

Report

1. Status

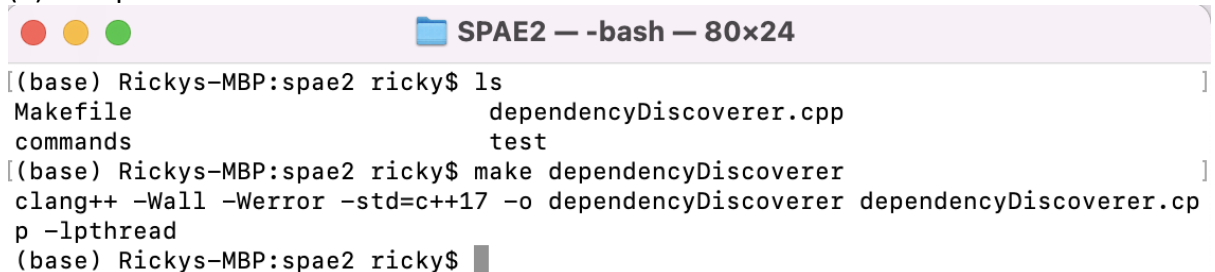
- My solutions works for all multithreaded solutions and can be adapted in the command line to support any number of threads below you can see I have tested 1,2,3,4,6,8 for the number of threads. I am not aware of any errors or problems with my implementation and ran the tests on the school Linux server.

2. Build

(a) Path:

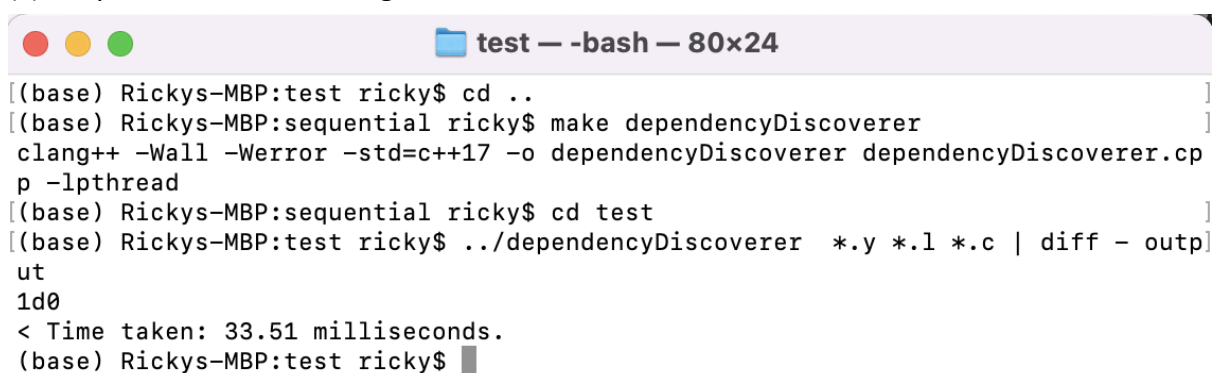
```
/users/level3/2465714a/c-files/spae2
[~bash-4.2$ cd c-files
[~bash-4.2$ cd spae2
[~bash-4.2$ pwd
/users/level3/2465714a/c-files/spae2
~bash-4.2$ █
```

(b) Compilation:



