

Московский государственный университет имени М. В. Ломоносова
Факультет вычислительной математики и кибернетики



Параллельные высокопроизводительные вычисления

Отчет по заданию №2
«Параллельная Сортировка Бэтчера»

Бирюков Андрей Матвеевич
528 группа

27 ноября 2022 года

1. Описание условия

Дана структура (или Класс)

```
Point{  
    float coord[2];  
    int index;  
} P[n1*n2];
```

Данная структура будет использоваться для работы с регулярной сеткой.

Точки данной сетки имеют координаты

$P[i * n2 + j].coord[0] = x(i, j)$,

$P[i * n2 + j].coord[1] = y(i, j)$, где $i = 0, \dots, n1 - 1, j = 0, \dots, n2 - 1$.

Индекс определяется соотношением $P[i*n2+j].index = i*n2+j$

Особенности: Для инициализации координат можете использовать функции, принимающие на вход параметры (i, j), то есть фактически каждая точка этой сетки однозначно определяется (i, j).

На входе: На каждом процессе одинаковое количество элементов структуры Point. (Если на некоторых процессах элементов структуры Point меньше чем во всех остальных, тогда необходимо ввести фиктивные элементы, например, с отрицательным значением индекса.)

Цель: Разработать и реализовать алгоритм, обеспечивающий параллельную сортировку методом Бэтчера массива или части массива структур Point, вдоль каждой из координат (x или y) в соответствии с заданным параметром. Следует реализовать и сортировку на каждом отдельном процессе и сеть сортировки Бэтчера.

На выходе: на каждом процессе одинаковое количество элементов структуры Point. Каждый элемент структуры Point одного процесса находится левее по координате x (или y) по сравнению с элементом структуры Point любого процесса с большим рангом, за исключением фиктивных элементов.

2. Описание метода решения

Реализация сортировки Бэтчера была основана на построенной сети сортировки в первом задании.

На каждом процессе генерируются вектора размером $(n1 * n2) / p$. Если на некоторых процессах элементов меньше, тогда незанятое пространство дополняется числами FLT_MAX.

Каждый процесс сортирует свой фрагмент с помощью стандартного алгоритма *std::sort*.

Сортировка проводилась при помощи алгоритма «сети обменной сортировки со слиянием Бэтчера», который изучался в процессе первого задания. Алгоритм описан в книге М. В. Якубовского «Введение в параллельные методы решения задач», 2013г, глава 5, страница 145.

В рамках описания алгоритма были реализованы функции В и S.

Слияние упорядоченных фрагментов, на основе конфигурируемых компараторов в функции S, было реализовано методом, который был рассказан на 5 лекции (слайд 74+ - https://lira.imamod.ru/msu202209/L05_20221017_YakobovskiyMV.pdf). В рамках слияния были реализованы функции *processes_interaction* и *Join*.

Для проверки правильности сортировки проверяется корректность сортировки внутри каждого процесса или при помощи MPI_Gather собираются все фрагменты на 0 процессе и проверяется соединенный массив.

Запуск программы производился при помощи написанного bash скрипта. (прикреплен в архиве).

3. Описание используемой вычислительной системы

Запуск программы проводился на вычислительном комплексе *IBM Blue Gene/P*.

IBM Blue Gene/P — массивно-параллельная вычислительная система, которая состоит из двух стоек, включающих 8192 процессорных ядер (2 x 1024 четырехъядерных вычислительных узлов), с пиковой производительностью 27,9 терафлопс (27,8528 триллионов операций с плавающей точкой в секунду).

