Architectural Support for Cloud Computing – Project report

Project description – Even if traffic doesn't always go smoothly, it can still be pretty beautiful to see cars navigating junctions and turning and stopping at traffic lights. This thought caused me to consider how crucial traffic movement is to human civilisation. After that, the nerd in me couldn't help but come up with a strategy to mimic traffic flow. I put a couple of weeks into a traffic flow project for my undergrad. I carefully considered each simulation method before settling on one. The primary goal of traffic simulation is to generate data outside of the real world. You can use a model run on software to anticipate traffic flow rather than trying novel approaches to managing traffic systems in the real world or gathering data using sensors. This facilitates quicker traffic system optimization and data collection. Real-world testing can be replaced by simulation, which is both cheaper and faster. Large datasets that might be challenging and expensive to obtain and process are necessary for training machine learning models. It is simple to adjust traffic simulation to generate the precise kind of data required.

Modules:

We must first mathematically model a transportation system before we can examine and optimize it. On the basis of input data (road network geometry, vehicles per minute, vehicle speed, etc.), such a model should accurately depict traffic flow.

According to the operating level, traffic system models are typically divided into three categories:

Microscopic models: depict each vehicle independently and try to mimic the actions of the drivers.

Macroscopic models: depict the overall movement of cars in terms of traffic density (number of vehicles per km) and traffic flow (vehicles per minute). They frequently resemble fluid flow.

Mesoscopic models: Modeling flow as "packets" of moving vehicles, mesoscopic models are hybrid models that combine aspects of microscopic and macroscopic models.

I'm going to use a microscopic model in this project.

Vehicle Generator

We have two choices for adding automobiles to our simulation:

Create a new instance of the Vehicle class and add it to the list of vehicles to manually add each vehicle to the simulation.

Stochastically add automobiles with predetermined probability.

We need to define a stochastic vehicle generator for the second choice.

Two restrictions serve as the definition of a stochastic vehicle generator:

Vehicle generation rate (vgr): (in vpm) specifies the average number of vehicles that should be added to the simulation each minute.

Vehicle configuration list(L): A list of tuples with a list item for each vehicle's configuration and probability.

L = [p1, v1, p2, v2, p3, v3,...]

Traffic Light

Placed at vertices, traffic signals have two distinct zones:

Slow down zone: A zone where cars reduce their top speed by employing the slow down factor is identified by a slow down distance and a slow down factor.

Stop zone: a space where cars stop that is identified by a stop distance. Through the use of a damping force.

Simulation

We'll take an object-oriented strategy. Every route and vehicle will have a class defined for them.

The following __init__ function will be utilized frequently in a variety of upcoming classes. Through the function set default config, the default configuration of the current class is set. and anticipates a dictionary, setting each entry's property as a property of the current class instance. In this manner, we can avoid worrying about future changes or modifying the __init__ procedures of various classes.

Data migrations strategies

Basic idea - We can create a flagged variable (migrate is the variable used in the implementation), which is a boolean variable. It can be set as “true” or “false” depending on the status of the job, if it needs migration or not.

When migrate = false, the process continues or the job is terminated. This is happens when there are no other available cars or if the data of the job is lost.

When migrate = true, then the car ID is changed to another car ID that is available.

- ✓ Before leaving the signal, the departing car can migrate the job to the data center controller (DC), and the DC chooses which car should be given the incomplete task that is being considered.
- ✓ The simplest approach is to assign the job that needs to be completed to the automobile that arrives closest to the departure time of the leaving car.

References:

1. https://www.researchgate.net/figure/Hypothetical-migration-plan-for-traffic-signal-control-system-in-the-area_fig4_260019992
2. https://vtechworks.lib.vt.edu/bitstream/handle/10919/34006/Ganta_S_T_2010.pdf?sequence=1&isAllowed=y
3. https://www.itsinternational.com/its8/feature/migrating-advanced-traffic-management-systems