Developing Soft and Parallel Programming Skills Using Project-Based Learning

Fall 2019

Kenny and the Rest
Abdiaziz Dakane, Soojie Chin, Nish Patel, Brett Krokoff, Kehinde Adedara

Planning and Scheduling

Assignee Name	**Email**	**Task**	**Duration(hours)**	**Dependency**	**Due Date**	**Note**
Abdiaziz Dakane	Adakane1@student.gsu.edu	Task5 : Presentation	2 - 5 hours	None	12/05/2019	Get it ready as soon as possibly so soojie can have time to get the report ready
Soojie Chin	Schin6@student.gsu.edu	Task 4: Report	2 - 3 hours	None	12/06/2019	Get the report ready before 3-4 hours of due date
Nish Patel	Npatel161@student.gsu.edu	Team coordinator: times and due dates	2 - 4 hours	None	12/05/2019	Going to take care of task 1 and 2
Brett Krokoff	Bkrokoff1@student.gsu.edu	Task 3a	2 - 3 hours	None	12/05/2019	Get it ready as soon as possibly so soojie can have time to get the report ready
Kehinde Adedara	Kadedara1@student.gsu.edu	Task 3b and task 6	2 - 5 hours	None	12/05/2019	Take your time and finish it before the due date.

Parallel Programming Skills

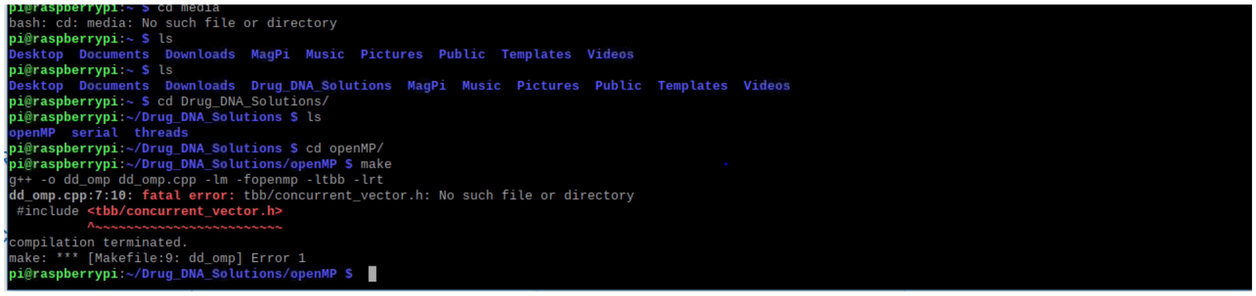
What are the basic steps (show all steps) in building a parallel program? Show at least one example.
The first step is to identify a set of tasks that can run concurrently. The next step is decomposition, which is splitting it up. The next step is assignment which assigns each task that needs to be completed. The next step is the processing where each task is processed. The final step is the mapping and scheduling, which executed the program. One example of a parallel program, is a program that uses the Master/Worker technique in order to find an approximation of pi.
What is MapReduce? MapReduce allows engineers to perform simple computations while ignoring the smaller details of programming. It helps with big data processing.

