Adaptive Resource Management for Analyzing Video Streams from Globally Distributed Network Cameras

Now-a-days cameras are deployed almost in all fields such as battle ground, agriculture fields, road side to monitor traffic etc. All this cameras will have internet connection via broadband network and the activities of these cameras are to send captured videos to centralized server for monitoring. In agriculture fields it will be used to monitor temperature and in road it will be used to monitor traffic to inform peoples about congestion. Maintaining centralized server to process and store such huge videos required lots of infrastructure and cost.

To reduce cost all organizations are using cloud services to process and monitor huge amount of video data but this usage will not come free of cost and customers of cloud has to pay to cloud service providers based on usage of cloud servers such as CPU time, number of cloud instances and Virtual Machine. If customer choose improper cloud services then it will charge more on the customer.

To overcome from above issues some cloud servers will choose Cloud Instances for processing which are nearer to them to reduce cost. But this existing technique will not using any algorithm to choose cloud which is having minimum cost without compromising quality. Here we are dealing with videos so all frames from video should be of good quality for monitoring purpose.

In propose technique using ‘BestFirstDecreasingSort’ algorithm we can choose cloud instances which are nearer and having minimum cost. To apply this concept first algorithm will make a list of all available cloud instances and their infrastructure with cost details and then algorithm will sort the entire object to choose cloud which is having closer distance and less cost.

While sorting this application ARMVAC will use location, instances type and cloud cost to sort the object in order to have minimum cost cloud.

To implement above concept we don’t have any camera or sensor to capture videos so i am directly uploading video to the application which will use OPENCV library to process the video to detect whether video frame contains person or not.

This application will take a video and then convert that video into frames and send to cloud for processing. Cloud will detect whether given frame from a video having a person or not.

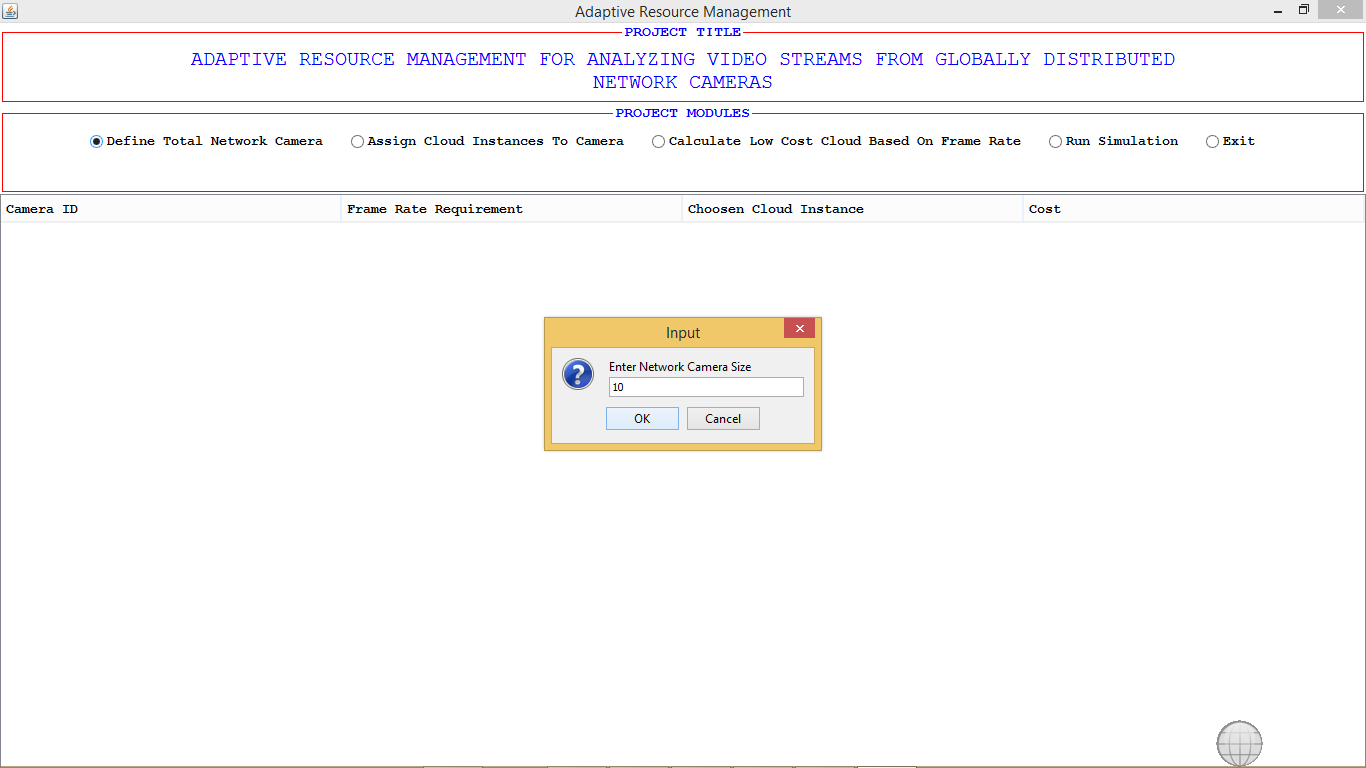
To run this application you can upload any video but here i am using one MP4 video for processing. I implement this project in the form of simulation where circles will represents as camera and they will use uploaded video frames to send to cloud for processing.

