



# Traffic Simulation on HPC platform

Yongqiang TIAN (City University of Hong Kong)  
Mentors: Kwai Wong (UTK), Cheng LIU (ORNL)



## Abstract

High performance computing platform is a powerful tool to help the scientists and engineers to model and solve complex problems. One potential large scale application is simulating traffic flow. Here, we aim to develop a prototype of traffic simulation software using RepastHPC. This project will be testing and compared with another famous micro-simulation software, TRANSIMS.

## Objective

There are several objectives:

- Using TRANSIMS to do some simple simulation.
- Developing the TranRepast using C++.
- Visualizing the output data using VisIt.
- Testing its performance, scalability and also efficiency.

## Developing environment

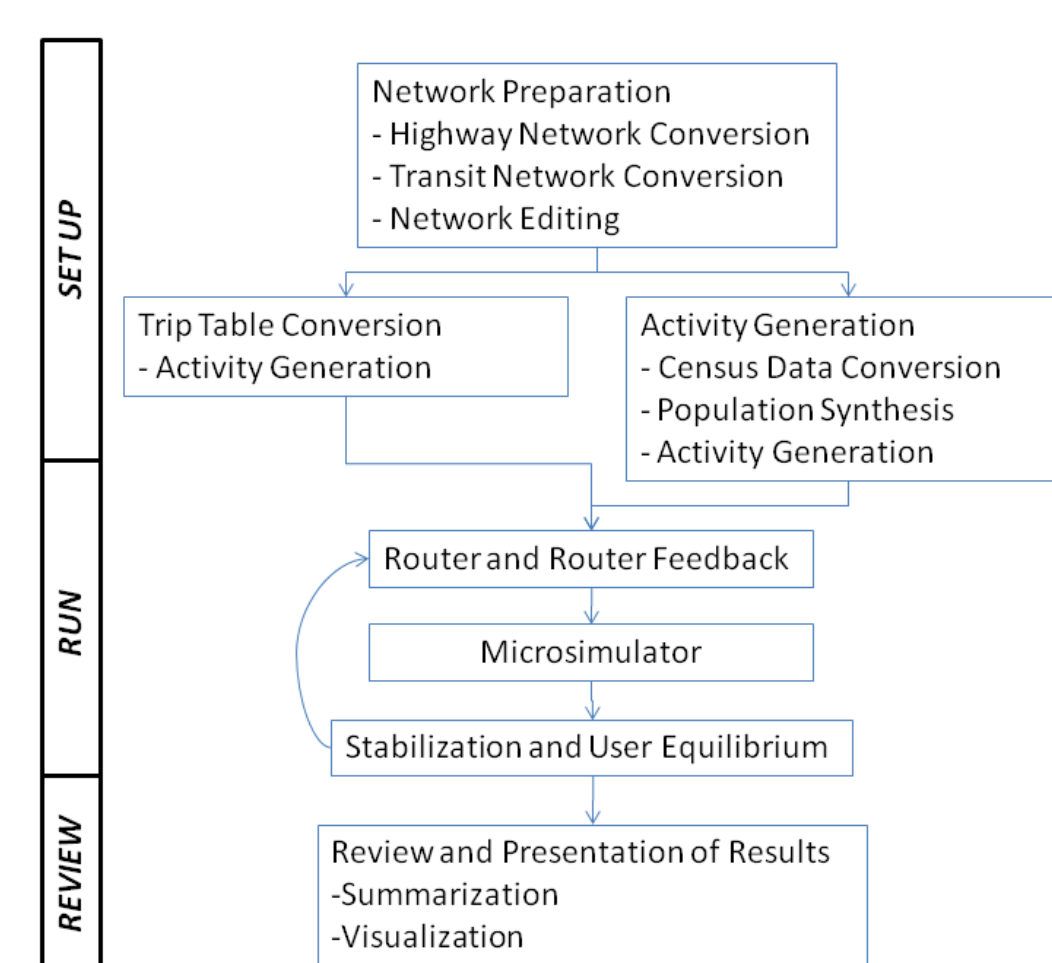
Hardware:

- csure1, a Linux (CentOS) machine for developing and testing in micro case.
- Beacon, an energy efficient cluster in NICS, will be used for large-scale testing.

