

Московский Государственный Университет им. М.В. Ломоносова

Факультет Вычислительной Математики и Кибернетики

Кафедра Суперкомпьютеров и Квантовой Информатики



## **Практикум на ЭВМ**

### **Отчёт № 3**

#### **Параллельная программа на MPI и OpenMP, реализующая однокубитное квантовое преобразование с шумами**

Работу выполнил

**Сайбель Т. А.**

Москва 2021

## Постановка задачи и формат данных

- 1) Реализовать параллельную программу на C++ с использованием MPI и OpenMP, которая выполняет квантовое преобразование n-Адамара с зашумленными вентилями над вектором состояний длины  $2^n$ , где  $n$  – количество кубитов. Использовать рекомендованную модель зашумления. Для работы с комплексными числами использовать стандартную библиотеку шаблонов.
- 2) Протестировать программу на системе Blue Gene/P.

Начальное состояние вектора генерируется случайным образом и нормируется (тоже параллельно).

**Формат командной строки:** <Число кубитов  $n$ > <Уровень шума  $\epsilon$ > <Количество потоков> <имя файла исходного вектора> <имя файла полученного вектора> <>

**Формат файла-вектора:** Вектор представляется в виде бинарного файла следующего формата:

<i>Тип</i>	<i>Значение</i>	<i>Описание</i>
Число типа int	$n$ – натуральное число	Число кубитов
Массив чисел типа complex<double>	$2^n$ – комплексных чисел	Элементы вектора

## Описание алгоритма

Однокубитная операция над комплексным входным вектором  $\{a_i\}$  размерности  $2^n$  задается двумя параметрами: комплексной матрицей  $\{u_{ij}\}$  размера  $2 \times 2$  (вентиль) и числом  $k$  от 1 до  $n$  (номер кубита, по которому проводится операция). Такая операция преобразует вектор  $\{a_i\}$  в  $\{b_i\}$  размерности  $2^n$ , где все элементы вычисляются по следующей формуле:

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

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

Зашумленный вентиль Адамара  $H_\epsilon$  определяется следующими формулами:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, H_\epsilon = HU(\theta), U(\theta) = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix}, \theta = e\xi, \xi \sim N(0, 1),$$

где  $\epsilon$  – это уровень шума.

Преобразование n-Адамара – это преобразование Адамара, выполненное последовательно n раз над вектором состояний, при этом кубит, по которому проводится преобразование изменяется от 1 до n. В качестве меры потери точности используется  $1-F$ , где F – мера точности (вероятность совпадения между идеальным и зашумленным векторами состояний). Мера точности вычисляется как квадрат модуля скалярного произведения соответствующих векторов.

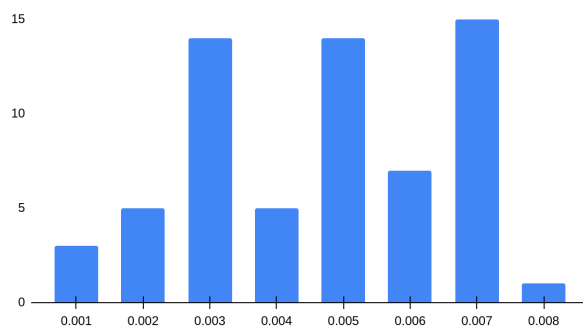
**Аппаратное обеспечение:** Исследования проводились на вычислительном комплексе Blue Gene/P.

**Анализ времени выполнения:** Для оценки времени выполнения программы использовалась функция `MPI_Wtime()`.

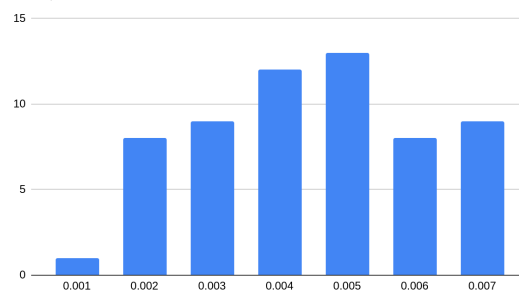
## Результаты выполнения

Количество кубитов (n)	Количество вычислительных узлов	Количество используемых ядер в узле	Время работы (сек)	Ускорение
28	1	1	41,08	1
		2	21,398794	1,919687203
		4	11,536339	3,560834247
		8	5,813737	7,065849556
	2	1	43,816157	1
		2	21,732912	2,016119929
		4	11,880787	3,68798439
		8	6,013621	7,286152054
	4	1	33,14957	1
		2	11,10364	2,985468729
		4	6,893902	4,808535137
		8	4,611833	7,187938071

