Student ID: 101062319
Name: 巫承威

Homework 2 - Report

• Design

(a) How do you parallelize each phase of Barnes-Hut algorithm?

I parallelized the segment of "computing net force & update each body's information" in Barnes-Hut algorithm. In my implementation of Barnes-Hut algorithm, it can be divided into several main segments as following descriptions:

1. Build "Quad Tree" structure of whole n-body system (implemented by sequential code)
2. Compute the net force of each body (implemented by parallel code)
3. Update each body's location and velocity (implemented by parallel code)

After building quad-tree structure of the whole n-body system, I let the #job to be the #body , which symbolizes the #left-jobs which would be done by the threads. Then, I wake up all threads by broadcasting. While the thread being waked up, it will first decrease the #left-job (since it will solve 1 job), then check whether the program is finished or not (it will exit if the program is finished). Then it will compute the net force of the specific body whose index is the #left-job(already minus 1). After getting the summation of the net force of this body, it will then update the information of it.

(b) How do you partition the task?

I partition the task into 2 parts. The first part is to build the quad-tree, which will construct the the whole system in the specific loop step. The second part is to wake up all other threads to do their own tasks including computing the net force done on body and updating the information of the body.

(c) How do you prevent from synchronization problem?

I malloc another body pointer(array) to store the updated information also called the new information. After every thread finishing their own task, I can get all the new locations and velocities of all bodies, then I only need to change the pointer of the new bodies and old bodies and just finish the updating process. Since I store the new information in another array, it can prevent the synchronization problem from wrongly using body's location or velocity.

(d) What technique do you use to reduce execution time and increase scalability?

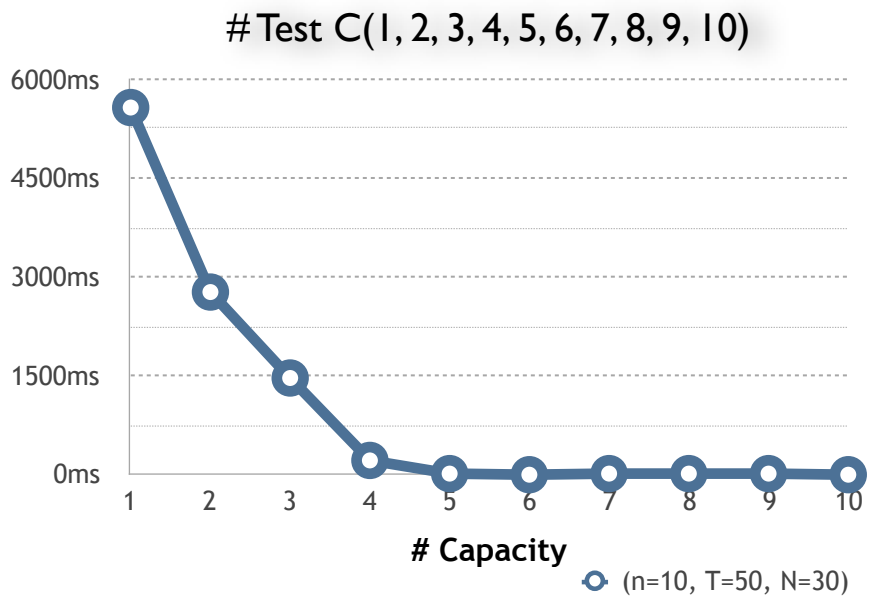
I store the n-body system information by pointer dynamically. And define the structs of "Quad-Tree" and "Body". They can increase the scalability at some degree. Also, I update the body information by only changing the pointer, not changing it body-by-body.

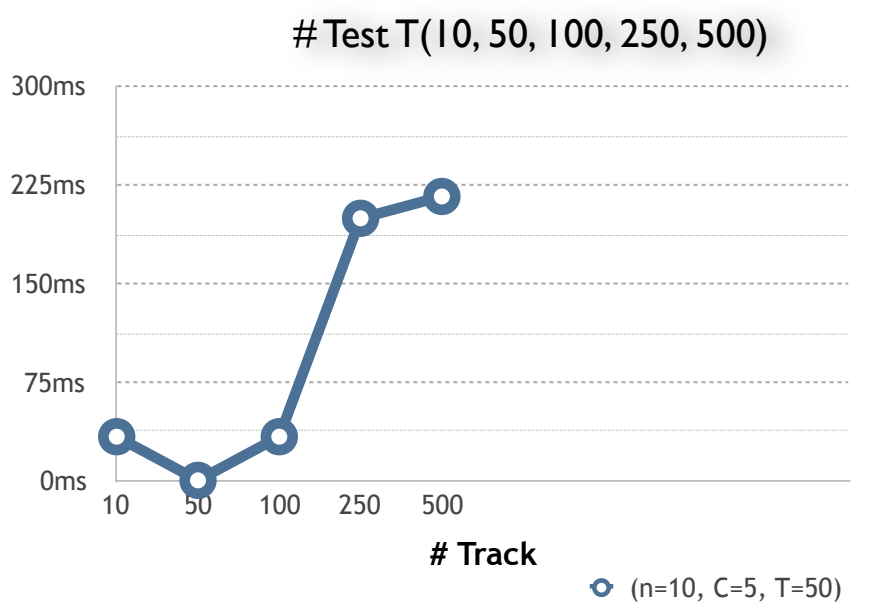
(e) Other efforts you've made in your program?

