Assignment5

1.安装 OpenCilk 并记录执行过程，了解并尝试 pre-build binaries 和 build from source code两种安装方式。

Pre-build binaries

下载对应的压缩包

```
wangwenqing@ubuntu:~/Desktop/OpenCilk-10.0.1-Linux/share/clang$ ./bash-autocompl
ete.sh
wangwenqing@ubuntu:~/Desktop/OpenCilk-10.0.1-Linux/share/clang$
```

运行对应的.sh文件

build from source code

出现报错：

1.fatal: could not create work tree dir 'infrastructure': No space left on device

虚拟机内存授权故障，重新设置properties,如还不行则需要重装系统

2.gnutls recv error

```
wangwenqing@ubuntu:~/Desktop$ git clone -b openclick/v1.0 https://github.com/Op
enClick/infrastructure
Cloning into 'infrastructure'...
fatal: unable to access 'https://github.com/OpenClick/infrastructure/': GnuTLS
recv error (-110): The TLS connection was non-properly terminated.
```

实验步骤

1. clone opencilk infrastructure repository

```
git clone -b opencilk/v1.0 https://github.com/OpenCilk/infrastructure
```

```
(base) wangwenqingdeMacBook-Pro:软件系统优化 wangwenqing$ git clone -b opencilk/v1.0 https://github.com/OpenCilk
/infrastructure
Cloning into 'infrastructure'...
remote: Enumerating objects: 208, done.
remote: Counting objects: 100% (208/208), done.
remote: Compressing objects: 100% (127/127), done.
remote: Total 208 (delta 99), reused 139 (delta 59), pack-reused 0
Receiving objects: 100% (208/208), 43.36 KiB | 121.00 KiB/s, done.
Resolving deltas: 100% (99/99), done.
Note: switching to '7918a23df3201d8721291059ae6841ad586482ca'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false
```

2. Run the following script to get the OpenCilk source code

```
infrastructure/tools/get $(pwd)/opencilk
```

```
+ git clone -b opencilk/v1.0 https://github.com/OpenCilk/opencilk-project $'/Users/wangwenqing/Desktop/软件系?\237?\230?\214\226/opencilk'
Cloning into '/Users/wangwenqing/Desktop/软件系统优化/opencilk'...
remote: Enumerating objects: 4237713, done.
remote: Counting objects: 100% (801/801), done.
remote: Compressing objects: 100% (502/502), done.
error: RPC failed; curl 56 LibreSSL SSL_read: SSL_ERROR_SYSCALL, errno 54
fatal: the remote end hung up unexpectedly
fatal: early EOF
fatal: index-pack failed
```

下载失败 查看网络 vpn等 重新下载

3. run the script to build opencilk

```
infrastructure/tools/build $(pwd)/opencilk $(pwd)/build
```

```
(base) wangwenqingdeMacBook-Pro:软件系统优化 wangwenqing$ infrastructure/tools/build $(pwd)/opencilk $(pwd)/build
2021年11月13日 星期六 15时56分03秒 CST
Building with 8 parallel jobs
```

2.阅读教程“Cilk Tutorial”，理解 cilk_spawn，cilk_sync，cilk_for 和 locks 等基本实现。

Click spawn:当 cilk_spawn 在函数调用之前创建并行工作，从而导致函数被spawn，最后的输出结果（如有）是乱序的。当一个函数产生另一个函数时，原始函数被称为父函数，而另一个函数被称为子函数。click_spawn不能作为另一个函数的参数。spawning 的语义与 C/C++ 函数或方法调用的不同之处在紧随spawn后的代码中，允许与child并行执行。

Click sync:cilk_sync 保证所有先前生成的任务在程序继续之前相互等待完成。

Click for:cilk_for 构造允许循环迭代并行运行的循环。cilk_for 将一个循环分成包含一个或多个循环迭代的块。 一旦循环中断，每个块都会在特定的执行线程上执行。

click for 有一些限制：1）不能更改循环体中的循环控制变量。2）不能在 C++ 的循环之外声明循环控制变量。

cilk_for 语句将循环分成多个较小的块，这些块在特定的执行线程上运行。 每个块中的最大迭代次数定义为粒度。 作为一个块运行的实际迭代次数通常小于粒度。

Locks:锁是防止多个线程同时更改变量的同步机制，可以有效消除数据竞争

有一些情况锁无法消除数据竞争。首先，可能会发生死锁，即所有线程都在互相等待。 其次，由于线程必须相互等待，代码的锁定部分被序列化，从而导致性能问题。 在 Cilk 中，通过减速器解决与锁相关的大多数问题