Характеристики системы:

- две стойки с вычислительными узлами и узлами ввода-вывода
- 1024 четырехъядерных вычислительных узла в каждой из стоек
- 16 узлов ввода-вывода в стойке (в текущей конфигурации активны 8, т.е. одна I/O-карта на 128 вычислительных узлов)
- выделенные коммуникационные сети для межпроцессорных обменов и глобальных операций
- программирование с использованием MPI, OpenMP/threads, POSIX I/O
- высокая энергоэффективность: ~ 372 MFlops/W (см. список Green500)
- система воздушного охлаждения
- латентность (ближайший сосед): 0.1 μ s (32-б. пакет), 0.8 μ s (256-б. пакет)
- Пропускная способность интерконнекта 425 MB/s

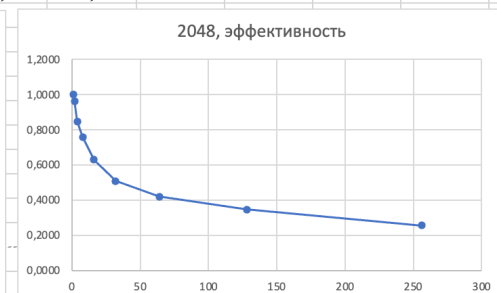
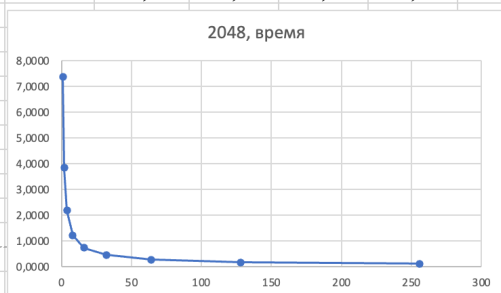
4. Результаты численных экспериментов

Сортировка проводилась на сетках 1024 x 1024, 2048 x 2048, 4096 x 4096, 6144 x 6144, 8192 x 8192 по координате x.

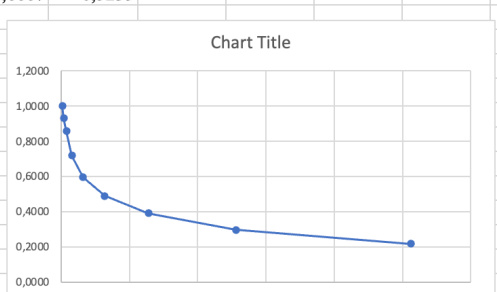
По результатам были составлены таблицы и графики, где:

- N – сторона сетки $n * n$ (ниже произведение)
- p – количество процессов
- T – время сортировки в секундах
- S – ускорение
- E – эффективность
- E max – максимально возможная эффективность
- S max – максимально возможное ускорение

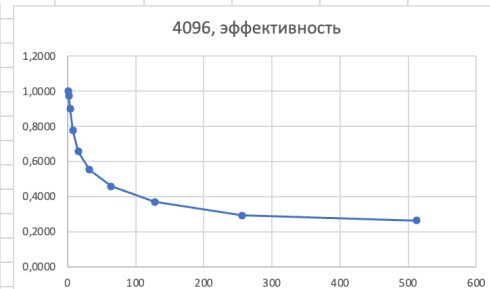
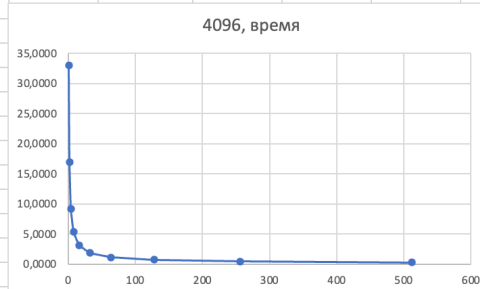
N	p	T	S	E	E max	S max	E / E max
2048,0000	1	7,3727	1,0000	1,0000	1,0000	1,0000	1,0000
4194304,0000	2	3,8323	1,9238	0,9619	1,0000	2,0000	0,9619
	4	2,1781	3,3850	0,8462	0,9565	3,8261	0,8847
	8	1,2155	6,0654	0,7582	0,8800	7,0400	0,8616
	16	0,7312	10,0827	0,6302	0,7857	12,5714	0,8020
	32	0,4538	16,2470	0,5077	0,6875	22,0000	0,7385
	64	0,2741	26,8933	0,4202	0,5946	38,0541	0,7067
	128	0,1664	44,2998	0,3461	0,5116	65,4884	0,6765
	256	0,1122	65,6995	0,2566	0,4400	112,6400	0,5833



