Parallel Programming: Linear Regression with RANSAC

Team33
109062206 張雅涵
109062211 張惇媛

Content

- **Problem Description**
- **Implementations**
  - **OpenMP**
  - **OpenACC**
  - **CUDA**
- **Experiment & Analysis**
- **Comparison**

Content

- **Problem Description**
- Implementations
  - OpenMP
  - OpenACC
  - CUDA
- Experiment & Analysis
- Comparison

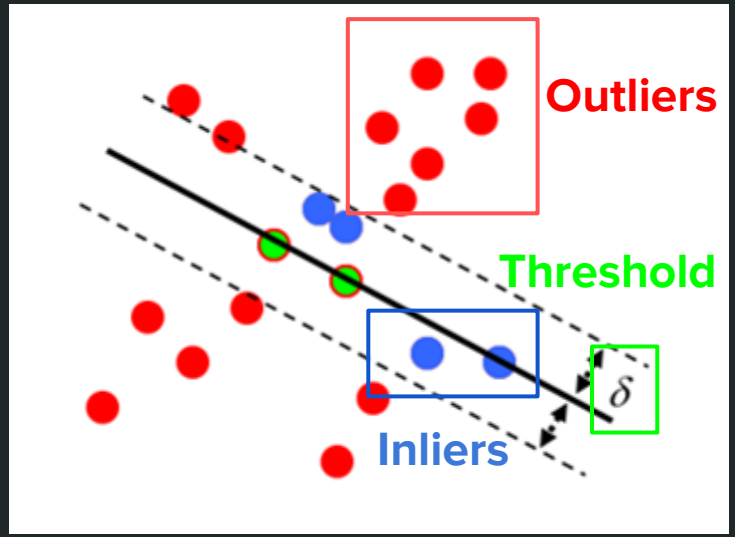
RANSAC: Random Sample Consensus

Goal:

- Robust regression algorithm
- Handles datasets with outliers

Parameters:

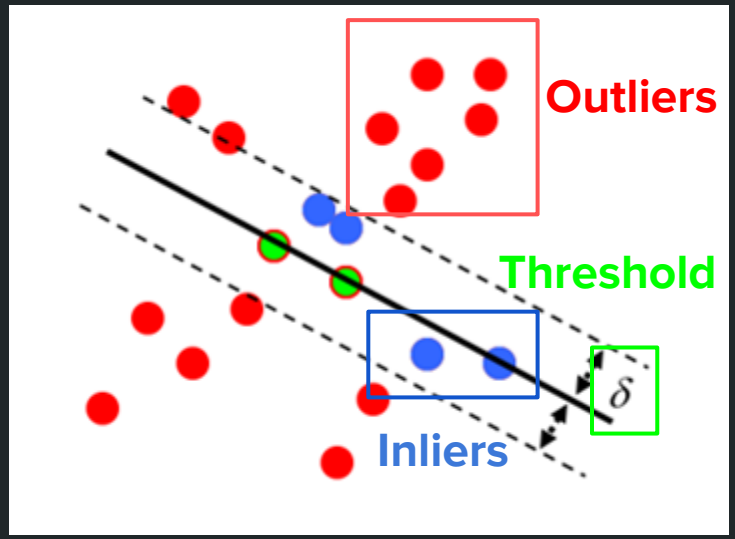
- number of iterations
- threshold



RANSAC:
Random Sample Consensus

Steps:

- Randomly selects 2 points for each iteration
- Fit line to the selected points
- Evalute inliers (points with error < threshold)
- Iterative process for best regression model selection (model with most inliers)
- use “Ordinary Least Squares” method for final regression model