3.阅读教程“Research and Teaching with OpenCilk”，了解 OpenCilk 的研发历程，尝试教程P31~P36 的 6 个 Demo，记录执行过程和结果。

Demo1 Compile and Run

报错 libtinfo.so.5: cannot open shared object file

```
wangwenqing@ubuntu:~/Desktop/OpenCilk-10.0.1-Linux$ ~/Desktop/OpenCilk-10.0.1-Li
nux/bin/clang fib.c -o fib -O3 -fopencilk -static
/home/wangwenqing/Desktop/OpenCilk-10.0.1-Linux/bin/clang: error while loading s
hared libraries: libtinfo.so.5: cannot open shared object file: No such file or
directory
```

解决方法：sudo apt-get install libncurses5

运行过程和输出

```
wangwenqing@ubuntu:~/Desktop/OpenCilk-10.0.1-Linux$ ~/Desktop/OpenCilk-10.0.1-Li
nux/bin/clang fib.c -o fib -O3 -fopencilk -static
wangwenqing@ubuntu:~/Desktop/OpenCilk-10.0.1-Linux$ ./fib 35
fib(35) = 9227465
wangwenqing@ubuntu:~/Desktop/OpenCilk-10.0.1-Linux$
```

Demo2 Using Cilksan

运行过程和输出

⚠️ 命令最好直接在虚拟机打一遍 复制黏贴会报错

```
wangwenqing@ubuntu:~/Desktop/OpenCilk-10.0.1-Linux$ ~/Desktop/OpenCilk-10.0.1-Li
nux/bin/clang nqueens.c -o nqueens -fopencilk -fsanitize=cilk -Og -g
clang-10: error: no such file or directory: '-o'
clang-10: error: no such file or directory: 'nqueens'
clang-10: error: no such file or directory: '-fopencilk'
clang-10: error: no such file or directory: '-fsanitize=cilk'
clang-10: error: no such file or directory: '-Og'
```

手打一遍即恢复正常

```
wangwenqing@ubuntu:~/Desktop/OpenCilk-10.0.1-Linux$ ~/Desktop/OpenCilk-10.0.1-Li
nux/bin/clang nqueens.c -o nqueens -fopencilk -fsanitize=cilk -Og -g
wangwenqing@ubuntu:~/Desktop/OpenCilk-10.0.1-Linux$ ./nqueens 12
Running Cilksan race detector.
Running ./nqueens with n = 12.
Race detected on location 7f9e967842b6
*     Read 499116 nqueens /home/wangwenqing/Desktop/OpenCilk-10.0.1-Linux/nqueen
s.c:62:3
|         `-to variable a (declared at /home/wangwenqing/Desktop/OpenCilk-10.0.1-
Linux/nqueens.c:48)
+     Call 499a0f nqueens /home/wangwenqing/Desktop/OpenCilk-10.0.1-Linux/nqueen
s.c:67:29
+    Spawn 49922c nqueens /home/wangwenqing/Desktop/OpenCilk-10.0.1-Linux/nqueen
s.c:67:29
|*   Write 4991fc nqueens /home/wangwenqing/Desktop/OpenCilk-10.0.1-Linux/nqueen
s.c:65:10
||        `-to variable b (declared at /home/wangwenqing/Desktop/OpenCilk-10.0.1-
Linux/nqueens.c:50)
\| Common calling context
 +    Call 499a0f nqueens /home/wangwenqing/Desktop/OpenCilk-10.0.1-Linux/nqueen
s.c:67:29
 +   Spawn 49922c nqueens /home/wangwenqing/Desktop/OpenCilk-10.0.1-Linux/nqueen
s.c:67:29
 +    Call 499a0f nqueens /home/wangwenqing/Desktop/OpenCilk-10.0.1-Linux/nqueen
```

```
     Spawn 49922c nqueens /home/wangwenqing/Desktop/OpenCilk-10.0.1-Linux/nqueen
s.c:67:29
      Call 499a0f nqueens /home/wangwenqing/Desktop/OpenCilk-10.0.1-Linux/nqueen
s.c:67:29
     Spawn 49922c nqueens /home/wangwenqing/Desktop/OpenCilk-10.0.1-Linux/nqueen
s.c:67:29
      Call 499a0f nqueens /home/wangwenqing/Desktop/OpenCilk-10.0.1-Linux/nqueen
s.c:67:29
     Spawn 49922c nqueens /home/wangwenqing/Desktop/OpenCilk-10.0.1-Linux/nqueen
s.c:67:29
      Call 499a0f nqueens /home/wangwenqing/Desktop/OpenCilk-10.0.1-Linux/nqueen
s.c:67:29
     Spawn 49922c nqueens /home/wangwenqing/Desktop/OpenCilk-10.0.1-Linux/nqueen
s.c:67:29
      Call 4996cd main /home/wangwenqing/Desktop/OpenCilk-10.0.1-Linux/nqueens.c
:100:9

1.518000
Total number of solutions : 14200

Cilksan detected 1 distinct races.
Cilksan suppressed 781409 duplicate race reports.

wangwenqing@ubuntu:~/Desktop/OpenCilk-10.0.1-Linux$
```