What is map and what is reduce? Map takes as input a function and a sequence of values, it then applies the function to each value in the sequence. Reduce combines all the elements of a sequence using a binary operation.
Why MapReduce? It used to process large amounts of raw data, that is so large it must be distributed across thousands of machines in order to be processed in a reasonable time.
Show an example for MapReduce.
An example of MapReduce is Google using it in order to process large amounts of data. For instance, they use MapReduce to process crawled documents or web request logs.
Explain in your own words how MapReduce model is executed?
It forks the input files into different pieces, then starts up multiple versions of the copy on many different machines. One copy, the master, assigns map tasks or reduce tasks to other machines, the ones assigned map tasks then reads the contents of the program and the in picks pairs out of the data and sends it to a user defined map function. Paist that buffer are written on local disks, and the locations of the buffers are sent back to the master who sends it to the ones responsible for reducing. The reduce workers read the buffered data and then sorts it, then the output is attached to a final output file. When all the tasks get completed, the master powers up the program and everything involved in the Map reduce call moves back to the user code.
List and describe three examples that are expressed as MapReduce computations.
Distributed Grep: The map function emits a line if it matches the given pattern. The reduce function is an identity function that just copies the supplied intermediate data to the output.
Count of URL Access Frequency: map function processes logs of web page requests and outputs . The reduce function adds together all values for the same URL and emits a pair.
Reverse Web-Link Graph: The map function outputs pairs for each link to target a url found in a page named “source”. The reduce function concatenates the list of all source URLS associated with the given target URL and emits the pair:
When do we use OpenMP, MPI, and Mapreduce(Hadoop), and why?
Programs use OpenMp so they can be compiled into multithreaded programs, so that threads can share threads. The reason you would use openmp is because it is easier than using pthreads since the compiler takes care of transforming the sequential code into parallel code. MPI would be used to help provide “point-to-point, collective, one-sided, and parallel I/O communication models”. MPI also sends messages of gigabyte size between processes. You would use MPI because it is a high level abstraction for parallel programming and programmers can easily construct parallel and distributed processing applications without a deep understanding of the underlying mechanisms. You would use MapReduce for big data processing because MapReduce handles data without loading all of it into the memory and in addition the program is executed in parallel. Hadoop also provides everything needed for distributed and parallel processing.
Explain what a Drug Design and DNA problem is.
Drug design involves designing small molecules that are complementary in shape and charge to the biomolecular target with which the molecules will bind to. Frequently, drug design relies on computer modeling techniques known as “computer-aided drug design”. However, drug design does not necessarily mean drugs and relates more to ligand design that identifies target DNA using virtual screenings. Using DNA as the target in the drug design experiment can lead to problems. This is due to the fact that there are several compounds that are known to negatively interact with DNA so these interactions can cause some unwanted effects and alter the DNA. Another problem deals with the minor groove binders of DNA. Minor groove binders tend to

interact with just small molecules and typically the number of minor grooves on a DNA molecule outnumber the major grooves which can have serious effects on gene transcription.

ARM Assembly Programming

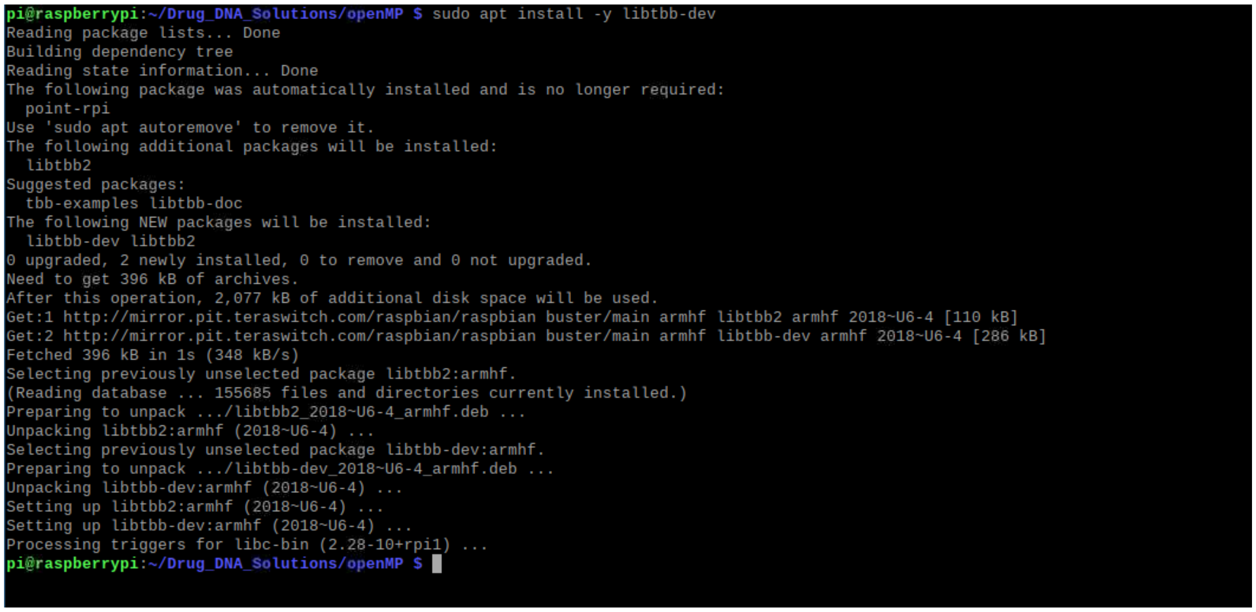
The serial program compiles alright, but the OpenMp had some issues.



```
pi@raspberrypi:~ $ cd media
bash: cd: media: No such file or directory
pi@raspberrypi:~ $ ls
Desktop  Documents  Downloads  MagPi  Music  Pictures  Public  Templates  Videos
pi@raspberrypi:~ $ ls
Desktop  Documents  Downloads  Drug_DNA_Solutions  MagPi  Music  Pictures  Public  Templates  Videos
pi@raspberrypi:~ $ cd Drug_DNA_Solutions/
pi@raspberrypi:~/Drug_DNA_Solutions $ ls
openMP  serial  threads
pi@raspberrypi:~/Drug_DNA_Solutions $ cd openMP/
pi@raspberrypi:~/Drug_DNA_Solutions/openMP $ make
g++ -o dd_omp dd_omp.cpp -lm -fopenmp -ltbb -lrt
dd_omp.cpp:7:10: fatal error: tbb/concurrent_vector.h: No such file or directory
 #include <tbb/concurrent_vector.h>
          ^~~~~~~~~~~~~~~~~~~~~~~~~
compilation terminated.
make: *** [Makefile:9: dd_omp] Error 1
pi@raspberrypi:~/Drug_DNA_Solutions/openMP $
```