I define some mathematical computation equations to be the type of macro such as computing the cube or square value of specific number. And also define some compare functions to be macro since it will be use in high frequency such like getting the maximum or minimum of the set of numbers.

• Performance analysis of Single Roller Coaster Car Problem

(a) Plot the average waiting time vs the input parameter C or T

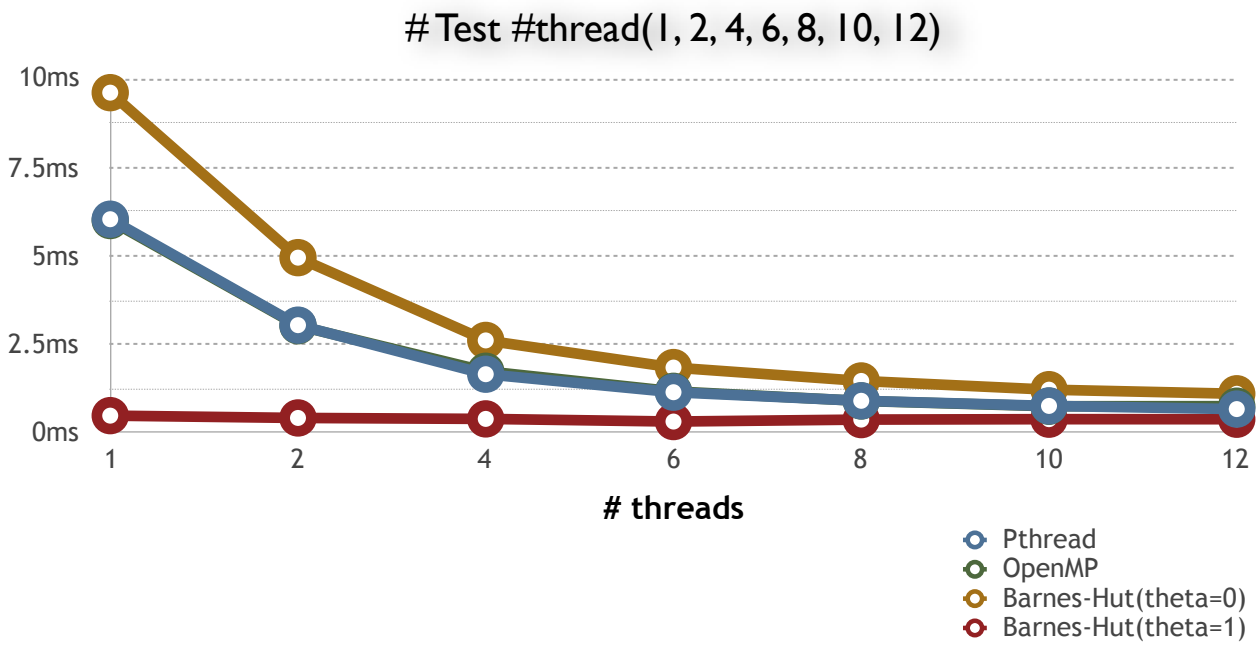




- Performance analysis of N-Body Problem

(a) Strong scalability (1 1 200 1 test2.txt 0 disable -0.3 -0.3 0.6 600)

