Project 2 N-body Simulation

1. Specification of Program

a. What is N-body simulation?

Textbook: Chapter 6

Wiki: https://en.wikipedia.org/wiki/N-body_simulation

Princeton: http://physics.princeton.edu/~fpretori/Nbody/intro.htm

Or Google "N-body"

b. Introduction of the program.

The default N-body number is 200, and iteration step is 10000. The default set is just for testing and watching the animation. In main() function, "nanosleep(&delay, &remaining)" can set the FPS of animation, you can uncomment this function to watch the animation slowly. But please remember to comment this function when you calculate the running time, because this function may slow the result.

Input "`./nbody 1000`" can directly change the N-body number.

For multithreading test, you should increase the N-body number to 1000, 2000, 5000 or larger, and decrease the iteration step to 100 or 200 for saving time. When N-body number is very small, the parallel result is not distinct.

Function "position_step" is the core function, which calculate the forces of bodies and update the position of bodies. You should parallelize this function. And other function will not be included in the time calculation.

c. Score rule

1. Use pthread or OpenMP to parallelize the core function. OpenMP is recommended, and it's also easier than pthread.

2. Try to use some optimization skill to improve the performance. If there exits false sharing, try to eliminate it. Besides, try to increase the cache hit rate, and reduce the times of memory access, or make the memory access continuous.

3. Bonus

Try to implement the Barnes-Hut algorithm.

https://en.wikipedia.org/wiki/Barnes%E2%80%93Hut_simulation

http://portillo.ca/nbody/barnes-hut/

d. If you have any question, please ask in QQ group or email: caijinjin4@sjtu.edu.cn.

2. How to run the program

For Windows:

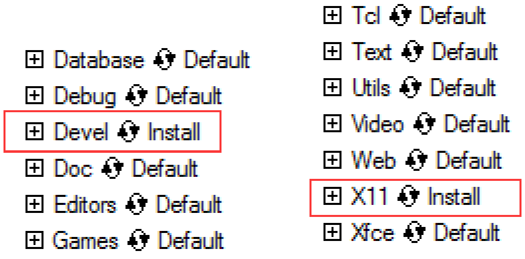
Because the GUI interface of this program is based on Xlib, and Windows do not support Xlib. At first, you need to install Cygwin. Here is the step.

a. Download Cygwin 64bit from http://cygwin.com/setup-x86_64.exe

or 32bit http://cygwin.com/setup-x86.exe

b. Install Cygwin.

- Choose “Install from Internet”
- Set the install path (root path for Cygwin) and package directory
- Choose “Direct Connection”
- Set the mirror site: http://mirrors.163.com/cygwin or http://mirrors.ustc.edu.cn/cygwin (Add this site), the default mirror site of Cygwin installer may be slow.
- Then choose the package, remember to install Devel and X11. May need about 10GB disk space.



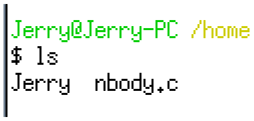
⊞ Database ⟳ Default
⊞ Debug ⟳ Default
⊞ Devel ⟳ Install
⊞ Doc ⟳ Default
⊞ Editors ⟳ Default
⊞ Games ⟳ Default

⊞ Tcl ⟳ Default
⊞ Text ⟳ Default
⊞ Utils ⟳ Default
⊞ Video ⟳ Default
⊞ Web ⟳ Default
⊞ X11 ⟳ Install
⊞ Xfce ⟳ Default

- Then wait until the installation completes, and there may be some “download incomplete” error during installation process, it is OK to ignore this error.

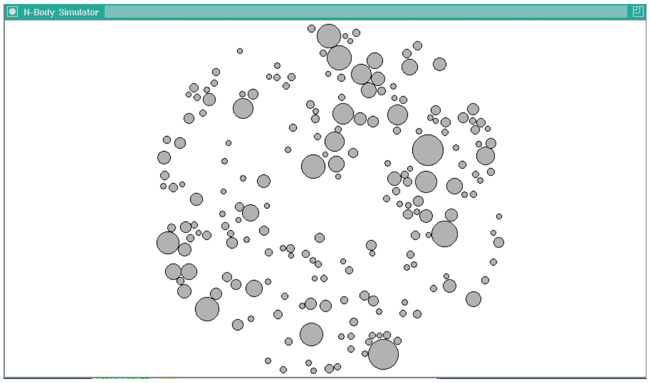
c. Compile and Run the program in Cygwin.