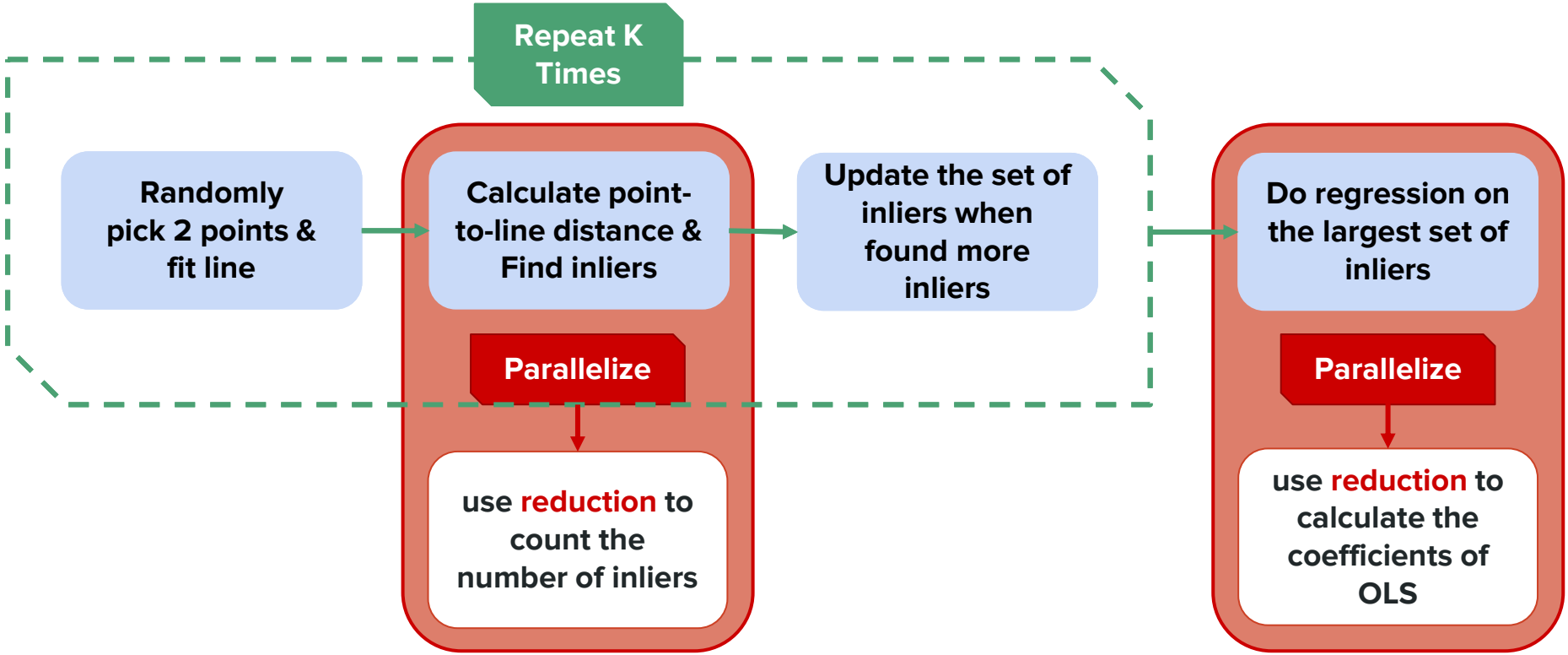
$$slope = \frac{\sum X_i Y_i - n * \mu_x * \mu_y}{\sum {X_i}^2 - n * {\mu_x}^2}$$

