Joe Subbiani
2377990s

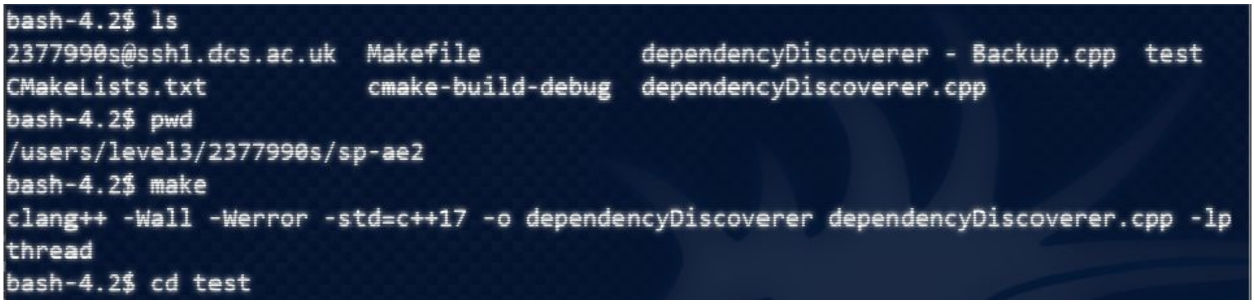
SP Assessed Exercise 2: Report

Status

My program compiles and will run with as many threads as specified by the environment variable `CRAWLER_TRHEADS`. If the environment variable is not set, the program will run with 2 threads. I have tested my program extensively and believe it is entirely thread safe and when run on the `test/` directory the output will always match the output of the provided `output` file.

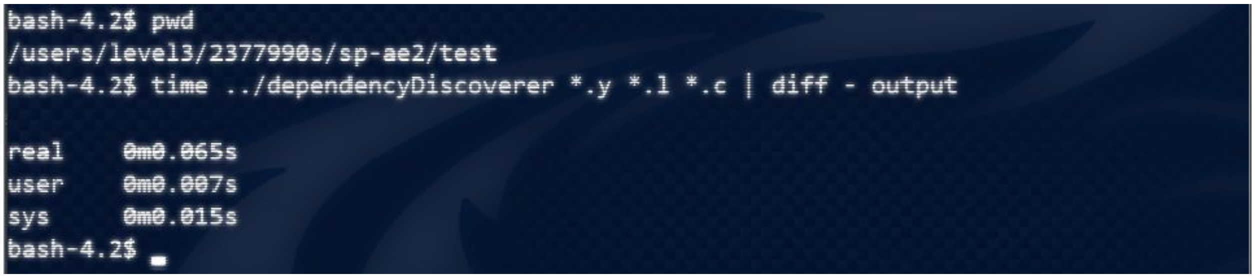
Build, Sequential and 1-Thread Runtimes.

The program was built and tested on my own machine as well as the Universities servers, specifically, stlinux03. Provided below are screenshots of my program being compiled (figure 1.), the sequential crawler being run and timed (figure 2.), and a screenshot of the program running on a single threaded crawler (figure 5.) is shown alongside the timed results of the program being run with multiple threads.



```
bash-4.2$ ls
2377990s@ssh1.dcs.ac.uk  Makefile          dependencyDiscoverer - Backup.cpp  test
CMakeLists.txt           cmake-build-debug  dependencyDiscoverer.cpp
bash-4.2$ pwd
/users/level3/2377990s/sp-ae2
bash-4.2$ make
clang++ -Wall -Werror -std=c++17 -o dependencyDiscoverer dependencyDiscoverer.cpp -lp
thread
bash-4.2$ cd test
```

Figure 1.



```
bash-4.2$ pwd
/users/level3/2377990s/sp-ae2/test
bash-4.2$ time ../dependencyDiscoverer *.y *.l *.c | diff - output

real    0m0.065s
user    0m0.007s
sys     0m0.015s
bash-4.2$ 
```

Figure 2.

Joe Subbiani
2377990s

Runtime with Multiple Threads.

CRAWLER_ THREADS	Elapsed Time - seconds					
	1	2	3	4	6	8
Execution 1	0.043	0.32	0.16	0.16	0.13	0.17
Execution 2	0.43	0.23	0.17	0.15	0.13	0.12
Execution 3	0.41	0.23	0.16	0.15	0.14	0.13
Median	0.42	0.275	0.165	0.155	0.135	0.125

Figure 3.

