**Parallel and Distributed Systems**

159735

Assignment one

π calculation

Wei Han

20015402

# Amdahl’s Law

A program which can be parallelized can be split up into two parts:

* A part cannot be parallelized
* Another part can be parallelized

T is called total time to execute program in serial.

The time T includes the time of both the non-parallelizable and parallelizable parts.

The non-parallelizable part is called B (serial part).

The parallizable part is T - B

**T = B + (T-B)**

**(T-B)** which can be sped up by executing it in parallel

(T-B)/N

The number of threads or CPUs is called N, which can speed up the parallelizable part.

T(N) = B + (T - B) / N

If N =1 T=1 then:

T(1) = B + (T(1)-B) which seems like just one single CPU

T(1)=T(1)=1

If N =2 T=1 B=0.6 then:

T(2) = B + (T(2)-B)/2

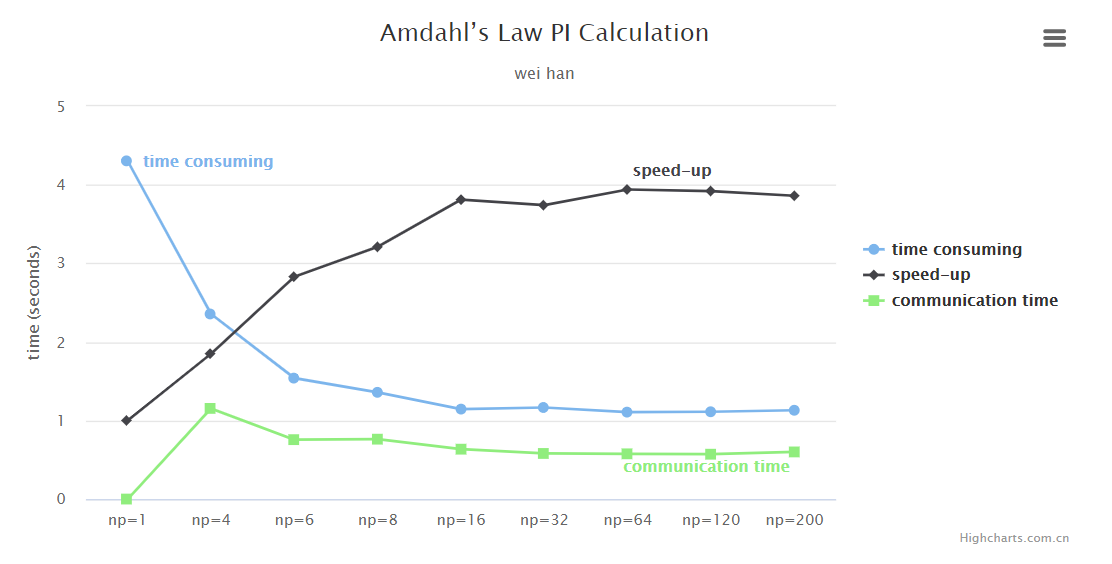
T(2) = 0.8

If N =4 T=1 B=0.6 then:

