Image Segmentation using K-Means Clustering with OpenMP

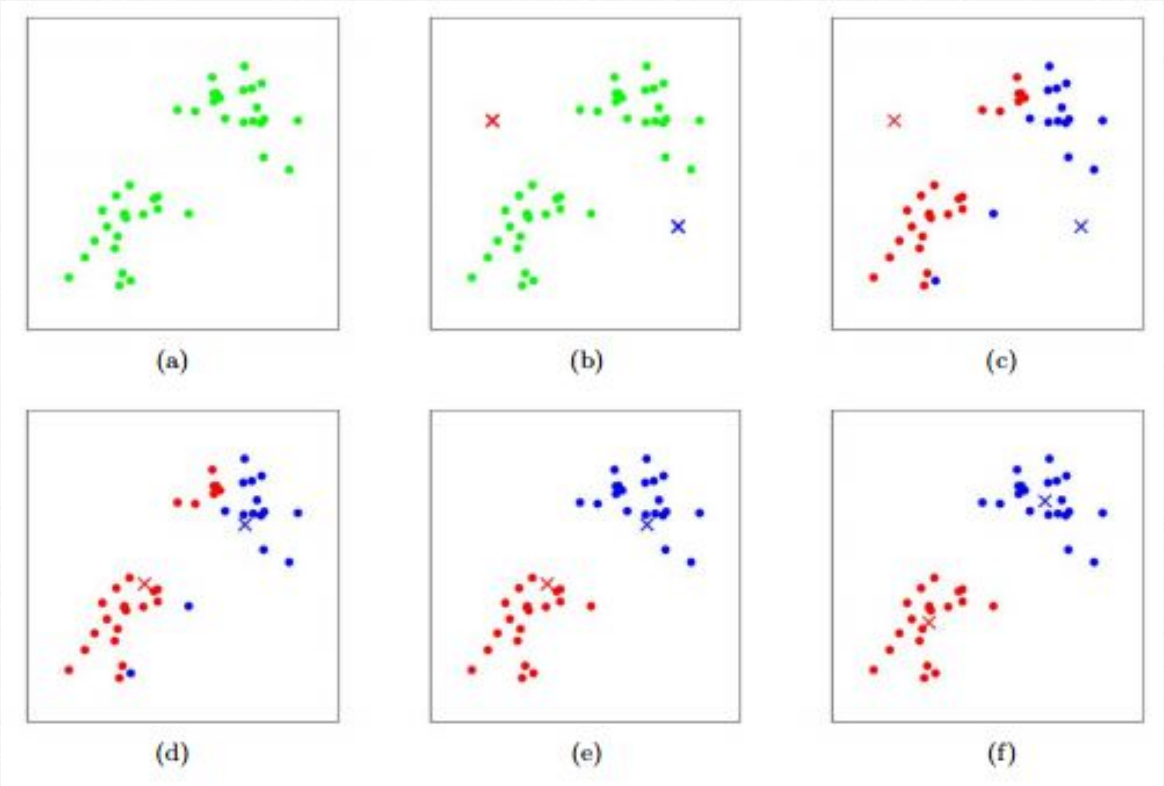
member : 310551154 林子恒、310554047 張方華、310552059曾宇廷

Outline

- Introduction/motivation
- Problem statement
- Proposed solution
- Evaluation
- Related work
- Contributions of each member
- Conclusion