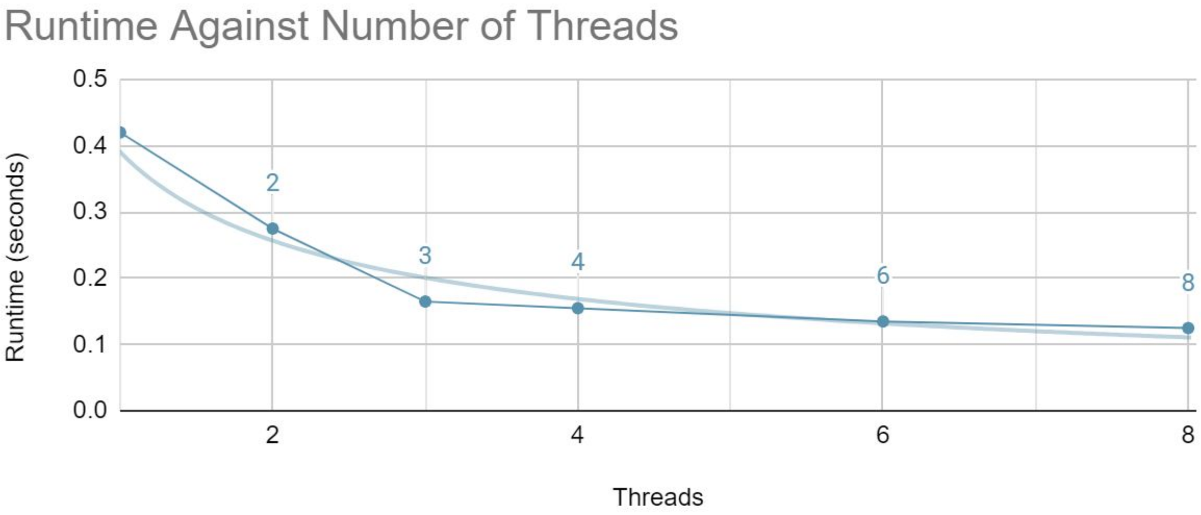
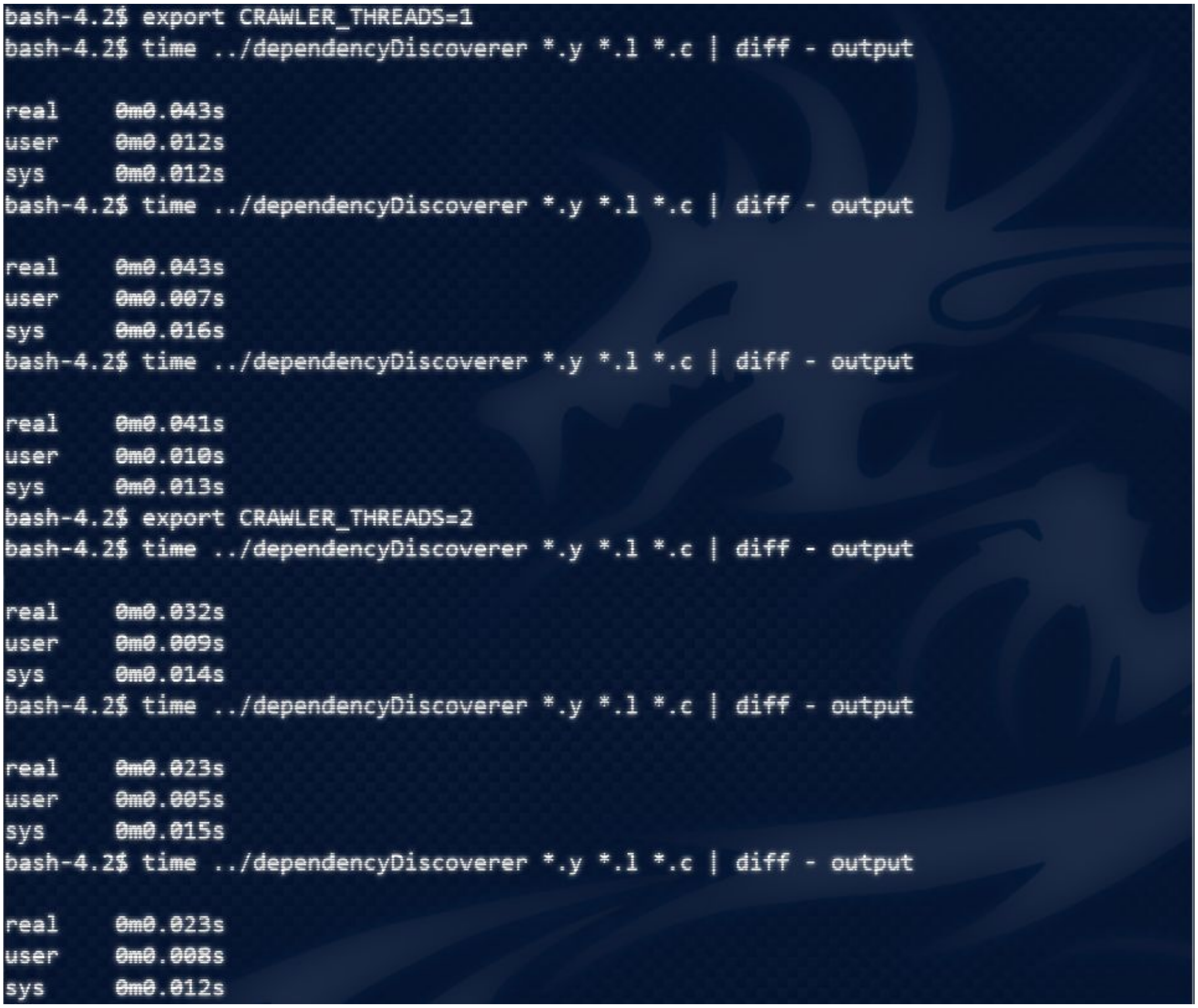


