Code with GPU and GPU offloading

PA5.cu

```
#include <iostream>
#include <vector>
#include <chrono>
#include <cuda_runtime.h>
using namespace std;
#define BLOCK_SIZE 1024


__global__ void getSumGPU(double* inputArray,double* outputArray, int len){
  __shared__ double partialSum[2*BLOCK_SIZE];
  unsigned int t = threadIdx.x;
  size_t  start = 2 * (size_t)blockIdx.x * BLOCK_SIZE;
  if (start + t < len)
    partialSum[t] = inputArray[start + t]; //first index
  else
    partialSum[t] = 0;
  if (start + BLOCK_SIZE + t < len)
    partialSum[BLOCK_SIZE + t] = inputArray[start + BLOCK_SIZE + t];
  else
    partialSum[BLOCK_SIZE + t] = 0;  //last index
  for (unsigned int stride = BLOCK_SIZE; stride >= 1; stride >>= 1) {
    __syncthreads();
    if (t < stride)
      partialSum[t] += partialSum[t + stride];
  }
  if (t == 0)
    outputArray[blockIdx.x] = partialSum[0];
}
```

```
double getSumCPU(vector<double>& array){
  int rows = array.size();
  double sum = 0.0;
  for(int i = 0; i<rows; i++){
      sum += array[i];
    }
  return sum;
}

int main(int argc, char* argv[]){
  int N = stoi(argv[1]);
  vector<double> array(N);
  for(int i = 0; i<N; i++){
    array[i] = i+1;
  }
  auto start = chrono::high_resolution_clock::now();
  double result = getSumCPU(array);
  auto end = chrono::high_resolution_clock::now();
  chrono::duration<double> elapsed = end - start;
  cout << "N: " << N << ",CPU Time: " << elapsed.count() <<
   "s, Result = " << result << endl;
  int numInputElements = N;
   int numOutputElements = (numInputElements + (2 * BLOCK_SIZE) - 1) / (2 * BLOCK_SIZE);
  double *d_input, *d_output;
  cudaMalloc((void**)&d_input, numInputElements * sizeof(double));
  cudaMalloc((void**)&d_output, numOutputElements * sizeof(double));
  cudaMemcpy(d_input, &array[0],
numInputElements*sizeof(double),cudaMemcpyHostToDevice);
```

```
 auto gpu_start = chrono::high_resolution_clock::now();
 double* GPUSum;
cudaMalloc((void**)&GPUSum, sizeof(double));
while (numInputElements > 1) {
      int gridSize = (numInputElements + 2 * BLOCK_SIZE - 1) / (2 * BLOCK_SIZE);
      dim3 dimGrid(gridSize, 1, 1);
      dim3 dimBlock(BLOCK_SIZE, 1, 1);

      getSumGPU<<<dimGrid, dimBlock>>>(d_input, d_output, numInputElements);
      cudaDeviceSynchronize();
      cudaMemcpy(d_input, d_output, gridSize * sizeof(double), cudaMemcpyDeviceToDevice);
      numInputElements = gridSize;
  }
double final_GPU_Sum;
cudaMemcpy(&final_GPU_Sum, d_output, sizeof(double), cudaMemcpyDeviceToHost);
auto gpu_end = chrono::high_resolution_clock::now();
chrono::duration<double> gpu_elapsed = gpu_end - gpu_start;
cout<< "GPU Result = " << final_GPU_Sum << endl;
cout << "GPU Time = " << gpu_elapsed.count() << "s" << endl;
cout << "Speedup = " << elapsed.count()/gpu_elapsed.count() << endl;
 return 0;
}
```

Console Log

Makefile Output -

Running PA5 CPU vs GPU...