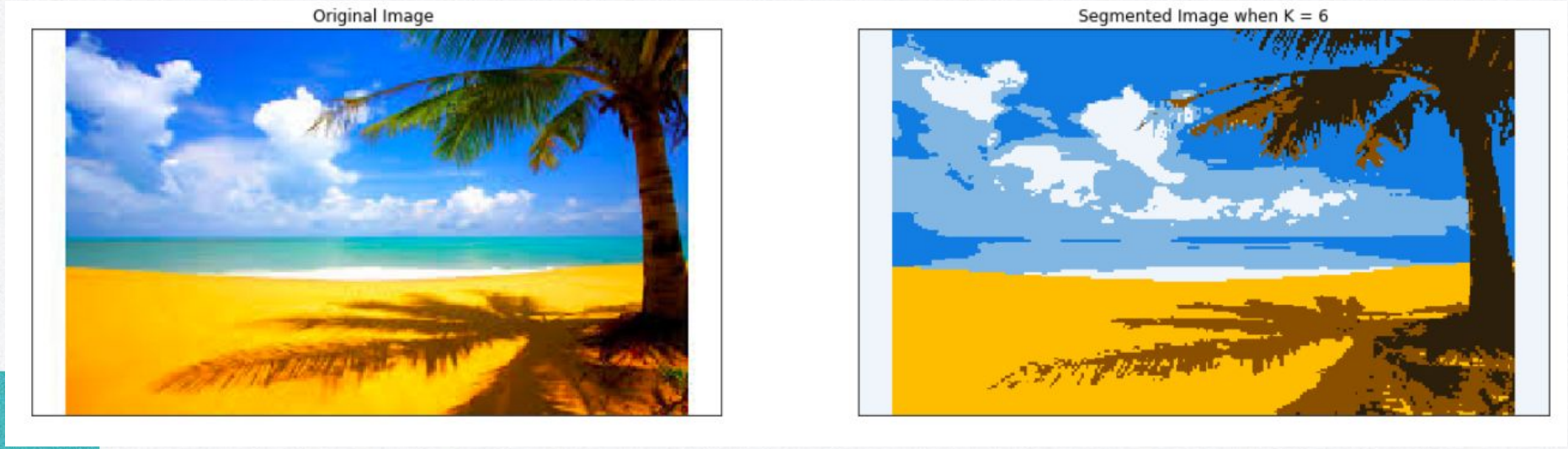
Introduction / motivation

kmeans



Introduction / motivation

K-means image segmentation / compression



1. findAssociatedCluster()
2. adjustClusterCenters()
3. applyFinalCluster()

