Homework 4 – finding the max of a function
EECS 587 parallel computing
Ethan Zhang (shuruiz@umich.edu)

In this problem, we are supposed to finding the max of $\{f(x): a \leq x \leq b\}$. The parallel algorithm used in the code file is briefly explained here:
First of all, the target max value of $f(x)$ is initialized to an infinite small value(-DBL_MAX). The interval is divided following the idea of breadth-first. It appears like:
[1,100]
[1,50], [50,100]
[1,25], [25,50], [50,75], [75,100]
…
……

Initialize the stack with interval [a, b]. The stack is a shared by all threads. For each thread, determine if its stack is empty or not, if empty, unlock it and set it to work on the next job, which is to generate intervals and find the max of it. The empty thread gets the work from the top of the stack. If the max value of the interval is greater than $M + \varepsilon$, generate new intervals. When new intervals are generated (in breadth-first way), they are pushed to the stack. Otherwise the thread is unlocked and set it work. When an interval is processed, it is removed from the stack with command *stack.pop().* Loop the whole process till the max values of all intervals are not greater than $current\_M + \varepsilon$. If the stack of master thread is empty, the process is finished.

The homework is run on PSC bridges computer for threads = 4, 16 and 28 (as required, 56 threads are not needed to run for this homework.).

The code file is named as (shuruiz.cpp)
Two compilers are used for the homework, which are: Intel C++ compiler icpc and g++.
icpc gets the best performance on PSC bridges.
The performance of icpc is :

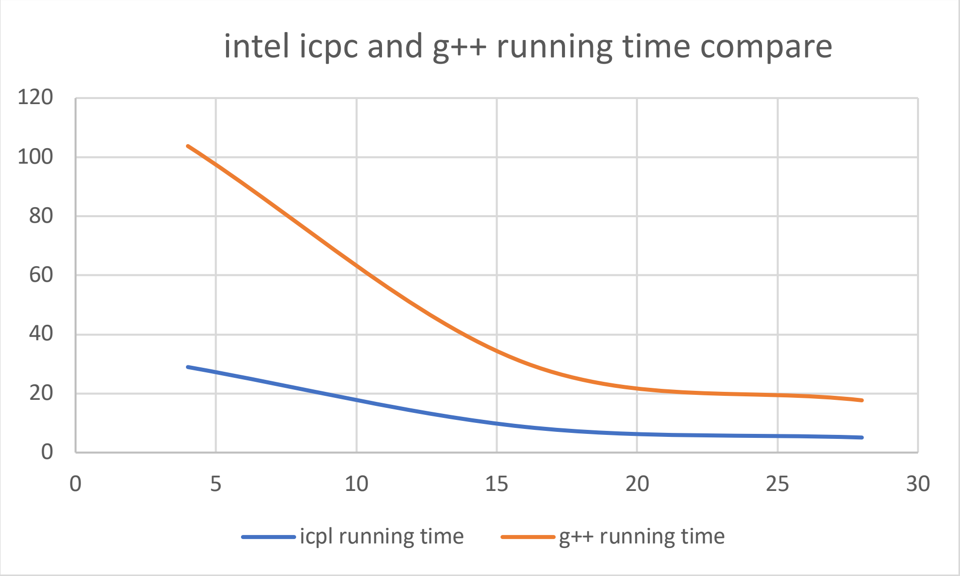
Intel icpc compiler results

p	1	4	16	28
final result	3.33333	3.33333	3.33333	3.33333
icpc running time	114.527	28.9485	8.69806	5.09614

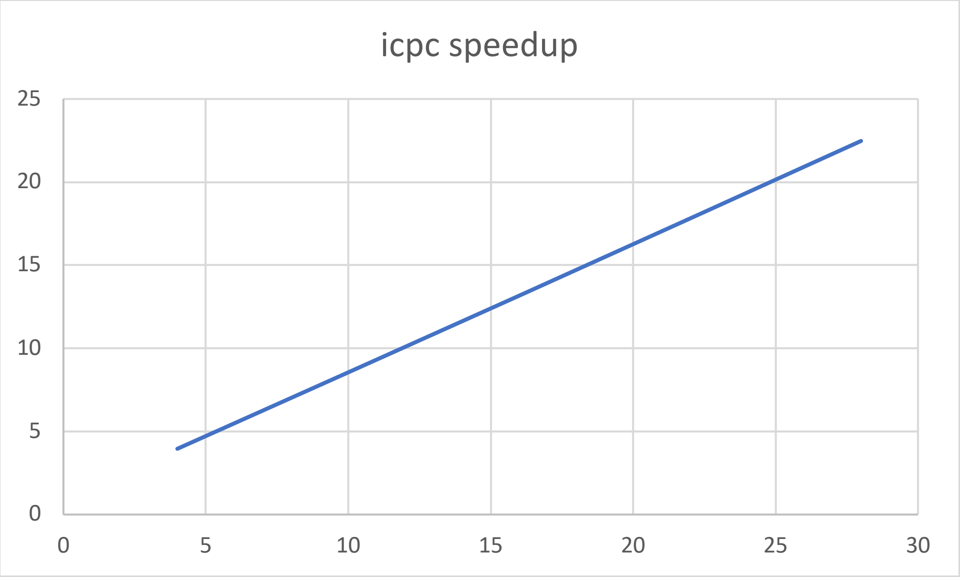
The performance for g++ is:

g++ compiler results

p	4	16	28
final result	3.33333	3.33333	3.33333
total running time	103.961	30.3953	17.6888



Speedup result of icpc, which is almost linear in (4, 16, 28) case.


icpc speedup