



Министерство науки и высшего образования Российской
Федерации
Федеральное государственное бюджетное образовательное
учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА «ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ
ТЕХНОЛОГИИ» (ИУ7)

ДОМАШНЕЕ ЗАДАНИЕ

Дисциплина: Программирование параллельных процессов

Студент

ИУ7И-31М

(Группа)

(Подпись, дата)

С.Р. Диас

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

А.П.Ковтушенко

(И.О. Фамилия)

Москва, 2024

Содержание

Задание	2
Выбор алгоритма	2
Загрузка процессоров.....	2
Зависимость времени вычислений от размерности задачи и количества процессов	2
Источники	6

Задание

На шахматной доске размером $N \times M$ построить путь обхода конем всех клеток доски. Каждую клетку следует посетить только один раз. Обосновать проектное решение (выбор алгоритма). Обеспечить равномерную загрузку процессоров. Результат вывести в текстовый файл. Исследовать зависимость времени счёта от размерности задачи и количества процессоров.

Выбор алгоритма

Был выбран алгоритм, описанный в статье «Optimal algorithms for constructing knight's tours on arbitrary $n \times m$ chessboards» [1]. Этот алгоритм состоит из следующих шагов:

1. Разделить таблицу на 4 подтаблицы и повторять процесс для каждой части, пока размер всех подтаблиц не станет 10×10 или меньше.
2. Для каждой подтаблицы выбрать заранее рассчитанное решение.
3. Объединить все подтаблицы (существуют специфические правила объединения подтаблиц).

Алгоритм был выбран по следующим причинам:

1. Алгоритм имеет линейную сложность.
2. Алгоритм предполагает разделение таблицы на множество подтаблиц, для каждой из которых решение можно рассчитывать независимо.
3. Алгоритм всегда находит решение, если оно существует.

Загрузка процессоров

Для распределения нагрузки на процессоры:

1. Главный процесс выполняет несколько итераций, пока количество подтаблиц не станет больше или равно количеству процессоров.
2. Каждый процесс получает S или $S + 1$ подтаблиц, где S — это количество подтаблиц, делённое на количество процессоров.
3. Каждый процесс решает задачу о ходе коня для своей подтаблицы.
4. Каждый процесс отправляет своё решение главному процессору (если используется MPI).
5. Главный процесс объединяет все полученные решения.

Зависимость времени вычислений от размерности задачи и количества процессоров

Были реализованы три версии выбранного алгоритма:

1. Последовательная.
2. Параллельная с использованием MPI.
3. Параллельная с использованием MPI + OpenMP.

Каждая версия была выполнена с использованием следующих размеров таблиц:

1. 100×101
2. 1000×1001

3. 5000 x 5005
4. 10000 x 10001

Результаты представлены в следующих таблицах и графиках:

	Версия алгоритма	Количество процессоров	Размер матрицы			
			100 x 101	1000 x 1001	5000 x 5005	10000 x 10001
Время в секундах	Последовательная	1	0.0079	0.765	19.8142	84.6715
	MPI	2	0.0045	0.4278	11.1468	48.7061
		4	0.0042	0.2663	6.4708	28.0229
		6	0.0094	0.3812	8.79271	40.6437
		8	0.009	0.5947	7.804	38.6621
	OpenMP	2	0.0589	0.3497	7.5285	31.6548
		4	0.0799	0.3453	6.6142	28.847
		6	0.3135	0.6257	8.2908	35.2838
		8	0.2951	0.6122	8.3759	33.9697

Таблица 1 Время выполнения в секундах

	Версия алгоритма	Количество процессоров	Размер матрицы			
			100 x 101	1000 x 1001	5000 x 5005	10000 x 10001
Ускорение	Последовательная	1	100%	100%	100%	100%
	MPI	2	176%	179%	178%	174%
		4	188%	287%	306%	302%
		6	84%	201%	225%	208%
		8	88%	129%	254%	219%
	OpenMP	2	13%	219%	263%	267%
		4	10%	222%	300%	294%
		6	3%	122%	239%	240%
		8	3%	125%	237%	249%

Таблица 2 Ускорение

	Версия алгоритма	Количество процессоров	Размер матрицы			
			100 x 101	1000 x 1001	5000 x 5005	10000 x 10001
E = Ускорение / Количество процессоров	Последовательная	1	100%	100%	100%	100%
	MPI	2	88%	89%	89%	87%
		4	47%	72%	77%	76%
		6	14%	33%	38%	35%
		8	11%	16%	32%	27%
	OpenMP	2	7%	109%	132%	134%
		4	2%	55%	75%	73%
		6	0%	20%	40%	40%
		8	0%	16%	30%	31%