Issue ran into while trying to compile the openMP file. But according to the issue it occurs that the we need to install the file/directory because it cant find it.

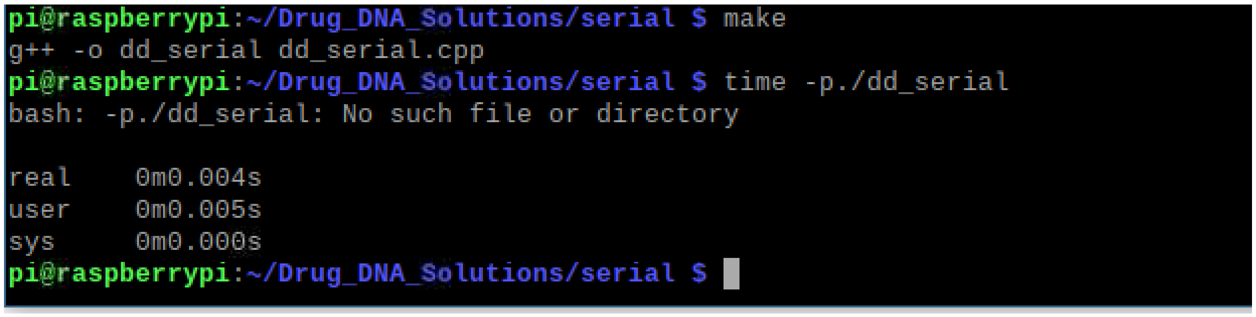
So we install the libtbb-dev library.



```
pi@raspberrypi:~/Drug_DNA_Solutions/openMP $ sudo apt install -y libtbb-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  point-rpi
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libtbb2
Suggested packages:
  tbb-examples libtbb-doc
The following NEW packages will be installed:
  libtbb-dev libtbb2
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 396 kB of archives.
After this operation, 2,077 kB of additional disk space will be used.
Get:1 http://mirror.pit.teraswitch.com/raspbian/raspbian buster/main armhf libtbb2 armhf 2018~U6-4 [110 kB]
Get:2 http://mirror.pit.teraswitch.com/raspbian/raspbian buster/main armhf libtbb-dev armhf 2018~U6-4 [286 kB]
Fetched 396 kB in 1s (348 kB/s)
Selecting previously unselected package libtbb2:armhf.
(Reading database ... 155685 files and directories currently installed.)
Preparing to unpack .../libtbb2_2018~U6-4_armhf.deb ...
Unpacking libtbb2:armhf (2018~U6-4) ...
Selecting previously unselected package libtbb-dev:armhf.
Preparing to unpack .../libtbb-dev_2018~U6-4_armhf.deb ...
Unpacking libtbb-dev:armhf (2018~U6-4) ...
Setting up libtbb2:armhf (2018~U6-4) ...
Setting up libtbb-dev:armhf (2018~U6-4) ...
Processing triggers for libc-bin (2.28-10+rpi1) ...
pi@raspberrypi:~/Drug_DNA_Solutions/openMP $
```

And we can finally compile the openMP program.

