



Практикум на ЭВМ

Отчёт № 3

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

Работу выполнил
Чепурнов А. В.

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

Формат командной строки: <Число кубитов n > <Уровень шума e > [<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 double	2^n пар чисел с плавающей точкой (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 2×2 (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:

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

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

Зашумленный вентиль Адамара H_e определяется следующими формулами:

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

где e – 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 IBM Polus.

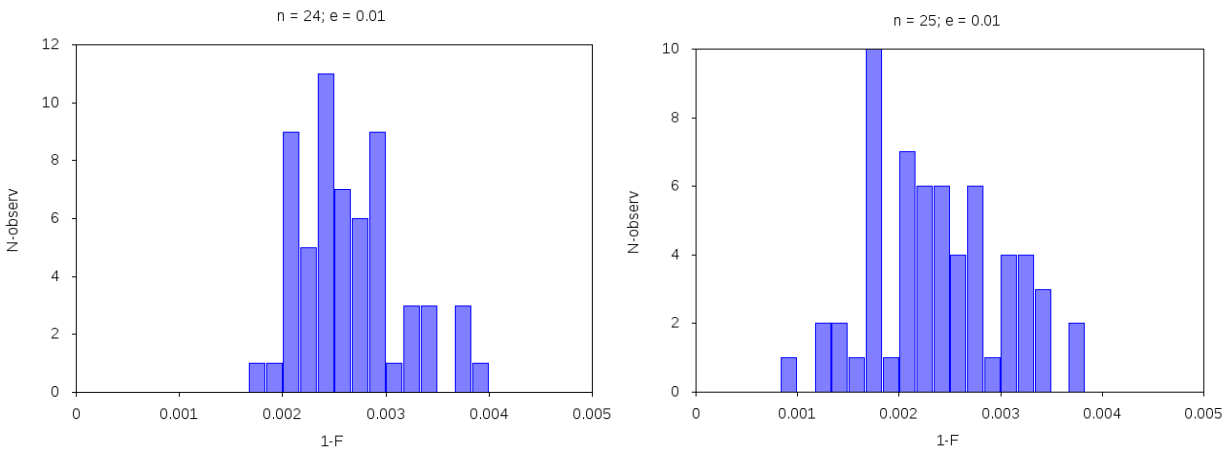
Анализ времени выполнения: Для оценки времени выполнения программы использовалась функция MPI_Wtime().

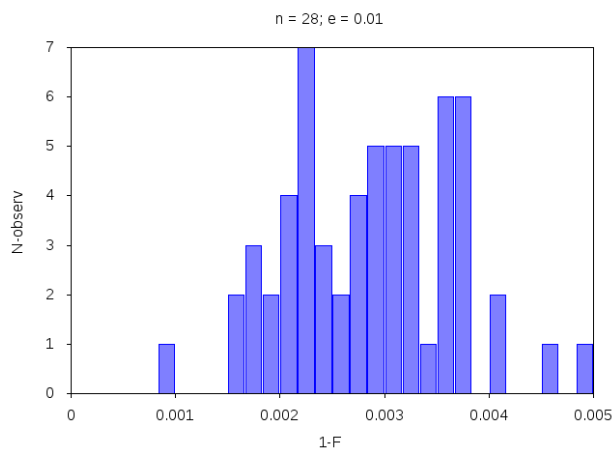
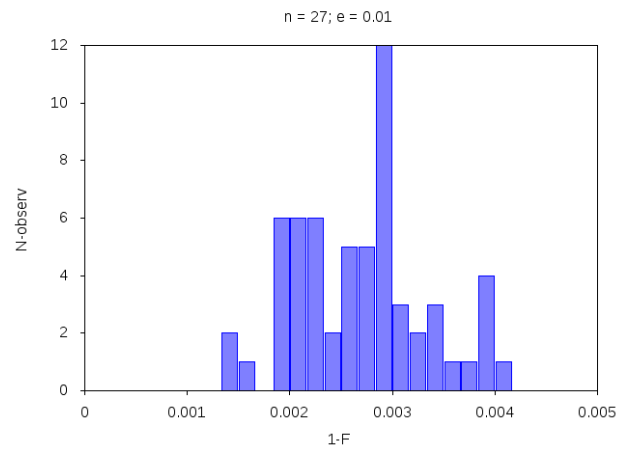
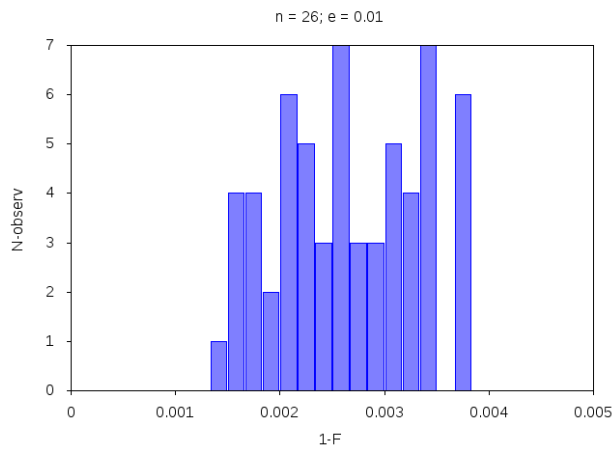
Анализ ускорения: Ускорение, получаемое при использовании параллельного алгоритма для p процессов, высчитывалось как отношение времени выполнения программы без распараллеливания к времени параллельного выполнения программы.