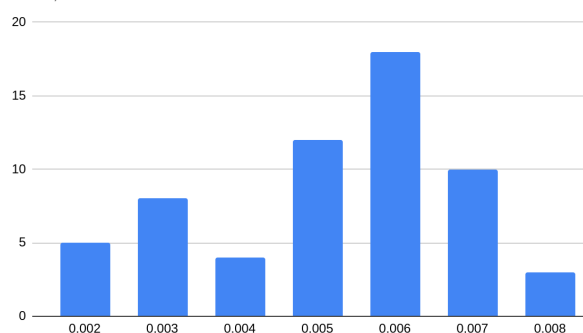
n=24; e=0.01



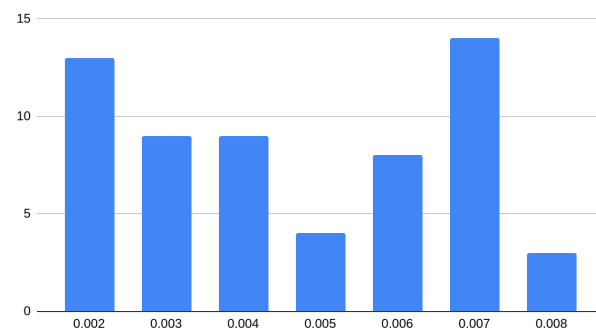
n=25; e=0.01



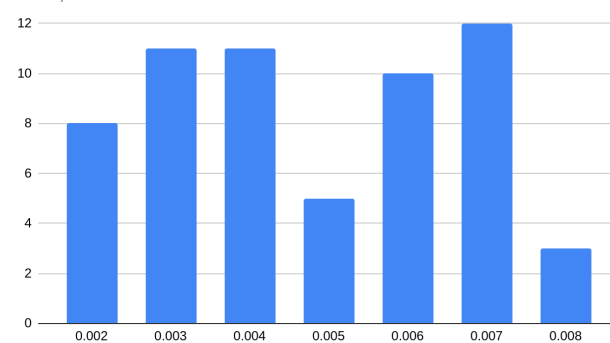
n=26; e=0.01



n=27; e=0.01



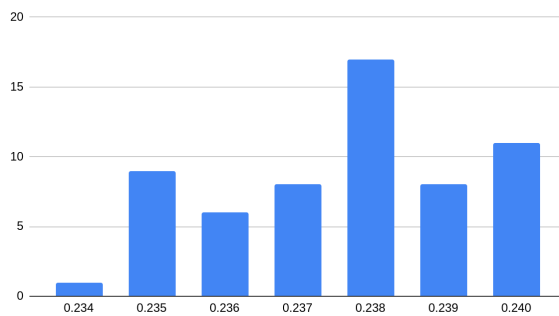
n=28; e=0.01



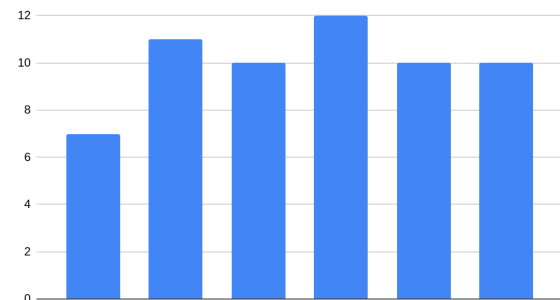
Количество кубитов	Среднее значение потери точности
24	0.00264463
25	0.00236435
26	0.00265423
27	0.00269367
28	0,00286371

е	Среднее значение потери точности
0.1	0.2352893
0.01	0.0026542
0.001	0,0000257

n=26; e=0.1



n=26; e=0.001



## Основные выводы

Распараллеливание ускоряет выполнение программы, но с увеличением числа процессов эффективность снижается из-за роста количества пересылок и накладных расходов на организацию параллелизма.