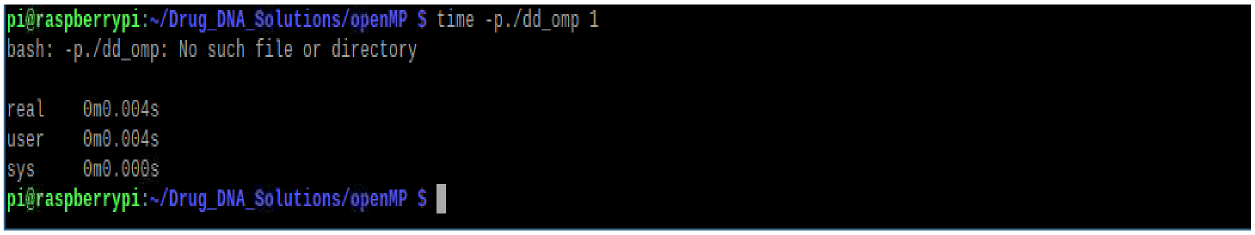
Measure Run-Time



```
pi@raspberrypi:~/Drug_DNA_Solutions/serial $ make
g++ -o dd_serial dd_serial.cpp
pi@raspberrypi:~/Drug_DNA_Solutions/serial $ time -p./dd_serial
bash: -p./dd_serial: No such file or directory

real    0m0.004s
user    0m0.005s
sys     0m0.000s
pi@raspberrypi:~/Drug_DNA_Solutions/serial $
```

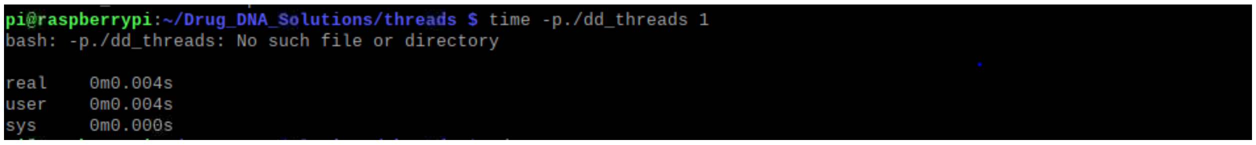
Time -p ./dd_omp 1



```
pi@raspberrypi:~/Drug_DNA_Solutions/openMP $ time -p./dd_omp 1
bash: -p./dd_omp: No such file or directory

real    0m0.004s
user    0m0.004s
sys     0m0.000s
pi@raspberrypi:~/Drug_DNA_Solutions/openMP $
```

Time -p ./dd_threads 1

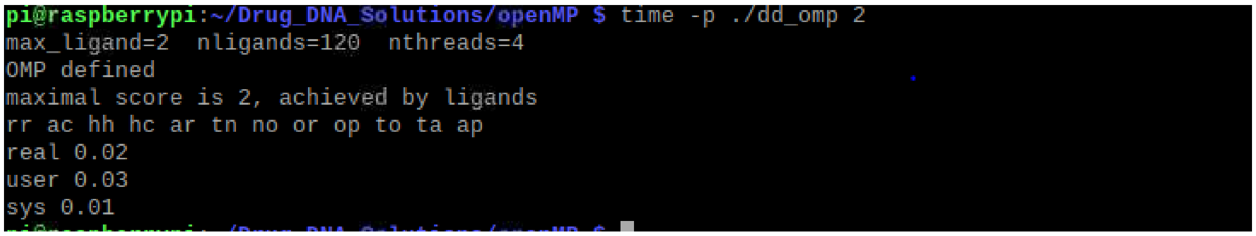


```
pi@raspberrypi:~/Drug_DNA_Solutions/threads $ time -p./dd_threads 1
bash: -p./dd_threads: No such file or directory

real    0m0.004s
user    0m0.004s
sys     0m0.000s
```

