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

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 объёма 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 OMP

Циклы инициализации и верификации данного алгоритмы были распараллелены с использованием следующей директивы:

```
#pragma omp parallel for num_threads(threads) shared(A) private(i,j,k)
```

Кроме того, в цикле верификации, происходит гонка тредов по отношению к переменной s, поэтому для корректного изменения переменной, была поставлена критическая секция:

```
#pragma omp critical
{
    s = s + A[i][j][k] * (i + 1) * (j + 1) * (k + 1) / (N * N * N);
}
```

Также было выяснено, что добавление директивы ordered в цикл верификации дает значительное ускорение.

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 eps была добавлена критическая секция, для корректного результата.

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

Результаты

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

OMP

Ниже приведена таблица, показывающая зависимость времени в секундах от размера N стороны трехмерного массива и количества тредов. Исследование проводилось на суперкомпьютере Polus.

Количество потоков / N	32	64	128	256
Оригинальная программа	0.056326	0.483458	14.302748	124.925766
2	0.100606	0.699319	5.260183	44.074349
4	0.115520	0.459126	3.446582	26.694542
8	0.133492	0.483331	2.437593	18.440645
16	0.192084	0.619872	2.451851	18.295865
32	0.536929	1.345616	3.727959	22.578850
64	0.992632	2.586013	5.591096	28.110741
128	2.551450	4.474325	10.212539	33.453963
160	2.831764	5.839857	12.993675	38.845848

Выводы

Реализована параллельной версия алгоритма релаксации Sor3D.
Версия, использующая OMP дает ускорение в ≈7 раз на на размере массива 256 по сравнению последовательной версией.

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