Московский Государственный Университет им. М.В. Ломоносова
Факультет Вычислительной Математики и Кибернетики
_____Кафедра Суперкомпьютеров и Квантовой Информатики_____

Спецкурс: системы и средства параллельного программирования

Отчёт № 1
Однокубитное квантовое преобразование, реализуемое с помощью параллельной программы на OpenMP

Работу выполнила
Домрачева Д. А.

Москва 2018

Постановка задачи и формат данных

Задача:

Разработать параллельную программу с использованием OpenMP, реализующую алгоритм однокубитного квантового преобразования. Тип данных – complex<double>.

- Определить максимальное количество кубитов, для которых возможна работа программы на системе Polus. Выполнить теоретический расчет и проверить его экспериментально.
- Протестировать программу на системе Polus для преобразования Адамара и трех различных номеров кубита k.

Формат командной строки:
<число кубитов n> <номер кубита k> <количество потоков>

Математическая постановка задачи

Однокубитное преобразование – преобразование вектора $\{ a_{i_1 i_2 \dots i_n} \}$ в вектор $\{ b_{i_1 i_2 \dots i_n} \}$, задающееся комплексной матрицей 2×2 (в частности, преобразование Адамара задается матрицей $U = \frac{1}{\sqrt{(2)}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$) и числом $1 \le k \le n$ – номером кубита, где элементы нового вектора вычисляются по формуле: $b_{i_1 i_2 \dots i_k \dots i_n} = \sum_{j_k=0}^{1} u_{i_k j_k} a_{i_1 i_2 \dots j_k \dots i_n}$.

Максимальное количество кубитов

Рассчитаем теоретически максимально возможное n, исходя из характеристик системы Polus. В доступе имеется 256 Гб оперативной памяти, следовательно:

$$256\,Гб = 2^8 \times 2^{30}\,байт = 2\,вектора\,длины \times 2^n\,по \times 2^4\,байт => n = 33.$$

Тестирование и результаты

На практике достигнуть n не получилось, так как помимо векторов в оперативной памяти должна храниться выполняемая программа и другие данные, поэтому максимальное n было принято равным 32.

Для пункта а: номер в списке группы (взят из google-таблицы) + 1 = 4.

а) k = 4

Количество кубитов	Количество процессов	Время работы (сек)	Ускорение
20	1	0.131911	1
	2	0.0677957	1.945
	4	0.0360418	3.664
	8	0.0215904	6.135
24	1	2.2233	1

	2	1.13713	1.955
	4	0.604661	3.675
	8	0.334551	6.646
28	1	39.7622	1
	2	19.9952	1.989
	4	10.6262	3.742
	8	5.81812	6.834
32	1	698.478	1
	2	351.165	1.989
	4	186.765	3.739
	8	100.884	6.929

b) k = 1

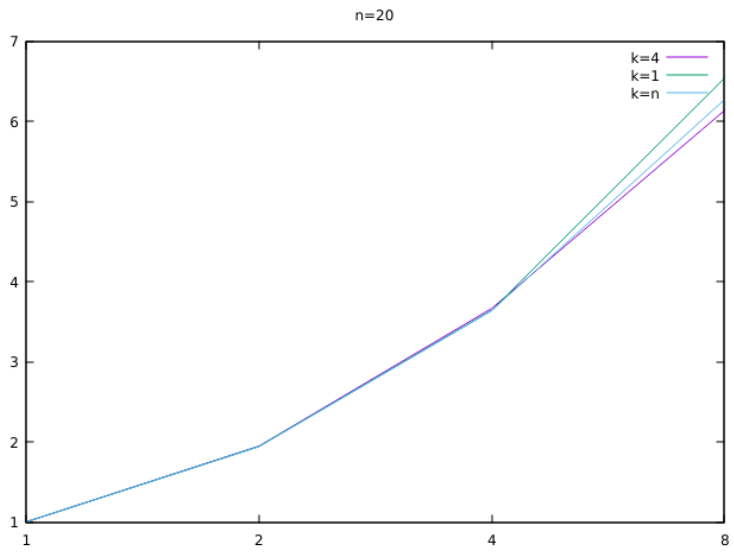
Количество кубитов	Количество процессов	Время работы (сек)	Ускорение
20	1	0.134273	1
	2	0.0692042	1.942
	4	0.0367729	3.641
	8	0.0205473	6.537
24	1	2.33588	1
	2	1.20078	1.947
	4	0.631853	3.697
	8	0.351026	6.655
28	1	39.9546	1
	2	20.2936	1.969
	4	10.7427	3.721
	8	6.01629	6.641
32	1	703.637	1
	2	371.93	1.892
	4	187.964	3.744
	8	101.502	6.932

