60002	1,729201
28	1	128,067	1
28	2	95,7812	1,337079
28	4	81,1047	1,579033
28	8	72,7552	1,760245
30	1	501,349	1
30	2	374,087	1,340194
30	4	317,79	1,577611
30	8	285,866	1,75379

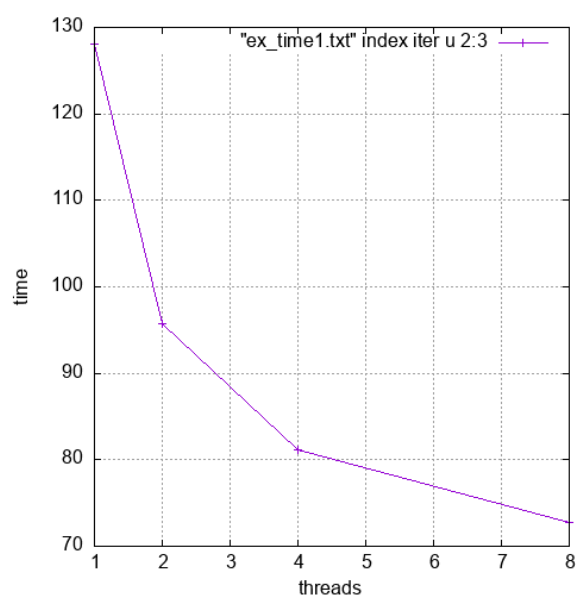
$n = 20$



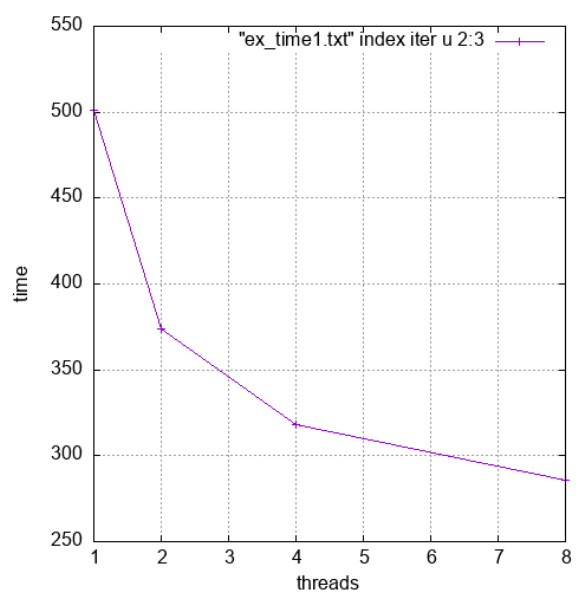
$n = 24$



n = 28



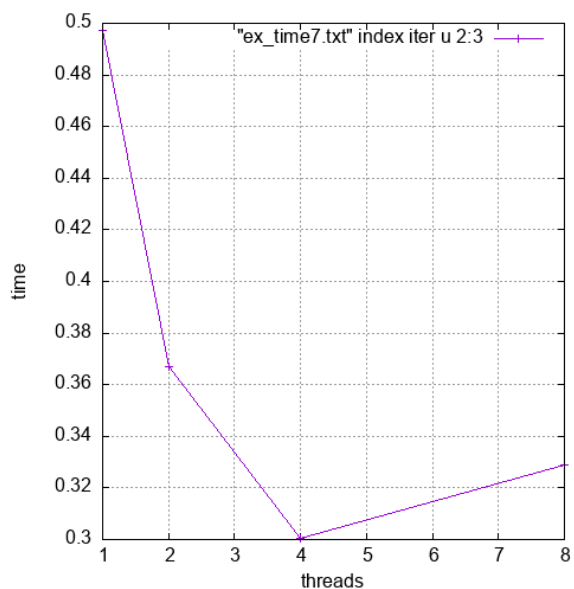
n = 30



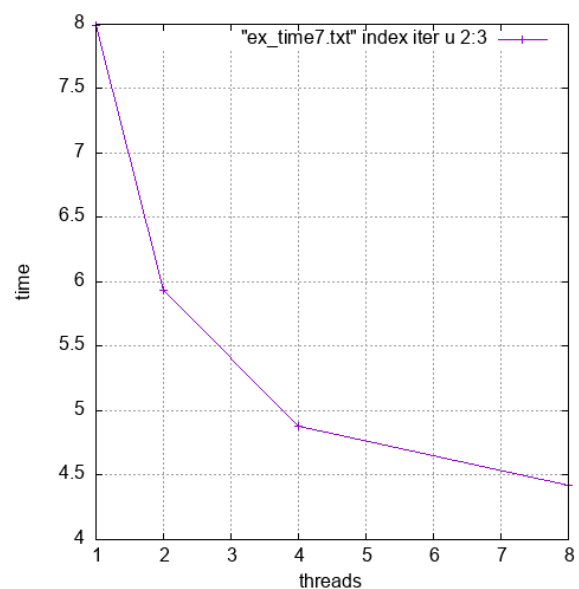
2)  $k = 7$

Количество кубитов	Количество процессов	Время работы (сек)	Ускорение
20	1	0,49747	1
20	2	0,366909	1,35584
20	4	0,300463	1,655678
20	8	0,329041	1,511878
24	1	7,99336	1
24	2	5,93673	1,346425
24	4	4,87726	1,639804
24	8	4,41824	1,809173
28	1	125,584	1
28	2	93,3483	1,345327
28	4	78,1869	1,606203
28	8	70,2742	1,787057
30	1	501,552	1
30	2	376,322	1,332774
30	4	312,286	1,606066
30	8	280,657	1,787064

