

ОТЧЕТ ПО ЗАДАНИЮ

«Реализация алгоритма 3D ADI с использованием графических процессоров»

Выполнил:
студент 201 группы
Лыфенко А. И.

Москва
2024

1 Постановка задачи

1. Реализовать параллельный алгоритм 3-х мерного ADI по данному последовательному алгоритму.
2. Оценить ускорение программы по отношению к последовательной версии.

2 Описание программы

При распараллеливании программы было создано 6 ядер:

```
__global__ void init_parallel(double *a)
```

Параллельно инициализирует массив.

```
__global__ void f1(double *a, int ii)
```

Используется для счета алгоритма.

```
__global__ void f2(double *a, int jj)
```

Используется для счета алгоритма.

```
__global__ void f3(double *a, int kk)
```

Используется для счета алгоритма.

```
__global__ void f_cp(double *a, double *tmp1)
```

Копирование массива.

```
__global__ void f4(double *a, double *tmp1, double *tmp2)
```

Для нахождения eps.

```
__global__ void f_cp_k_i_j(double *a, double *tmp3)
```

Переупорядочивает данные в массиве на k, i, j.

```
__global__ void f_cp_j_k_i(double *a, double *tmp3)
```

Переупорядочивает данные в массиве на j, k, i.

Список функций:

```
double adi_parallel(double* a)
```

Запускает параллельный счет алгоритма. Возвращает полученное eps.

```
void init_seq(double *a)
```

Последовательно инициализирует массив.

```
double adi_seq(double* a)
```

Запускает последовательный счет алгоритма. Возвращает полученное eps.

```
void print_benchmark(struct timeval startt, struct timeval endt)
```

Выводит результат теста.

3 Результаты работы программы на различных входных данных

Последовательное выполнение:

ADI Benchmark Completed.
Size = 100 x 100 x 100
Iterations = 100
Time in seconds = 0.38
Operation type = double precision
Verification = SUCCESSFUL
END OF ADI Benchmark

ADI Benchmark Completed.
Size = 384 x 384 x 384
Iterations = 100
Time in seconds = 29.74
Operation type = double precision
Verification = SUCCESSFUL
END OF ADI Benchmark

ADI Benchmark Completed.
Size = 500 x 500 x 500
Iterations = 100
Time in seconds = 66.37
Operation type = double precision
Verification = SUCCESSFUL
END OF ADI Benchmark

ADI Benchmark Completed.
Size = 500 x 500 x 500
Iterations = 200
Time in seconds = 132.80
Operation type = double precision
Verification = SUCCESSFUL
END OF ADI Benchmark

ADI Benchmark Completed.
Size = 100 x 200 x 300
Iterations = 100
Time in seconds = 3.04
Operation type = double precision
Verification = SUCCESSFUL
END OF ADI Benchmark

Параллельное выполнение:

ADI Benchmark Completed.
Size = 100 x 100 x 100
Iterations = 100
Time in seconds = 0.32
Operation type = double precision
Verification = SUCCESSFUL
END OF ADI Benchmark

ADI Benchmark Completed.
Size = 384 x 384 x 384
Iterations = 100
Time in seconds = 3.73
Operation type = double precision
Verification = SUCCESSFUL
END OF ADI Benchmark

ADI Benchmark Completed.
Size = 500 x 500 x 500
Iterations = 100
Time in seconds = 8.40
Operation type = double precision
Verification = SUCCESSFUL
END OF ADI Benchmark

ADI Benchmark Completed.
Size = 500 x 500 x 500
Iterations = 200
Time in seconds = 16.98
Operation type = double precision
Verification = SUCCESSFUL
END OF ADI Benchmark

ADI Benchmark Completed.
Size = 100 x 200 x 300
Iterations = 100
Time in seconds = 0.83
Operation type = double precision
Verification = SUCCESSFUL
END OF ADI Benchmark

4 Сравнение скорости работы алгоритмов

Последовательный алгоритм:

Теоретическая оценка: $O(I * nx * ny * nz)$

Параллельный алгоритм:

Теоретическая оценка: $O(I * \max(nx, ny, nz) * ???)$