增加 –fcilktool=cilkscale 编译符来用Cilkscale 计算 work and span

```
wangwenqing@ubuntu:~/Desktop/OpenCilk-10.0.1-Linux$ ~/Desktop/OpenCilk-10.0.1-Li
nux/bin/clang qsort.c -o qsort -fopencilk -fcilktool=cilkscale -O3
wangwenqing@ubuntu:~/Desktop/OpenCilk-10.0.1-Linux$ ./qsort 10000000
Sorting 10000000 integers
All sorts succeeded
tag,work (seconds),span (seconds),parallelism,burdened_span (seconds),burdened_p
arallelism
,2.42906,0.205818,11.802,0.206113,11.7851
wangwenqing@ubuntu:~/Desktop/OpenCilk-10.0.1-Linux$
```

Demo3: Analyze a Region

报错：use of undeclared identifier 'wsp_t'

```
wangwenqing@ubuntu:~/Desktop/OpenCilk-10.0.1-Linux$ ~/Desktop/OpenCilk-10.0.1-Li
nux/bin/clang qsort.c -o qsort -fopencilk -fcilktool=cilkscale -O3
qsort.c:84:3: error: use of undeclared identifier 'wsp_t'
  wsp_t start,end;
  ^
qsort.c:112:2: error: use of undeclared identifier 'start'
 start = wsp_getworkspan();
 ^
qsort.c:112:10: warning: implicit declaration of function 'wsp_getworkspan' is i
nvalid in C99 [-Wimplicit-function-declaration]
 start = wsp_getworkspan();
         ^
qsort.c:114:2: error: use of undeclared identifier 'end'
 end = wsp_getworkspan();
 ^
qsort.c:135:1: warning: implicit declaration of function 'wsp_dump' is invalid i
```

补充头文件 #include <cilk/cilkscale.h>

运行过程和输出

```
wangwenqing@ubuntu:~/Desktop/OpenCilk-10.0.1-Linux$ vim qsort.c
wangwenqing@ubuntu:~/Desktop/OpenCilk-10.0.1-Linux$ ~/Desktop/OpenCilk-10.0.1-Li
nux/bin/clang qsort.c -o qsort -fopencilk -fcilktool=cilkscale -O3
wangwenqing@ubuntu:~/Desktop/OpenCilk-10.0.1-Linux$ ./qsort 10000000
Sorting 10000000 integers
All sorts succeeded
tag,work (seconds),span (seconds),parallelism,burdened_span (seconds),burdened_p
arallelism
sample_qsort,2.39329,0.106702,22.4298,0.107076,22.3513
,2.49517,0.208576,11.9629,0.208951,11.9414
wangwenqing@ubuntu:~/Desktop/OpenCilk-10.0.1-Linux$
```