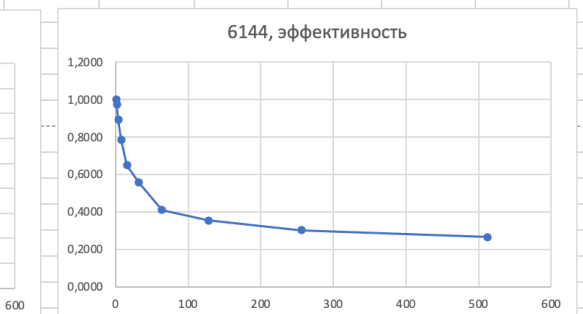
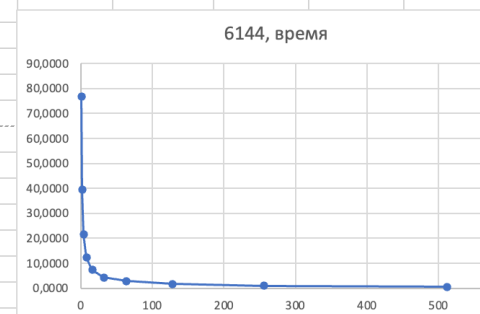
N	p	T	S	E	E max	S max	E / E max
1024,0000	1	1,6520	1,0000	1,0000	1,0000	1,0000	1,0000
1048576,0000	2	0,8876	1,8613	0,9307	1,0000	2,0000	0,9307
	4	0,4821	3,4268	0,8567	0,9524	3,8095	0,8995
	8	0,2875	5,7463	0,7183	0,8696	6,9565	0,8260
	16	0,1736	9,5159	0,5947	0,7692	12,3077	0,7732
	32	0,1057	15,6245	0,4883	0,6667	21,3333	0,7324
	64	0,0660	25,0271	0,3910	0,5714	36,5714	0,6843
	128	0,0435	37,9889	0,2968	0,4878	62,4390	0,6084
	256	0,0296	55,8406	0,2181	0,4167	106,6667	0,5235



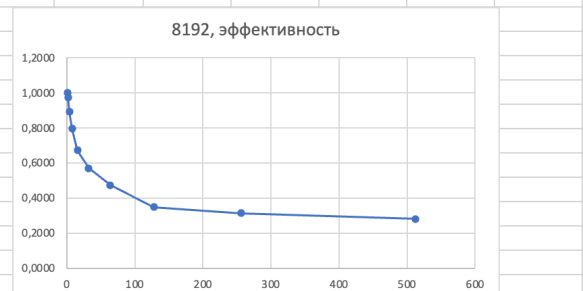
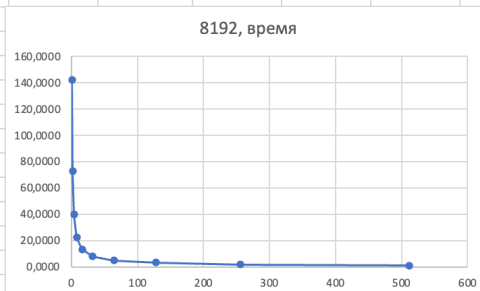
N	p	T	S	E	E max	S max	E / E max												
4096,0000	1	32,9491	1,0000	1,0000	1,0000	1,0000	1,0000												
16777216,0000	2	16,9295	1,9463	0,9731	1,0000	2,0000	0,9731												
	4	9,1512	3,6005	0,9001	0,9600	3,8400	0,9376												
	8	5,2995	6,2174	0,7772	0,8889	7,1111	0,8743												
	16	3,1333	10,5159	0,6572	0,8000	12,8000	0,8216												
	32	1,8642	17,6746	0,5523	0,7059	22,5882	0,7825												
	64	1,1223	29,3593	0,4587	0,6154	39,3846	0,7455												
	128	0,6954	47,3821	0,3702	0,5333	68,2667	0,6941												
	256	0,4417	74,5944	0,2914	0,4615	118,1538	0,6313												
	512	0,2442	134,9168	0,2635	0,4000	204,8000	0,6588												