Таблица 3 E = Ускорение / Количество процессоров

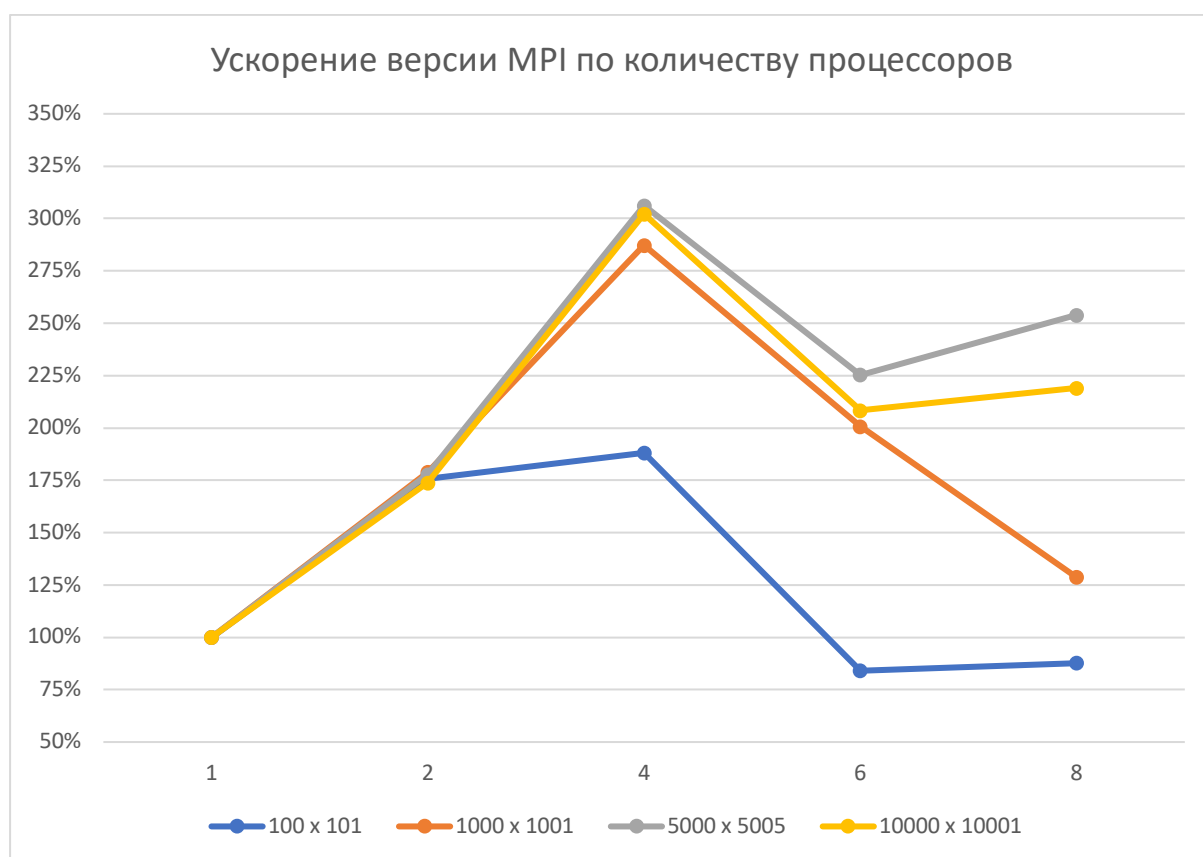


Рис. 1 Ускорение версии MPI

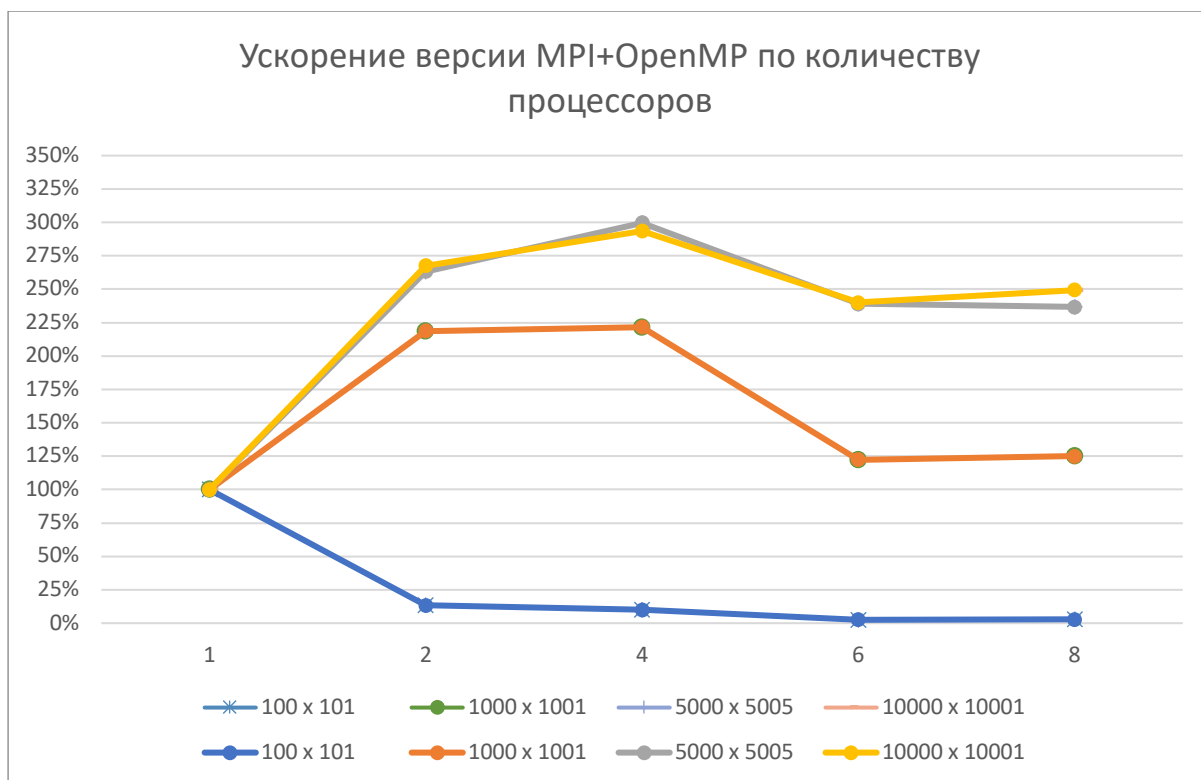