可以注意到多了sample_qsort这一行输出

Demo 4: Download cilkscaler visualizer

```
wangwenqing@ubuntu:~/Desktop/OpenCilk-10.0.1-Linux$ git clone https://github.com
/OpenCilk/productivity-tools.git
Cloning into 'productivity-tools'...
remote: Enumerating objects: 602, done.
remote: Counting objects: 100% (125/125), done.
remote: Compressing objects: 100% (82/82), done.
remote: Total 602 (delta 55), reused 83 (delta 42), pack-reused 477
Receiving objects: 100% (602/602), 398.26 KiB | 101.00 KiB/s, done.
Resolving deltas: 100% (320/320), done.
wangwenqing@ubuntu:~/Desktop/OpenCilk-10.0.1-Linux$
```

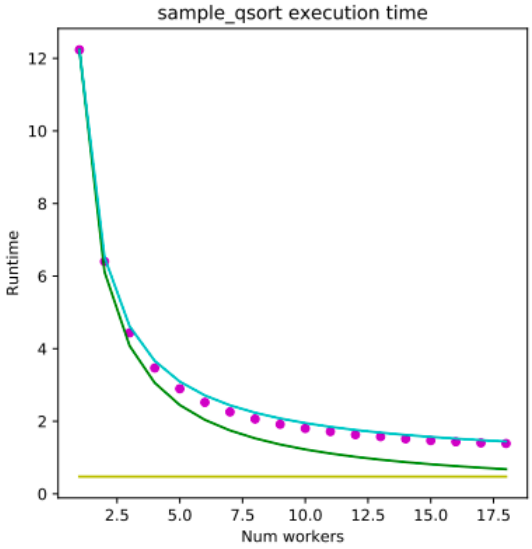
Demo 5:Cilkscaler visualizer

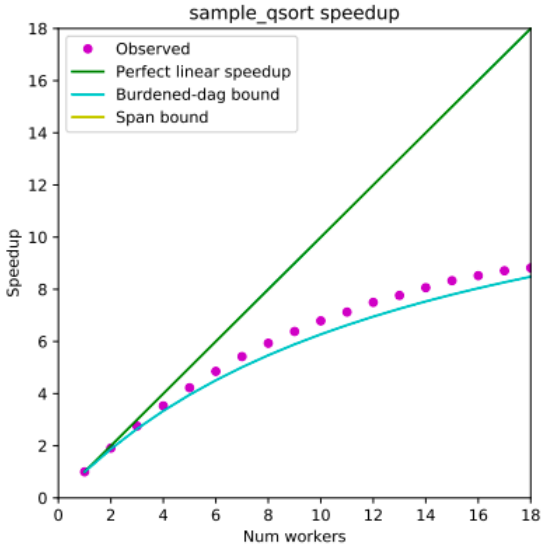
```
wangwenqing@ubuntu:~/Desktop/OpenCilk-10.0.1-Linux/productivity-tools/Cilkscale_
vis$ python3 cilkscale.py -c ~/Desktop/OpenCilk-10.0.1-Linux/qsort -b ~/Desktop/
OpenCilk-10.0.1-Linux/qsort-bench --args 10000000 -rplot 0,1
Namespace(args=['10000000'], cilkscale='/home/wangwenqing/Desktop/OpenCilk-10.0.
1-Linux/qsort', cilkscale_benchmark='/home/wangwenqing/Desktop/OpenCilk-10.0.1-L
inux/qsort-bench', cpu_counts=None, output_csv='out.csv', output_plot='plot.pdf'
, rows_to_plot='0,1')
WARNING:cilkscale.py:matplotlib required to generate plot.

>> STDOUT (/home/wangwenqing/Desktop/OpenCilk-10.0.1-Linux/qsort 10000000)
Sorting 10000000 integers
All sorts succeeded
<< END STDOUT

>> STDERR (/home/wangwenqing/Desktop/OpenCilk-10.0.1-Linux/qsort 10000000)
<< END STDERR

INFO:runner:Generating scalability data for 1 cpus.
INFO:runner:CILK_NWORKERS=1 taskset -c 0 /home/wangwenqing/Desktop/OpenCilk-10.0
.1-Linux/qsort-bench 10000000
```

