

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

**Практикум по курсу
"Суперкомпьютеры и параллельная обработка данных"**

Разработка параллельной версии метода релаксации

ОТЧЕТ

323 группы

ВМК МГУ

Кислов Евгений Витальевич

Москва, 2020

Содержание

Постановка задачи	3
Описание алгоритма	3
<i>Последовательный алгоритм.....</i>	<i>3</i>
<i>Параллельный алгоритм с использованием MPI</i>	<i>3</i>
Результаты	3
OMP.....	4
Выводы	5

Постановка задачи

Дана последовательная версия алгоритма релаксации.

Необходимо:

- 1) Реализовать ленточный алгоритм перемножения матриц с использованием технологий OpenMP и MPI.
- 2) Сравнить их эффективность.
- 3) Исследовать масштабируемость полученной параллельной программы: построить графики зависимости времени исполнения от числа ядер/процессоров для различного объема входных данных.

Описание алгоритма

Последовательный алгоритм

Основной цикл данного алгоритма включает в себя цикл, в котором вычисляются новые значения элементов $a(i,j,k) = (a(i-1,j,k) + a(i+1,j,k) + a(i,j-1,k) + a(i,j+1,k) + a(i,j,k-1) + a(i,j,k+1)) / 6$ и выполняется редукция по операции максимума.

Параллельный алгоритм с использованием MPI

Основной цикл(в функции `relax()`) был изменен таким образом, чтоб текущая итерация не зависела от соседних 6-ти итераций. Это достигается тем, что обход трехмерного массива идет “волной”. Кроме того, так же, как и в методе верификации, при изменении переменной `ers` была добавлена критическая секция, для корректного результата.

Данные изменения позволили достичь того же результата, который был на выходе у последовательного алгоритма.

Память между процессами пересылалась с помощью функций `MPI_Send`, `MPI_Recv`

Результаты

Ниже приведен 3D график зависимости времени работы от размера матриц и количества ядер/процессов. Тестирование проводилось на Polus.

OMP

Ниже приведена таблица, показывающая зависимость времени в секундах от размера N стороны трехмерного массива и количества процессов. Исследование проводилось на супер-компьютере Polus.

Количество процессов/N	32	64	96	128
Оригинальная программа	0.145876	1.153717	9.226888	73.749571
2	0.112577	0.870627	6.938802	55.536046
4	0.068572	0.516434	4.067354	33.075641
8	0.040347	0.285317	2.217781	17.542633
16	0.031333	0.206625	1.387350	10.252184

Выводы

1. Реализована параллельная версия алгоритма релаксации Sor3D.
2. Версия, использующая MPI дает ускорение в ≈ 7 раз на на размере массива 256 по сравнению последовательной версией.
3. Версия, разработанная на MPI, как и ожидалось, работает медленнее, чем версия, разработанная на OMP из-за того, что время на создание процесса в системе и на прочие накладные расходы значительно превышает время накладных расходов на треды.

Код и графики можно найти в репозитории <https://github.com/ew-kislov/skipod>