Результаты выполнения

Количество кубитов (n)	Количество вычислительных узлов	Количество используемых ядер в узле	Время работы (сек)	Ускорение
28	1	1	41,078991	1
		2	21,398794	1,919687
		4	11,536339	3,560834
		8	7,113737	5,774601
	2	1	43,816157	0,937531
		2	21,732912	1,890174
		4	11,880787	3,457598
		8	7,413621	5,541016
	4	1	33,14957	1,239201
		2	11,10364	3,699597
		4	6,893902	5,958743
		8	4,451833	9,227433

При фиксированной точности $\epsilon = 0.01$:

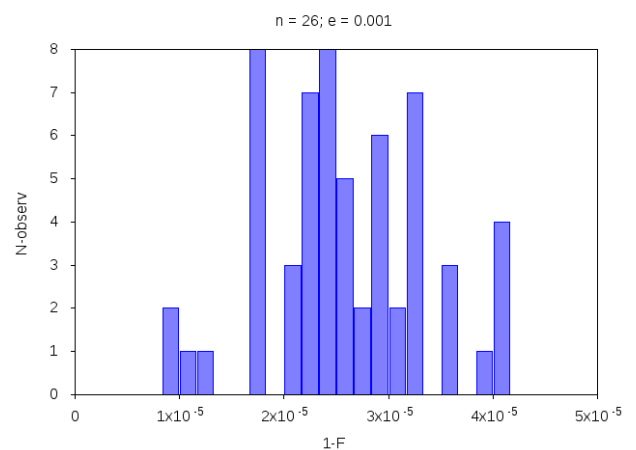
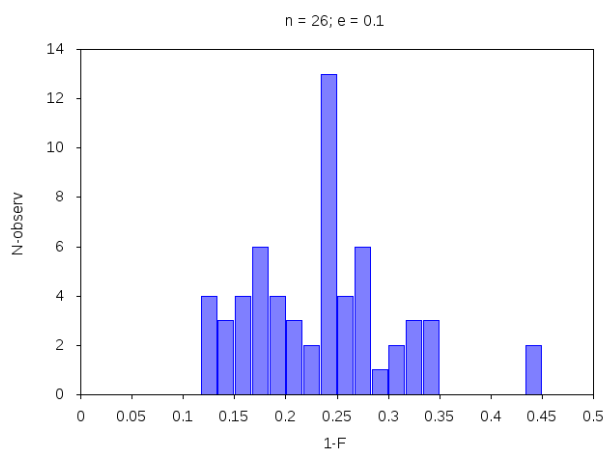




Количество кубитов	Среднее значение потерь точности
24	0,002643717
25	0,00236435
26	0,002654017
27	0,002693367
28	0,002863717

При фиксированном количестве кубитов $n = 26$:

e	Среднее значение потерь точности
0.1	0,235289383
0.01	0,002654017
0.001	0,000025734



Для построения каждого распределения проводилось 60 экспериментов.