- Open the Cygwin Terminal
- Input “`startx`”
- Go the directory of “nbody.c” (The default root path is the install path)



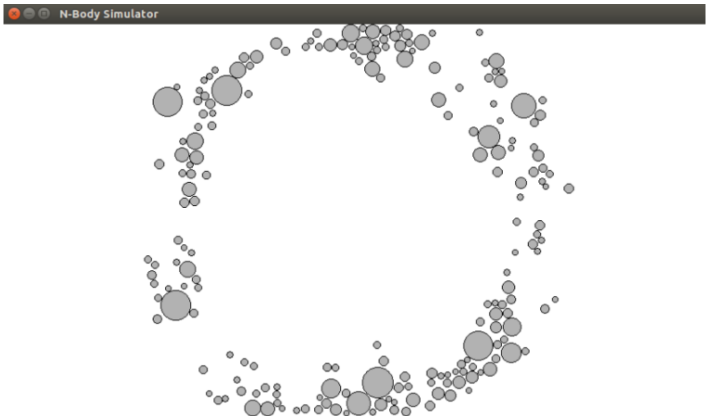
```
Jerry@Jerry-PC /home
$ ls
Jerry  nbody.c
```

- Input “`gcc nbody.c -o nbody -lm -lX11`” to compile (The command is the same as Linux)
- Input “`./nbody.exe`” to run



For Linux: (Linux is recommended)

- Compile the program: “`gcc nbody.c -o nbody -lm -lX11`”
  if error: X11/Xlib.h no such file or directory, you should install X11 library.
  - Ubuntu: apt-get install libX11-dev
  - CentOS: yum install libX11-devel
  - Other: just search Google “libX11 + System Name”
- Run the program: “./nbody”



For Mac:

You may should install XQuartz to get the X11 lib support. I’m sorry for that I don’t have the Mac environment, so I can’t provide any installation instructions here, and you need to install X11 lib by yourselves.

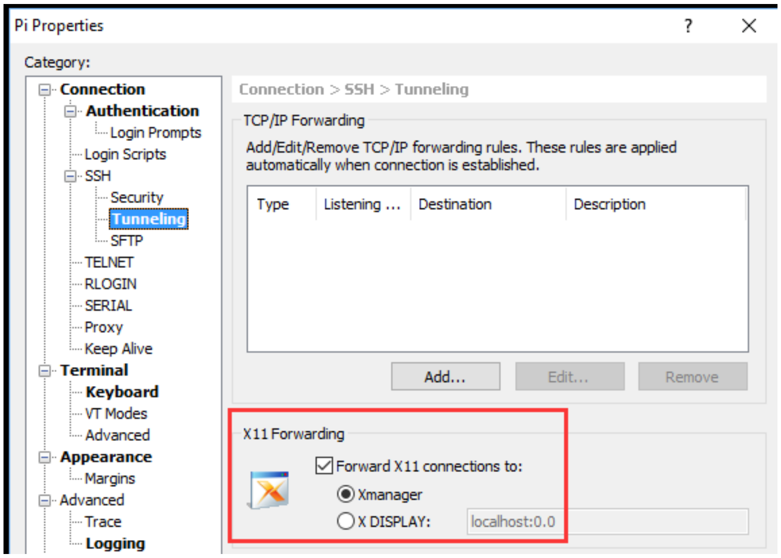
For Pi supercomputer:

- Load Module Path
  unset MODULEPUATH; module use /lustre/usr/modulefiles/pi
- Load gcc
  module load purge
  module load gcc
- Compile the program: “`gcc nbody.c -o nbody -lm -lX11`”
- Run the program on login node: “`./`nbody”

Important Note:

a. Run program on login node only for test, because long time running may affect other users, please stop the program as soon as possible.)

b. To watch the GUI, if you're using Xshell, you need to install Xmanager first. And then turn on the X11 Forwarding.



Add/Edit/Remove TCP/IP forwarding rules. These rules are applied automatically when connection is established.

X11 Forwarding

☑ Forward X11 connections to:

● Xmanager

○ X DISPLAY: localhost:0.0

- Run the program on the computing nodes:

  a. Set "`#define NOT_RUN_ON_PI 1`" to "`#define NOT_RUN_ON_PI 0`"

  Computing node do not support X11, so program will only output the calculation time.

  b. Recompile the program

  c. Write the slurm scripts "nbody.slurm"

  (Example below, more information on https://pi.sjtu.edu.cn/doc/samples/)

```
#!/bin/bash

#SBATCH --job-name=nbody
#SBATCH --partition=cpu
#SBATCH -n 1
#SBATCH --output=%j.out
#SBATCH --error=%j.err

unset MODULEPUATH; module use /lustre/usr/modulefiles/pi
module purge
module load gcc

./nbody
```

  d. Submit your job "`sbatch -p cpu nbody.slurm`"

  e. Wait for result.