Screen shots

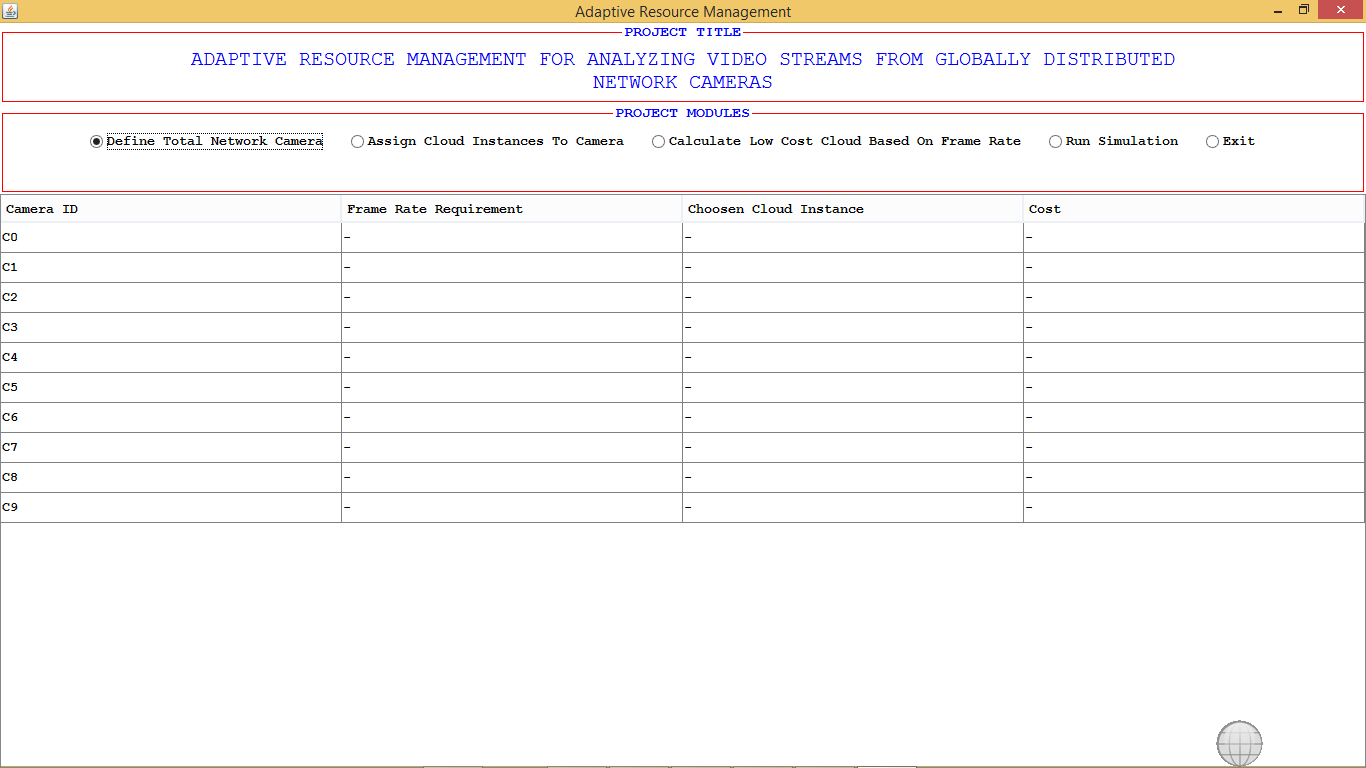
Double click on ‘run.bat’ file to get below screen