Problem statement

A **megapixel** (**MP**) is a million pixels

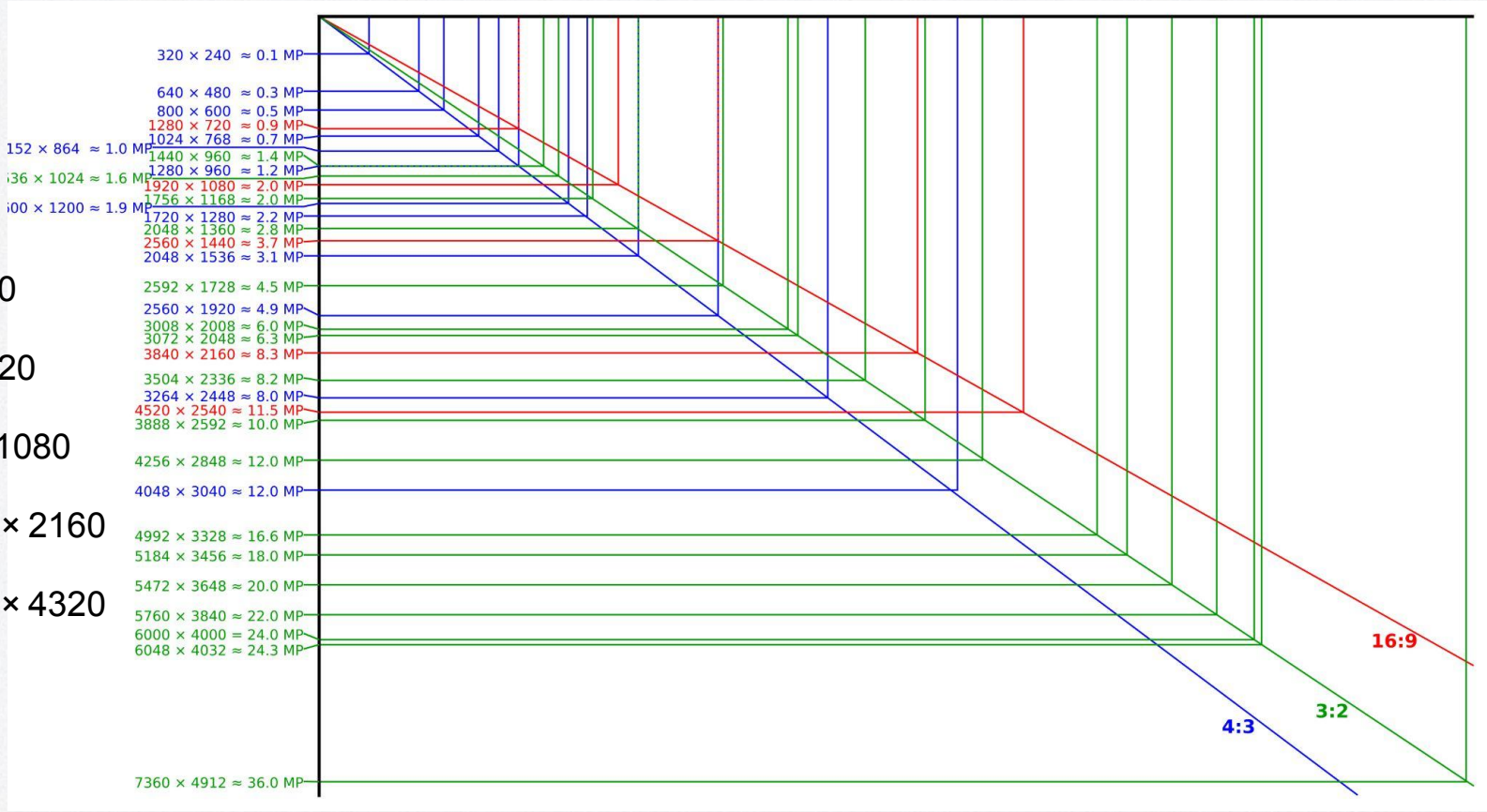
480p = 720 × 480

720p = 1280 × 720

1080p = 1920 × 1080

4K UHD = 3840 × 2160

8K UHD = 7680 × 4320