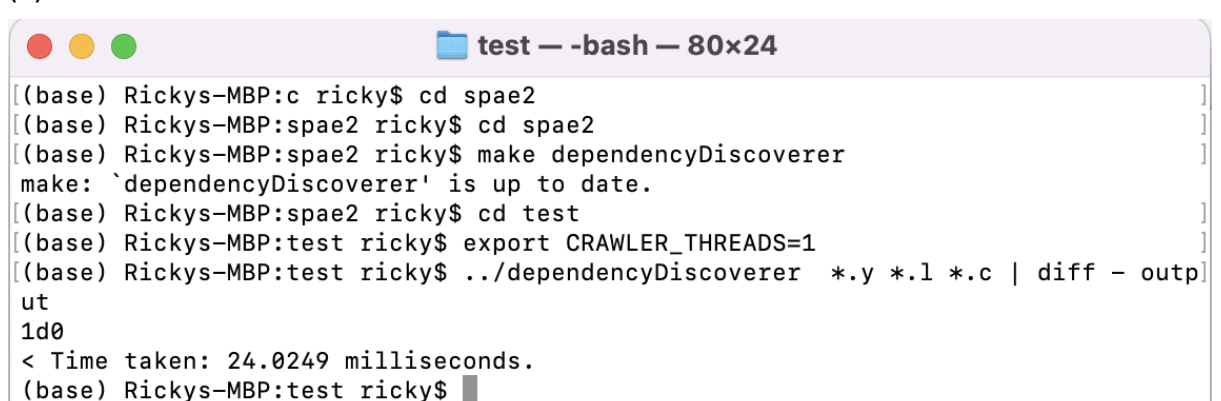
```
[(base) Rickys-MBP:spae2 ricky$ ls
Makefile                                dependencyDiscoverer.cpp
commands                               test
[(base) Rickys-MBP:spae2 ricky$ make dependencyDiscoverer
clang++ -Wall -Werror -std=c++17 -o dependencyDiscoverer dependencyDiscoverer.cp
p -lpthread
(base) Rickys-MBP:spae2 ricky$ █
```

(c) Sequential crawler testing:



```
[(base) Rickys-MBP:test ricky$ cd ..
[(base) Rickys-MBP:sequential ricky$ make dependencyDiscoverer
clang++ -Wall -Werror -std=c++17 -o dependencyDiscoverer dependencyDiscoverer.cp
p -lpthread
[(base) Rickys-MBP:sequential ricky$ cd test
[(base) Rickys-MBP:test ricky$ ../dependencyDiscoverer *.y *.l *.c | diff - outp
ut
1d0
< Time taken: 33.51 milliseconds.
(base) Rickys-MBP:test ricky$ █
```

(e) Runtime with one thread:



```
[(base) Rickys-MBP:c ricky$ cd spae2
[(base) Rickys-MBP:spae2 ricky$ cd spae2
[(base) Rickys-MBP:spae2 ricky$ make dependencyDiscoverer
make: `dependencyDiscoverer' is up to date.
[(base) Rickys-MBP:spae2 ricky$ cd test
[(base) Rickys-MBP:test ricky$ export CRAWLER_THREADS=1
[(base) Rickys-MBP:test ricky$ ../dependencyDiscoverer *.y *.l *.c | diff - outp
ut
1d0
< Time taken: 24.0249 milliseconds.
(base) Rickys-MBP:test ricky$ █
```

3.

(a) Run times for 1,2,3,4,6,8 threads:

School server:

```
[~bash-4.2$ ls ]
c-files perl5 ]
[~bash-4.2$ cd c-files ]
[~bash-4.2$ cd spae2 ]
[~bash-4.2$ ls ]
commands dependencyDiscoverer dependencyDiscoverer.cpp Makefile test ]
[~bash-4.2$ cd test ]
[~bash-4.2$ ssh stlinux03 ]
2465714a@stlinux03's password: ]
Last login: Thu Nov 25 15:56:56 2021 from ssh1 ]
School of Computing Science 2019 centos7 stlinux v3 ]
[~bash-4.2$ cd c-files ]
[~bash-4.2$ cd spae2 ]
[~bash-4.2$ ls ]
commands dependencyDiscoverer dependencyDiscoverer.cpp Makefile test ]
[~bash-4.2$ cd test ]
[~bash-4.2$ export CRAWLER_THREADS=1 ]
[~bash-4.2$ ../dependencyDiscoverer *.y *.l *.c | diff - output ]
1d0 ]
< Time taken: 28.361 milliseconds. ]
[~bash-4.2$ export CRAWLER_THREADS=2 ]
[~bash-4.2$ ../dependencyDiscoverer *.y *.l *.c | diff - output ]
1d0 ]
< Time taken: 19.5303 milliseconds. ]
[~bash-4.2$ export CRAWLER_THREADS=3 ]
[~bash-4.2$ ../dependencyDiscoverer *.y *.l *.c | diff - output ]
1d0 ]
< Time taken: 19.0042 milliseconds. ]
[~bash-4.2$ export CRAWLER_THREADS=4 ]
[~bash-4.2$ ../dependencyDiscoverer *.y *.l *.c | diff - output ]
1d0 ]
< Time taken: 19.9924 milliseconds. ]
[~bash-4.2$ export CRAWLER_THREADS=5 ]
[~bash-4.2$ ../dependencyDiscoverer *.y *.l *.c | diff - output ]
1d0 ]
< Time taken: 17.9848 milliseconds. ]
[~bash-4.2$ export CRAWLER_THREADS=6 ]
[~bash-4.2$ ../dependencyDiscoverer *.y *.l *.c | diff - output ]
1d0 ]
< Time taken: 17.9298 milliseconds. ]
[~bash-4.2$ export CRAWLER_THREADS=8 ]
[~bash-4.2$ ../dependencyDiscoverer *.y *.l *.c | diff - output ]
1d0 ]
< Time taken: 17.7463 milliseconds. ]
[~bash-4.2$ pwd ]
/users/level3/2465714a/c-files/spae2/test ]
[~bash-4.2$ █ ]
```