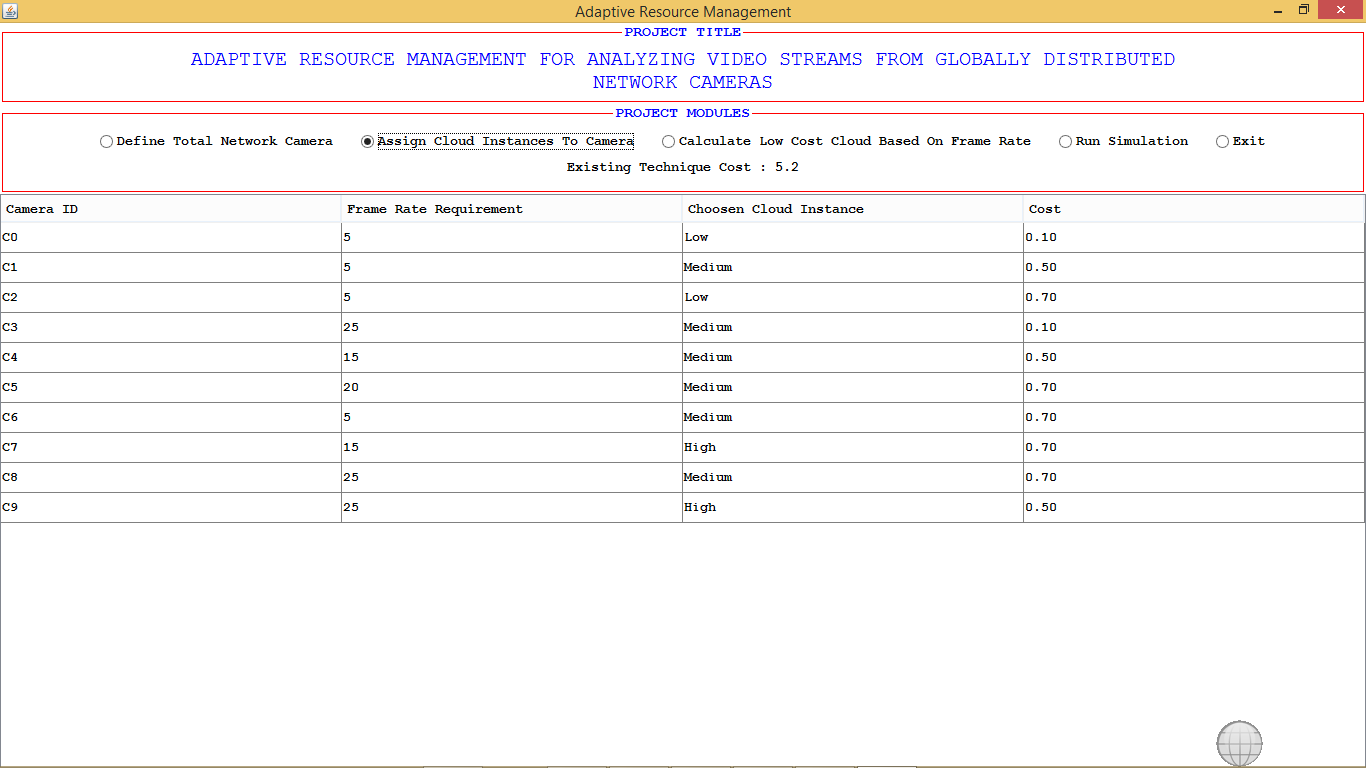
In above screen click on ‘Define Total Network Camera’ radio button to enter number of cameras for this project simulation



