PC Assignment 8

22610073 Vedant Amit Panari

Problem Statement:

LU Decomposition using OpenAcc

Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h> // Include for time measurements

#define N 4 // Matrix size

// Function for LUP Decomposition
void lup_decomposition(float A[N][N], int P[N]) {
int i, j, k, pivot;
float maxA, absA, temp;

// Initialize permutation matrix
for (i = 0; i < N; i++) P[i] = i;

// LUP Decomposition with OpenACC parallelization
for (k = 0; k < N; k++) {
pivot = k;
maxA = fabs(A[k][k]);

// Find pivot
#pragma acc parallel loop reduction(max:maxA)
for (i = k + 1; i < N; i++) {
absA = fabs(A[i][k]);
if (absA > maxA) {
maxA = absA;
pivot = i;
}
}

// Swap rows if needed
if (pivot != k) {
for (j = 0; j < N; j++) {
temp = A[k][j];
A[k][j] = A[pivot][j];
A[pivot][j] = temp;
}
int tmp = P[k];
P[k] = P[pivot];
```

```
P[pivot] = tmp;
}

// Compute multipliers
#pragma acc parallel loop
for (i = k + 1; i < N; i++) {
A[i][k] /= A[k][k];
for (j = k + 1; j < N; j++) {
A[i][j] -= A[i][k] * A[k][j];
}
}
}
}

// Function to print matrix
void print_matrix(float A[N][N]) {
for (int i = 0; i < N; i++) {
for (int j = 0; j < N; j++) {
printf("%8.4f ", A[i][j]);
}
printf("\n");
}
}

int main() {
float A[N][N] = {
{2, 3, 1, 5},
{6, 13, 5, 19},
{2, 19, 10, 23},
{4, 10, 11, 31}
};
int P[N];

// Start time for CPU execution (gcc)
clock_t start, end;
double cpu_time_used;

// Timing for CPU execution
start = clock();

printf("Original Matrix:\n");
print_matrix(A);

// Perform LUP decomposition on CPU
lup_decomposition(A, P);

// End time for CPU execution
end = clock();
cpu_time_used = ((double)(end - start)) / CLOCKS_PER_SEC;

printf("\nLUP Decomposed Matrix (LU):\n");
```

```
print_matrix(A);

printf("\nPermutation Vector:\n");
for (int i = 0; i < N; i++) {
printf("%d ", P[i]);
}
printf("\n");

printf("\nCPU Execution Time: %f seconds\n", cpu_time_used);

// Timing for GPU execution (pgcc)
// Use OpenACC data region to offload computation to GPU
#pragma acc data copy(A, P)
{
// Start time for GPU execution
start = clock();

// Perform LUP decomposition on GPU with OpenACC
lup_decomposition(A, P);

// End time for GPU execution
end = clock();
cpu_time_used = ((double)(end - start)) / CLOCKS_PER_SEC;

printf("\nGPU Execution Time: %f seconds (OpenACC with pgcc)\n", cpu_time_used);
}

return 0;
}
```

Command:

```
it@it-OptiPlex-3050:~$ pgcc -acc -Minfo=accel -o lup_acc lup_acc.c
```

Output:

```
it@it-OptiPlex-3050:~$ ./22610073-lup_acc
Original Matrix:
  2.0000   3.0000   1.0000   5.0000
  6.0000  13.0000   5.0000  19.0000
  2.0000  19.0000  10.0000  23.0000
  4.0000  10.0000  11.0000  31.0000

LUP Decomposed Matrix (LU):
  2.0000   3.0000   1.0000   5.0000
  3.0000   4.0000   2.0000   4.0000
  1.0000   4.0000   1.0000   2.0000
  2.0000   1.0000   7.0000   3.0000

Permutation Vector:
0 1 2 3

CPU Execution Time: 0.079969 seconds

GPU Execution Time: 0.000112 seconds (OpenACC with pgcc)
```