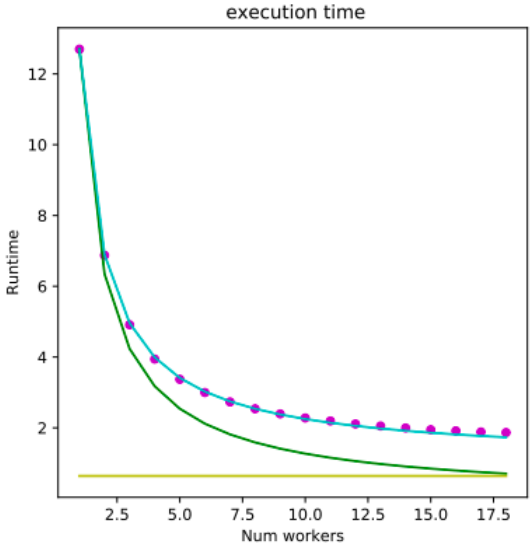
安装matplotlib

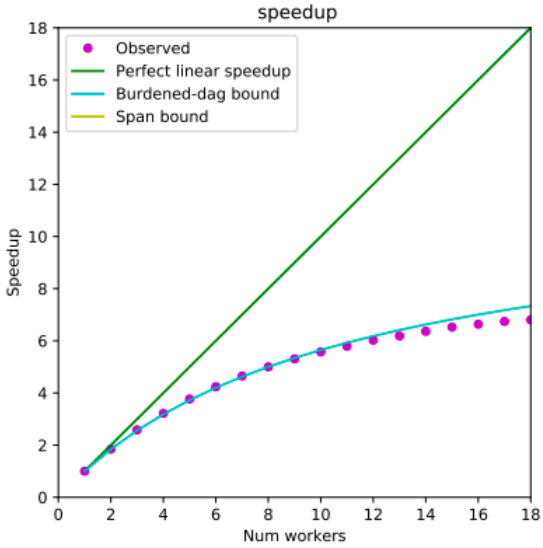
```
OpenCilk-10.0.1-Linux/qsort-bench --args 10000000 -rplot 0,1
Namespace(args=['10000000'], cilkscale='/home/wangwenqing/Desktop/OpenCilk-10.0.
1-Linux/qsort', cilkscale_benchmark='/home/wangwenqing/Desktop/OpenCilk-10.0.1-L
inux/qsort-bench', cpu_counts=None, output_csv='out.csv', output_plot='plot.pdf'
, rows_to_plot='0,1')

>> STDOUT (/home/wangwenqing/Desktop/OpenCilk-10.0.1-Linux/qsort 10000000)
Sorting 10000000 integers
All sorts succeeded
<< END STDOUT

>> STDERR (/home/wangwenqing/Desktop/OpenCilk-10.0.1-Linux/qsort 10000000)
<< END STDERR

INFO:runner:Generating scalability data for 1 cpus.
INFO:runner:CILK_NWORKERS=1 taskset -c 0 /home/wangwenqing/Desktop/OpenCilk-10.0
.1-Linux/qsort-bench 10000000
INFO:plotter:Generating plot
wangwenqing@ubuntu:~/Desktop/OpenCilk-10.0.1-Linux/productivity-tools/Cilkscale_
vis$ ls
cilkscale.py    out.csv  plot.pdf    __pycache__
CMakeLists.txt  out.pdf  plotter.py  runner.py
wangwenqing@ubuntu:~/Desktop/OpenCilk-10.0.1-Linux/productivity-tools/Cilkscale_
vis$
```

得到的performance图片


sample_qsort execution time
Runtime
Num workers


sample_qsort speedup
Observed
Perfect linear speedup
Burdened-dag bound
Span bound
Speedup
Num workers


execution time
Runtime
Num workers


speedup
Observed
Perfect linear speedup
Burdened-dag bound
Span bound
Speedup
Num workers