In my project, we split the initial problem into huge number of independent subproblems and solve them with available workers. Here two steps must be defined-

1) The definition of the tasks and

2)The task assigned to each worker. Here we have used static task assignment where each worker does a fixed part of problem which is dehazing an image.

Motivation - Hadoop could be used as a solution for parallel computations. But in Hadoop systems, we have to install software's (like Apache Spark) for parallel computations which would make it system dependent. We wanted to implement it in a new way such that the computations are system independent. So this motivated us in doing this project without Hadoop.

Objective - Images of outdoor scenes are usually degraded by the turbid medium in the atmosphere. So, the main objective of our project is to accurately determine which areas are hazy and dehaze the hazy areas and to complete all rendering in a reasonable amount of time. Here each worker would independently perform their tasks without communicating with other tasks.

Our method to convert the Hazed image to dehazed image is by using

Dark Channel Prior Method

The dark channel prior method consists of a process called soft matting where we use Preconditioned Conjugate Gradient (PCG) algorithm which takes about 10-20 sec to process a 600*400 pixel image on a PC with a 3.0GHz Intel Pentium 4 processor. The dark channel prior method can unveil the details and recover vivid color information even in very dense haze regions. But it is a kind of statistic, which may not work for some particular images. When the scene objects are inherently similar to atmospheric light and no shadow is cast on them, the dark channel prior is invalid. Our method will underestimate the transmission for these objects.

Main **Challenges** that we have faced here:

- One of major challenges while dealing with embarrassingly parallel computing is synchronization between workers. This scenario boils down to the situation that once an image is selected for processing, no other worker should process the same image.
- Load imbalance- Different amounts of works across processors and the different speeds of the processors cause load imbalance.
- Communication cost between server and client.
- The parallel speedup of any program is limited by the time needed for any sequential portions of the program to be completed.

Technical Requirements

1- Xampp Stack
2- Mysql server
3- FFmpeg

Installation

Ubuntu - (https://linuxize.com/post/how-to-install-ffmpeg-on-ubuntu-18-04/) This works for all versions for Ubuntu.

LinuxSystems (https://www.ostechnix.com/install-ffmpeg-linux/)

(https://itsfoss.com/ffmpeg/)

Language Requirement

1- PHP
2- Javascript
3- Html

Libraries Used

1- OpenCV
2- Javascript utility library

Explanation of File Directory of the code

1) index.php – This is used for uploading of the images(Hazed) which have been extracted from the video. This acts as the **Server** page.
2) Lib folder- This folder consists of the libraries that are used for the hazing algorithm to run. ie opencv.js and util.js
3) zip.php – This is to download all the images(dehazed images) in zipped format into the computer.
4) delete.php – This used to clean all the images in the c folder/out and also clean the data base in server.
5) C folder – This is used as **Client** page. This is where all the magic happens parallely.
6) C folder/in - This is where all the uploaded images have been stored.
7) C folder/out - This is where all processed images(Dehazed Images) are stored.
8) C folder/dehaze.php – Here the conversion of the images code is written. All the load balancing for images is done here.
9) C folder/test.php – Here the updating of the database is done ie Database is updated whether the image is processed or not.
10) C folder/out.php – Here the image in the infolder is removed and respective same dehazed image is stored in the output image.
11) C folder/zip.php – This is to download all the images(dehazed images) in zipped format into the computer.

FFmpeg

How to run FFmpeg from a webpage

Kindly please refer this

https://www.ibm.com/developerworks/library/os-php-commandline/index.html

FFmpeg shell script code

1) Give the path of the folder where the final images are stored.
   Eg: opt/lampp/htdocs/MajPro/c/out
2) ffmpeg -r 1/5 -pattern_type glob -i '*.png' -c:v libx264 output.mp4

This finds automatically all the images that are sequentially available in the folder and converts it a video file.

Note: Please give **Permissions**(open to all) to all the files and folders in the server or else there would be problems in accessing the images.