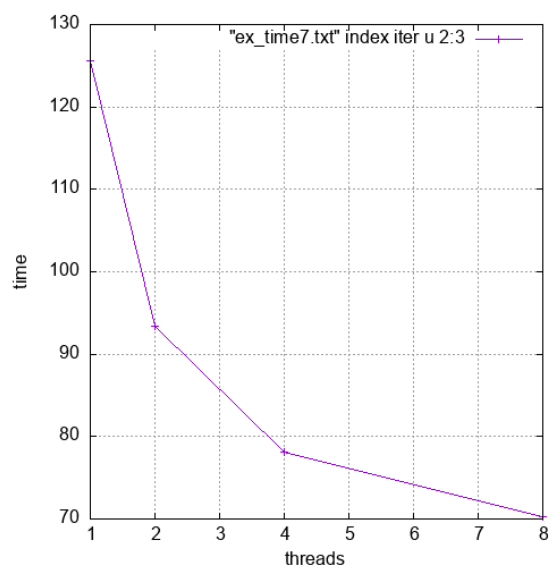
$n = 20$



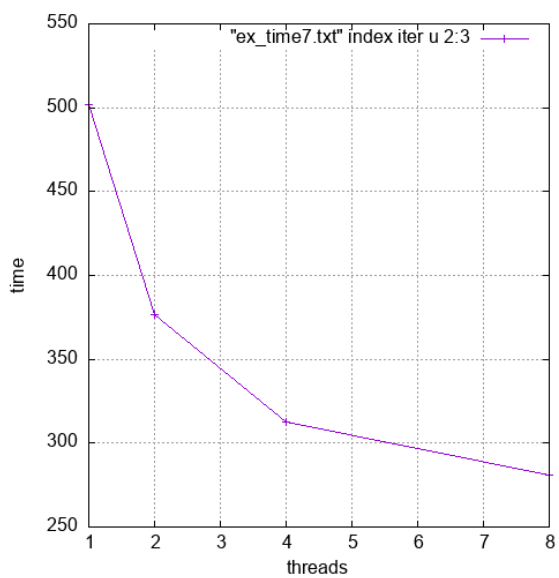
$n = 24$



$n = 28$



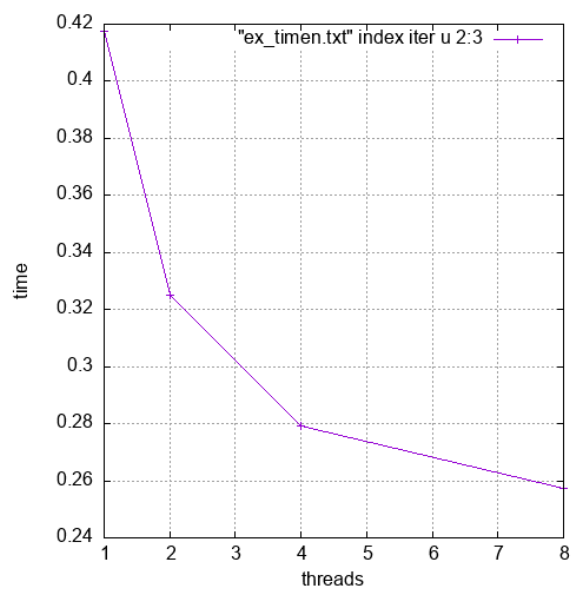
$n = 30$



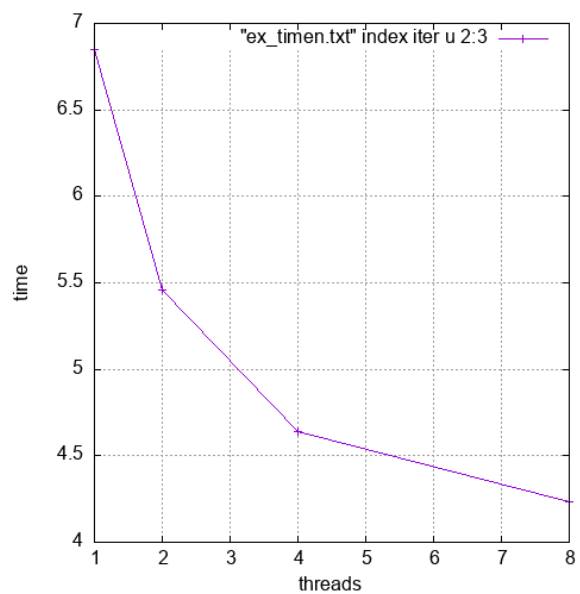
3)  $k = n$

Количество кубитов	Количество процессов	Время работы (сек)	Ускорение
20	1	0,417485	1
20	2	0,325147	1,283988
20	4	0,279092	1,495869