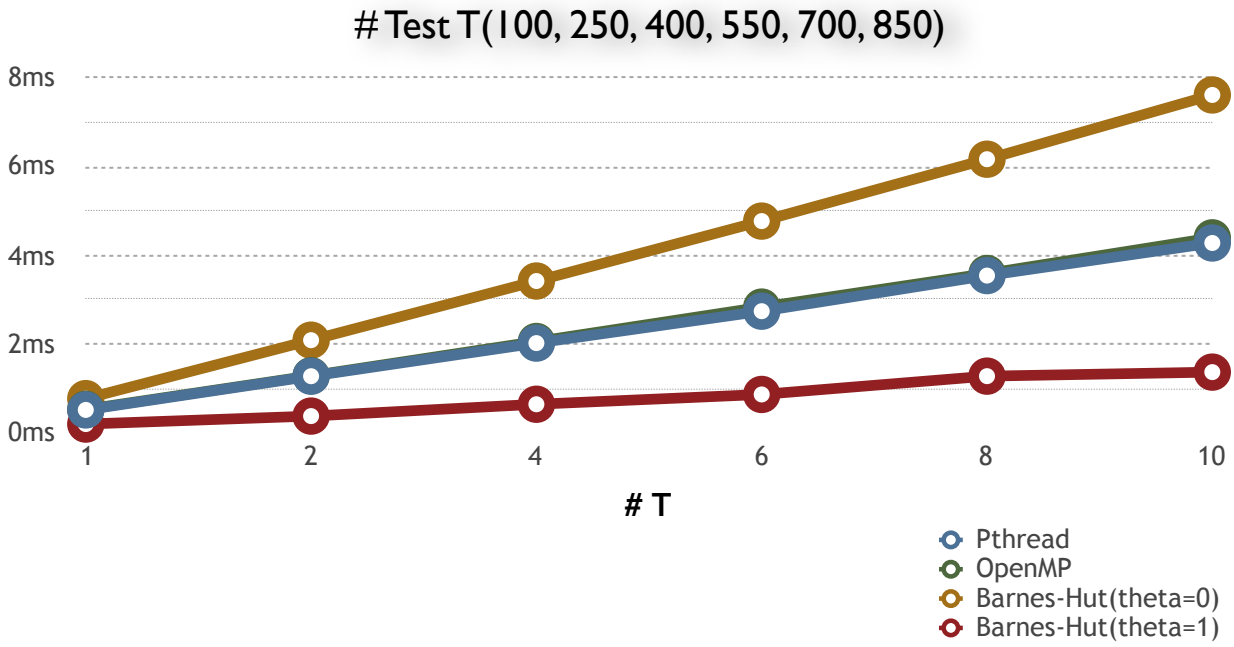
i. Fix *N*, *T*, and manipulates *#threads*



(b) Weak scalability

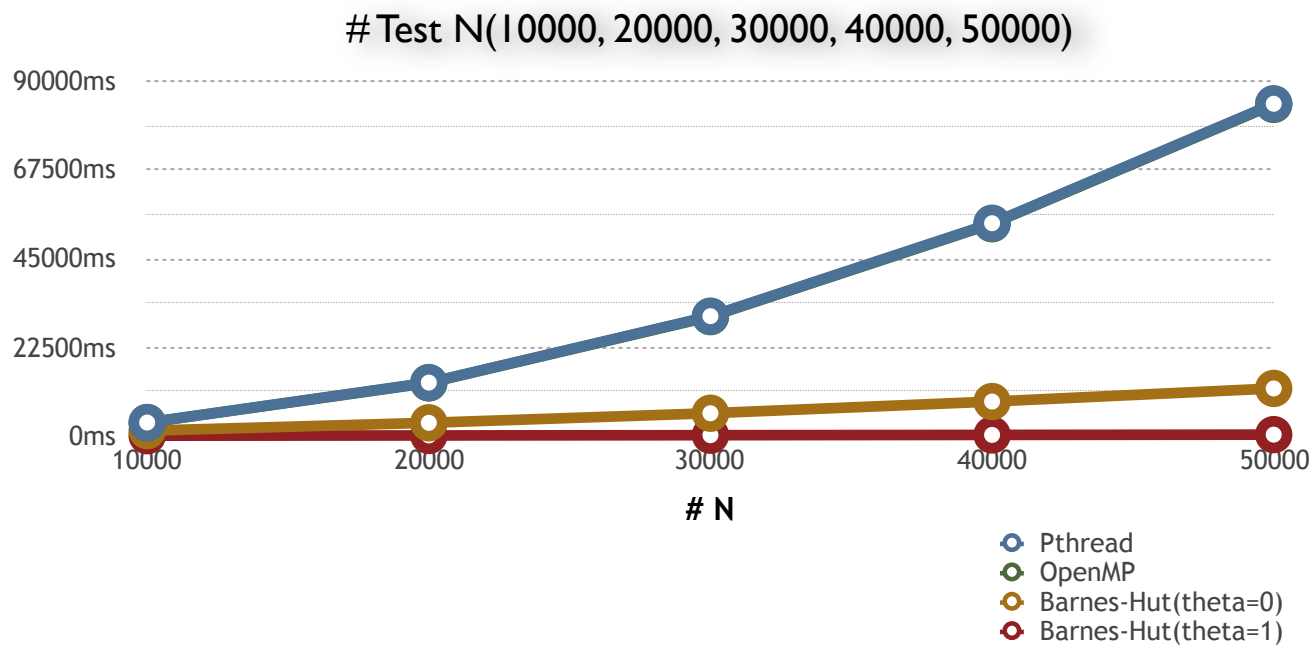
i. Fix *N*, *#threads*, and manipulates *T*