T(4) = B + (T(4)-B)/4

T(4) = 0.7

# My program and data



The following images were captured on the mighty cluster,

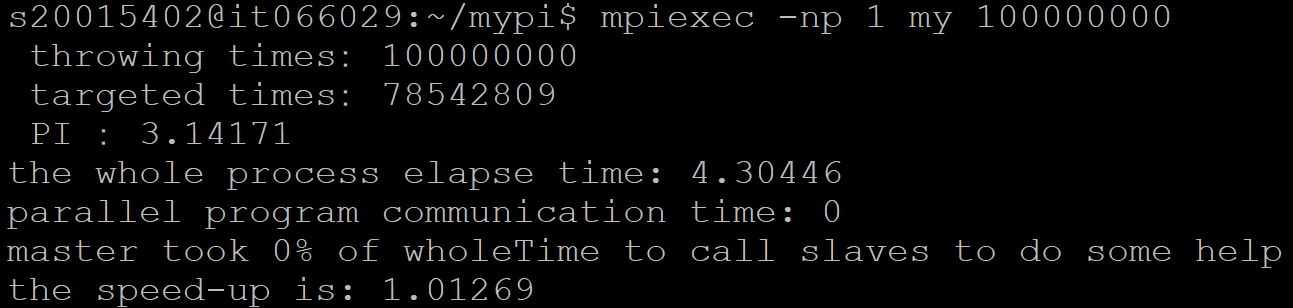
“my” is the name of my cpp program

When np=1 throwing 100 million times,

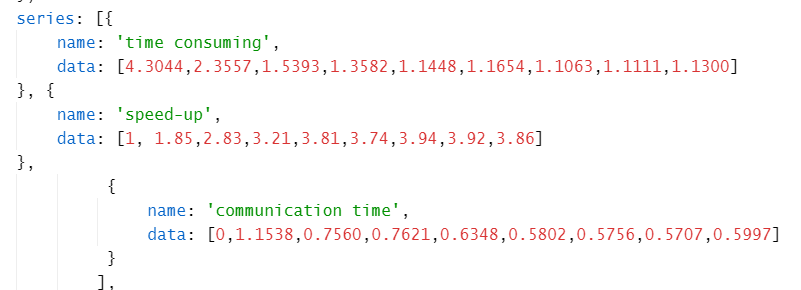
time consuming nearly: 4.3044 seconds

parallel program communication time: 0 second

the speed-up is almost: 1

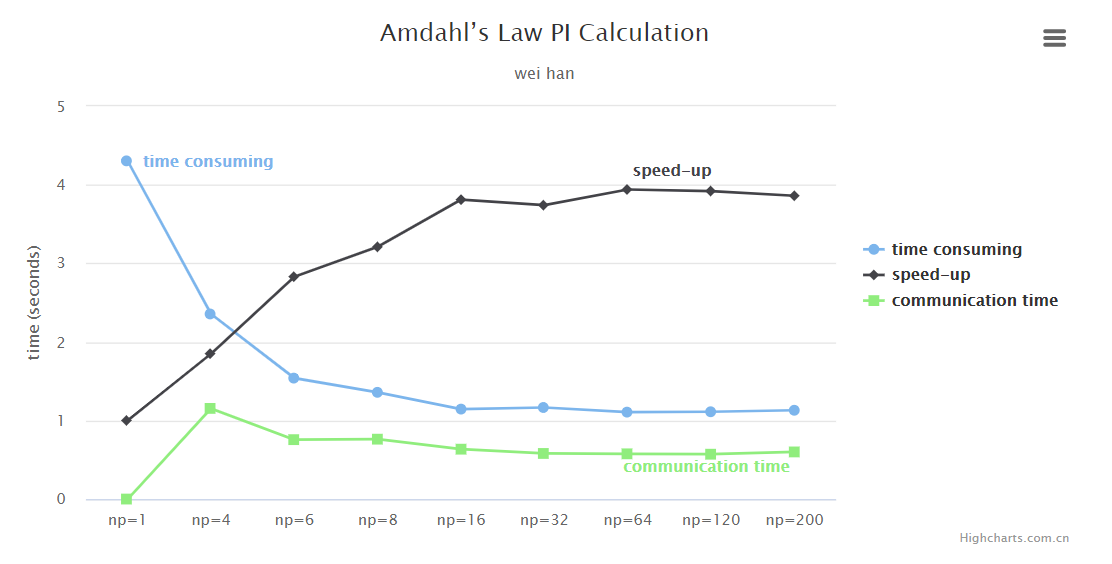


As we can see, with the increasing number of processors, the time consuming of whole process is decreasing , which is shorten by more processors .



However, when the number of processors get the limit such as 60 or 100 and more, the whole time consuming is almost steady, which is reaching the limitation. In addition, with number of processors is increasing, the time consuming is increasing as well.

# Parallel speed-ups



When we want to optimize the sequential part of a program, we need a factor O to represent that.

T(O,N) = B/O + (1-(B/O))/N

B/O is the time of sequential part.

The factor O can shorten the program execution time in sequential part.

If the time of the old version of the program is T, then the speedup will be:

SpeedUp(SU) = T / T(O,N)

When we set T = 1 then, SU= 1/ (B/O + (1-(B/O))/N)

If B=0.6 ,O=2, N=4

SU =2.1052…..the result varies according to these parameters.

It means that the original speed has been speed up by “Speed-ups” at least 2 times.

**The SpeedUp is almost 4,** which means that if we made the sequential program become parallel program by adding more processors, we can shorten the time consuming of running program to calculate PI.

Actually, the SpeedUp is what I expected, we just need to spend nearly 25% of the original time to calculate PI value.

The communication time is always kept on about 50% of the whole time consuming. Actually, I though I would be higher than before with the increased number of processors, but it didn’t. Lucky me.