COL 380 - Assignment 1

cs1190405

January 2023

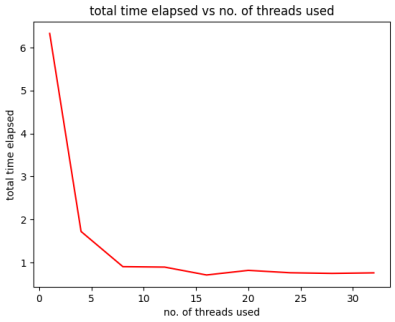
1 Question 1

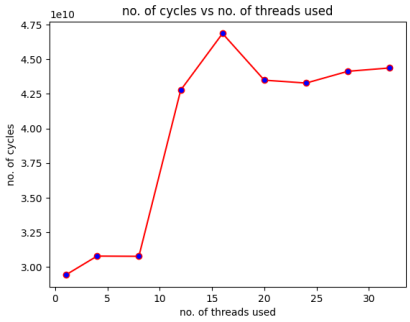
For this assignment, I connected to the css2 server and used perf for analysis.

2 Question 2

1. ***perf stat*** - The perf stat command instruments and summarizes key CPU counters (PMCs).
   The *perf stat* command was run for *./classiffy rfile dfile 1009072 <no. of threads used >*. The no. of repititions was not specified and the default value of 6 was used. The following observations was observed:

No. of Threads used	total time elapsed	no. of cycles
1	6.302815603	29,47,63,99,326
4	1.685299279	30,27,23,58,594
8	0.947380728	31,65,52,09,295
12	0.900934003	42,98,15,42,122
16	0.701129524	46,64,64,44,306
20	0.815482245	43,11,37,69,491
24	0.751055407	43,19,75,75,453
28	0.740205913	43,85,35,84,892
32	0.747876676	44,29,33,37,255





(a) Graph of time elapsed vs threads used (b) Graph of no. of cycles vs threads used

2. ***perf record***

   (a) ***perf record*** - The *perf record* command was run to create the *perf.data* file which was used for analysis.

   (b) ***perf report*** - The *perf report* command was used to inspect the file generated inn the previous step and then the file was annotated for hotspot analysis.
   These commands were found in the during the hotspot analysis of classify event.:

      i. $37.68|\downarrow jg93$

      ii. $9.37|cmp(\%r11, \%rax, 8), \%edx$

      iii. 14.81 - $\uparrow jge40$
      $18.50 - 93 : lea0x1(\%rax), \%r13$

   iv. 14.56 - $\uparrow jmp8a$

   (c) The instructions that takes the maximum time is:
   jg 93 **(37.68% of time)**

   (d) It maps to the the bool within functions defined in the classify.h
   bool within(int val) const // Return if val is within this range return(lo *leq* val val *leq* hi);

   (e) The make file was changed slightly to allow the perf report to show the source code along with the assembly code. This was done using the ***- g*** flag during compiling.

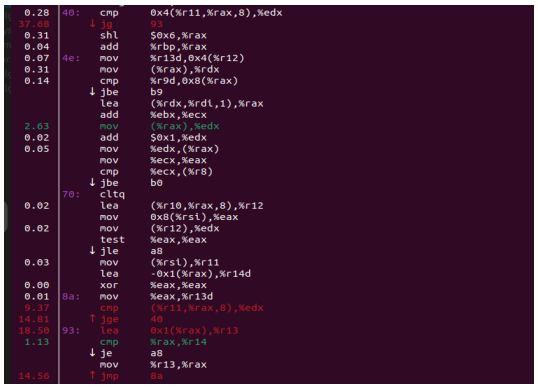


Figure 2: Hotspots found

3. **Hotspot Analysis**

   (a) Perf report was run with the *-g* flag in the makefile to show the source code with the assembly code and the output was saved as perf_2_1.data

   (b)

Figure 3: 3.2 - Top Hotspot

(c) The above assembly code refers to the boolean *within* function in the *classify.h* file.

(d) perf record was run to record branch instructions, branch misses, cache misses, page faults, cpu cycles. The following command was used to record the events and for analysis: *perf record -e branch-instructions,branch-misses,cache-misses,page-faults,cpu-cycles make run*

4. **Memory Profiling**

(a) the perf mem record was used to record the memory resources used by different functions and the data was stored in perf_3.data

(b)