$$intercept = \mu_y - slope * \mu_x$$

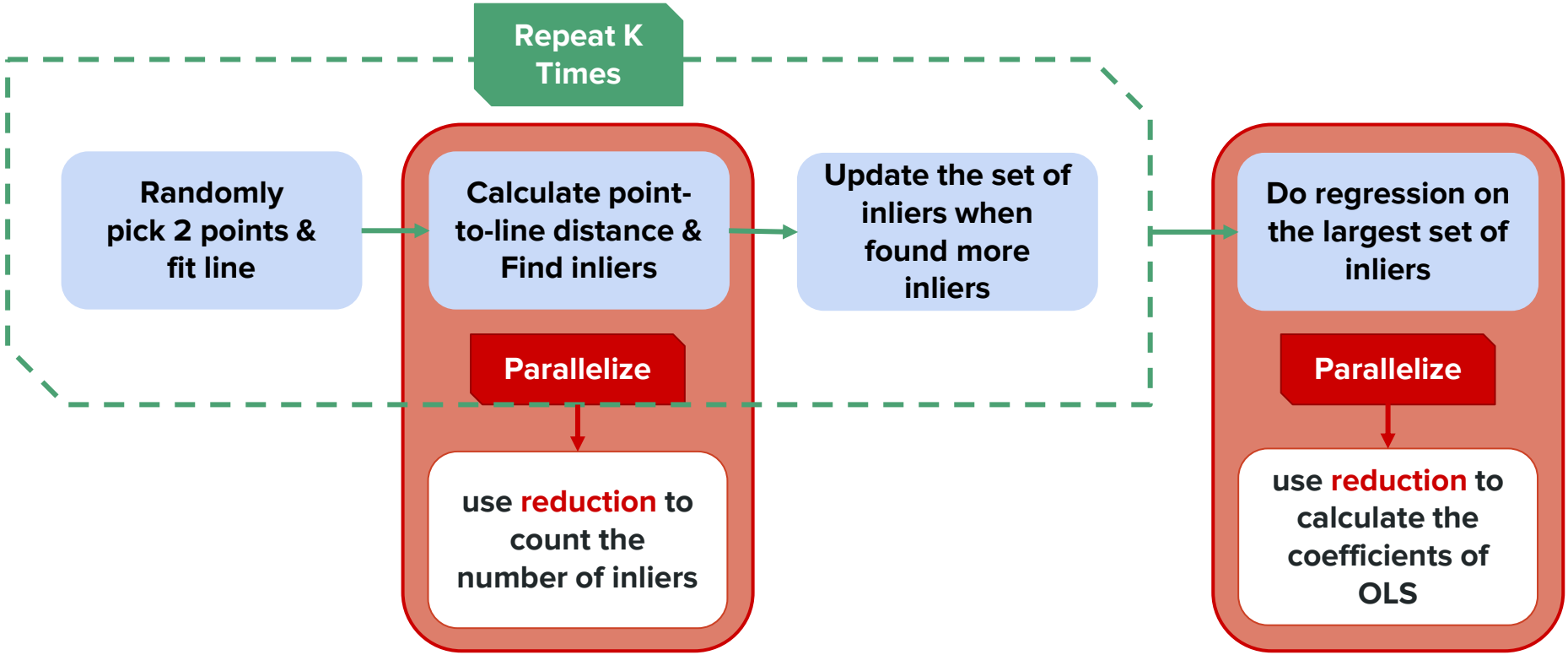
Content

- Problem Description
- **Implementations**
  - **OpenMP**
  - **OpenACC**
  - **CUDA**
- Experiment & Analysis
- Comparison