Running N=2^10 array size

Running N=2^11 array size

Running N=2^12 array size

Running N=2^13 array size

Running N=2^14 array size

Running N=2^15 array size

Running N=2^16 array size

Running N=2^17 array size

Running N=2^18 array size

Running N=2^19 array size

Running N=2^20 array size

Running N=2^21 array size

Running N=2^22 array size

Running N=2^23 array size

Running N=2^24 array size

Running N=2^25 array size

Running N=2^26 array size

Running N=2^27 array size

Running N=2^28 array size

Running N=2^29 array size

Running N=2^30 array size

Code Output -

N: 1024,CPU Time: 2.174e-06s, Result = 524800

GPU Result = 524800

GPU Time = 0.00854512s

Speedup = 0.000254414

N: 2048,CPU Time: 4.278e-06s, Result = 2.09818e+06

GPU Result = 2.09818e+06

GPU Time = 0.0074499s

Speedup = 0.000574236

N: 4096,CPU Time: 8.507e-06s, Result = 8.39066e+06

GPU Result = 8.39066e+06

GPU Time = 0.0086051s

Speedup = 0.000988599

N: 8192,CPU Time: 1.6962e-05s, Result = 3.35585e+07

GPU Result = 3.35585e+07

GPU Time = 0.00645866s

Speedup = 0.00262624

N: 16384,CPU Time: 3.3854e-05s, Result = 1.34226e+08

GPU Result = 1.34226e+08

GPU Time = 0.00735273s

Speedup = 0.00460428

N: 32768,CPU Time: 6.7698e-05s, Result = 5.36887e+08

GPU Result = 5.36887e+08

GPU Time = 0.00861531s

Speedup = 0.00785787

N: 65536,CPU Time: 0.000135247s, Result = 2.14752e+09

GPU Result = 2.14752e+09

GPU Time = 0.00698456s

Speedup = 0.0193637

N: 131072,CPU Time: 0.000270474s, Result = 8.59e+09

GPU Result = 8.59e+09

GPU Time = 0.00668376s

Speedup = 0.0404674

N: 262144,CPU Time: 0.000540907s, Result = 3.43599e+10

GPU Result = 3.43599e+10

GPU Time = 0.00958501s

Speedup = 0.0564326

N: 524288,CPU Time: 0.00108187s, Result = 1.37439e+11

GPU Result = 1.37439e+11

GPU Time = 0.00771357s

Speedup = 0.140256

N: 1048576,CPU Time: 0.00217384s, Result = 5.49756e+11

GPU Result = 5.49756e+11

GPU Time = 0.00955751s

Speedup = 0.227448

N: 2097152,CPU Time: 0.00433697s, Result = 2.19902e+12

GPU Result = 2.19902e+12

GPU Time = 0.00706873s

Speedup = 0.613544

N: 4194304,CPU Time: 0.00868877s, Result = 8.7961e+12

GPU Result = 8.7961e+12

GPU Time = 0.00606459s

Speedup = 1.43271

N: 8388608,CPU Time: 0.0175367s, Result = 3.51844e+13

GPU Result = 3.51844e+13

GPU Time = 0.00860528s

Speedup = 2.0379

N: 16777216,CPU Time: 0.0348497s, Result = 1.40737e+14

GPU Result = 1.40737e+14

GPU Time = 0.00857768s

Speedup = 4.06283