Figure 4.

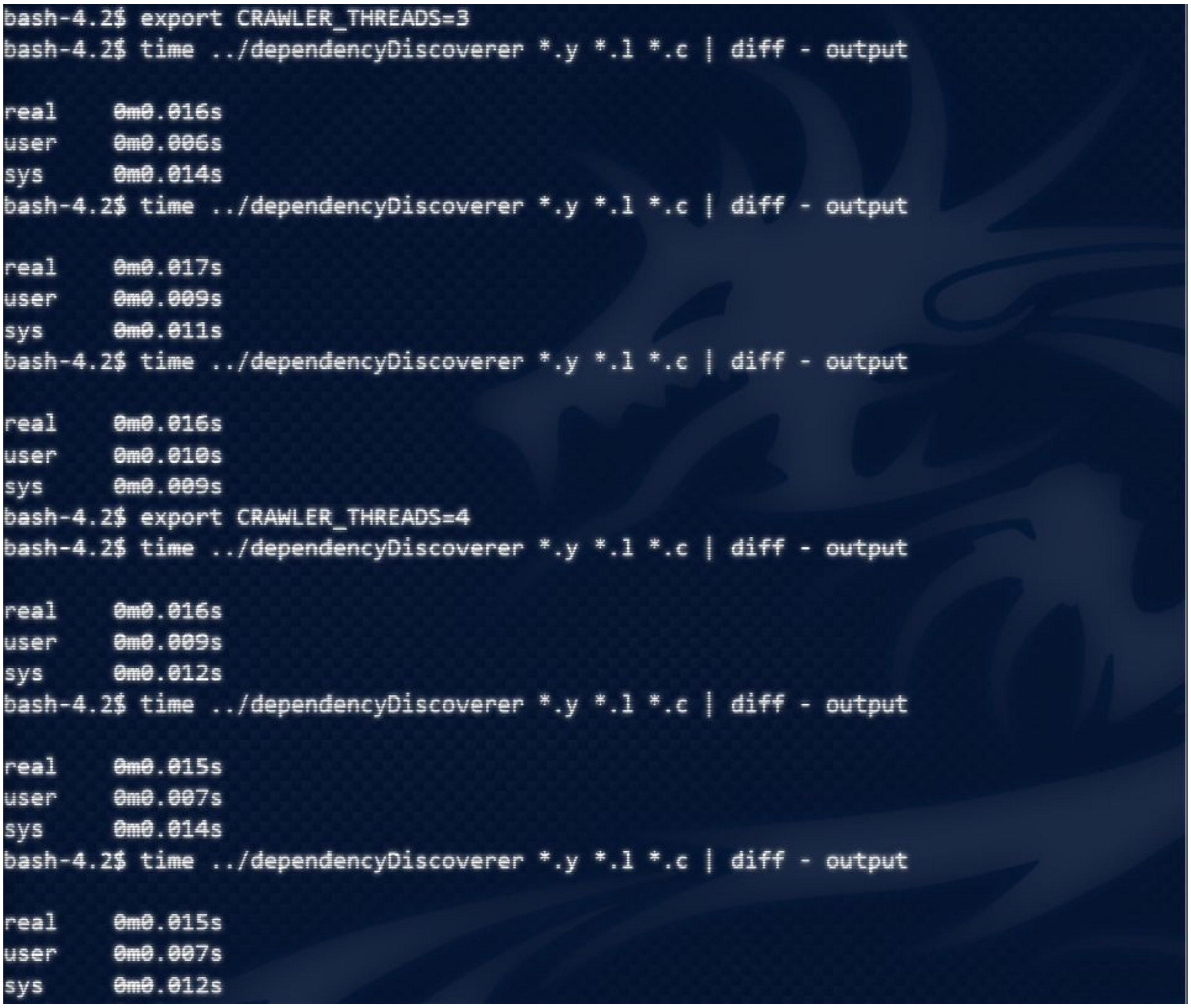
The results shown in figure 3. and 4. Clearly demonstrate that as you allow the program to utilize more threads, the run time of the program will reduce. Proof of these results are shown below in figures 5., 6. and 7.