Рис. 2 Ускорение версии OpenMP

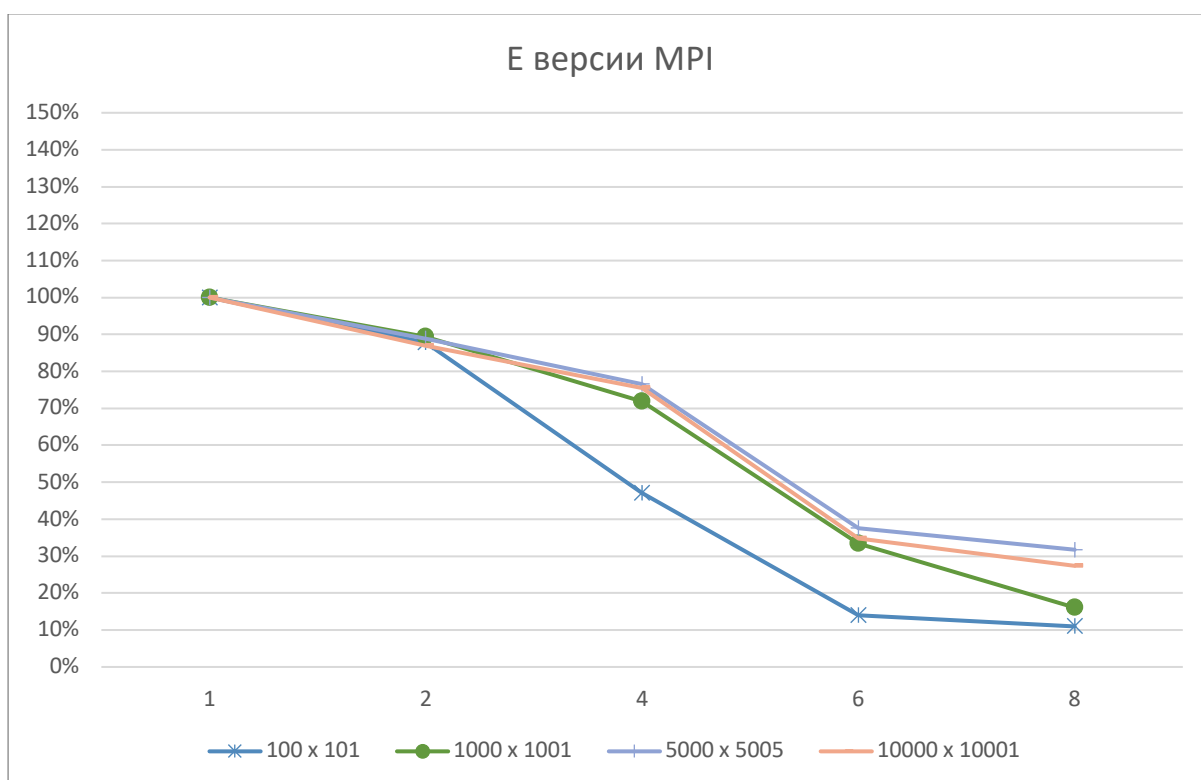


Рис. 3 Е версии MPI

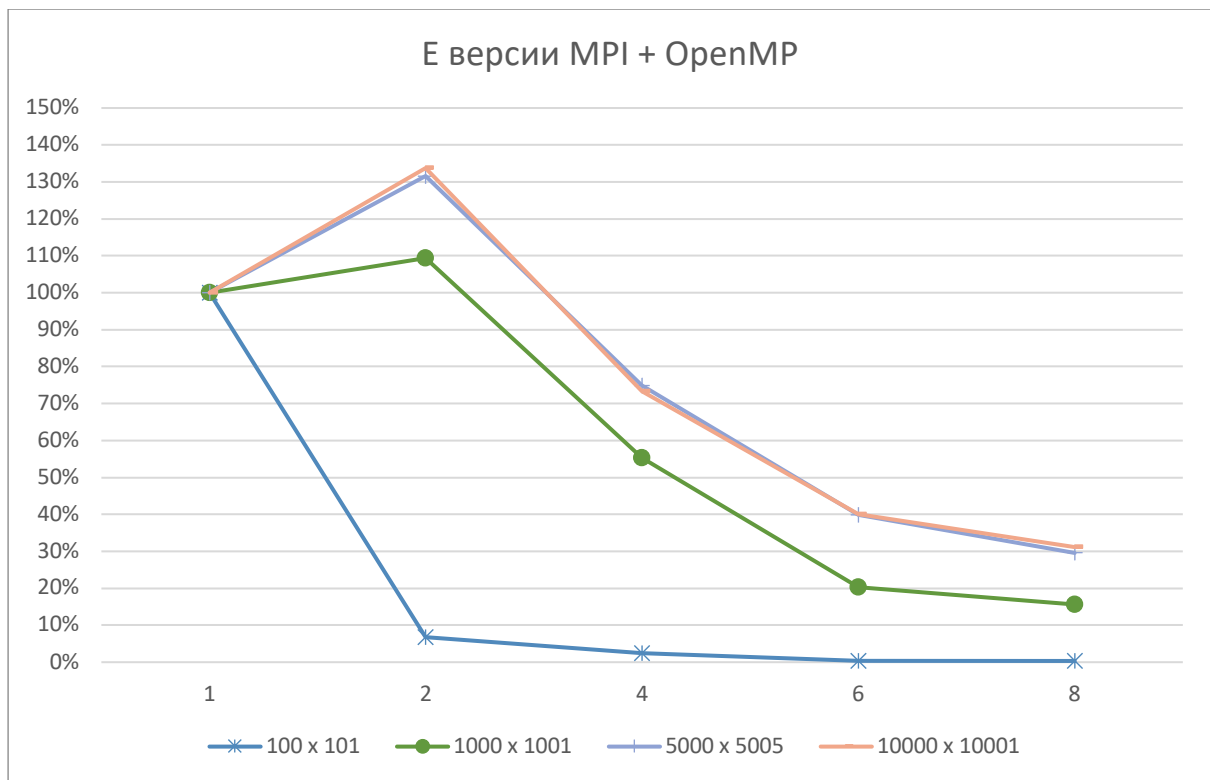


Рис. 4 Е версии MPI + OpenMP

Источники

1. Lin S. S., Wei C. L. Optimal algorithms for constructing knight's tours on arbitrary $n \times m$ chessboards // Discrete applied mathematics. – 2005. – Т. 146. – №. 3. – С. 219-232.