Implementation - OpenMP

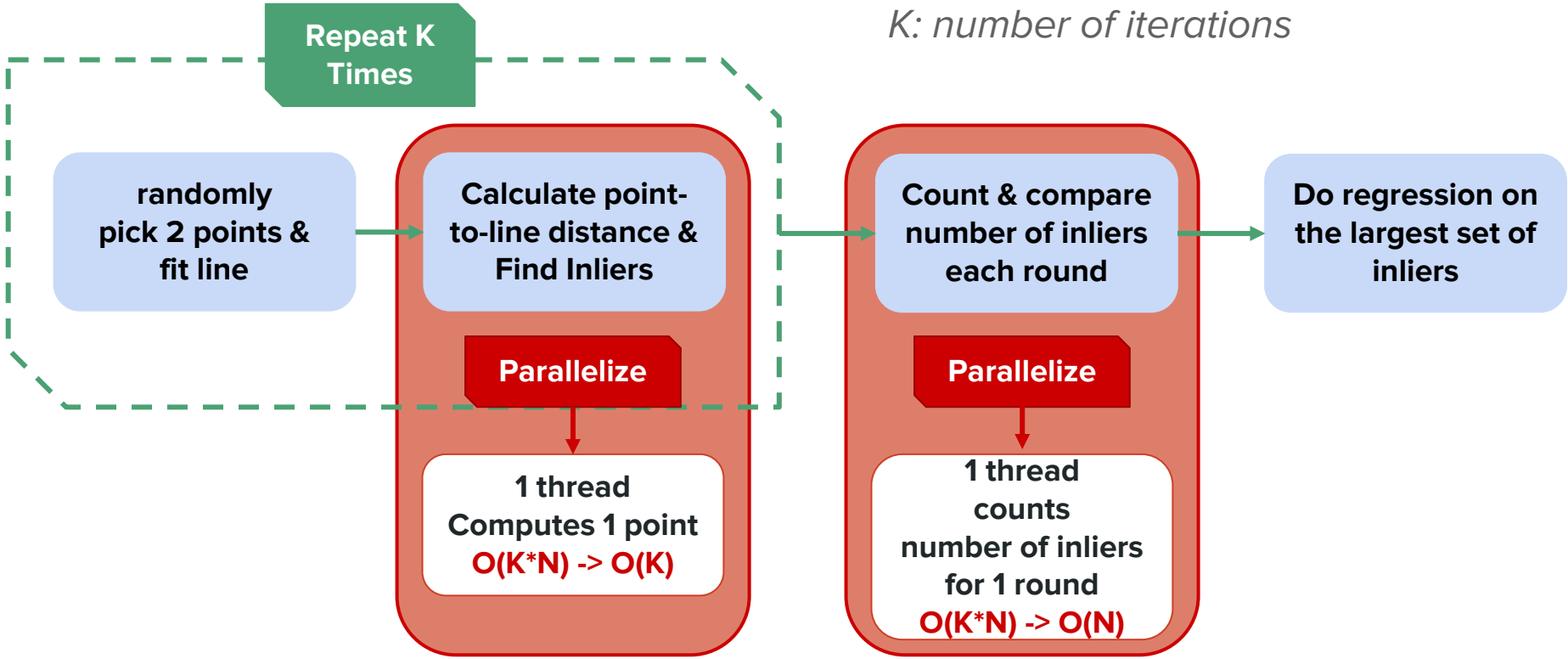


Implementation - OpenACC



Implementation - CUDA

N: number of points
K: number of iterations



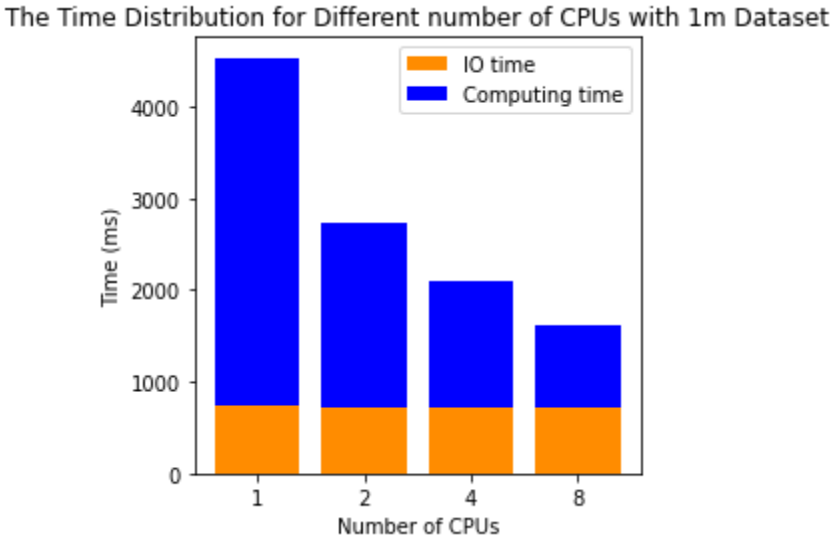
Content

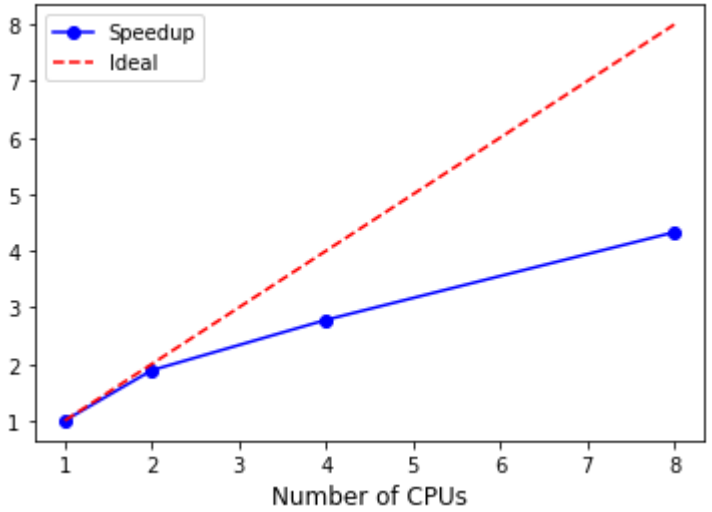
- Problem Description
- Implementations
  - OpenMP
  - OpenACC
  - CUDA
- **Experiment & Analysis**
- Comparison

