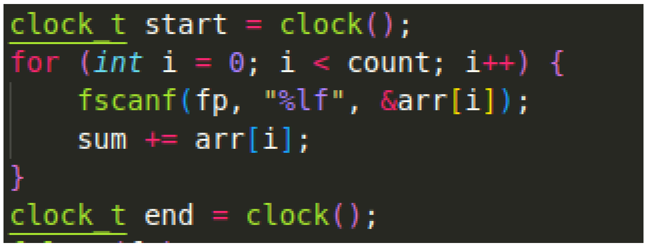
HPC CUDA C TUTORIAL 8 REPORT

CS22B2015 – HARSHITH B

1. Introduction

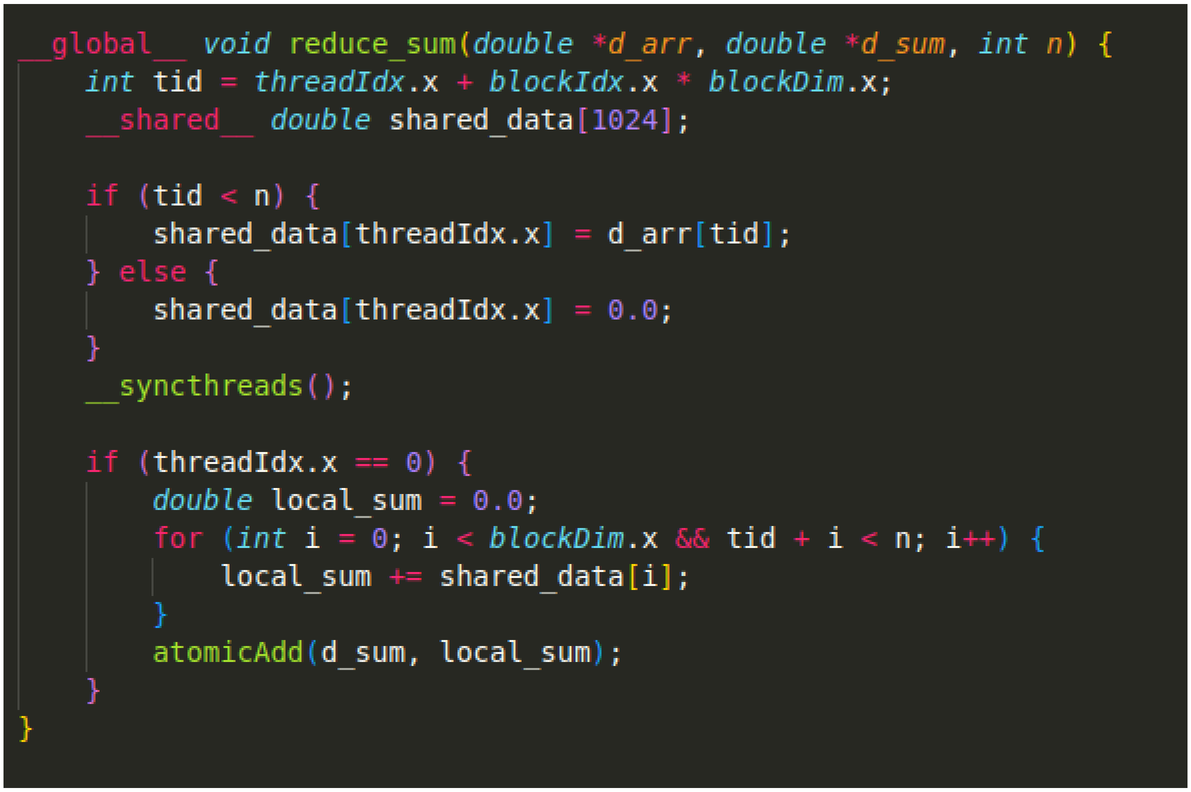
This report analyzes the performance of serial and parallel implementations for calculating the sum of 1.5 million floating-point double numbers. A serial code written in C and a parallel code written in CUDA (.cu) were developed to perform this computation, and their execution times are compared to evaluate efficiency and potential speedup.