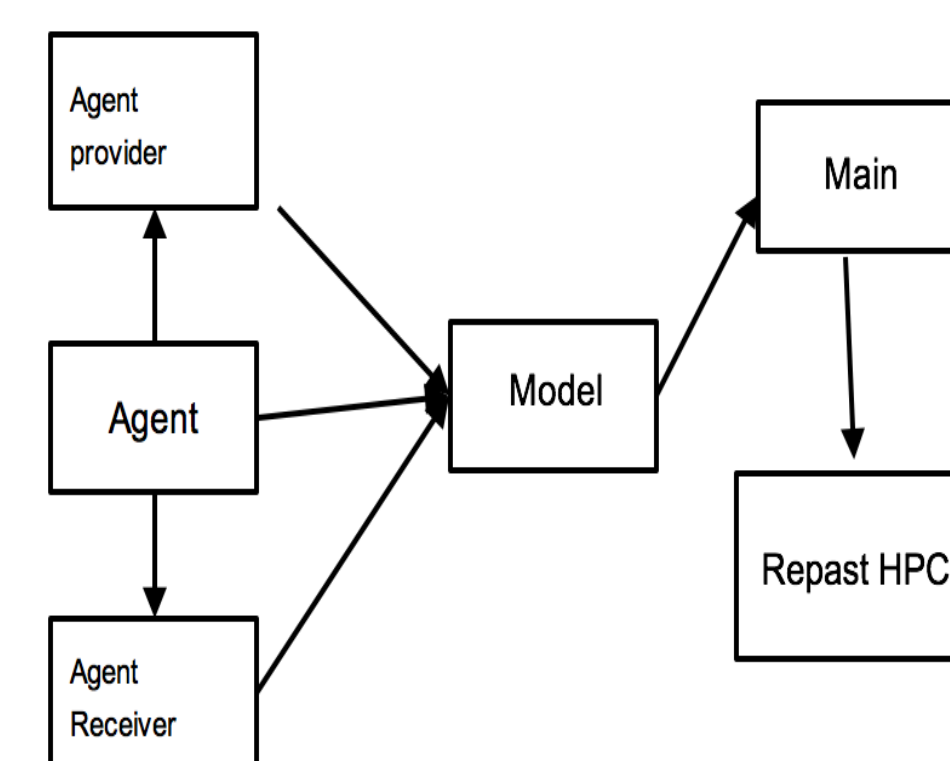
Software:

- Boost, MPICH, MPI

## Structure

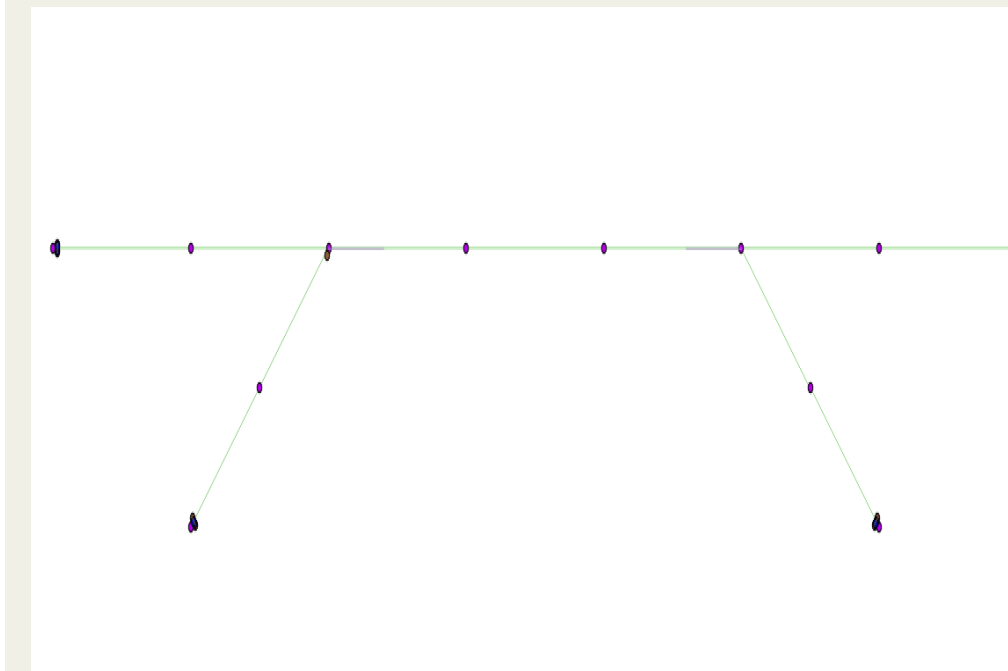


TRANSIMS

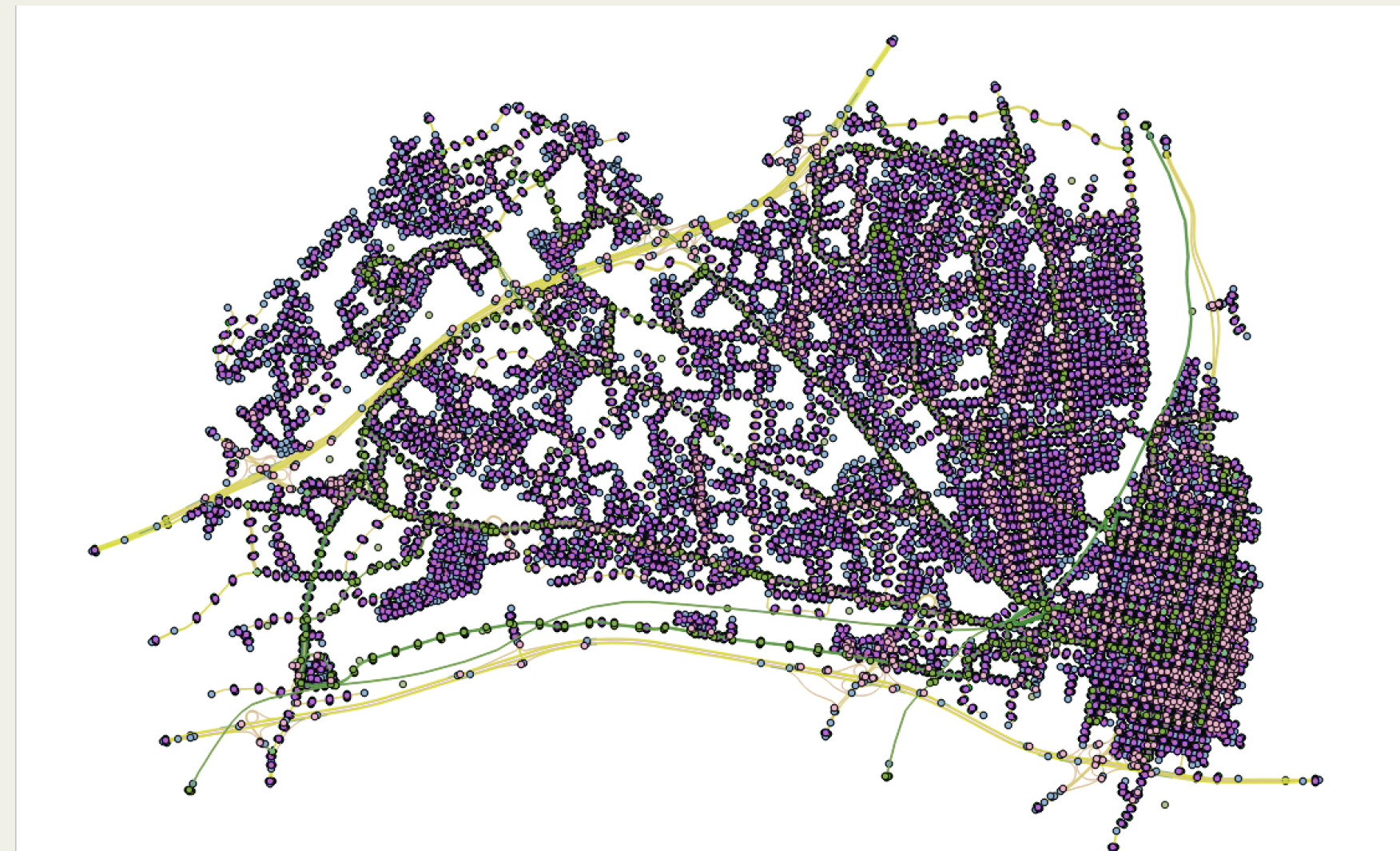


TranRepast

## Case Study



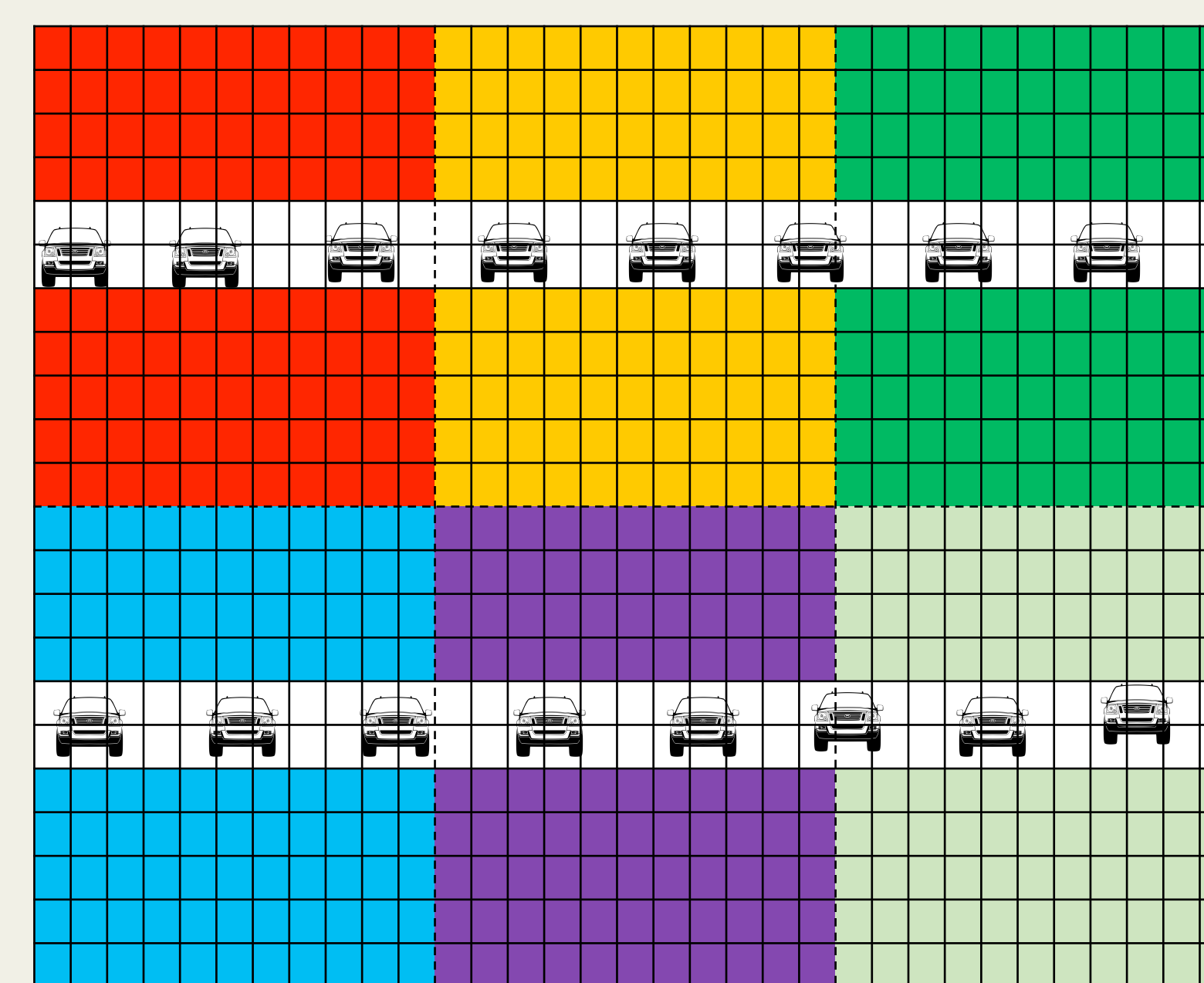
Tiny Case



Alexandria, VA

There are two cases used for case study. One is created by manual, which is for design and analysis. Another one is from Alexandria, VA. In each test case, there are several kinds of nodes. One is the entry or exit point for the vehicle, and another is the signal node, which can be considered as the traffic light in real world. The links between these nodes represent the road in reality. The output data will describe the time spent by each vehicle in its own trip and also its position at each moment.

## Model Used in RepastHPC



This grid map is the model used in RepastHPC. The map is divided into six regions. Each region will be assigned to a process and each process can only see and control its own region. The margin of the region is called buffer, which is shared by two processes. The white grids are roads and the grids at the most left are the entry points while the most right are the exit point. The vehicle will move in the road, step by step. Their speeds decide how many grids they can move in each step. The output data will describe the positions of these vehicles at each moment.

## Algorithm of TranRepast

1. Initialization (only once)
  1. Read configuration files, set environment variables.
  2. Create grid context.
  3. Initial agents, add these into the context.
2. For each agent:
  1. Get neighborhoods' positions
  2. Decide next behavior, including: move forward or change lane, accelerate or decelerate.
  3. Act according to the above decision. (after synchronization).
3. Update context:
  1. remove old agent
  2. add new agent
4. Save data to file and repeat step 2,3,4.

## Future work

There are quite a lot of work that can be done later.

- Supporting more kinds of vehicles.
- Supporting more kinds of situation, like crossroads, highways.
- Formatting input/output data so it can work with more professional traffic or geographic visualization software.
- Improve the performance, especially the performance on HPC

## References

Collier, N.T., and M.J. North, "Parallel agent-based simulation with Repast for High Performance Computing" in Simulation: Transactions of the Society for Modeling and Simulation International. Forthcoming.

Elefteriadou, L., AN INTRODUCTION TO TRAFFIC FLOW THEORY, 2014

## Acknowledgements

Support from City University of Hong Kong, Oak Ridge National Laboratory, Joint Institute for Computational Sciences and University of Tennessee are gratefully acknowledged.