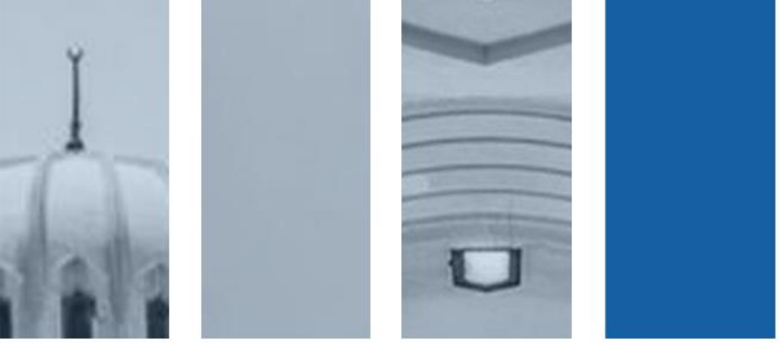


RESEARCH & PROJECT SUBMISSIONS

Program: CESS

Course Code: CSE 317

Course Name: Parallel and cluster computing

Examination Committee

Dr. Mohamed Taher

Ain Shams University
Faculty of Engineering
Spring Semester – 2020

AIN SHAMS UNIVERSITY
FACULTY OF ENGINEERING

CSE 317, Parallel and cluster computing
, SPRING 2020

Student Personal Information for Group Work

Student Names:	Student Codes:
Mahmoud Mohamed Abdelhamid	15P1053
Basil Osama Eloraby	16P6025

Plagiarism Statement

I certify that this assignment / report is my own work, based on my personal study and/or research and that I have acknowledged all material and sources used in its preparation, whether they are books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. I also certify that this assignment / report has not been previously been submitted for assessment for another course. I certify that I have not copied in part or whole or otherwise plagiarized the work of other students and / or persons.

Signature/Student Name: **Mahmoud Mohamed – Basil Osama** **Date:** **08/06/2020**

Submission Contents

1- Introduction

A thread is a single sequence stream within in a process. Because threads have some of the properties of processes, they are sometimes called *lightweight processes*. In this report we are discussing the implementation of 2-D heat transfer problem through both sequential and parallel (multi-threaded) solutions through different performance aspects.

2- POSIX Framework

POSIX threads or pthreads is a C programming library to provide multi-threading functionality to your C program, like the ability to create one or more thread, thread synchronization, the ability to control these threads and much more.

We can think of a thread that it's more like a process or a set of instructions with one specific difference that threads unlike processes can share the same memory space.

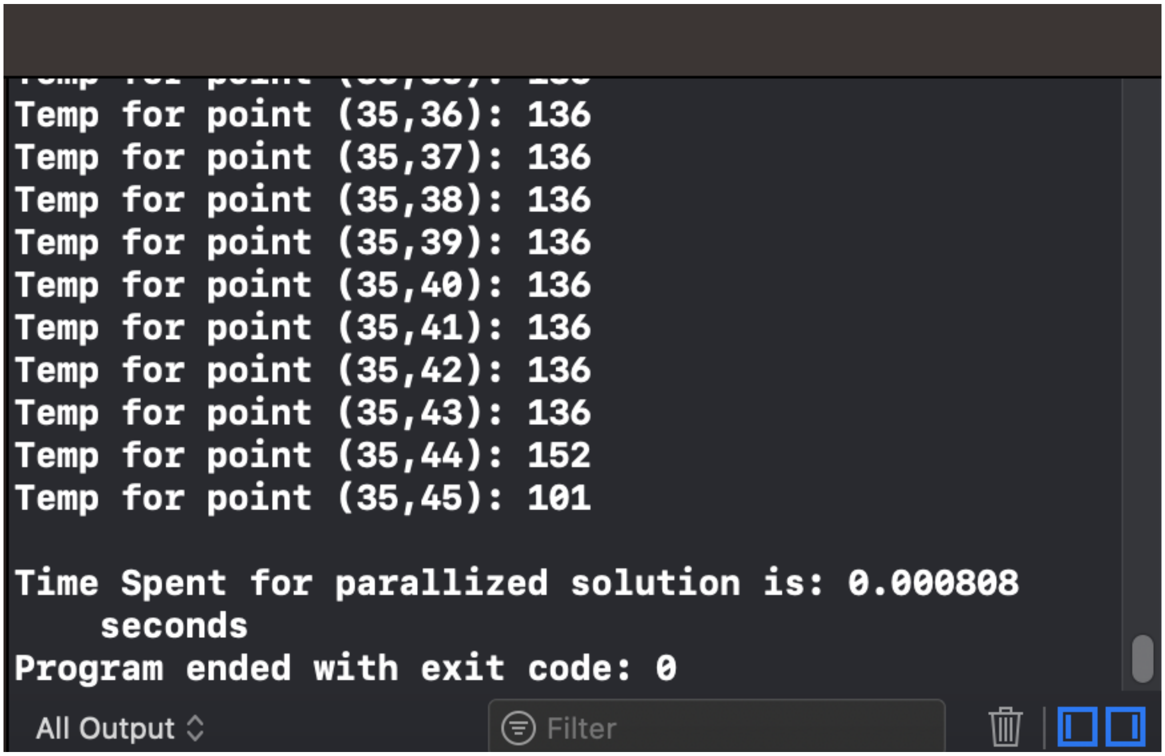
3- Code Design

a- The sequential solution:
Our sequential solution is a simple one threaded iterative solution which converts the initial temperatures into a 2D array with a provided size and apply the heat transfer equation until a temperature smaller than the provided threshold is reached.

b- The Multi-Threaded solution:
The multi-threaded solution is more or less like the sequential solution except for that it divides any temperature 2D grid into two halves each to be solved on a separate thread, while each thread locks the column it's working on using mutex to avoid any wrong results.