2. Serial Code Snippet

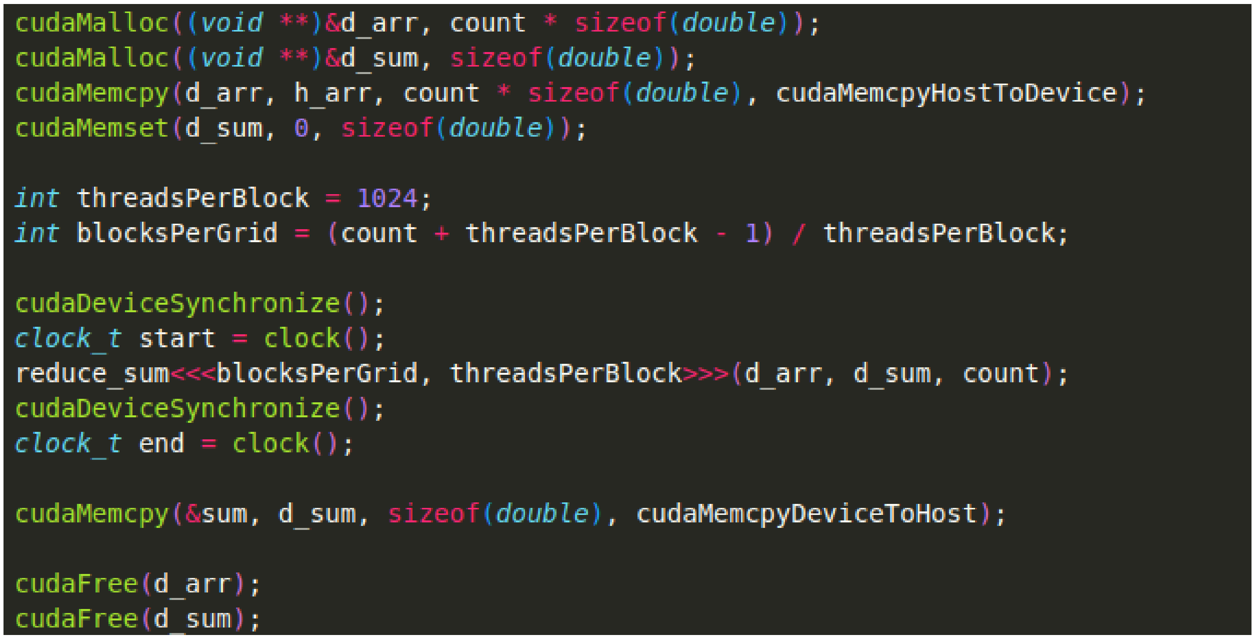


```
clock_t start = clock();
for (int i = 0; i < count; i++) {
    fscanf(fp, "%lf", &arr[i]);
    sum += arr[i];
}
clock_t end = clock();
```

3. Parallel Code Snippet

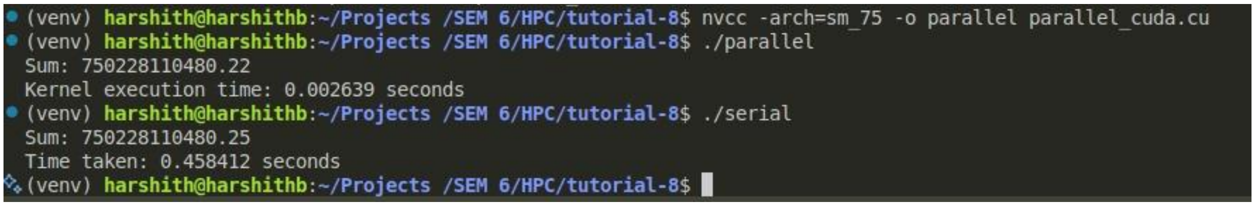


```
__global__ void reduce_sum(double *d_arr, double *d_sum, int n) {
    int tid = threadIdx.x + blockIdx.x * blockDim.x;
    __shared__ double shared_data[1024];

    if (tid < n) {
        shared_data[threadIdx.x] = d_arr[tid];
    } else {
        shared_data[threadIdx.x] = 0.0;
    }
    __syncthreads();

    if (threadIdx.x == 0) {
        double local_sum = 0.0;
        for (int i = 0; i < blockDim.x && tid + i < n; i++) {
            local_sum += shared_data[i];
        }
        atomicAdd(d_sum, local_sum);
    }
}
```



```
cudaMalloc((void **)&d_arr, count * sizeof(double));
cudaMalloc((void **)&d_sum, sizeof(double));
cudaMemcpy(d_arr, h_arr, count * sizeof(double), cudaMemcpyHostToDevice);
cudaMemset(d_sum, 0, sizeof(double));

int threadsPerBlock = 1024;
int blocksPerGrid = (count + threadsPerBlock - 1) / threadsPerBlock;

cudaDeviceSynchronize();
clock_t start = clock();
reduce_sum<<<blocksPerGrid, threadsPerBlock>>>(d_arr, d_sum, count);
cudaDeviceSynchronize();
clock_t end = clock();

cudaMemcpy(&sum, d_sum, sizeof(double), cudaMemcpyDeviceToHost);

cudaFree(d_arr);
cudaFree(d_sum);
```

4. Terminal Output Screenshot



```
(venv) harshith@harshithb:~/Projects /SEM 6/HPC/tutorial-8$ nvcc -arch=sm_75 -o parallel parallel_cuda.cu
(venv) harshith@harshithb:~/Projects /SEM 6/HPC/tutorial-8$ ./parallel
Sum: 750228110480.22
Kernel execution time: 0.002639 seconds
(venv) harshith@harshithb:~/Projects /SEM 6/HPC/tutorial-8$ ./serial
Sum: 750228110480.25
Time taken: 0.458412 seconds
(venv) harshith@harshithb:~/Projects /SEM 6/HPC/tutorial-8$ 
```

5. Serial and Parallel Code Execution Time and Speedup Calculation

- **Serial Code Execution Time:** 0.458412 seconds
- **Parallel Code Execution Time:** 0.002639 seconds

Speedup Calculation:

Speedup = Serial Execution Time / Parallel Execution Time

Speedup = 173.706707086