Proposed Solution

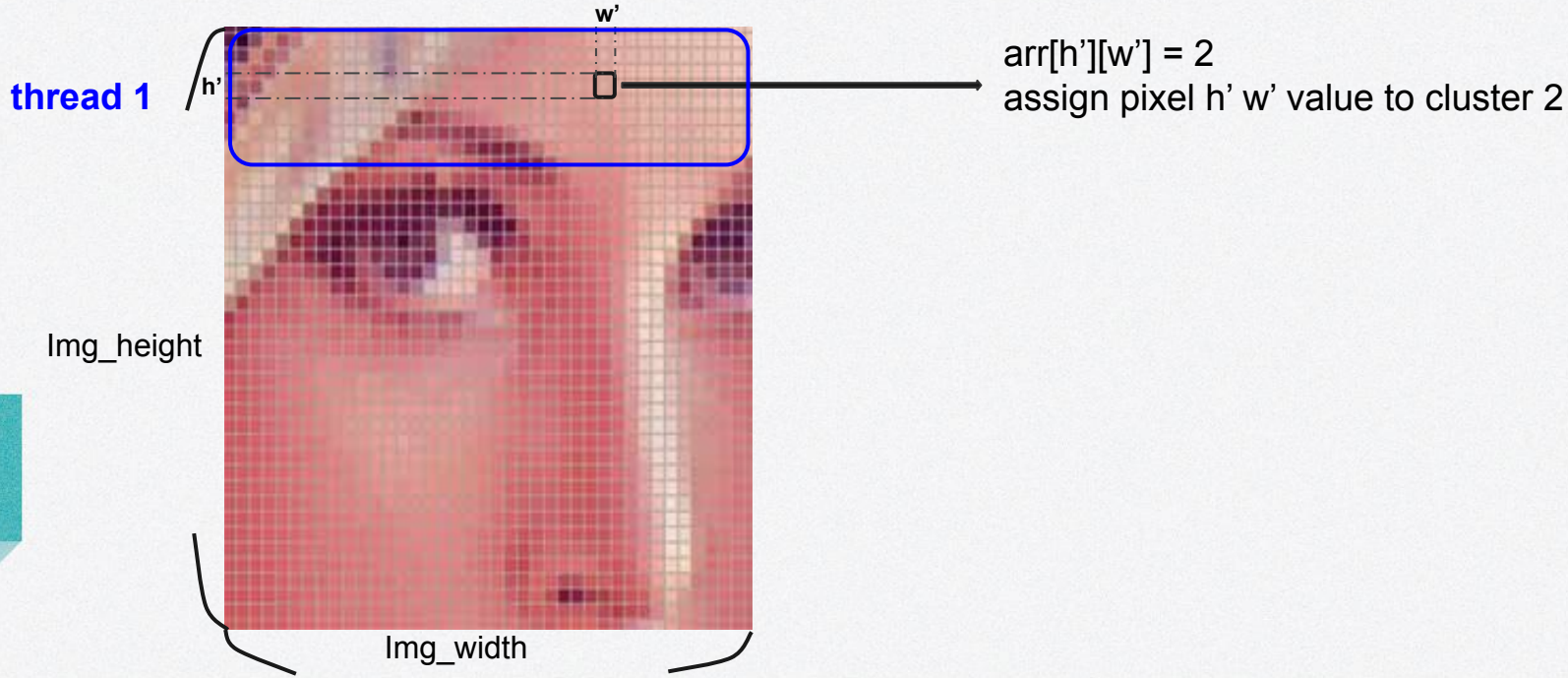
```
threshold = 0.001

while diffChange > threshold  do
    findAssociatedCluster()
    diffchange = adjustClusterCenter()
    applyFinalClusterToImage()
end
```

