Assignment 1:tetrahedral nummbers

Gaurav Choudekar - CS22BTECH11015

December 3, 2023

1 Introduction

This report is regarding assignment 1 conducted on finding tetrahedral numbers with different number of processes.

2 About Code

The code is written in C.all tetrahedral numbers are found till given input number and represented in OutMain file with their process numbers,also,for each process different log file is made to represent numbers it checks and if it is tetrahedral number or not.

3 Input format

A file is given as input,name of file is modified accordingly.there are only two numbers in file seperated by space bar,first represent upper limit to find tetrahedral numbers and second one to represent number of processes used.

4 Basic idea of resource usage

For each process,one node is made and in it with help of array,all numbers are stored.also tet attribute is used in it to store number of numbers in that array.For each process,their is different memory space buffer between it and main function.While computing in each process if it is tetrahedral number or not,if it is,then it is stored locally in array then communicate with main process after computing for all numbers for that process.

5 Code Flow

5.1 Main Function

In the main function take input from file and make our nodes ready according to that,they are first initialised,then numbers are evenly distributed in it.after each fork,we send that specific node to process function for checking tetrahedral numbers.for each process,when it is used by fork,seperate shared memory buffer is made with unique name.we make sure that only parent process calls fork,so,child process executes exit after it computes for all numbers in it.wait is used to make sure all child process gets executed before parent,after wait,we print one by one numbers stored in all memory buffers in output file.

5.2 Process Function

Aim of process function is to check each number in node if it is tetrahedral number or not.first,it will open a link with parent process with the same name parent opens for its child.we iterate numbers in node-array one by one and checks with the help of formula that if it is tetrahedral number or not.if it is tetrahedral number,then we store it in local array.process function creates a seperate file for each child process and stores in it result of every number check.at the end,after checking for all numbers,we communicate with main process our locally stored array for tetrahedral numbers.

5.3 Accuracy Check complications

5.4 Accuracy check

Before giving big inputs,i gave small inputs so that it became easy to verify from each output file if it is giving correct output or not.whenever i got error or wrong output,i gave printf statements in between to check if that function or part of function is reached or not,also if needed,i printed variable values to check if their correct value is reached or not to confirm that code till that part is correct.at the end,when i became confident about small values,and later big values i cross-checked my output values/result with peers to become double sure about it.

5.5 complications

First of all while coding for this problem,i was giving direct input in code through variables rather than file input,so after shaping it in required format,i got many segmentation faults and wrong values for given inputs.this was my first time working with shared memory,so it was little interesting and difficult for me to use it and give parameters to it.As i used node,i would get segmentation fault or no output for wrong result,i required to correct it and keep the current sizes of variables.it was little difficult for me to record execution time,so i used some online resources to see what can i use for it.

5.6 comments in my code

I am intentionally not deleting commented parts of my code so that it indicates my printf statements where i was needed to use them for accuracy check and dealing with complications.

6 Graphs

Include the graphs as specified in the assignment.

