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

Отчёт
Однокубитное преобразование вектора-состояния

Работу выполнил
Имашев В.Р.

Москва 2021

Задание

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. Для работы с комплексными числами возможно использование стандартной библиотеки шаблонов.

Определить максимальное количество кубитов, для которых возможна работа программы на системе Polus. Выполнить теоретический расчет и проверить его экспериментально.

Протестировать программу на системе Polus. В качестве теста использовать преобразование Адамара по номеру кубита:
a) который соответствует номеру в списке группы плюс 1.
b) 1
c) n

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

Описание алгоритма

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

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

Итак, дана комплексная матрица:
и *k* - номер индекса от 1 до *n* (номер кубита).

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

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

Тестирование на Polus и расчет максимального числа кубитов

Тестирование программы проводилось на Polus. На следующей странице приведена таблица, содержащая информацию о результатах выполнения программы на вычислительном комплексе.

Объем оперативной памяти вычислительного комплекса 256 Гб = $2^8 \cdot 2^{30}$ байт. Учитывая, что sizeof(double) = 8 байт, а тип complexd хранит в себе два элемента типа double (действительная и мнимая части комплексного числа), то sizeof(complexd) = 2^4 байт. Так как вектор состояний состоит из 2^n элементов типа complexd, то, пренебрегая остальными расходами оперативной памяти, получаем верхнюю оценку: n < 34.

Замер времени производился с помощью функции omp_get_wtime().

Результаты выполнения

Количество кубитов	Количество процессоров	Время работы программы(сек)			Ускорение		
		k =1	k = 8	k = n	k = 1	k = 8	k = n
20	1	0,1239510	0,0933440	0,0933187	1,0000000	1,0000000	1,0000000
	2	0,0637496	0,0467116	0,0466794	1,9443416	1,9983045	1,9991409
	4	0,0423556	0,0332707	0,0459866	2,9264371	2,8055917	2,0292585
	8	0,0235531	0,0170099	0,0170317	5,2626194	5,4876278	5,4791183
	160	0,0122506	0,007375	0,00730542	10,11795341	12,65681356	12,77389938
24	1	1,8314000	1,8333700	1,4933700	1,0000000	1,0000000	1,0000000
	2	0,7612040	0,7646970	0,7625260	2,4059253	2,3975117	1,9584513
	4	0,3741870	0,3837470	0,4965240	4,8943443	4,7775487	3,0076492
	8	0,2489030	0,2443070	0,5994030	7,3578864	7,5043695	2,4914290
	160	0,0485891	0,0814751	0,047674	37,69158103	22,50221233	31,32462139
28	1	24,5510000	24,0445000	24,1510000	1,0000000	1,0000000	1,0000000
	2	15,0338000	21,3335000	11,9898000	1,6330535	1,1270771	2,0142955
	4	6,1577100	6,2646100	5,9912300	3,9870341	3,8381479	4,0310587
	8	3,5601200	3,2521200	3,2979500	6,8961159	7,3934849	7,3230340
	160	1,0288345	0,9421239	0,9723211	23,8629245	25,5215901	24,83850242
максимальное (удалось при 30)	1	103,793238	97,790523	96,887222	1,0000000	1,0000000	1,0000000
	2	51,117445	49,630346	49,500590	2,030485639	1,970377619	1,957294287
	4	32,737021	27,343853	26,914656	3,170515668	3,576325655	3,599794179
	8	13,689873	13,188012	15,002064	7,581753169	7,41510722	6,458259477
	160	7,143421	7,810027	6,902296	14,52990633	12,52115044	14,03695553

Выводы

Номер преобразуемого кубита не повлиял на время работы программы. При росте числа кубитов наблюдается высокий рост времени работы программы (например, при 8 нитях на 24 кубитах – 0,24 сек, а при 28 кубитах уже почти 4 сек). Наилучшее ускорение наблюдается на 24 кубитах при 160 нитях (почти в 40 раз).