Proposed Solution

findAssociatedCluster()

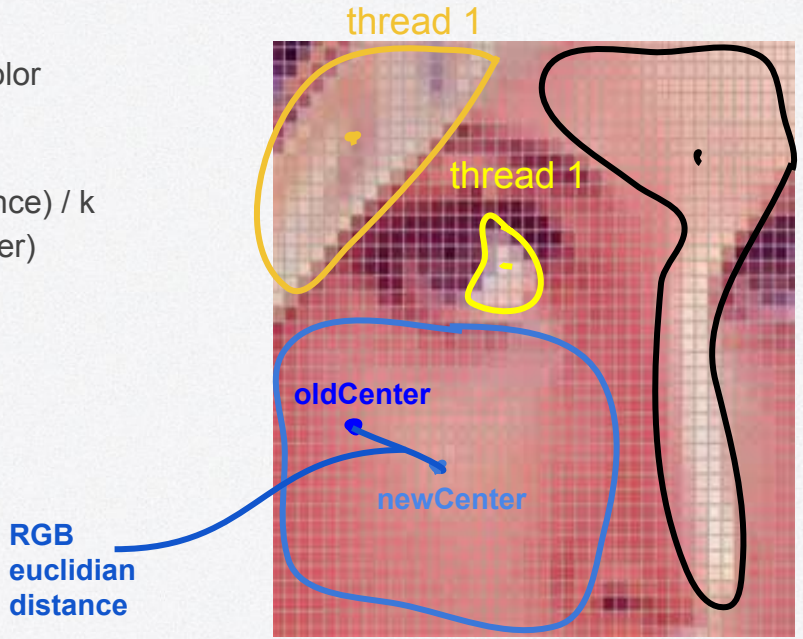
1. Assign **Img_height/P** to each processor to find associated cluster for each pixel
2. Store the reallocated cluster number **k'** in a dynamic 2d array of Img_height * Img_width size



Proposed Solution

adjustClusterCenters()

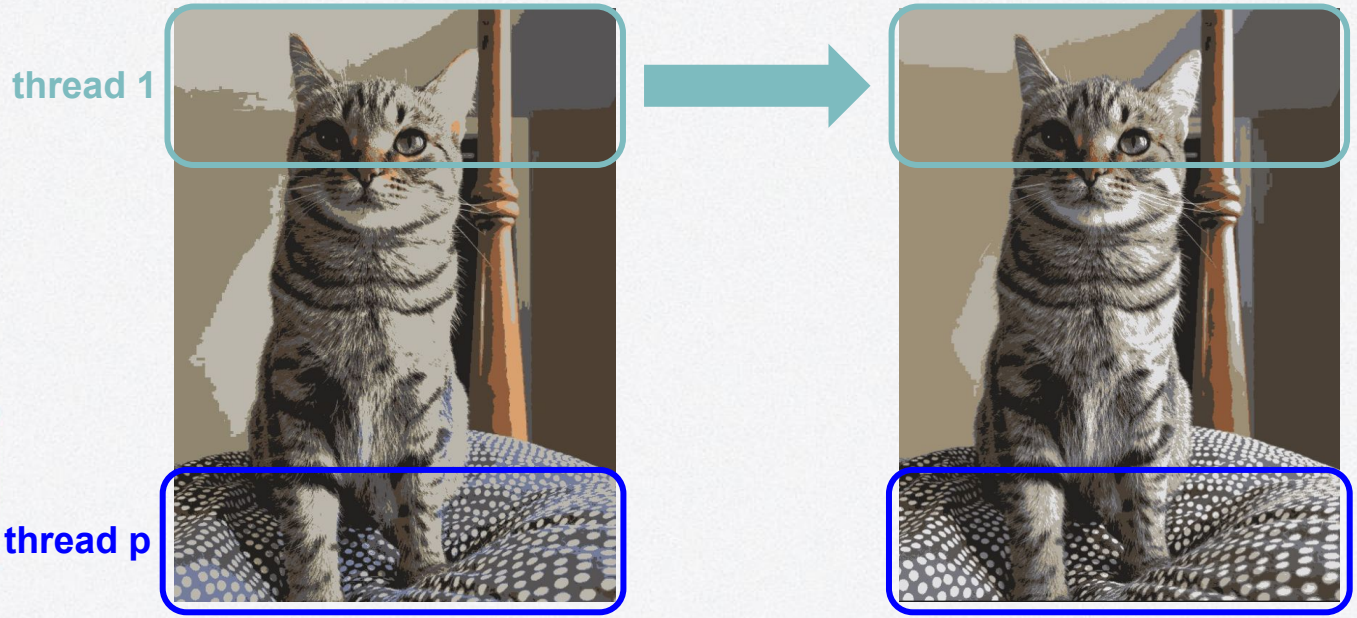
1. Assign **K/P** to each processor to compute the center color euclidian distance
2. Update new center to each cluster
3. meanNewCenter = (sum up newCenter euclidian distance) / k
4. Calculate diffChange = abs(oldCenter - meanNewCenter)