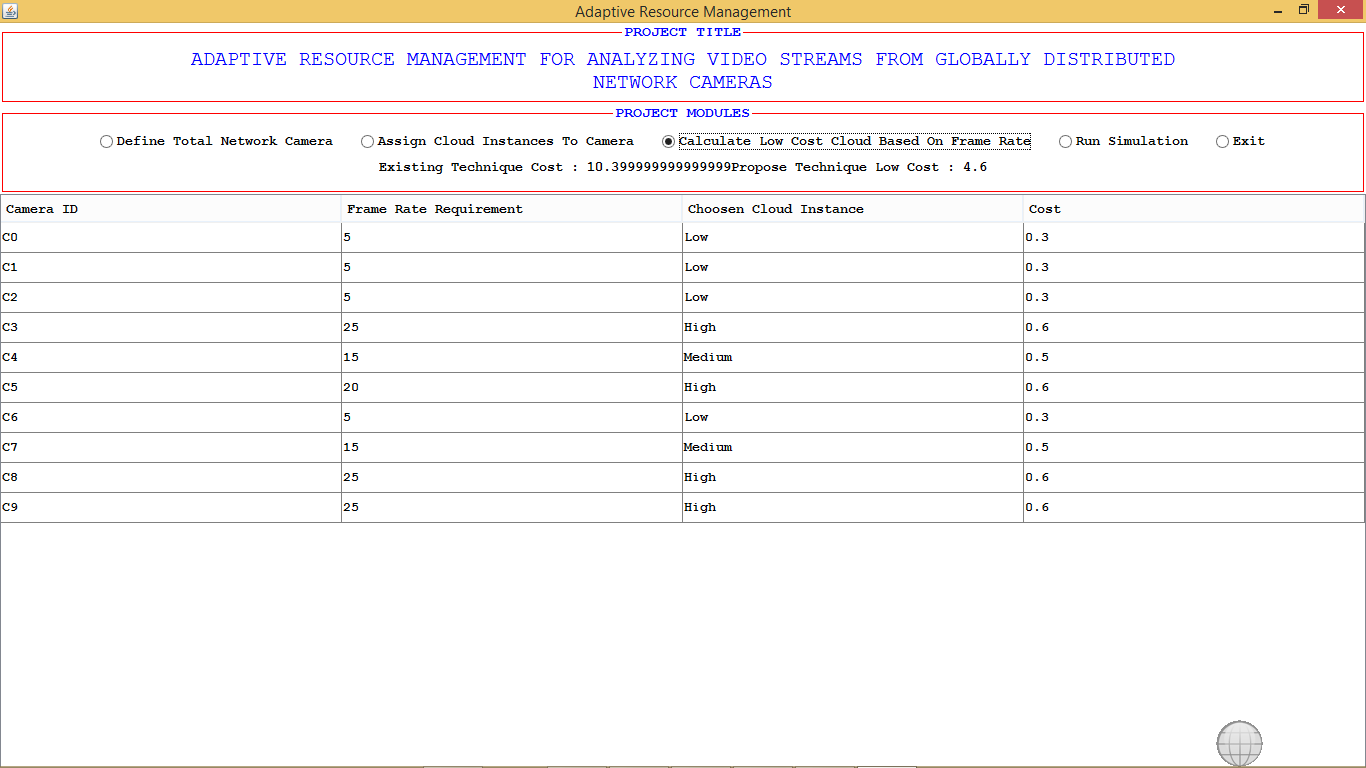
In above screen i am giving number of cameras as 10. Now click ok to get below screen



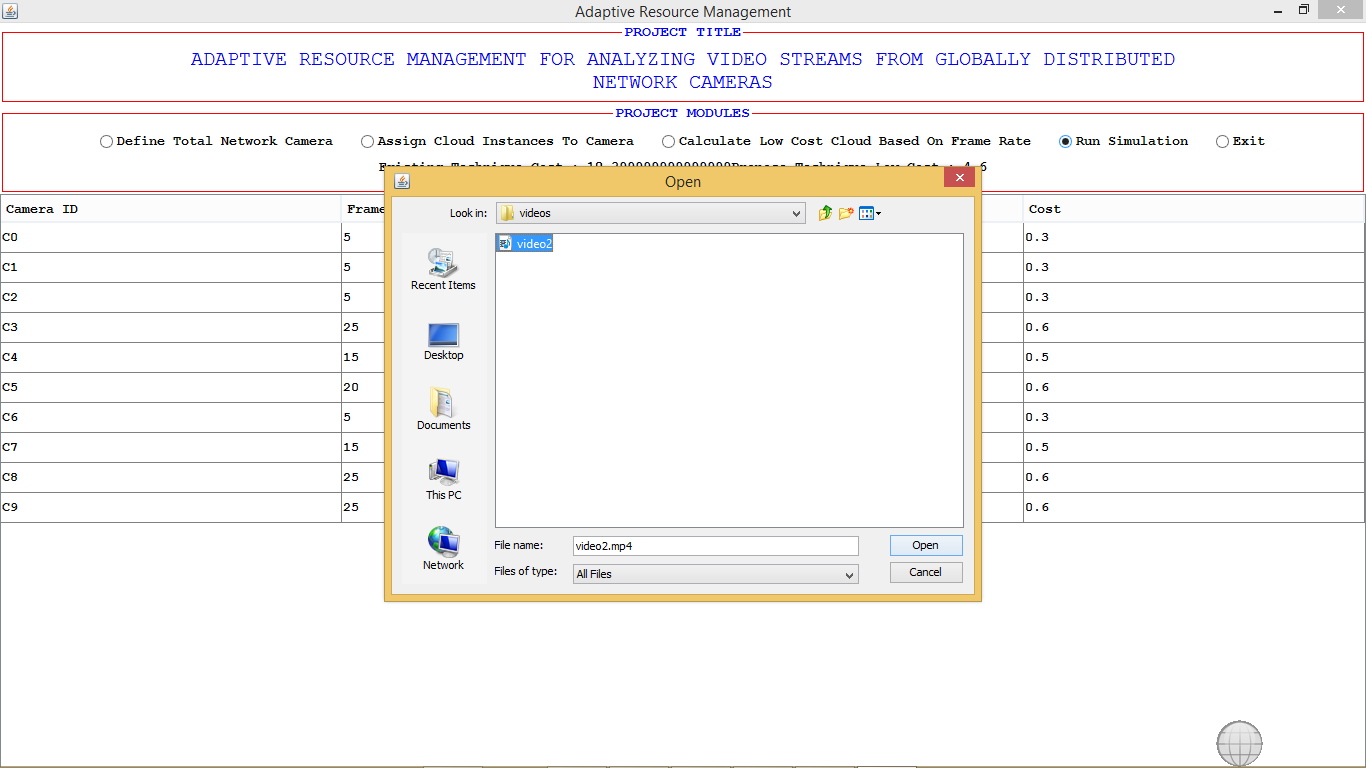
In above screen total 10 cameras are created from 0 to 9 but they don’t have any assigned cloud and frame rate requirement or cloud cost. Click on ‘Assign Cloud Instances To Camera’ radio button to assign this cameras to cloud instances