Experiment setting

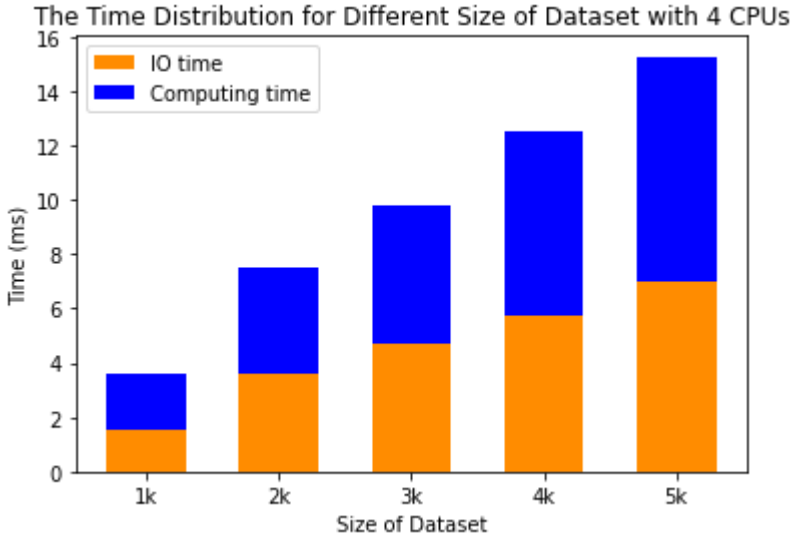
- Environment: Apollo and Hades
- Testcases:
  - number of points => 1k, 2k, 3k, 4k, 5k, 1m
  - file size => 16 KB, 32 KB, 48 KB, 64 KB, 80 KB, 16 MB