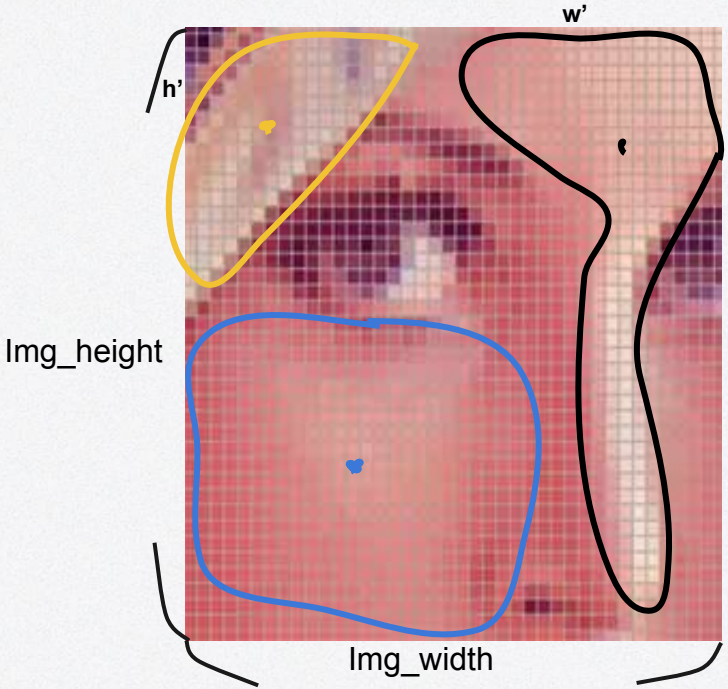
Proposed Solution

applyFinalClusterToImage()

