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

Практикум на ЭВМ: 6 семестр.

Отчёт № 2.

Анализ параллельной программы на MPI, реализующей однокубитное квантовое преобразование

Работу выполнил

Федоров В. В.

Москва 2021

Постановка задачи и формат данных.

Задача: Реализовать параллельный алгоритм однокубитного квантового преобразования с использованием MPI, проанализировать зависимость времени выполнения программы от числа кубитов, номера преобразуемого кубита и числа нитей на системе Polus.

Формат командной строки: Первый аргумент — единственный символ, означающий один из двух режимов работы:

- `g` — вектор состояния инициализируется случайными значениями. Дальнейшие аргументы командной строки: <число кубитов n> <номер преобразовываемого кубита k> <имя файла для вывода времени выполнения>
- `f` — вектор состояния считывается из файла, вектор-результат также записывается в файл. Дальнейшие аргументы командной строки: <номер преобразовываемого кубита k> <имя файла для считывания вектора> <имя файла для записи вектора>
    - Вектор записывается в файл в бинарном формате — сначала идут 4 бита числа кубитов n, затем — 2^n пар восьмибайтовых чисел с плавающей точкой, соответствующих действительной и мнимой части соответствующего элемента вектора.

Описание программы

В предположении что число процессов $p=2^s$, где $s\in\mathbb{N}, s\leq\frac{n}{2}$ разделим вектор состояния на 2^s+1 блоков одного размера, тогда каждому процессу достается 2 блока. В таком случае либо блоки можно будет разделить между процессами так, чтобы элементы двух блоков конкретного процесса были попарно соседними по k-му кубиту, либо соседние элементы уже находятся в одном и том же блоке, и тогда блоки можно распределить между процессорами как угодно, но для простоты будем выдавать процессам пары соседних блоков. Рассмотрим пример с n = 4 и s = 2:

Номер блока	Номер элемента
0	0000
	0001
1	0010
	0011
2	0100
	0101
3	0110
	0111
4	1000
	1001
5	1010
	1011
6	1100
	1101

7	1110
	1111

k\pr	0	1	2	3
1	0,4	1,5	2,6	3,7
2	0,2	1,3	4,6	5,7
3	0,1	2,3	4,5	6,7
4	0,1	2,3	4,5	6,7

Таким образом, составив пользовательский тип MPI и заполнив массив сдвигов, можно распределить вектор состояния между процессами.

Тестирование

Программа тестировалась при помощи режима «f» на 5 заранее сгенерированных файлах с n = 16 для всех k от 1 до 16 и для всех количеств процессов от 1 до 16, равных степени двойки.

Результаты выполнения

Результаты для k = 1

Кол-во кубитов	Кол-во нитей	Время работы, с	Ускорение	Эффективность
25	1	3,62268	1,000000	1,000000
	2	1,90078	1,905891	0,952946
	4	1,16596	3,107036	0,776759
	8	0,737274	4,913614	0,614202
	16	0,502079	7,215359	0,450960
26	1	6,84707	1,000000	1,000000
	2	3,71309	1,844036	0,922018
	4	2,05984	3,324079	0,831020
	8	1,38629	4,939133	0,617392
	16	0,865623	7,909991	0,494374
27	1	14,6193	1,000000	1,000000
	2	7,63231	1,915449	0,957724
	4	4,31039	3,391642	0,847911
	8	2,69613	5,422328	0,677791
	16	2,07385	7,049353	0,440585

Результаты для k = 13

Кол-во кубитов	Кол-во нитей	Время работы, с	Ускорение	Эффективность
25	1	3,56405	1,000000	1,000000
	2	1,93638	1,840574	0,920287

	4	1,09994	3,240222	0,810056
	8	0,705744	5,050061	0,631258
	16	0,510674	6,979110	0,436194
26	1	7,0877	1,000000	1,000000
	2	3,76441	1,882818	0,941409
	4	2,1989	3,223293	0,805823
	8	1,45129	4,883724	0,610466
	16	0,995078	7,122758	0,445172
27	1	13,7044	1,000000	1,000000
	2	7,41727	1,847634	0,923817
	4	4,68312	2,926340	0,731585
	8	2,8776	4,762441	0,595305
	16	1,91683	7,149512	0,446845

Результаты для k = n

Кол-во кубитов	Кол-во нитей	Время работы, с	Ускорение	Эффективность
25	1	3,44666	1,000000	1,000000
	2	1,86464	1,848432	0,924216
	4	1,17342	2,937277	0,734319
	8	0,731877	4,709343	0,588668
	16	0,536817	6,420549	0,401284
26	1	6,8547	1,000000	1,000000
	2	4,02094	1,704751	0,852375
	4	2,33394	2,936965	0,734241
	8	1,51025	4,538785	0,567348
	16	1,00457	6,823517	0,426470
27	1	15,9937	1,000000	1,000000
	2	8,61516	1,856460	0,928230
	4	4,65193	3,438078	0,859520
	8	2,97121	5,382891	0,672861
	16	1,92467	8,309840	0,519365

Основные выводы.

Какой-либо заметной разницы во времени выполнения между различными k не выявлено, зависимость эффективности от числа процессов довольно типична.