Experiment - OpenMP

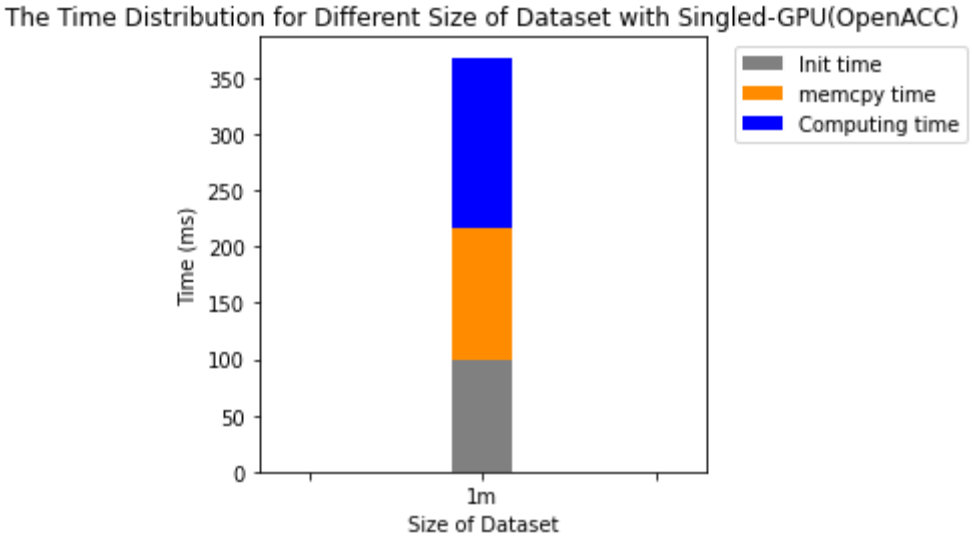




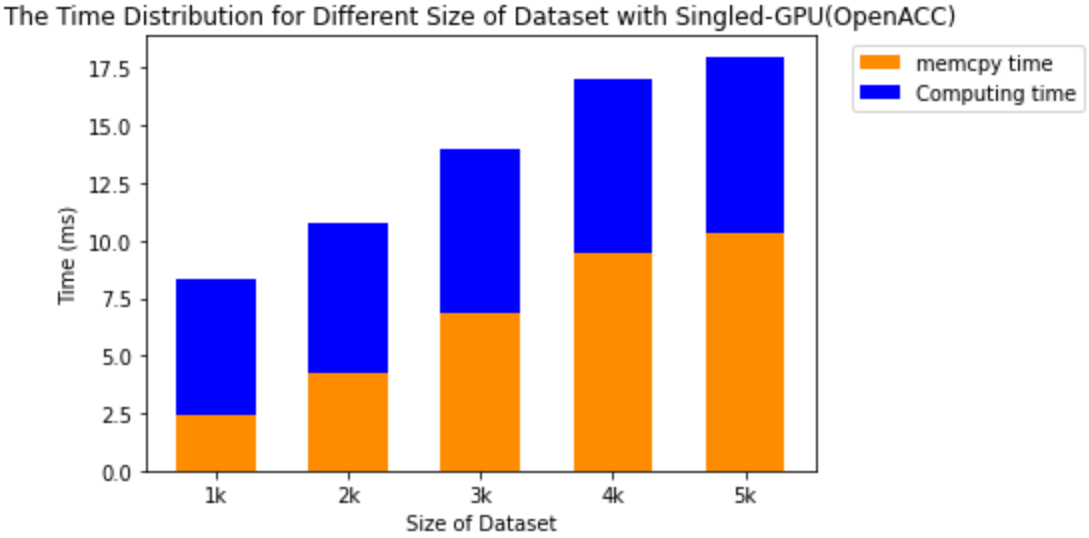
Experiment - OpenMP



Experiment - OpenACC

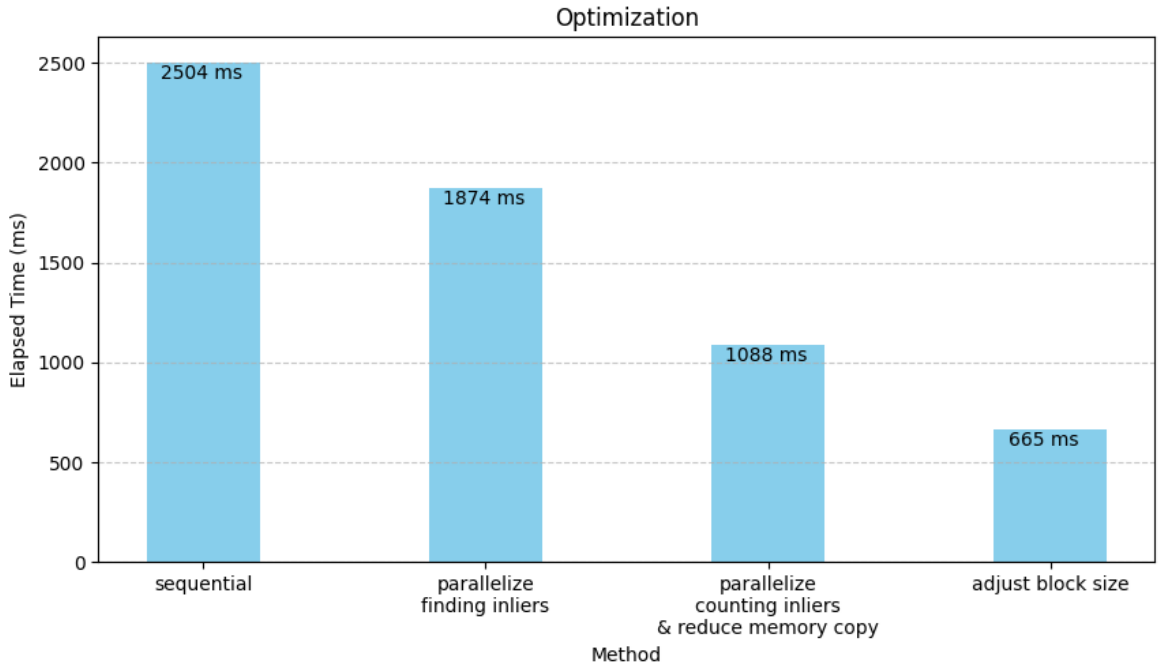


Experiment - OpenACC