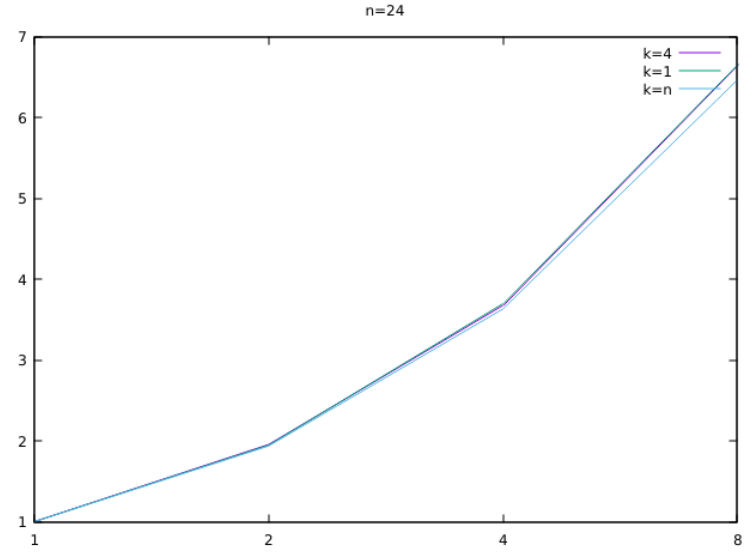
c) k = n

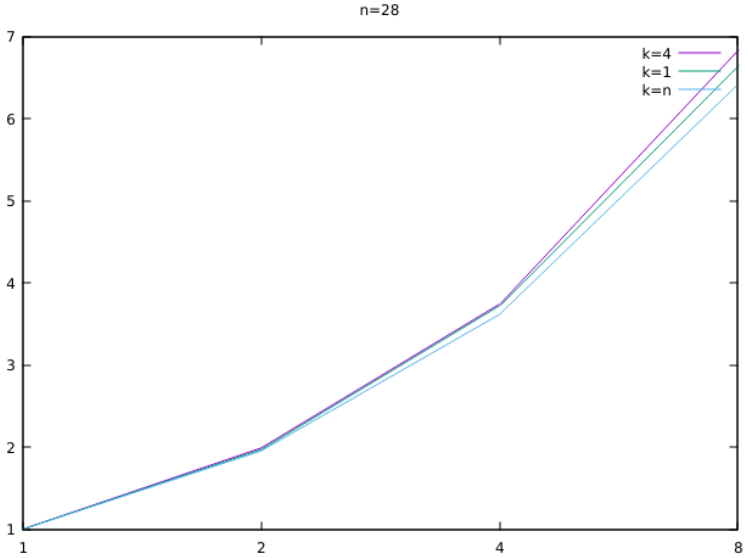
Количество кубитов	Количество процессов	Время работы (сек)	Ускорение
20	1	0.116066	1

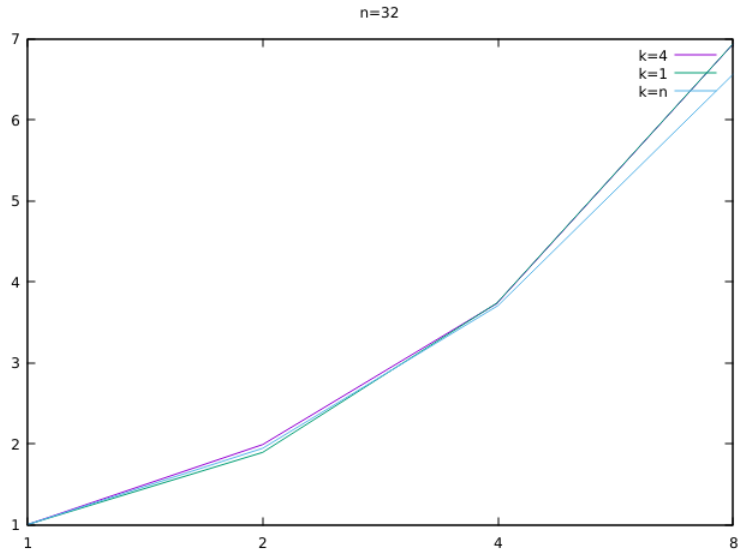
	2	0.0598536	1.939
	4	0.0319883	3.636
	8	0.0184708	6.271
24	1	1.96494	1
	2	1.01498	1.936
	4	0.541044	3.632
	8	0.304091	6.464
28	1	32.798	1
	2	16.7934	1.953
	4	9.0708	3.616
	8	5.11285	6.415
32	1	540.775	1
	2	278.472	1.942
	4	146.006	3.704
	8	82.4097	6.562

Графики зависимости ускорения от числа процессов для n = 20, 24, 28, 32 и k = 1, 4, n:




n=24
k=4
k=1
k=n
7
6
5
4
3
2
1
1
2
4
8


n=28
k=4
k=1
k=n
7
6
5
4
3
2
1
1
2
4
8


n=32
k=4
k=1
k=n
7
6
5
4
3
2
1
1
2
4
8