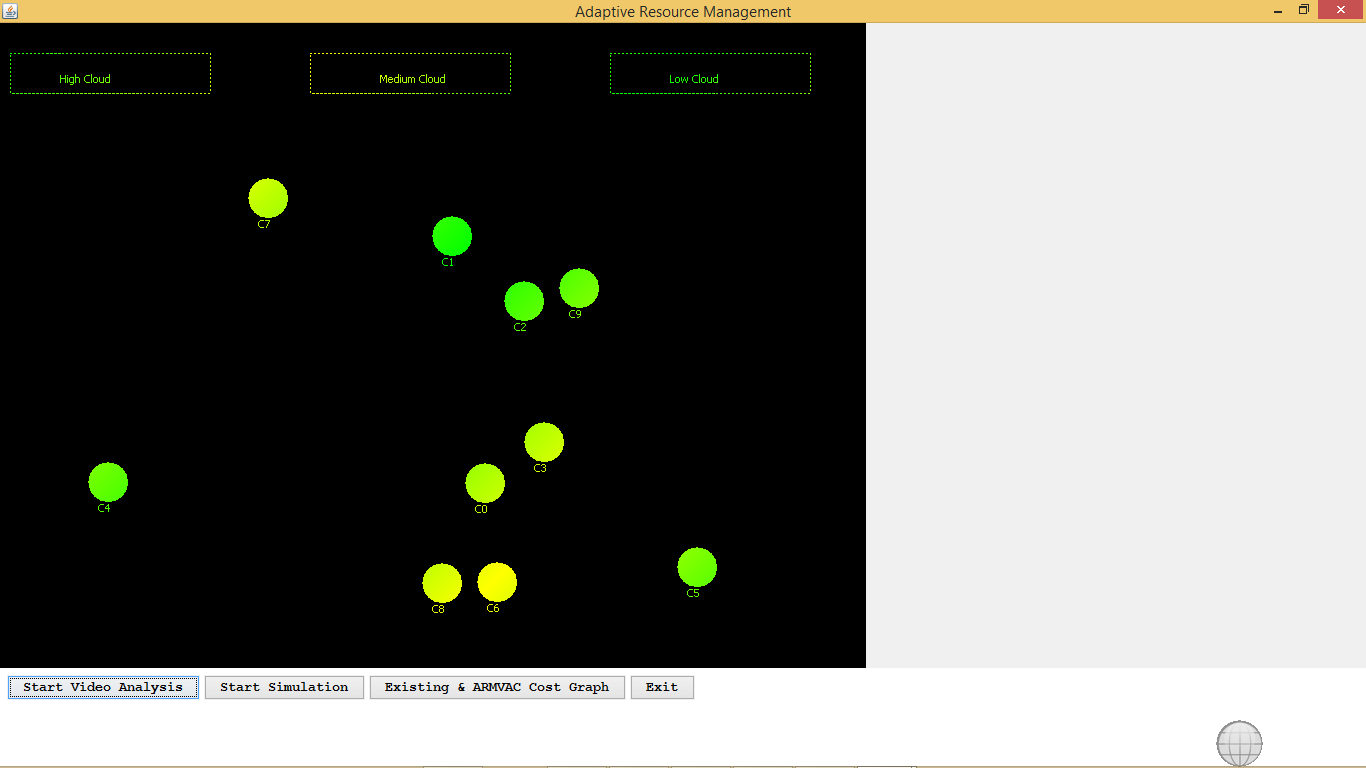
In above screen base on Frame requirements cloud instances and cost assign to each camera. Now to assign lowest cost cloud to each camera base on frame requirement and closeness of cloud click on ‘Calculate Low Cost Cloud Base on Frame Rate’ button to get below screen. In above screen we can see with existing technique for each hour cost is 5.2