N	p	T	S	E	E max	S max	E / E max												
6144,0000	1	76,6994	1,0000	1,0000	1,0000	1,0000	1,0000												
37748736,0000	2	39,4262	1,9454	0,9727	1,0000	2,0000	0,9727												
	4	21,5255	3,5632	0,8908	0,9618	3,8472	0,9262												
	8	12,2203	6,2764	0,7845	0,8935	7,1480	0,8781												
	16	7,3993	10,3658	0,6479	0,8075	12,9201	0,8023												
	32	4,3017	17,8299	0,5572	0,7157	22,9013	0,7786												
	64	2,9223	26,2465	0,4101	0,6266	40,1015	0,6545												
	128	1,6950	45,2504	0,3535	0,5452	69,7803	0,6485												
	256	0,9923	77,2925	0,3019	0,4734	121,1869	0,6378												
	512	0,5653	135,6817	0,2650	0,4115	210,6755	0,6440												



N	p	T	S	E	E max	S max	E / E max												
8192,0000	1	142,1810	1,0000	1,0000	1,0000	1,0000	1,0000												
67108864,0000	2	72,9579	1,9488	0,9744	1,0000	2,0000	0,9744												
	4	39,7905	3,5732	0,8933	0,9630	3,8519	0,9277												
	8	22,3455	6,3628	0,7954	0,8966	7,1724	0,8871												
	16	13,2102	10,7630	0,6727	0,8125	13,0000	0,8279												
	32	7,8073	18,2112	0,5691	0,7222	23,1111	0,7880												
	64	4,6933	30,2945	0,4734	0,6341	40,5854	0,7464												
	128	3,1819	44,6843	0,3491	0,5532	70,8085	0,6311												
	256	1,7755	80,0812	0,3128	0,4815	123,2593	0,6497												
	512	0,9895	143,6865	0,2806	0,4194	214,7097	0,6692												



5. Анализ полученных результатов

Анализ проводился на основе статьи М. В. Якубовского:
<https://lira.imamod.ru/FondProgramm/Sort/ParallelSort.pdf>

Большая часть вычисленных данных представлены в таблицах из пункта 4

Максимальное значение коэффициента эффективности использования
вычислительной мощности дается выражением:

$$E^{\max}(n, p) = \frac{t(n, 1)}{pt(n, p)} = \frac{\log_2 n}{\log_2 n + s_p - \log_2 p} \approx \frac{1}{1 + \log_n p (\log_2 p - 1) / 2}$$

Число шагов слияния:

$$s_p \approx \frac{\lceil \log_2 p \rceil (\lceil \log_2 p \rceil + 1)}{2}$$

$s(1) = 0, s(2) = 1, s(4) = 3, s(8) = 6, s(16) = 10, s(32) = 15, s(64) = 21, s(128) = 28, s(256) = 36, s(512) = 45$

T1/[N*log(N)]:

1024: 7.8773e-08

2048: 7.9899e-08

4096: 8.1830e-08

6144: 8.0725e-08

8192: 8.1487e-08

6. Используемые материалы:

1. М. В. Якубовского «Введение в параллельные методы решения задач», 2013г
2. <https://lira.imamod.ru/FondProgramm/Sort/ParallelSort.pdf>
3. https://lira.imamod.ru/msu202209/L05_20221017_YakobovskiyMV.pdf

7. Приложение

Код сортировки реализован на языке C++: файл `batcher.cpp`

Excel файл с подсчетом результатов: `results.xlsx`

`run.sh` – `bash` скрипт, при помощи которого запускалась программа на суперкомпьютере.