1. Assign **Img_height/P** to each processor to write Image



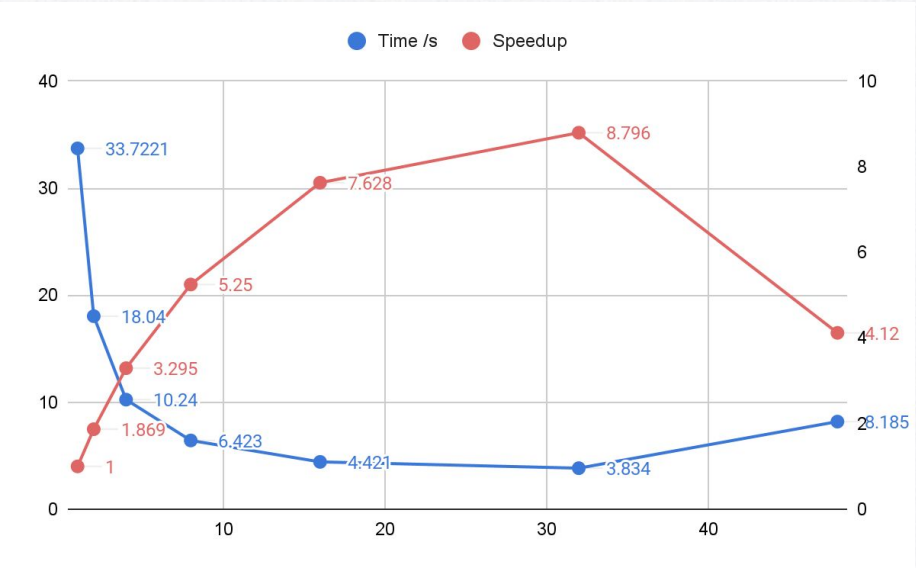
Proposed Solution

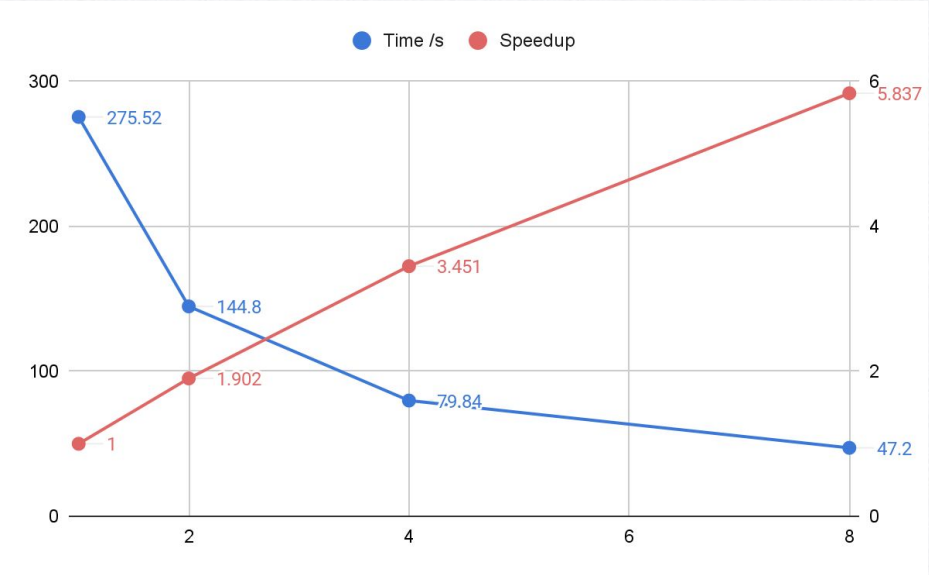


Evaluation

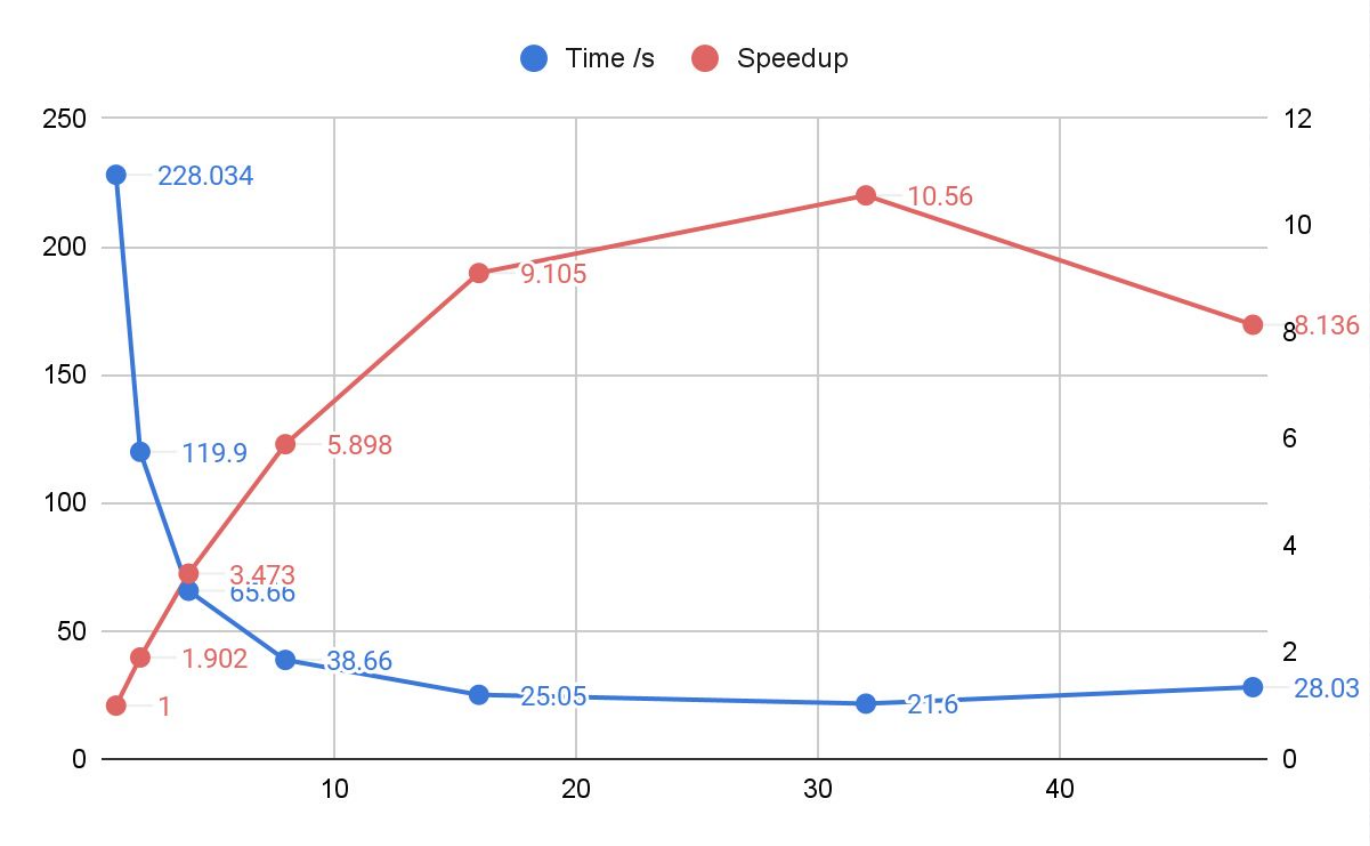
- The time proportion of *"findAssociatedCluster"* will decrease as the number of threads increases, but the time proportion of *"applyFinalClusterToImage"* will increase as the number of threads increases.
- The bottleneck of acceleration is about when threads = 16, because too many threads will increase the communication cost.
- The number of pixels affects computation time, but the pixel value affects the number of convergence rounds.

