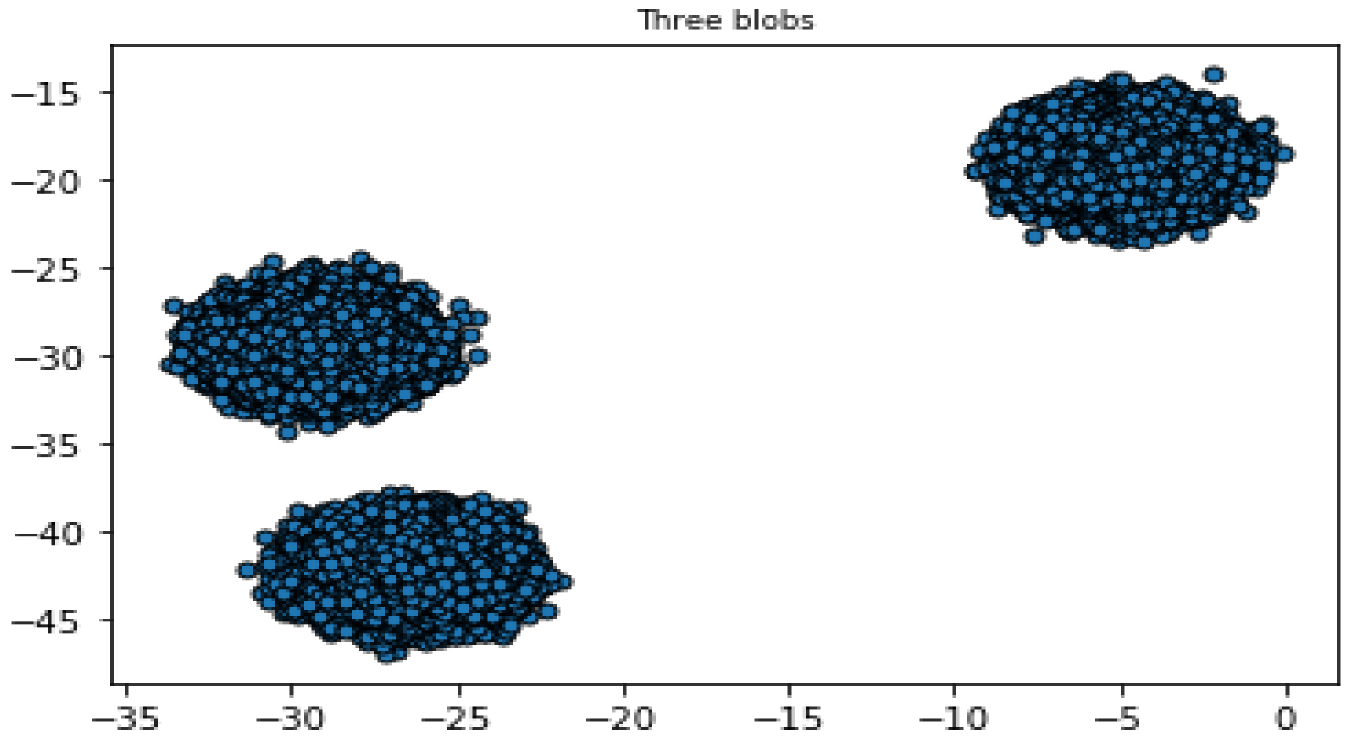
Dhirubhai Ambani
Institute of Information and Communication Technology

# K-means Clustering

| Sr.No. | Name | Roll No |
|---|---|---|
| 1 | Maulik Soni | 201911009 |
| 2 | Shirpad Bhat | 201911003 |

## ❖ Input/Output:



## ❖ **Algorithm Analysis:**

- If N is the number of data points to be grouped into K clusters, and each data point is of D dimensions and takes l iterations, then the time complexity is O(N*K*D*I).
- In serial code for larger problem size single thread takes more time to make the clusters whereas in parallel code multiple threads can able to do same thing in quite less time.

## ❖ **Parallelization Strategy:**

- Kernel-1: It takes  data array as input and generates new clusters based on given centroids(Initially randomly generated) by computing distance(uses Euclidian metric) between data points and centroids.
- Kernel-2: It takes the clusters and calculates the sum for each cluster. Eventually it helps in finding new mean of each centroid.