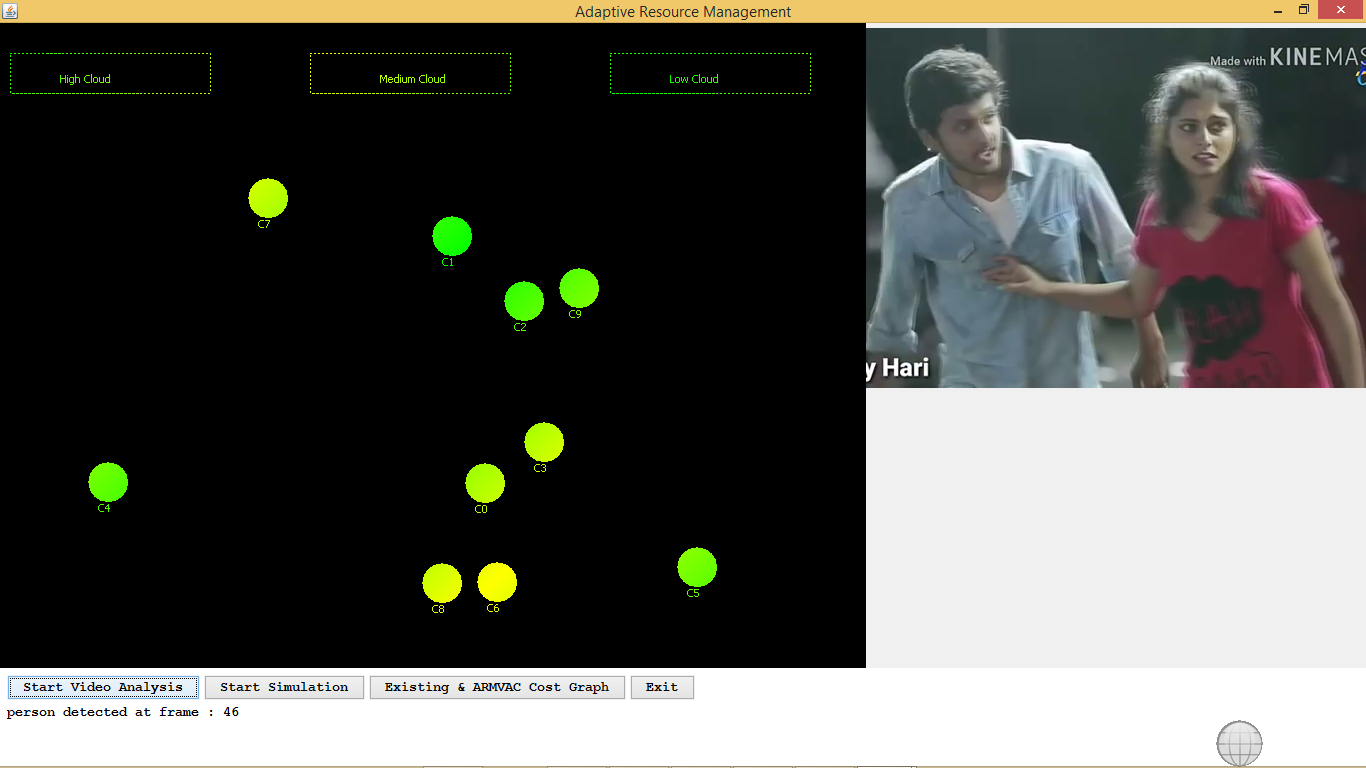
In above screen we allocate low cost cloud to camera and with propose technique we can see per hour required cost is 4.6$. So propose ARMVAC technique charge less compare to all existing technique. Now click on ‘Run Simulation’ button to upload video and to see simulation