(7 1 100 1 test2.txt 0 disable -0.3 -0.3 0.6 600)



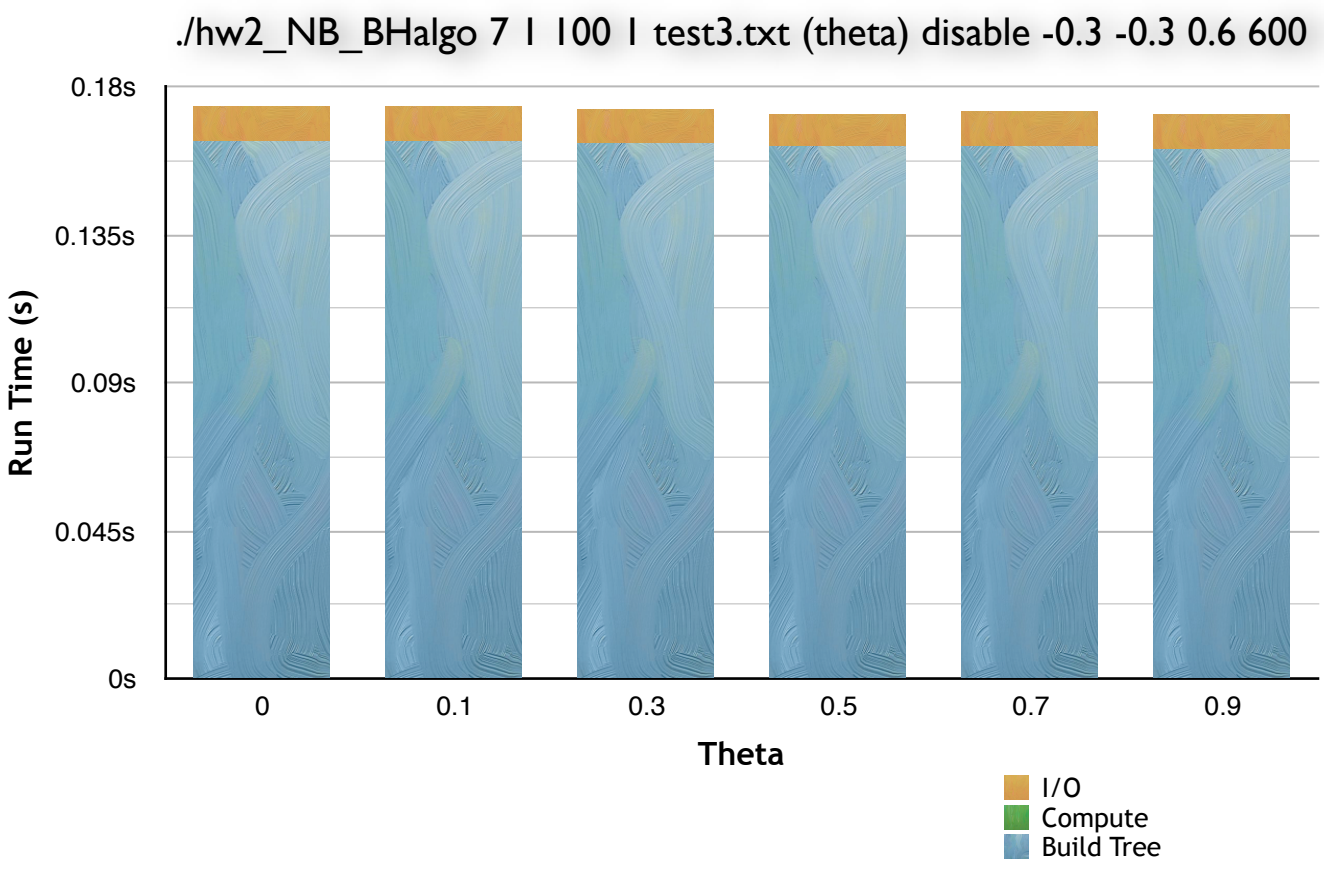
ii. Fix *T*, *#threads*, and manipulates *N*

(7 1 100 1 test2.txt 0 disable -0.3 -0.3 0.6 600)



(c) Barnes-Hut algorithm

i. Show the time of three phases (I/O, building tree, and computing) based on different θ



- Experience

  (a) What have you learned from this assignment?

  I learned several parallel languages and how to implement the Barnes-Hut algorithm.

  (b) What difficulty did you encounter when implementing this assignment?

I can't solve the bling-bling graph produced by Xwindow……
Since I use C to implement the Barnes-Hut algorithm, I must use so many pointers to implement my functions…… And this caused so many bugs while I was debugging.
So I will chose C++ first next time……
I think the most difficult part is to build the quad-tree and how to parallelize the algorithm.