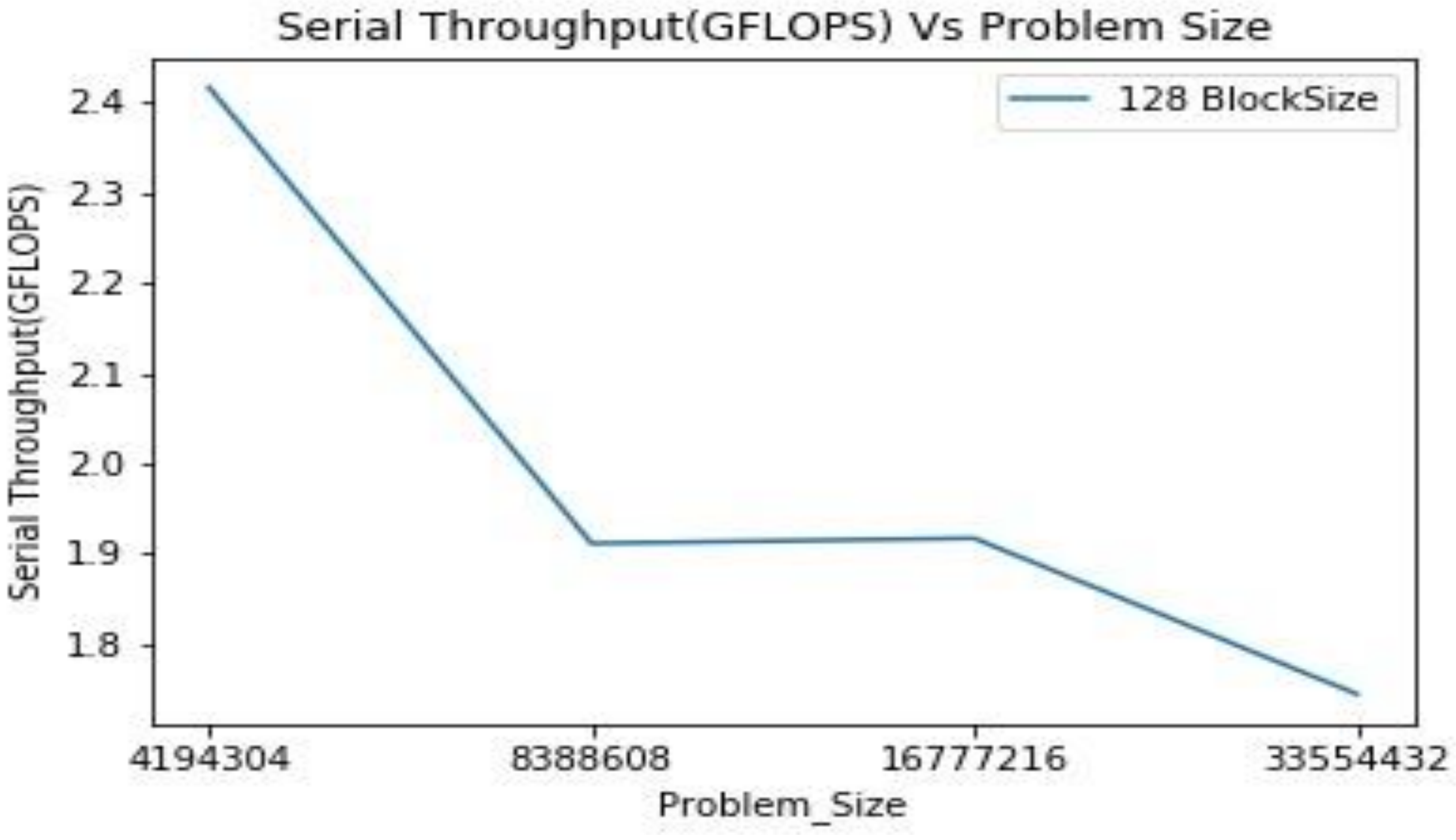
## ❖ **Optimization Strategy:**

- Privatisation is used for atomic add(Histogram) operation to reduce memory access time to update particular location.
- Reduction algorithm is used for summing individual clusters and in place reduction to avoid memory transfer between Host and Gpu.
- Sequentially address gives coalesced data access and it is bank conflict free.
- Used ternary operator to reduce control divergence.