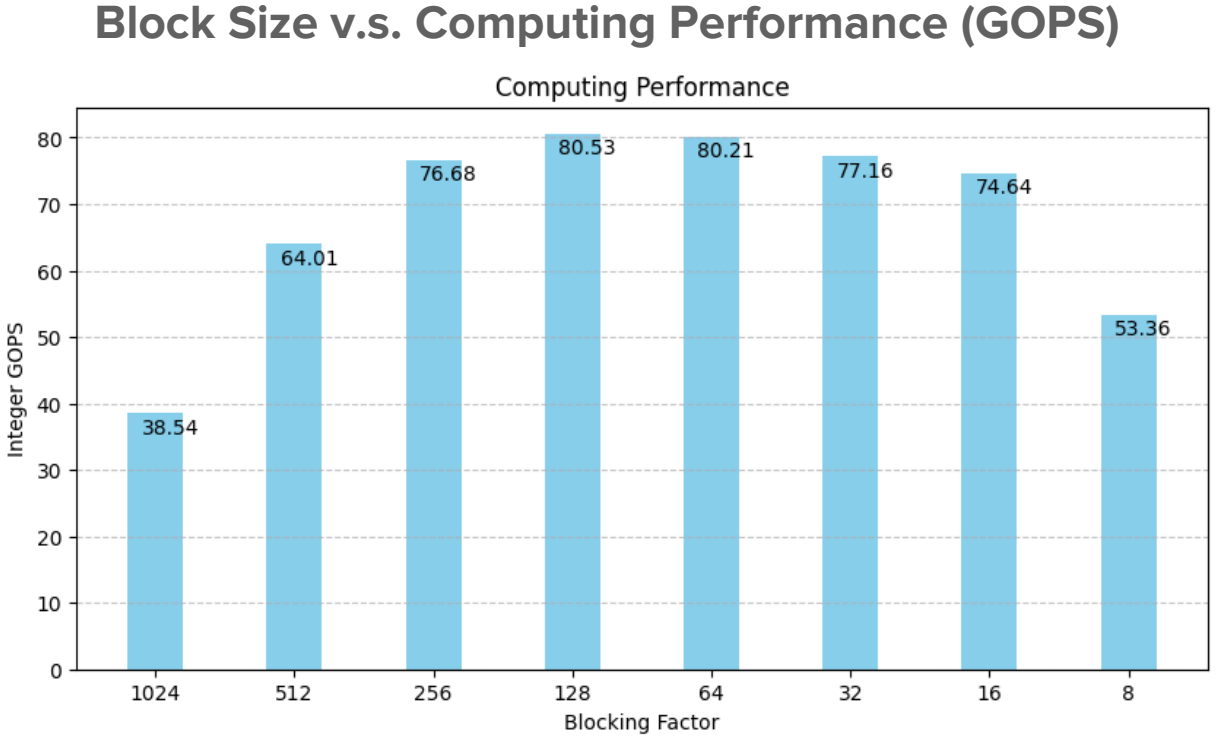
Experiment - CUDA

Optimazation *Test Data: 1M points*



Experiment - CUDA

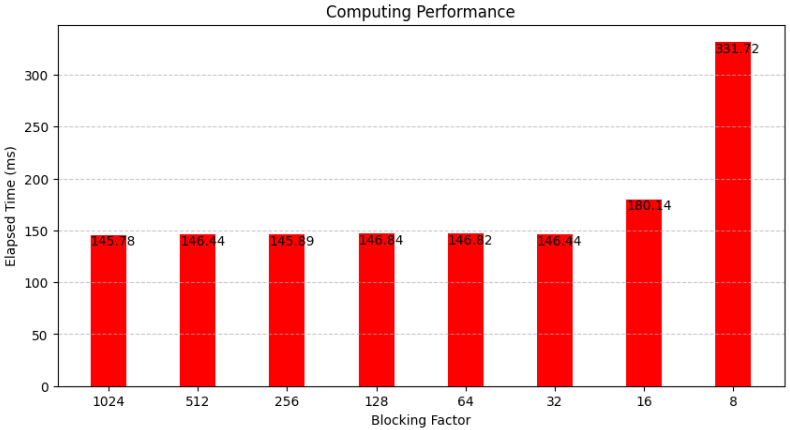
GOPS =
total kernel integer instruction
/ total kernel time



