Project 2 - DB

Teammates

Minghan Zhu (mz2716)
Zeyu Liu (zl3087)
Tushar Arora (ta2673)

1. The machine characteristics for the for the google cloud are -

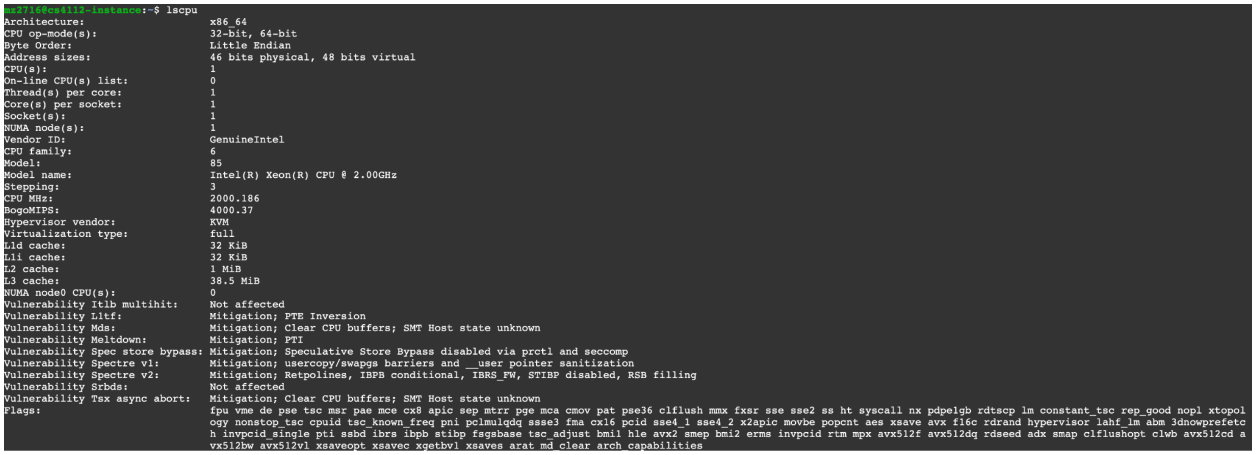

Figure 1: `lscpu` command result

   a. GCP VM type: n1-standard-n
   b. CPU(s): 1 Physical CPU Core
   c. Memory (GB): 3.75
   d. L1d cache: 32 KiB
   e. L1i cache: 32 KiB
   f. L2 cache: 1 MiB

Chart of the five functions

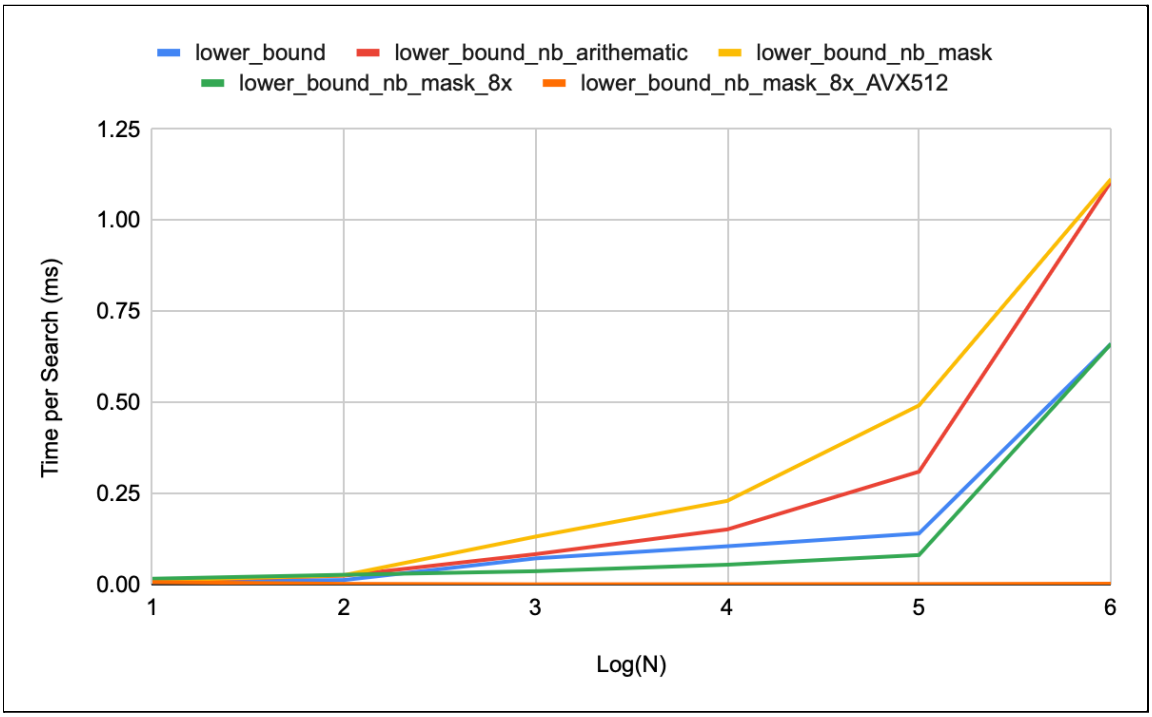

Figure 2: Measured performance of the five routines as N varied from 10 to 10^6

Executing command - ./db4112 N 1 1 1 1000
R = 1000
where the value of "N" was varied from 10 to 1000000 (10^6)

From the graph, it is very evident that the method, lower_bound_nb_mask_8x_AVX512 performs exceptionally well when the value of N starts increasing. The performance metrics of all the methods is almost the same for small numbers, between 10 and 100 but after N=100 the performance metrics of the lower_bound_nb_mask decreases exponentially. This method takes the most time to complete the scanning and this was also very evident while we were running this method for analyzing the performance. In some of the cases, it took over 30 mins to finish the scan.

But a general analysis of the graph signifies that the 8x methods have a comparatively better performance than all the other methods. The lower_bound_nb_mask_8x method outperforms all

the other methods (except AVX512) for all values above N=100. One of the highlighted behaviors in the above graph is that after N=100000, all the methods (except AVX512) get an exponential growth in the time taken to run the scan which is visible from the slope of the individual line graphs. This could possibly be because now the number of values is the dominant number in the algorithm and this becomes a huge factor when going to data dependency. Clearly at this point the advantage of moving from control dependency to data dependency does not provide the optimized results anymore.