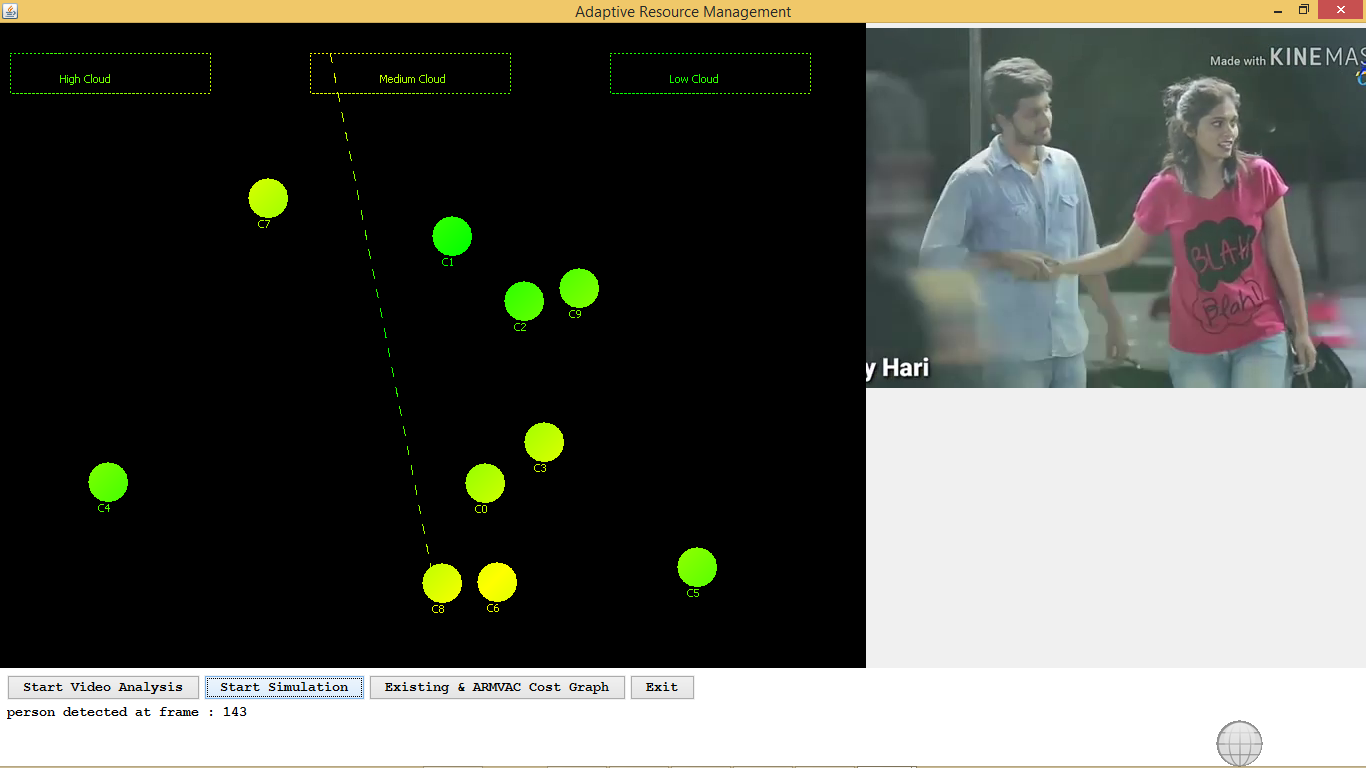
In above screen I uploading one video and after video upload will get below screen



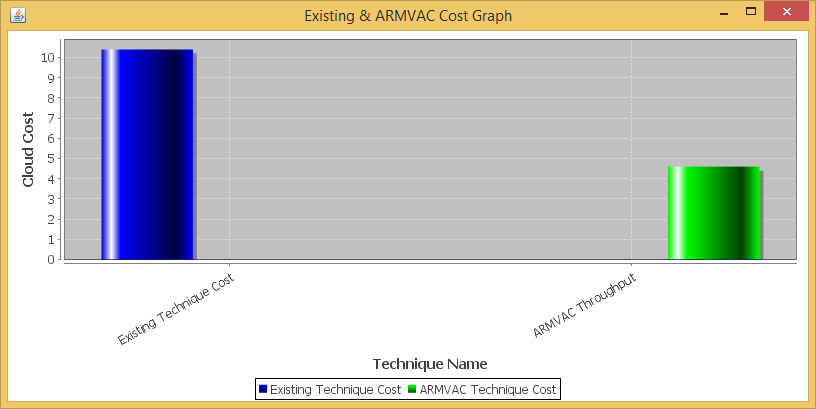
In above screen all circles represents cameras and click on ‘Start Video Analysis’ to play video and then click on ‘Start Simulation’ button to send that video frame to cloud instances. All rectangles represents as cloud instances.