Having additional cores process the input data results in more data being processed in the same amount of time when compared with a sequential crawler. Each time we increase the number of cores, the processing time decreases. There is some variability as seen in figure 3. Where there appears to be an anomalous result. I did not include this result in the median as it did not seem representative of the data. Whilst this anomaly is still faster than the sequential execution, it is clear that, as we increase the number of cores, the reduction in processing time becomes less significant. This could be due to the size of the dataset not being large enough for the processing time to outweigh the creation and joining of all the threads. This could also be due to the number of cores the server has available to utilise. If there are only 4 cores then having more threads will not be utilising more processing power. This is just speculation though as to why the reduction in processing time is less significant depending on the number of threads we use.



```
bash-4.2$ export CRAWLER_THREADS=1
bash-4.2$ time ../dependencyDiscoverer *.y *.l *.c | diff - output

real    0m0.043s
user    0m0.012s
sys     0m0.012s
bash-4.2$ time ../dependencyDiscoverer *.y *.l *.c | diff - output

real    0m0.043s
user    0m0.007s
sys     0m0.016s
bash-4.2$ time ../dependencyDiscoverer *.y *.l *.c | diff - output

real    0m0.041s
user    0m0.010s
sys     0m0.013s
bash-4.2$ export CRAWLER_THREADS=2
bash-4.2$ time ../dependencyDiscoverer *.y *.l *.c | diff - output

real    0m0.032s
user    0m0.009s
sys     0m0.014s
bash-4.2$ time ../dependencyDiscoverer *.y *.l *.c | diff - output

real    0m0.023s
user    0m0.005s
sys     0m0.015s
bash-4.2$ time ../dependencyDiscoverer *.y *.l *.c | diff - output

real    0m0.023s
user    0m0.008s
sys     0m0.012s
```