Evaluation - Small, K = 32

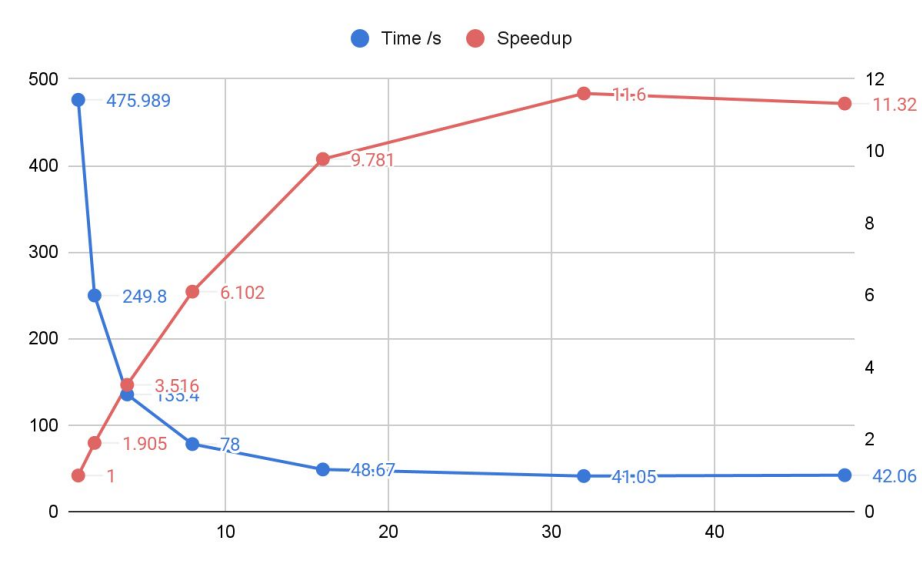


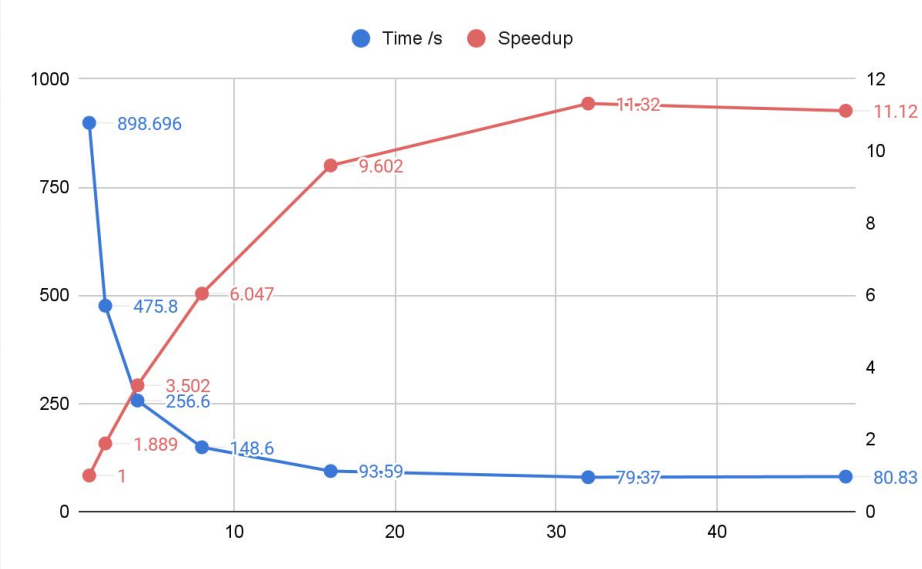


Evaluation - Small, K = 32

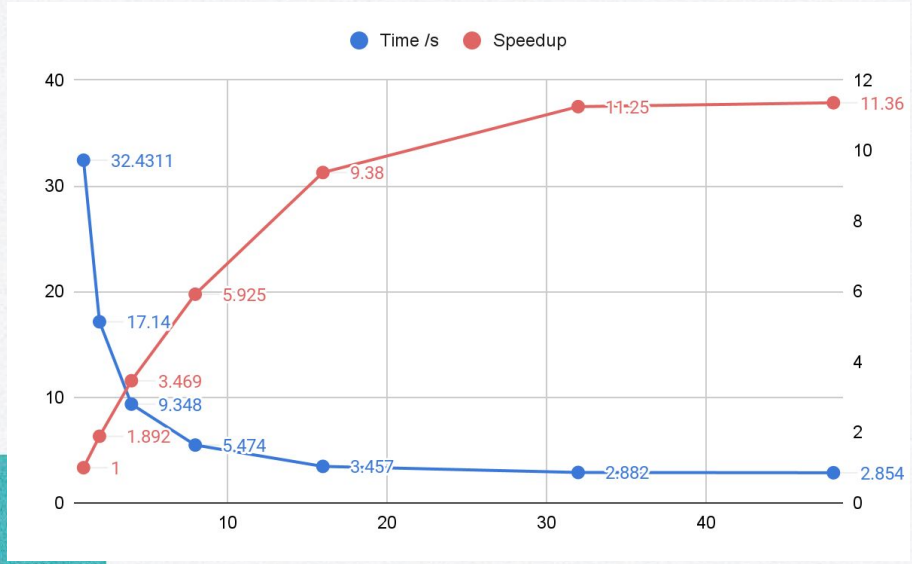


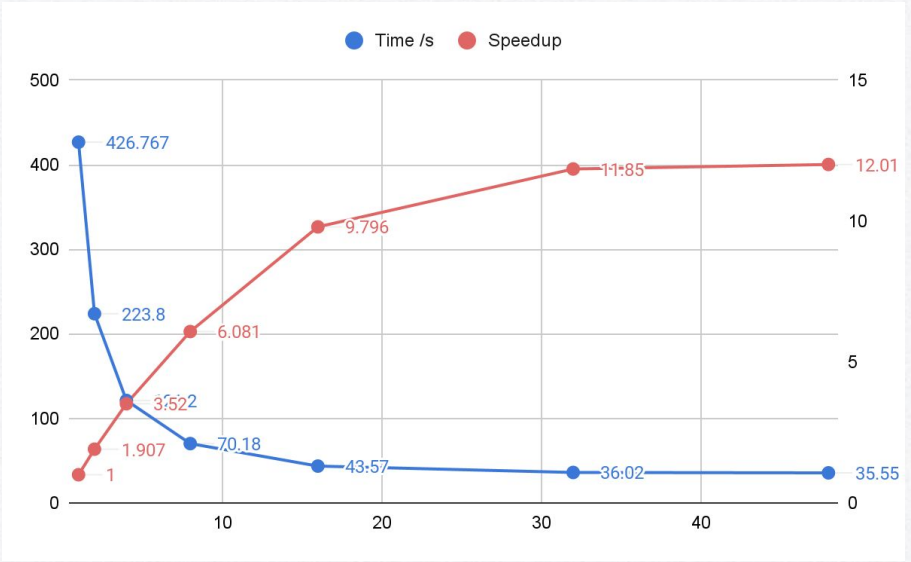
Evaluation - Medium, K = 32





Evaluation - Big, K = 32





Conclusion

1. 顏色較相近的圖片會收斂較快
2. 要先針對程式中的運算貧頸做平行化
3. 編譯器不要使用 -O3，要使用 -O0
4. 在寫 serial 的時候就要考慮 parallel 版本
5. 以 Row 來做平行在 thread 較多時會提升較多

Q&A