N: 33554432,CPU Time: 0.0695161s, Result = 5.6295e+14

GPU Result = 5.6295e+14

GPU Time = 0.00926558s

Speedup = 7.50262

N: 67108864,CPU Time: 0.139139s, Result = 2.2518e+15

GPU Result = 2.2518e+15

GPU Time = 0.00716672s

Speedup = 19.4146

N: 134217728,CPU Time: 0.277954s, Result = 9.0072e+15

GPU Result = 9.0072e+15

GPU Time = 0.00922489s

Speedup = 30.1309

N: 268435456,CPU Time: 0.556686s, Result = 3.60288e+16

GPU Result = 3.60288e+16

GPU Time = 0.01134s

Speedup = 49.0904

N: 536870912,CPU Time: 1.13623s, Result = 1.44115e+17

GPU Result = 1.44115e+17

GPU Time = 0.0113645s

Speedup = 99.9803

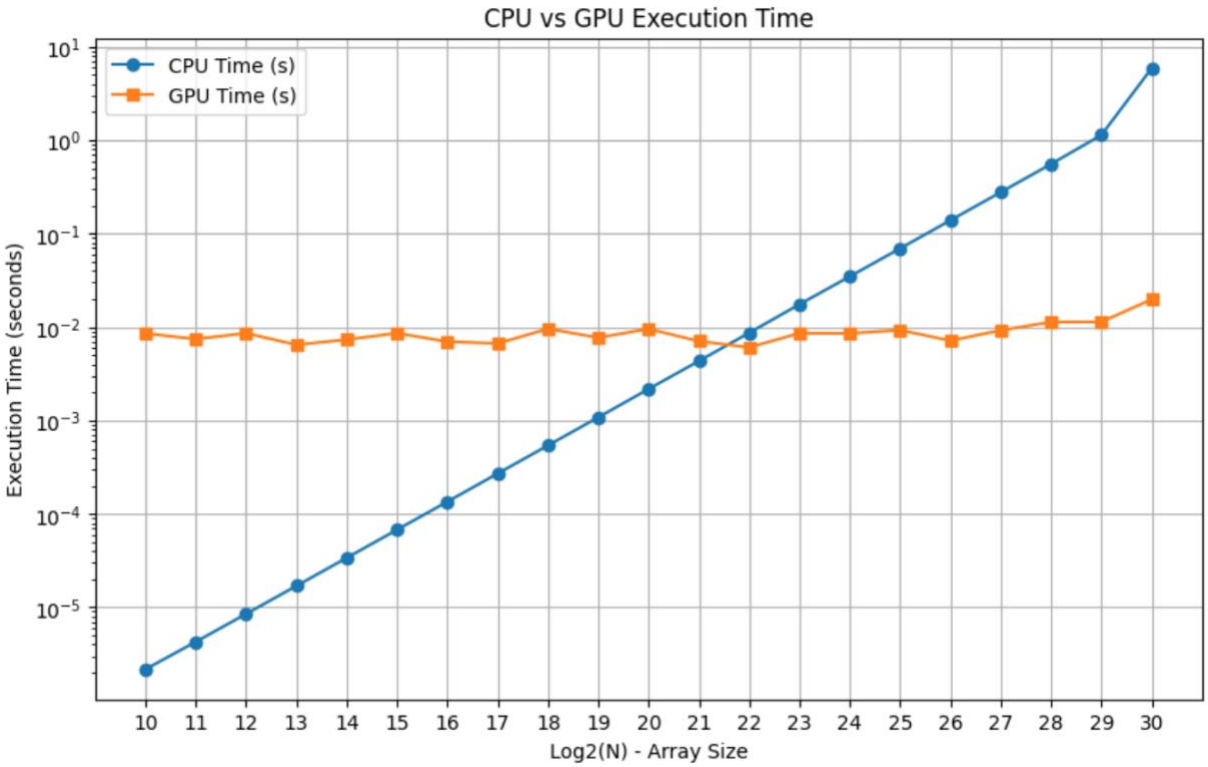
N: 1073741824,CPU Time: 5.90211s, Result = 5.76461e+17

GPU Result = 5.76461e+17

GPU Time = 0.019811s

Speedup = 297.921

Results



As seen from the CPU vs GPU Execution Time vs Execution Time plot, I am seeing breakeven around 2^22 N (4194304). The GPU graph stays constant for almost all values of N, while for CPU there is linear increase in time taken with increase in N.

Below attached is the time comparison and speed up table for all values of N.

N	CPU Time (s)	CPU Result	GPU Time (s)	GPU Result	Speed Up
1024	2.174E-06	524800	0.00854512	524800	0.000254414
2048	4.278E-06	2098180	0.0074499	2098180	0.000574236
4096	8.507E-06	8390660	0.0086051	8390660	0.000988599
8192	1.6962E-05	33558500	0.00645866	33558500	0.00262624
16384	3.3854E-05	134226000	0.00735273	134226000	0.00460428
32768	6.7698E-05	536887000	0.00861531	536887000	0.00785787
65536	0.000135247	2147520000	0.00698456	2147520000	0.0193637
131072	0.000270474	8590000000	0.00668376	8590000000	0.0404674
262144	0.000540907	34359900000	0.00958501	34359900000	0.0564326
524288	0.00108187	137439000000	0.00771357	137439000000	0.140256
1048576	0.00217384	549756000000	0.00955751	549756000000	0.227448
2097152	0.00433697	2199020000000	0.00706873	2199020000000	0.613544
4194304	0.00868877	8796100000000	0.00606459	8796100000000	1.43271

8388608	0.0175367	35184400000000	0.00860528	35184400000000	2.0379
16777216	0.0348497	140737000000000.0	0.00857768	140737000000000.0	4.06283
33554432	0.0695161	562950000000000.0	0.00926558	562950000000000.0	7.50262
67108864	0.139139	2251800000000000.0	0.00716672	2251800000000000.0	19.4146
134217728	0.277954	9007200000000000.0	0.00922489	9007200000000000.0	30.1309
268435456	0.556686	3.60288E+16	0.01134	3.60288E+16	49.0904
536870912	1.13623	1.44115E+17	0.0113645	1.44115E+17	99.9803
1073741824	5.90211	5.76461E+17	0.019811	5.76461E+17	297.921