Experiment - CUDA

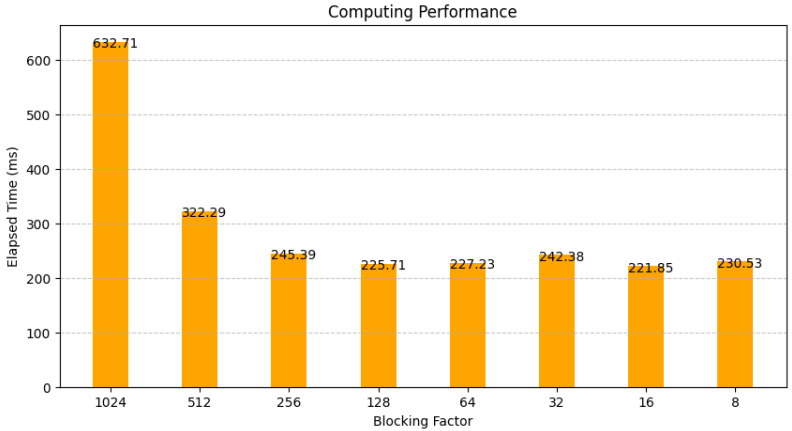
Block Size v.s. Computing Performance (GOPS)

Kernel 1
Find Inliers



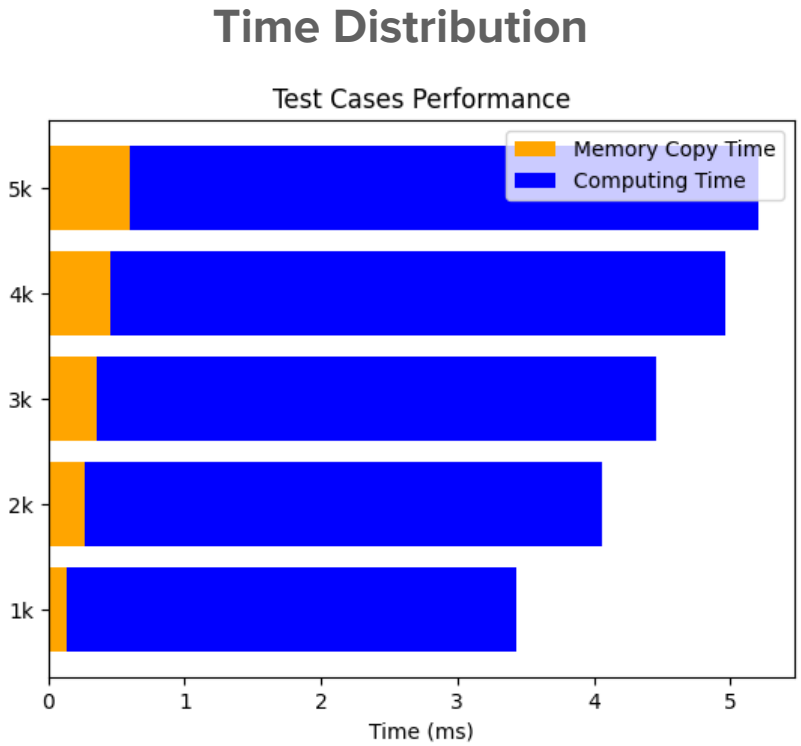
Number of Threads = N (1m)
Number of Block = N/Block Size

Kernel 2
Count inliers



Number of Threads = K (2000)
Number of Block = K/Block Size

Experiment - CUDA



Content

- Problem Description
- Implementations
  - OpenMP
  - OpenACC
  - CUDA
- Experiment & Analysis
- **Comparison**

Comaprison

Data size: 1M dataset
Number of Iteration: 2000
Only consider the computing time and communication time

Methods	Sequential	OpenMP	OpenACC	CUDA
Time (ms)	4037	879 (with 8 CPUs)	270	649

Comaprison

Methods	Sequential	OpenMP	OpenACC (parallelize regression)	CUDA (parallelize inliers count)
Time (ms)	4037	879 (with 8 CPUs)	270	649

OpenMp:
Pros -
No memory copying
Cons -
Less parallel resources

Cuda/OpenACC:
Pros -
More computing resources
Cons -
Long HtoD memory copy time

Conclusion

1. Successfully implemented parallelized RANSAC on GPU

2. Reduced Memory Copy Time

3. Parallel algorithm can be applied on other RANSAC applications