In above screen we can see video is playing and in button we can see person detected at 46th frame. Now click on ‘Start Simulation’ button to start sending frames to cloud

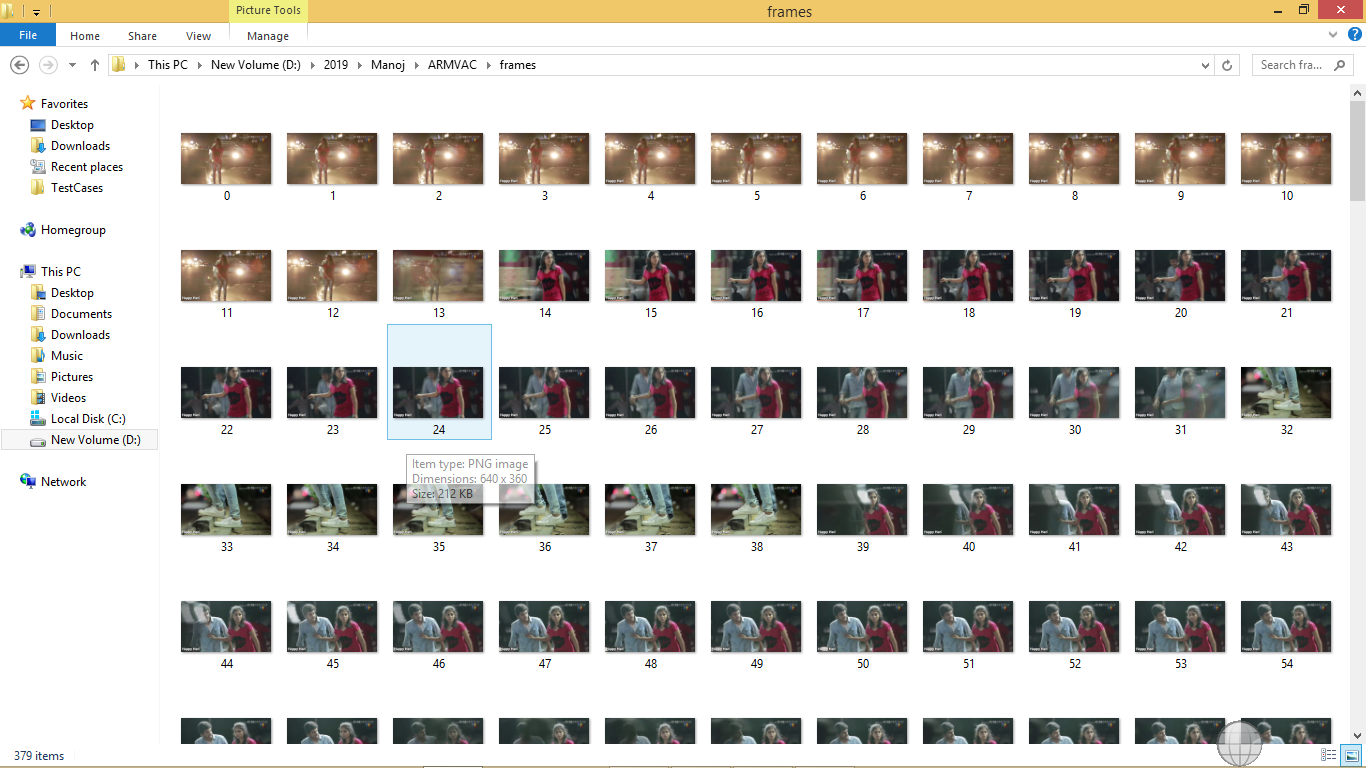


In above screen we can see camera is sending frame to nearest cloud instance. Now click on ‘Existing & ARMVAC Cost Graph’ button to view cost details in graph



In above graph x-axis represents technique name and y-axis represents cost taken by that technique and we can see ARMVAC took less cost.

All generated frames from video can be seen inside frames folder. See below screen



So by choosing nearest cloud instances with less cost we can adaptively use and manage resources of cloud