Text Histogram

Questions

1. Using shared memory on the intermediate privatized historgrams reduces the latency of accessing locations especially when it is heavily contended i.e. when multiple threads have the same ASCII character. This will improve the throughput of the atomic operations with faster memory access.

2. Nsys Compute Profiler: Although the profiler shows that we have an achieved occupancy of 88.72%,

```
[junseo@r40a-09.sif TextHistogram]$ ./text_histogram.sh
==PROF== Connected to process 924024 (/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module7/TextHistogr
am/bin/release/TextHistogram_Solution)
[TIME][Generic][Importing data and creating memory on host][/home/warehouse/junseo/cuda-code-repo-joonsu
uh/Module7/TextHistogram/solution.cu: 54-58] Elapsed time: 839.119 ms
The input length is 1000000000
The number of bins is 128
[TIME][GPU][Allocating GPU memory.][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module7/TextHistogram
/solution.cu: 63-69] Elapsed time: 3.65363 ms
[TIME][GPU][Copying input memory to the GPU.][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module7/Tex
tHistogram/solution.cu: 71-78] Elapsed time: 124.709 ms
Launching kernel
==PROF== Profiling "histogram_private_kernel" - 0: 0%....50%....100% - 8 passes
[TIME][Compute][Performing CUDA computation][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module7/Text
Histogram/solution.cu: 83-92] Elapsed time: 1297.85 ms
[TIME][Copy][Copying output memory to the CPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module7/Te
xtHistogram/solution.cu: 94-100] Elapsed time: 0.063746 ms
[TIME][GPU][Freeing GPU Memory][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module7/TextHistogram/sol
ution.cu: 102-107] Elapsed time: 2.7643 ms
The solution is correct
==PROF== Disconnected from process 924024
==PROF== Report: /home/warehouse/junseo/cuda-code-repo-joonsuuh/Module7/TextHistogram/profile.ncu-rep
```

Figure 1: Nsys Compute Profiler

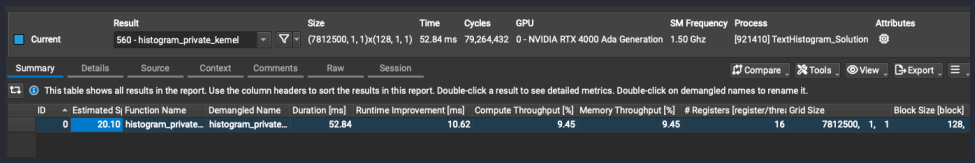

Figure 2: Ncu Profiler

   the GPU SOL throughput shows us the peak sustained rate was less than 10%. This is probably due to the large overhead of privatization where the number of launched thread blocks is smaller than the number that can be simultaneously launched by the GPU leading to serialization. This could be improved with thread coarsening or having a thread go through multiple input characters.

3. In the histogram adding step `atomicAdd(&bins_s[input[tid]], 1u);` has to read from global memory once per thread, so for an input length N we have N global memory reads.

4. In the for loop we are incrementing by `blockDim.x` up to the number of bins, so there are `numBins`$*$ number of blocks global memory writes.

5. In shared memory, we have N atomic operations since it happens along the histogram adding step in the privatized part. In global memory which is the final lines of code on the kernel function, we have `numBins` $\times$ `numBlocks` atomic operations.

6. For a histogram without privatization or shared memory, there would be N atomic operations performed in global memory. This ratio of the speed up compared to the privatized version is

$$N : \texttt{numBins} \times \texttt{numBlocks}$$
$$N : \texttt{numBins} \times \frac{N}{\texttt{threads per block}}$$
$$1 : \frac{\texttt{numBins}}{\texttt{threads per block}}$$

   So for 128 bins and 1024 threads per block, the speed up is $1 : 128/1024 = 8 : 1$.

OUPUT

```
[junseo@r40a-09.sif TextHistogram]$ ./text_histogram.sh
[TIME][Generic][Importing data and creating memory on host][/home/warehouse/junseo/cu
da-code-repo-joonsuuh/Module7/TextHistogram/solution.cu: 54-58] Elapsed time: 853.211
 ms
The input length is 1000000000
The number of bins is 128
[TIME][GPU][Allocating GPU memory.][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Mo
dule7/TextHistogram/solution.cu: 63-69] Elapsed time: 3.34549 ms
[TIME][GPU][Copying input memory to the GPU.][/home/warehouse/junseo/cuda-code-repo-j
oonsuuh/Module7/TextHistogram/solution.cu: 71-78] Elapsed time: 126.88 ms
Launching kernel
[TIME][Compute][Performing CUDA computation][/home/warehouse/junseo/cuda-code-repo-jo
onsuuh/Module7/TextHistogram/solution.cu: 83-92] Elapsed time: 41.0969 ms
[TIME][Copy][Copying output memory to the CPU][/home/warehouse/junseo/cuda-code-repo-
joonsuuh/Module7/TextHistogram/solution.cu: 94-100] Elapsed time: 0.037892 ms
[TIME][GPU][Freeing GPU Memory][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module
7/TextHistogram/solution.cu: 102-107] Elapsed time: 2.26625 ms
The solution is correct
```

Figure 3: Dataset 14 success ouput