

Shouri Addepally
ECE 220 Honors
05/03/2020

MP 3 Honors: Report

Questions:

1. When I used 'THRESHOLD' = 1024, I ended up getting a segmentation fault. A possible explanation as to why this happened is because pthread_create fails to create any new threads due to a machine limitation. Once pthread_join is called, there is no pthread_t for the pthread_join to operate on, therefore resulting in a segmentation fault. From repeated trials, I observed that my machine limitation was approximately when 'THRESHOLD' = 2600.
2. When I ran ./merge_sort_parallel with a 'THRESHOLD'=1, I got a segmentation fault. While running the program with valgrind, the size of the pthread_t structure was of size 8 and 97 pthread_t threads were allocated with 'THRESHOLD'=1. From this information, we can conclude that a segmentation fault occurred because there were too many threads that were being generated and not enough memory allocated for the threads to operate on successfully.

```
Command: ./merge_sort_parallel
Thread 97:
Use of uninitialised value of size 8
at 0x400ABB: split_ll (merge_sort.c:38)
by 0x4008DA: merge_sort_thread_wrapper (merge_sort_parallel.c:10)
by 0x4E3EEA4: start_thread (in /usr/lib64/libpthread-2.17.so)
by 0x51518DC: clone (in /usr/lib64/libc-2.17.so)

Invalid write of size 8
at 0x400ABB: split_ll (merge_sort.c:38)
by 0x4008DA: merge_sort_thread_wrapper (merge_sort_parallel.c:10)
by 0x4E3EEA4: start_thread (in /usr/lib64/libpthread-2.17.so)
by 0x51518DC: clone (in /usr/lib64/libc-2.17.so)
Address 0x8 is not stack'd, malloc'd or (recently) free'd

Process terminating with default action of signal 11 (SIGSEGV)
Access not within mapped region at address 0x8
at 0x400ABB: split_ll (merge_sort.c:38)
by 0x4008DA: merge_sort_thread_wrapper (merge_sort_parallel.c:10)
by 0x4E3EEA4: start_thread (in /usr/lib64/libpthread-2.17.so)
by 0x51518DC: clone (in /usr/lib64/libc-2.17.so)
```

3. The threshold that generated the benchmark with a reasonable speedup ($\geq 1.4X$) was approximately in the range of 'THRESHOLD'=125000.

```
[shouria2@linux-a2 mp3h]$ python benchmark.py
Benchmark success!
Time taken for serial:      1.17274188995
Time taken for parallel:    0.815626859665
Speedup:                    1.44 X
[shouria2@linux-a2 mp3h]$
```

4. As the number of threads began to increase, I began to see diminishing return at around a threshold value of 'THRESHOLD' = 325000. A reason as to why I began to see diminishing returns is because the pthread_create and pthread_join operations are both 'costly' and take up system memory and time to spawn and stop from spawning. Because the generation of threads plays a role in the system memory, at a higher threshold the cost of generating new threads began to play a role in slowing down the speedup rate.