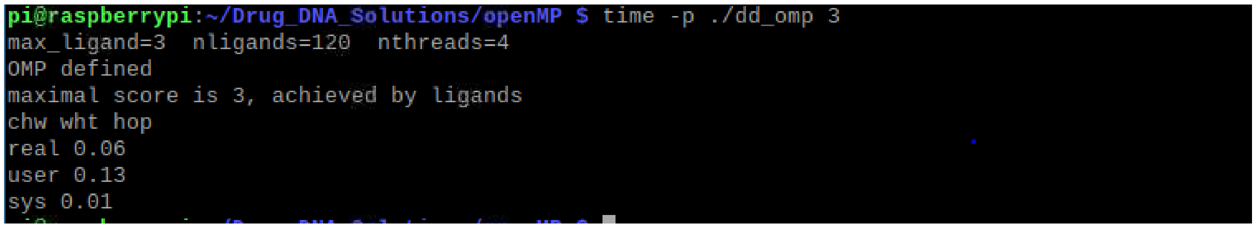
Use for real time:

Omp 2



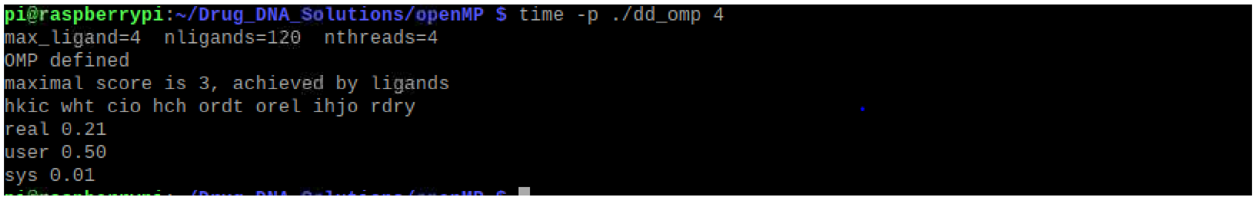
```
pi@raspberrypi:~/Drug_DNA_Solutions/openMP $ time -p ./dd_omp 2
max_ligand=2  nligands=120  nthreads=4
OMP defined
maximal score is 2, achieved by ligands
rr ac hh hc ar tn no or op to ta ap
real 0.02
user 0.03
sys 0.01
```

Omp 3



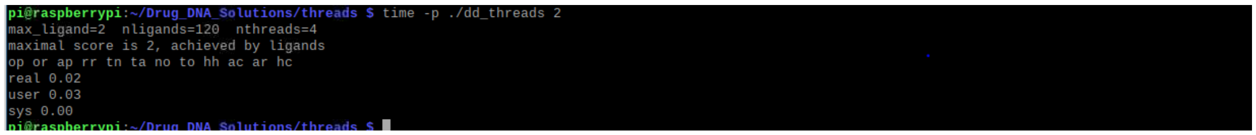
```
pi@raspberrypi:~/Drug_DNA_Solutions/openMP $ time -p ./dd_omp 3
max_ligand=3  nligands=120  nthreads=4
OMP defined
maximal score is 3, achieved by ligands
chw wht hop
real 0.06
user 0.13
sys 0.01
```

Omp 4

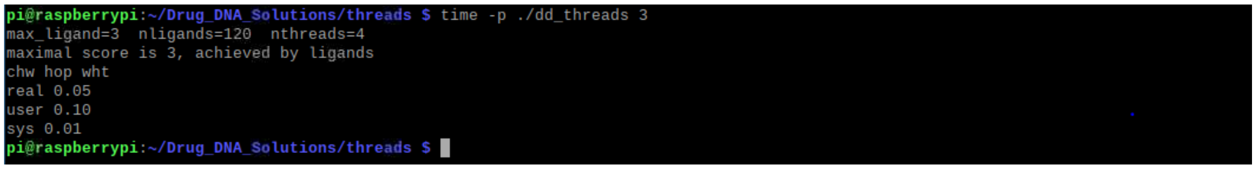


```
pi@raspberrypi:~/Drug_DNA_Solutions/openMP $ time -p ./dd_omp 4
max_ligand=4  nligands=120  nthreads=4
OMP defined
maximal score is 3, achieved by ligands
hkic wht cio hch ordt orel ihjo rdry
real 0.21
user 0.50
sys 0.01
```