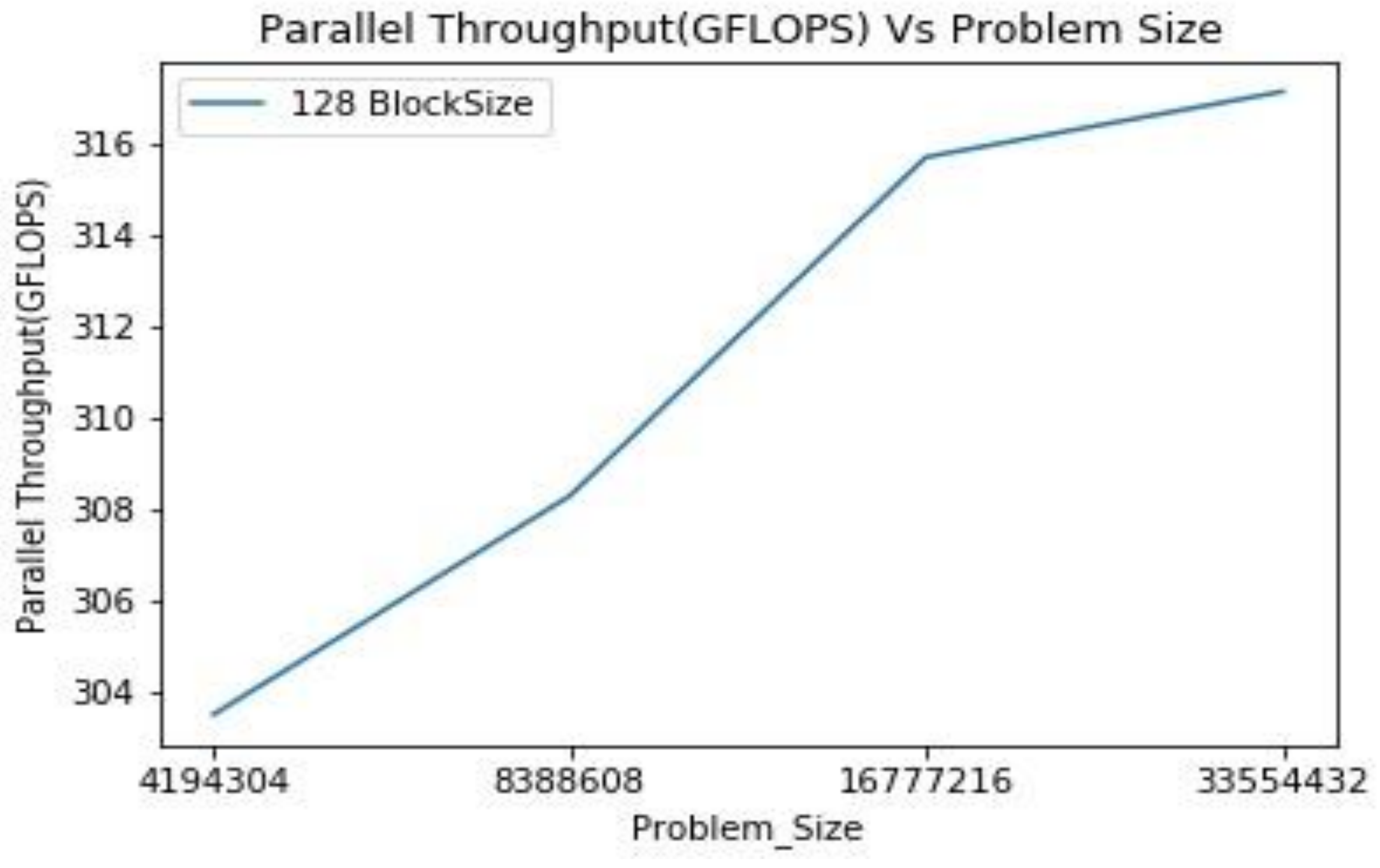
## ❖ Performance Analysis:

- Peak Throughput of GPU: 1.43 TFLOPS
- Max achieved throughput: 0.3 TFLOPS (128 Thread Block)
- 128 Thread Block gives maximum Thorough-put because it uses maximum thread blocks(16 as per the specification of ) per SM.
- Peak Speedup = 183 (128 Thread Block)
- Average SM efficiency of 70% for kernel-2.

