**GENERATE HISTOGRAM USING PTHREADS**

**Harshvardhan Agrawal (hka25)**

**Chirag Jain (csj43)**

(Makefile is given to run the code).

Steps to Parallelize the implementation of histogram generation:

* In order to parallelise the code, we need to break the data into chunks on which a respective thread will perform computation.
* Create a structure that holds the thread\_id, start index for the chunk of data of respective thread and the end index for chunk of data of the respective thread.
* We also declare the input data array globally and a 2D array for storing local histograms of each thread.
* We will sum the local histograms of each thread and store it in the local histogram of thread 0.
* When we deploy the threads we compute the local histograms of within each thread.
* After computing the local histograms we form a tree like formation of the threads to compute the final histogram.
* For e.g. In the case of 8 threads, we will have 3 levels (i.e. log2(8)). In the first level 1->0, 3->2, 5->4,....,7->6. In the second level 2->0, 6->4 and in the last level 4->0.
* This way we have the final histogram stored in same location as local histogram of thread 0.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number of Threads** | **Input Size** | **Serial Time** | **Parallel Time** | **Speedup** |
| 2 | 1,000,000 | 0.0095 | 0.0095 | 1 |
|  | 10,000,000 | 0.0941 | 0.0596 | 1.58 |
|  | 100,000,000 | 0.9424 | 0.591 | 1.6 |
| 4 | 1,000,000 | 0.0094 | 0.0086 | 1.1 |
|  | 10,000,000 | 0.0941 | 0.0368 | 2.56 |
|  | 100,000,000 | 0.9407 | 0.2631 | 3.58 |
| 8 | 1,000,000 | 0.0094 | 0.0074 | 1.28 |
|  | 10,000,000 | 0.0941 | 0.0258 | 3.65 |
|  | 100,000,000 | 0.9403 | 0.1836 | 5.13 |
| 16 | 1,000,000 | 0.0094 | 0.0063 | 1.5 |
|  | 10,000,000 | 0.0941 | 0.0283 | 3.33 |
|  | 100,000,000 | 1.249 | 0.1002 | 12.47 |