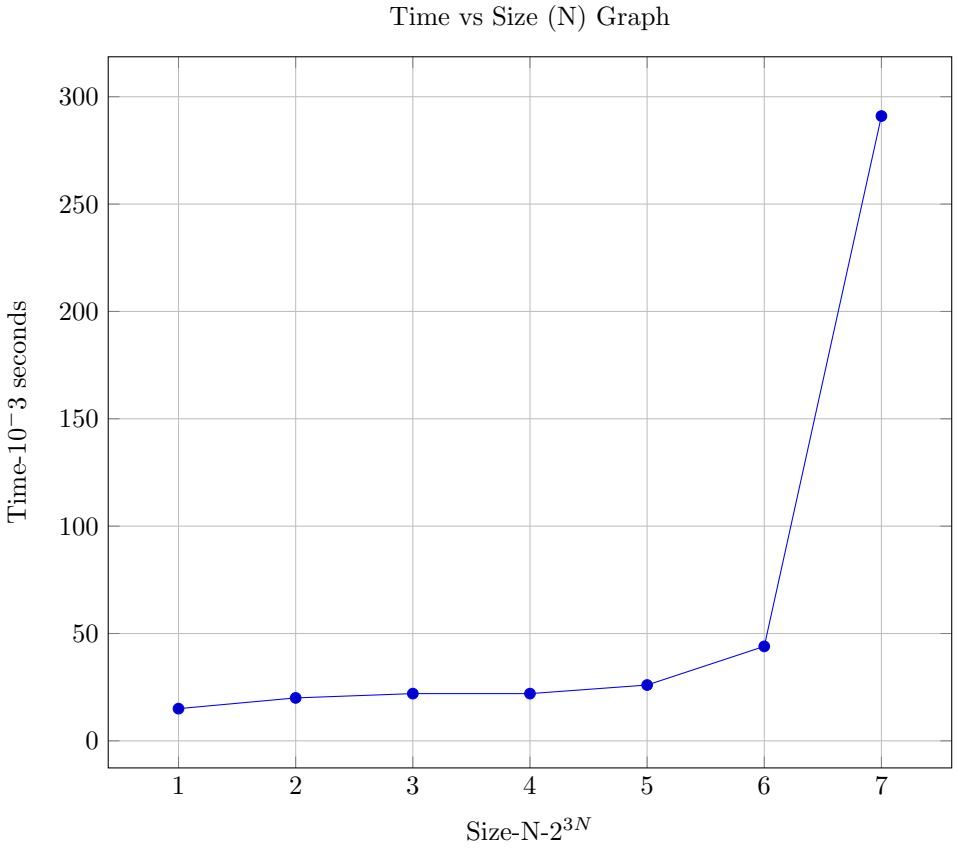

Figure 1: Time vs Size (N) Graph

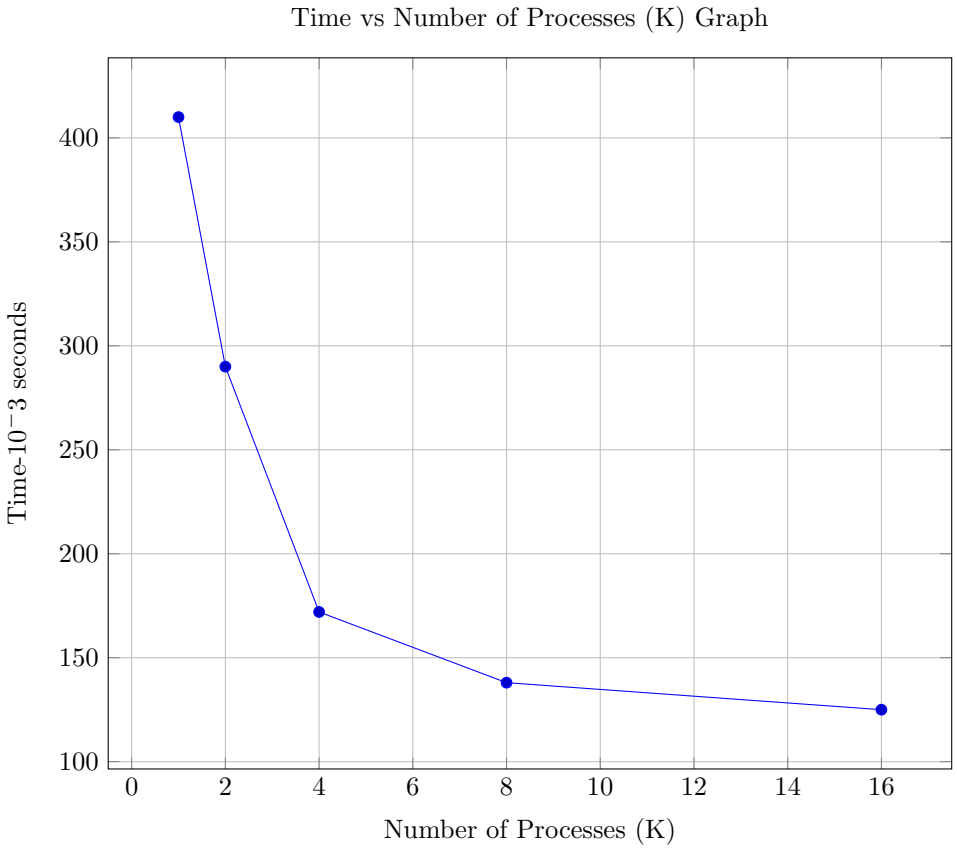

Figure 2: Time vs Number of Processes (K) Graph

7 Output analysis

for fixed number of processes,as we increase number of operations,that is,number of numbers to compute, time required for execution/their computation increases.For fixed number of numbers,as we increase number of processes,time required for their exection decrases.this is due to that fact that one process requires to compute less number of numbers.this facts can be seen easily with the help of graphs.as number of numbers to compute increases,time execution increases more and more rapidly.also,tetrahedral number distribution decreases in particular range for higher and higher values.

8 Improvement scope

Output obtained is not equal in different processes,which means it could be divided more properly between different processes.memory allocation can be more proper,that is,it can be allocated as per needed,not how much we can.due to these there is scope for better time complexity and space comp.memory cleanup and error handling/check could be more proper.