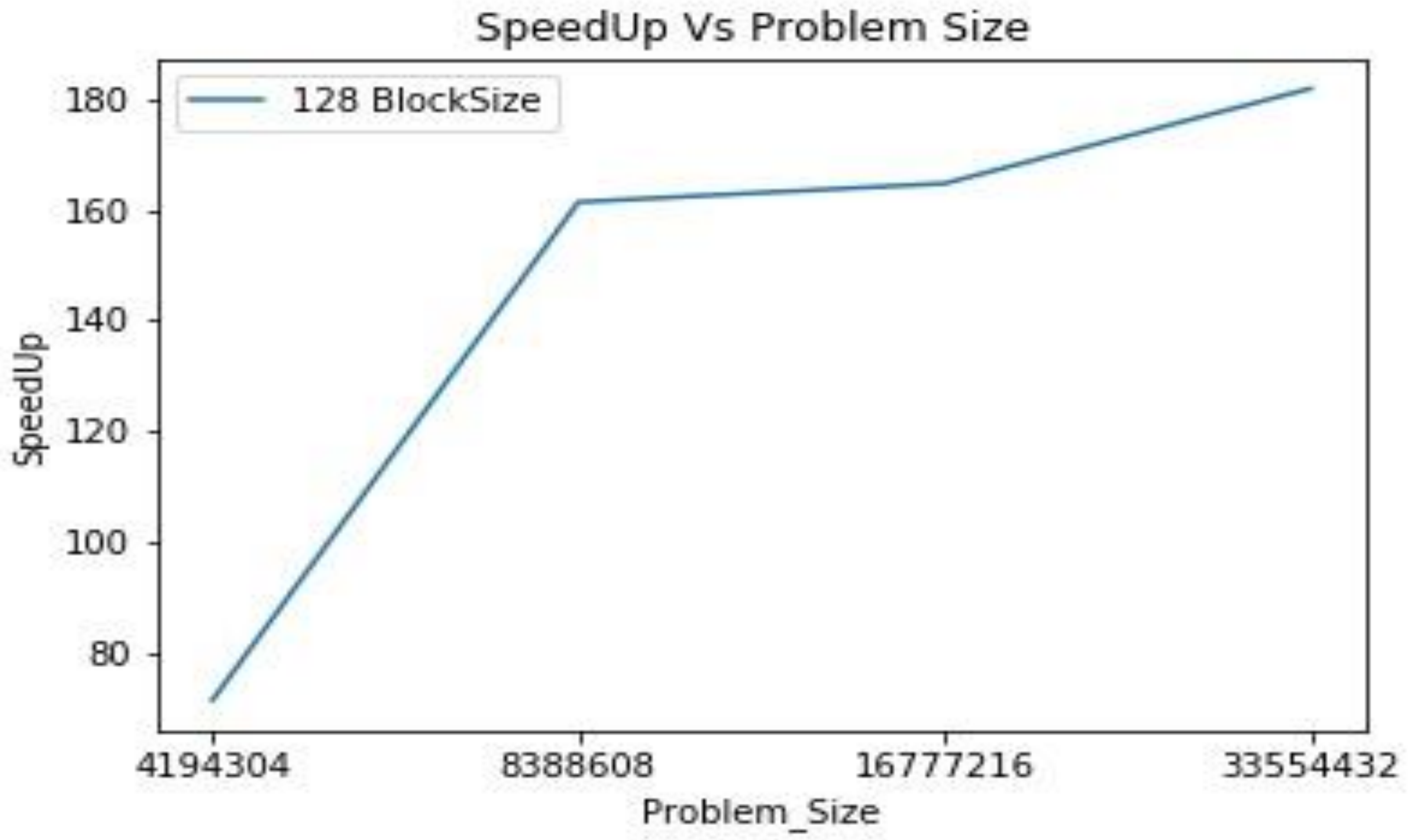
## ❖ **Serial Throughput Vs Problem size:**



## ❖ Parallel Throughput Vs Problem size:



## ❖ **Speedup Vs Problem size:**



## ❖ **Conclusion:**

- We conclude that parallel implementation of K-means Clustering for larger problem size can significantly reduce execution time and increases throughput contrary to serial implementation where throughput is decreasing as problem size increases.