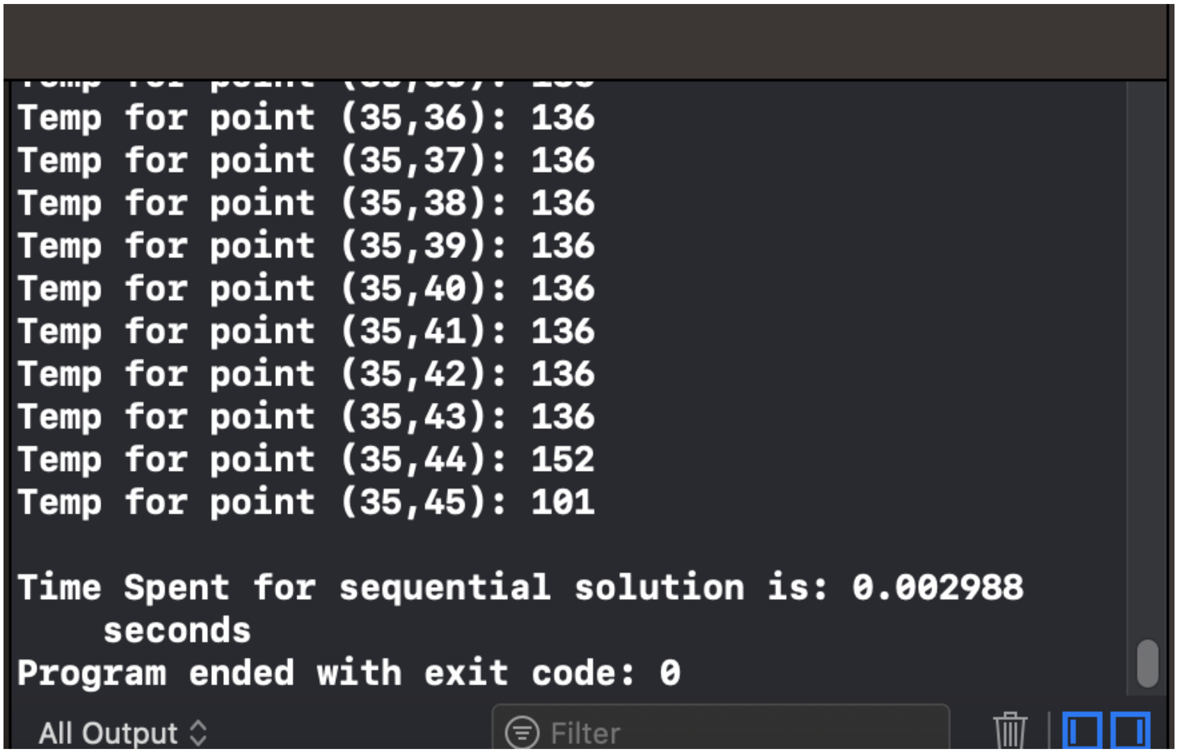
4- Performance analysis

The sequential solution showed a better performance for small sized grids, while the parallelized solution showed a good speed up for larger grids, attached below the results of both solutions on a 36*46.



```
Temp for point (35,36): 136
Temp for point (35,37): 136
Temp for point (35,38): 136
Temp for point (35,39): 136
Temp for point (35,40): 136
Temp for point (35,41): 136
Temp for point (35,42): 136
Temp for point (35,43): 136
Temp for point (35,44): 152
Temp for point (35,45): 101

Time Spent for parallized solution is: 0.000808
    seconds
Program ended with exit code: 0
```



```
Temp for point (35,36): 136
Temp for point (35,37): 136
Temp for point (35,38): 136
Temp for point (35,39): 136
Temp for point (35,40): 136
Temp for point (35,41): 136
Temp for point (35,42): 136
Temp for point (35,43): 136
Temp for point (35,44): 152
Temp for point (35,45): 101

Time Spent for sequential solution is: 0.002988
    seconds
Program ended with exit code: 0
```

a- Why multi-threading?

Threads are popular way to improve application through parallelism. For example, in a browser, multiple tabs can be different threads. MS word uses multiple threads, one thread to format the text, other thread to process inputs, etc.
Threads operate faster than processes due to following reasons:
1) Thread creation is much faster.
2) Context switching between threads is much faster.
3) Threads can be terminated easily
4) Communication between threads is faster.

5- Teamwork

We did a lot of pair programming sessions through TeamViewer due to the current situations and a lot of brainstorming to figure out the algorithm.

a- Bassil’s work:
- Worked on implementing the sequential algorithm.
- Worked on dividing the grid in order to be sent to each thread and covering the arrays corner cases.
- Supported debugging for the threaded solution.

b- Mahmoud’s work:
- Worked on implementing the threaded solution.
- Worked on temperature values threshold case.
- Worked on synchronizing the threads using mutex.

6- References

- http://www.csc.villanova.edu/~mdamian/threads/posixthreads.html - pthreads-definition
- https://www.geeksforgeeks.org/multithreading-c-2/
- Class lecture 5 videos.
- http://www.flipkart.com/computer-systems-programmer-s-perspective-2nd/p/itmdx5gnnz8ynpsh?pid=9780136108047&affid=sandeepgfg