

Concurrent Dependency Discoverer

Status:

The Program I submitted is a multi-threaded solution and runs without errors produces the expected output from the test file provided.

Build, and sequential & 1-thread runtimes:

Sequential:

```
Documents/Comp Sci 1/Systems_Programming/Cw2 - Copy$ make dependencyDiscoverer
clang++ -Wall -Werror -std=c++17 -o dependencyDiscoverer dependencyDiscoverer.cpp -lpthread
Documents/Comp Sci 1/Systems_Programming/Cw2 - Copy$ time ./dependencyDiscoverer -Itest test/*.c test/*.l
test/*.y > temp
real    0m1.126s
user    0m0.032s
sys     0m0.091s
```

1-thread:

```
Documents/Comp Sci 1/Systems_Programming/Cw2(1_thread)$ make dependencyDiscoverer
clang++ -Wall -Werror -std=c++17 -o dependencyDiscoverer dependencyDiscoverer.cpp -lpthread
Documents/Comp Sci 1/Systems_Programming/Cw2(1_thread)$ time ./dependencyDiscoverer -Itest test/*.c test/*.l
*.l test/*.y > temp
real    0m0.579s
user    0m0.024s
sys     0m0.098s
```

Runtime with Multiple Threads:

Screenshot:

```
Documents/Comp Sci 1/Systems_Programming/Cw2$ make dependencyDiscoverer
clang++ -Wall -Werror -std=c++17 -o dependencyDiscoverer dependencyDiscoverer.cpp -lpthread
Documents/Comp Sci 1/Systems_Programming/Cw2$ export CRAWLER_THREADS=1
Documents/Comp Sci 1/Systems_Programming/Cw2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*
.y > temp
real    0m0.574s
user    0m0.002s
sys     0m0.118s
Documents/Comp Sci 1/Systems_Programming/Cw2$ export CRAWLER_THREADS=2
Documents/Comp Sci 1/Systems_Programming/Cw2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*
.y > temp
real    0m0.242s
user    0m0.029s
sys     0m0.085s
Documents/Comp Sci 1/Systems_Programming/Cw2$ export CRAWLER_THREADS=3
Documents/Comp Sci 1/Systems_Programming/Cw2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*
.y > temp
real    0m0.231s
user    0m0.002s
sys     0m0.150s
Documents/Comp Sci 1/Systems_Programming/Cw2$ export CRAWLER_THREADS=4
Documents/Comp Sci 1/Systems_Programming/Cw2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*
.y > temp
real    0m0.206s
user    0m0.053s
sys     0m0.109s
Documents/Comp Sci 1/Systems_Programming/Cw2$ export CRAWLER_THREADS=6
Documents/Comp Sci 1/Systems_Programming/Cw2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*
.y > temp
real    0m0.163s
user    0m0.021s
sys     0m0.153s
Documents/Comp Sci 1/Systems_Programming/Cw2$ export CRAWLER_THREADS=8
Documents/Comp Sci 1/Systems_Programming/Cw2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*
.y > temp
real    0m0.156s
user    0m0.040s
sys     0m0.137s
```

Experiment:

CRAWLER_ THREADS	1	2	3	4	6	8
	Elapsed Time	Elapsed Time	Elapsed Time	Elapsed Time	Elapsed Time	Elapsed Time
Execution 1	0.080s	0.040s	0.024s	0.028s	0.019s	0.020
Execution 2	0.053s	0.030s	0.023s	0.021s	0.020s	0.019
Execution 3	0.050s	0.028s	0.023s	0.021s	0.022s	0.019s
Median	0.061s	0.033s	0.023s	0.023s	0.020s	0.019s

From running my program with different thread counts, I conclude that increasing the thread count can significantly reduce the runtime of my code. This reduction in run time reduces the more threads I add (for example the difference between 1 and 2 threads is much greater than the difference between 7 and 8 threads).

I can see from my experiments on the school sever that there can be lots of variation on the Elapsed times. Only by taking the median of multiple results can we see the clear benefits of making the program concurrent.