



Практикум на ЭВМ: 6 семестр.

Отчёт № 1.

**Анализ параллельной программы на OpenMP,
реализующей однокубитное квантовое преобразование**

Работу выполнил

Федоров В. В.

Постановка задачи и формат данных.

Задача: Реализовать параллельный алгоритм однокубитного квантового преобразования с использованием OpenMP, оценить максимально возможное число кубитов, на котором система Polus может выполнить программу, проанализировать зависимость времени выполнения программы от числа кубитов, номера преобразуемого кубита и числа нитей.

Формат командной строки: <количество кубитов n> <номер кубита k, над которым выполняется преобразование> <число нитей>

Оценка максимального числа кубитов

Размер общей оперативной памяти системы Polus составляет 256 Гб. Т.к. `sizeof(double) == 8`, то `sizeof(complexd) == 16`, следовательно, размер массива для вектора состояний n кубитов равен $16 * 2^n$. Обозначим за ε все остальные затраты программы по памяти — они не зависят от n. Получим неравенство:

$$16 * 2^n + \varepsilon \leq 256 * 1024^3$$

$$2^n < 2^{34}$$

$$n_{\max} = 33$$

Результаты выполнения

На практике в виду временных ограничений на системе Polus максимальное n, на котором удалось протестировать программу, равно 30.

Результаты для k = 1

Кол-во кубитов	Кол-во нитей	Время работы, с	Ускорение	Эффективность
20	1	0,090729	1,000000	1,000000
	2	0,044946	2,018609	1,009304
	4	0,022644	4,006792	1,001698
	8	0,011845	7,659494	0,957437
	16	0,011934	7,602819	0,475176
	32	0,007701	11,781778	0,368181
	64	0,008558	10,601969	0,165656
	128	0,008558	10,601548	0,082825
	160	0,009944	9,123921	0,057025
24	1	1,428130	1,000000	1,000000
	2	0,718396	1,987943	0,993971
	4	0,360347	3,963208	0,990802
	8	0,189141	7,550610	0,943826
	16	0,165418	8,633462	0,539591
	32	0,108348	13,180954	0,411905
	64	0,092415	15,453527	0,241461
	128	0,091219	15,656127	0,122313

	160	0,081902	17,437102	0,108982
28	1	22,885500	1,000000	1,000000
	2	11,546400	1,982046	0,991023
	4	5,828720	3,926334	0,981583
	8	3,069280	7,456309	0,932039
	16	2,060340	11,107633	0,694227
	32	1,644190	13,919012	0,434969
	64	1,348370	16,972715	0,265199
	128	0,980996	23,328841	0,182257
	160	0,908224	25,198079	0,157488
30	1	97,262900	1,000000	1,000000
	2	49,116700	1,980241	0,990120
	4	24,638200	3,947646	0,986912
	8	12,745900	7,630917	0,953865
	16	8,066340	12,057873	0,753617
	32	6,158720	15,792713	0,493522
	64	5,119280	18,999332	0,296865
	128	3,387770	28,710007	0,224297
	160	3,282350	29,632093	0,185201

Результаты для k = 13

Кол-во кубитов	Кол-во нитей	Время работы, с	Ускорение	Эффективность
20	1	0,089420	1,000000	1,000000
	2	0,044967	1,988574	0,994287
	4	0,022702	3,938929	0,984732
	8	0,011844	7,549942	0,943743
	16	0,011928	7,496458	0,468529
	32	0,006998	12,778210	0,399319
	64	0,008386	10,663569	0,166618
	128	0,008720	10,254528	0,080114
	160	0,010628	8,414020	0,052588
24	1	1,430410	1,000000	1,000000
	2	0,718047	1,992084	0,996042
	4	0,361283	3,959251	0,989813
	8	0,189097	7,564425	0,945553
	16	0,181014	7,902206	0,493888

	32	0,108478	13,186176	0,412068
	64	0,094700	15,104678	0,236011
	128	0,079881	17,906829	0,139897
	160	0,083562	17,117989	0,106987
28	1	22,906300	1,000000	1,000000
	2	11,568300	1,980092	0,990046
	4	5,795170	3,952654	0,988163
	8	3,017110	7,592133	0,949017
	16	2,084890	10,986815	0,686676
	32	1,617420	14,162246	0,442570
	64	1,357910	16,868791	0,263575
	128	0,971263	23,584034	0,184250
	160	0,938806	24,399397	0,152496
30	1	91,899900	1,000000	1,000000
	2	46,273600	1,986011	0,993006
	4	23,538900	3,904171	0,976043
	8	12,100400	7,594782	0,949348
	16	7,649320	12,014127	0,750883
	32	6,079820	15,115563	0,472361
	64	4,999900	18,380348	0,287193
	128	3,569550	25,745514	0,201137
	160	3,045350	30,177122	0,188607

Результаты для k = n

Кол-во кубитов	Кол-во нитей	Время работы, с	Ускорение	Эффективность
20	1	0,089277	1,000000	1,000000
	2	0,044875	1,989451	0,994725
	4	0,022624	3,946084	0,986521
	8	0,011852	7,532937	0,941617
	16	0,011893	7,506462	0,469154
	32	0,007001	12,751086	0,398471
	64	0,008182	10,911595	0,170494
	128	0,008724	10,233870	0,079952
	160	0,009316	9,582756	0,059892
24	1	1,428410	1,000000	1,000000
	2	0,717783	1,990030	0,995015

	4	0,359608	3,972131	0,993033
	8	0,188000	7,597926	0,949741
	16	0,166292	8,589770	0,536861
	32	0,108490	13,166283	0,411446
	64	0,095821	14,907144	0,232924
	128	0,081405	17,546892	0,137085
	160	0,083033	17,203002	0,107519
28	1	23,039300	1,000000	1,000000
	2	11,571300	1,991073	0,995536
	4	5,846770	3,940518	0,985129
	8	3,030090	7,603504	0,950438
	16	2,140580	10,763111	0,672694
	32	1,669480	13,800285	0,431259
	64	1,371710	16,796043	0,262438
	128	1,056410	21,809051	0,170383
	160	0,900442	25,586656	0,159917
30	1	92,254600	1,000000	1,000000
	2	46,490600	1,984371	0,992186
	4	23,369400	3,947667	0,986917
	8	12,319200	7,488684	0,936086
	16	7,475220	12,341389	0,771337
	32	5,758970	16,019288	0,500603
	64	4,935360	18,692578	0,292072
	128	3,358700	27,467353	0,214589
	160	3,299000	27,964413	0,174778

Основные выводы.

Распределение эффективности нитей типично для системы Polus — она близка к единице при числе нитей до восьми, а затем резко падает в связи с добавлением большего числа ядер. Время выполнения для разных k варьировалось незначительно, однако в среднем быстрее всего программа работала при $k = n$, на втором месте — $k = 13$.