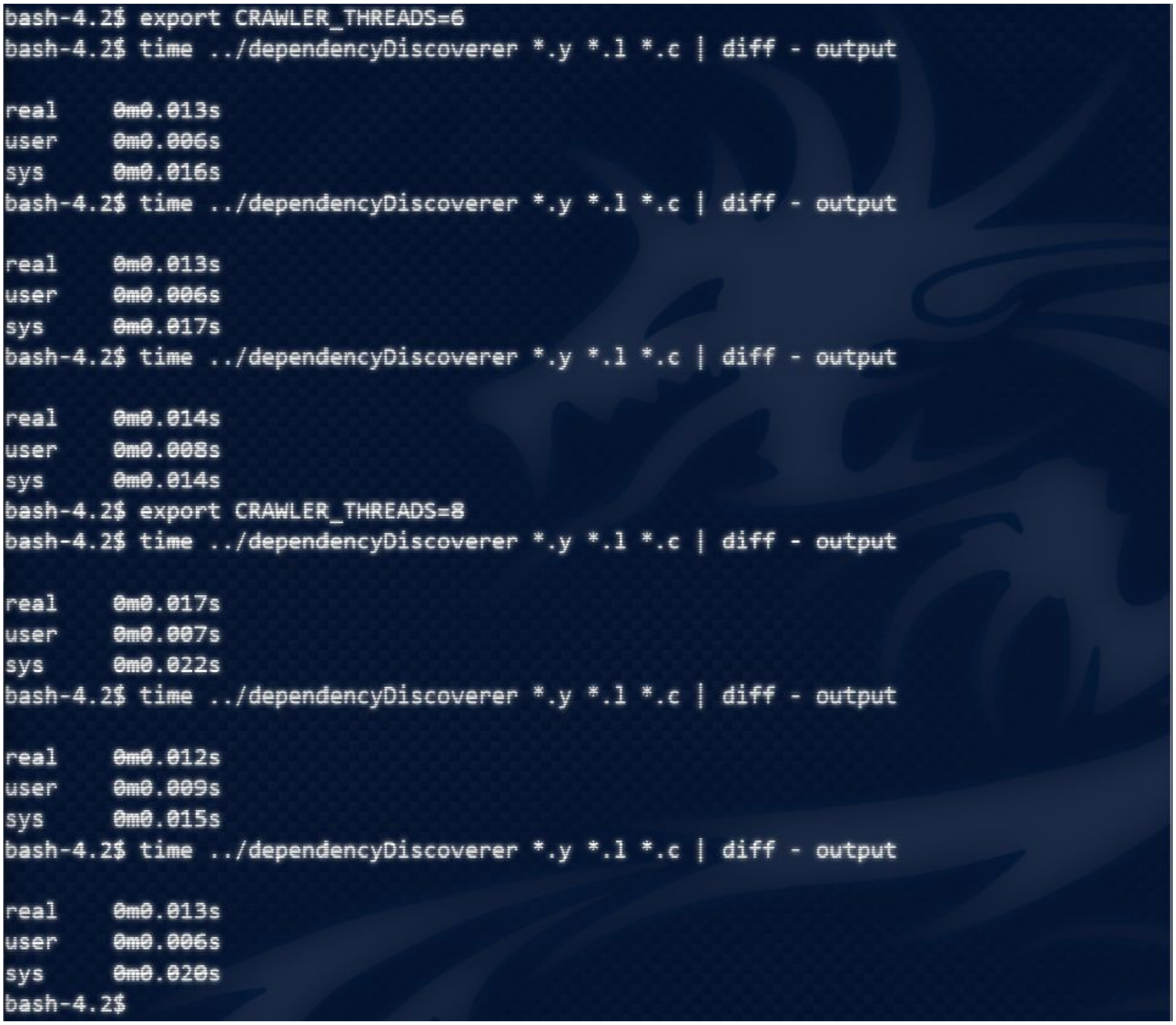
Figure 5.



```
bash-4.2$ export CRAWLER_THREADS=3
bash-4.2$ time ../dependencyDiscoverer *.y *.l *.c | diff - output

real    0m0.016s
user    0m0.006s
sys     0m0.014s
bash-4.2$ time ../dependencyDiscoverer *.y *.l *.c | diff - output

real    0m0.017s
user    0m0.009s
sys     0m0.011s
bash-4.2$ time ../dependencyDiscoverer *.y *.l *.c | diff - output

real    0m0.016s
user    0m0.010s
sys     0m0.009s
bash-4.2$ export CRAWLER_THREADS=4
bash-4.2$ time ../dependencyDiscoverer *.y *.l *.c | diff - output

real    0m0.016s
user    0m0.009s
sys     0m0.012s
bash-4.2$ time ../dependencyDiscoverer *.y *.l *.c | diff - output

real    0m0.015s
user    0m0.007s
sys     0m0.014s
bash-4.2$ time ../dependencyDiscoverer *.y *.l *.c | diff - output

real    0m0.015s
user    0m0.007s
sys     0m0.012s
```

Figure 6.

Joe Subbiani
2377990s



```
bash-4.2$ export CRAWLER_THREADS=6
bash-4.2$ time ../dependencyDiscoverer *.y *.l *.c | diff - output

real    0m0.013s
user    0m0.006s
sys     0m0.016s
bash-4.2$ time ../dependencyDiscoverer *.y *.l *.c | diff - output

real    0m0.013s
user    0m0.006s
sys     0m0.017s
bash-4.2$ time ../dependencyDiscoverer *.y *.l *.c | diff - output

real    0m0.014s
user    0m0.008s
sys     0m0.014s
bash-4.2$ export CRAWLER_THREADS=8
bash-4.2$ time ../dependencyDiscoverer *.y *.l *.c | diff - output

real    0m0.017s
user    0m0.007s
sys     0m0.022s
bash-4.2$ time ../dependencyDiscoverer *.y *.l *.c | diff - output

real    0m0.012s
user    0m0.009s
sys     0m0.015s
bash-4.2$ time ../dependencyDiscoverer *.y *.l *.c | diff - output

real    0m0.013s
user    0m0.006s
sys     0m0.020s
bash-4.2$
```

Figure 7.