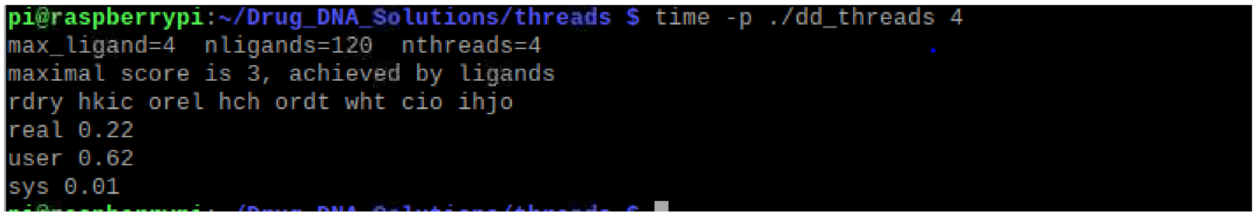
Threads 2,3,and 4



```
pi@raspberrypi:~/Drug_DNA_Solutions/threads $ time -p ./dd_threads 2
max_ligand=2  nligands=120  nthreads=4
maximal score is 2, achieved by ligands
op or ap rr tn ta no to hh ac ar hc
real 0.02
user 0.03
sys 0.00
pi@raspberrypi:~/Drug_DNA_Solutions/threads $
```



```
pi@raspberrypi:~/Drug_DNA_Solutions/threads $ time -p ./dd_threads 3
max_ligand=3  nligands=120  nthreads=4
maximal score is 3, achieved by ligands
chw hop wht
real 0.05
user 0.10
sys 0.01
pi@raspberrypi:~/Drug_DNA_Solutions/threads $
```



```
pi@raspberrypi:~/Drug_DNA_Solutions/threads $ time -p ./dd_threads 4
max_ligand=4  nligands=120  nthreads=4
maximal score is 3, achieved by ligands
rdry hkic orel hch ordt wht cio ihjo
real 0.22
user 0.62
sys 0.01
```

What approach is fastest?

- According to our results, the executions have similar results, but with the openMP, we are getting a little bit faster result than the reset.
- Perhaps, this is because of the structure of the openMP and its ability to divide executions between cores.

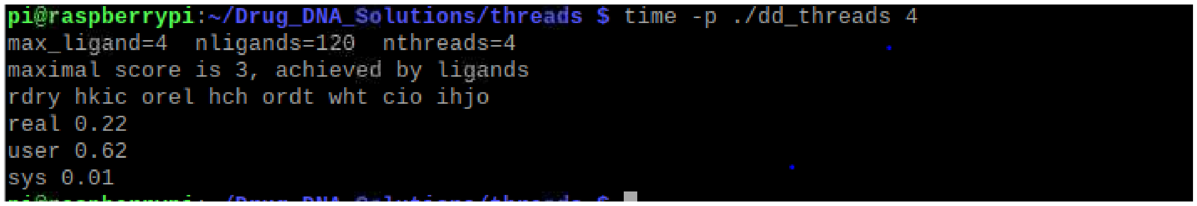
Determine the number of lines in each file. How does the C++ implementations compare to the openMP implementations?

- We can say that the C++ implementation is more detailed and simplified, but we have to understand the purpose of openMP is minimizing code while reducing run time complexity at the same time. But since the C++ implementation is more explanatory, in a way that other programmers can look into your program and suggest what you are writing.

Increasing threads to 5. What is the un time?

- Real – 0. 45
- User – 1.00
- Sys - 0. 3

Increase the maximum lingand and return each program



```
pi@raspberrypi:~/Drug_DNA_Solutions/threads $ time -p ./dd_threads 4
max_ligand=4  nligands=120  nthreads=4
maximal score is 3, achieved by ligands
rdry hkic orel hch ordt wht cio ihjo
real 0.22
user 0.62
sys 0.01
```

I don’t know if right, but run time is about the same as the rest.

Appendix

YouTube link: https://www.youtube.com/watch?v=qxYp5v_UlE0&feature=youtu.be
Slack link: https://app.slack.com/client/TNAGW4TJ9/CNCCR06S3