Local system:

```
test — -bash — 89x41
[(base) Rickys-MBP:c ricky$ cd spae2 ]
[(base) Rickys-MBP:spae2 ricky$ cd spae2 ]
[(base) Rickys-MBP:spae2 ricky$ make dependencyDiscoverer ]
make: `dependencyDiscoverer' is up to date.
[(base) Rickys-MBP:spae2 ricky$ cd test ]
[(base) Rickys-MBP:test ricky$ export CRAWLER_THREADS=1 ]
[(base) Rickys-MBP:test ricky$ ../dependencyDiscoverer *.y *.l *.c | diff - output ]
1d0
< Time taken: 24.0249 milliseconds.
[(base) Rickys-MBP:test ricky$ export CRAWLER_THREADS=2 ]
[(base) Rickys-MBP:test ricky$ ../dependencyDiscoverer *.y *.l *.c | diff - output ]
1d0
< Time taken: 19.2642 milliseconds.
[(base) Rickys-MBP:test ricky$ export CRAWLER_THREADS=3 ]
[(base) Rickys-MBP:test ricky$ ../dependencyDiscoverer *.y *.l *.c | diff - output ]
1d0
< Time taken: 4.41562 milliseconds.
[(base) Rickys-MBP:test ricky$ export CRAWLER_THREADS=4 ]
[(base) Rickys-MBP:test ricky$ ../dependencyDiscoverer *.y *.l *.c | diff - output ]
1d0
< Time taken: 3.50084 milliseconds.
[(base) Rickys-MBP:test ricky$ export CRAWLER_THREADS=6 ]
[(base) Rickys-MBP:test ricky$ ../dependencyDiscoverer *.y *.l *.c | diff - output ]
1d0
< Time taken: 3.75733 milliseconds.
[(base) Rickys-MBP:test ricky$ export CRAWLER_THREADS=8 ]
[(base) Rickys-MBP:test ricky$ ../dependencyDiscoverer *.y *.l *.c | diff - output ]
1d0
< Time taken: 1.92179 milliseconds.
[(base) Rickys-MBP:test ricky$ ]
```

(b) Experiment

CRAWLER_	1	2	3	4	6	8
THREADS	Elapsed Time	Elapsed Time	Elapsed Time	Elapsed Time	Elapsed Time	Elapsed Time
Execution 1	24.0	19.3	4.4	3.5	3.8	1.9
Execution 2	24.2	17.2	11.4	2.3	2.3	1.8
Execution 3	22.0	18.0	8.8	3.6	3.5	2.0
Median	24.0	17.2	8.8	3.5	3.5	1.9

[Units = milliseconds]

(c) Discussion

When the number of threads is increased the elapsed computation time decreases, this benefit decreases as you get to a higher number of threads. This is likely due to the processor only having a certain number of cores normally 4 cores - thus 8 with hyper threading so after 4 and 8 threads the same benefits are not observed. So overall using threads to concurrently works through queues is a lot faster and effective method than a traditional sequential implementation.