20	8	0,257516	1,6212
24	1	6,846	1
24	2	5,45611	1,25474
24	4	4,64035	1,47532
24	8	4,23184	1,617736
28	1	110,189	1
28	2	86,2942	1,276899
28	4	73,5132	1,498901
28	8	68,5235	1,608047
30	1	440,56	1
30	2	344,609	1,278434
30	4	296,505	1,485843
30	8	273,242	1,612344

$n = 20$

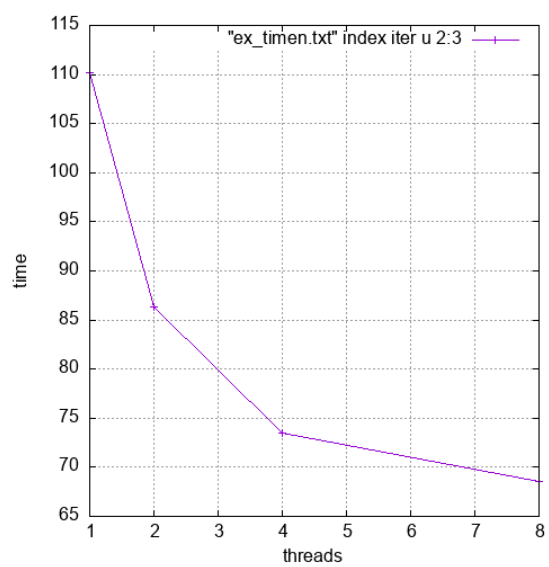


$n = 24$





n = 28



n = 30

