



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	0,045954	1,974350	0,987175
	4	0,023142	3,920521	0,980130
	8	0,012143	7,471474	0,933934
	16	0,011921	7,611182	0,475699
	32	0,006986	12,987702	0,405866
	64	0,009165	9,899185	0,154675
	128	0,008146	11,138117	0,087017
	160	0,010353	8,763387	0,054771
24	1	1,451250	1,000000	1,000000
	2	0,731958	1,982696	0,991348
	4	0,369511	3,927488	0,981872
	8	0,195087	7,438989	0,929874
	16	0,182957	7,932192	0,495762
	32	0,108117	13,422958	0,419467
	64	0,091711	15,824234	0,247254
	128	0,084625	17,149248	0,133979

	160	0,079054	18,357658	0,114735
28	1	23,556500	1,000000	1,000000
	2	11,821000	1,992767	0,996384
	4	5,984600	3,936186	0,984047
	8	3,114280	7,564028	0,945503
	16	2,107040	11,179902	0,698744
	32	1,648070	14,293386	0,446668
	64	1,339960	17,580002	0,274688
	128	1,017280	23,156358	0,180909
	160	0,899757	26,180958	0,163631
30	1	99,199200	1,000000	1,000000
	2	49,632900	1,998658	0,999329
	4	25,113600	3,950019	0,987505
	8	13,218900	7,504346	0,938043
	16	8,041650	12,335677	0,770980
	32	6,430370	15,426671	0,482083
	64	4,986330	19,894231	0,310847
	128	3,524780	28,143373	0,219870
	160	3,307140	29,995464	0,187472

Результаты для k = 13

Кол-во кубитов	Кол-во нитей	Время работы, с	Ускорение	Эффективность
20	1	0,090968	1,000000	1,000000
	2	0,045857	1,983751	0,991876
	4	0,023172	3,925734	0,981434
	8	0,012122	7,504550	0,938069
	16	0,011966	7,602198	0,475137
	32	0,006991	13,012274	0,406634
	64	0,006729	13,519267	0,211239
	128	0,008859	10,268022	0,080219
	160	0,010076	9,027817	0,056424
24	1	1,451820	1,000000	1,000000
	2	0,732075	1,983157	0,991579
	4	0,369442	3,929764	0,982441
	8	0,192920	7,525503	0,940688
	16	0,182512	7,954655	0,497166

	32	0,108273	13,408883	0,419028
	64	0,090368	16,065660	0,251026
	128	0,079394	18,286361	0,142862
	160	0,082006	17,703762	0,110649
28	1	23,261800	1,000000	1,000000
	2	11,805600	1,970404	0,985202
	4	6,014940	3,867337	0,966834
	8	3,108380	7,483577	0,935447
	16	2,125690	10,943176	0,683949
	32	1,641810	14,168387	0,442762
	64	1,377120	16,891629	0,263932
	128	1,066280	21,815846	0,170436
	160	0,967533	24,042384	0,150265
30	1	94,205700	1,000000	1,000000
	2	47,672500	1,976102	0,988051
	4	23,916600	3,938925	0,984731
	8	12,237900	7,697865	0,962233
	16	7,488970	12,579260	0,786204
	32	5,708600	16,502417	0,515701
	64	5,039030	18,695205	0,292113
	128	3,490420	26,989789	0,210858
	160	3,206790	29,376947	0,183606

Результаты для k = n

Кол-во кубитов	Кол-во нитей	Время работы, с	Ускорение	Эффективность
20	1	0,090708	1,000000	1,000000
	2	0,045776	1,981569	0,990784
	4	0,023149	3,918403	0,979601
	8	0,012098	7,498008	0,937251
	16	0,011970	7,578000	0,473625
	32	0,006994	12,968887	0,405278
	64	0,010873	8,342567	0,130353
	128	0,008364	10,845661	0,084732
	160	0,009121	9,944908	0,062156
24	1	1,451730	1,000000	1,000000
	2	0,730650	1,986902	0,993451

	4	0,369294	3,931096	0,982774
	8	0,190646	7,614794	0,951849
	16	0,182439	7,957345	0,497334
	32	0,108171	13,420695	0,419397
	64	0,095595	15,186270	0,237285
	128	0,082504	17,595853	0,137468
	160	0,082228	17,654935	0,110343
28	1	23,252600	1,000000	1,000000
	2	11,915800	1,951409	0,975705
	4	5,988970	3,882571	0,970643
	8	3,109770	7,477273	0,934659
	16	2,076680	11,197007	0,699813
	32	1,620480	14,349205	0,448413
	64	1,342210	17,324115	0,270689
	128	1,003230	23,177736	0,181076
	160	0,867702	26,797910	0,167487
30	1	94,154500	1,000000	1,000000
	2	47,920300	1,964814	0,982407
	4	23,957100	3,930129	0,982532
	8	12,301900	7,653655	0,956707
	16	7,428730	12,674374	0,792148
	32	6,118050	15,389626	0,480926
	64	5,028960	18,722460	0,292538
	128	3,304050	28,496693	0,222630
	160	3,189950	29,515980	0,184475

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 — $k = 13$.