Traffic simulation using Repast HPC Report

Yongqiang(Victor) TIAN
Email: yongqtian2-c@my.cityu.edu.hk

Jul-Aug 2016

Contents

1 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 begin with testing and understanding the another famous micor-simulation software, TRANSIMS[1]. After that, a simple prototype is developed based on Repast HPC[2].

2 Introduction

TRANSIMS is a famous micro transportation simulation software developed by Argonne national laboratory. It consists of around 50 individual modules. Besides the modules which are designed for doing the simulation step by step, some are in charge of assigning route for each vehicle, some focus converting data so that other professional software can use these data. This software has several version, however, the most common and stable one, version 4, is serial code. The performance and efficiency cannot satisfy the demand. Parallelization may be approach to solve this problem.

3 Trial in TRANSIMS

TRANSIMS has four version avaibale. Following is a table which showes their difference.

Version	Parallel	Platform	Test case	Document
4	NO	Linux and Windows	Yes	Yes
5	YES	Linux and Windows	No	Limited
6	YES	Windows	No	No
7	YES	Windows	No	NO

Version 6 and 7 are controlled by MS VS project, and they are easy to compile. However since they can only be run in Windows platform, while we hope to use Linux since most HPC platform are using Linux, we have to discard them. So we mainly focus on verion 5.

This is our original plan:

1. Using TRANSIMS, more precisely, the parallel version, to do the traffic flow microsimulation on Hong Kong geographic information.

2. Using ADEVS to do the mesosimulation based on TRANSIMS.

3. Using Repast HPC to develop a prototype for traffic simulation.

The first problem I faced is that it is really hard to build Version 5. The developer team of TRANSIMS using CMAKE to make sure it can be built in both Windows and Linux platform. This increases the difficulty to compile it. Although TRANSIMS version 5 has three options for building at last, which uses different parallel libraries, only one of them can be built successfully.

After that special version 5 is built, the next problem occurred is that there is not example data for me to verify it is really executable and it can give correct result. Some

example case from serial version is used to have a try, but later it is proved this way doesnt work. The difference between serial version and parallel version is not only in the way how the computation is executed, but also the I/O data format, and more important, the design and structure are different.

At last, TRANSIMS version 4 is compiled and tested. Compared with version 5, It is easy to complied and testing, after several bugs are found and fixed. There are two cases and some official documents for this version on its website. These two case, 'tinycase1' and 'alexandria' are tested. Following are their maps.

Figure 1: tinycase

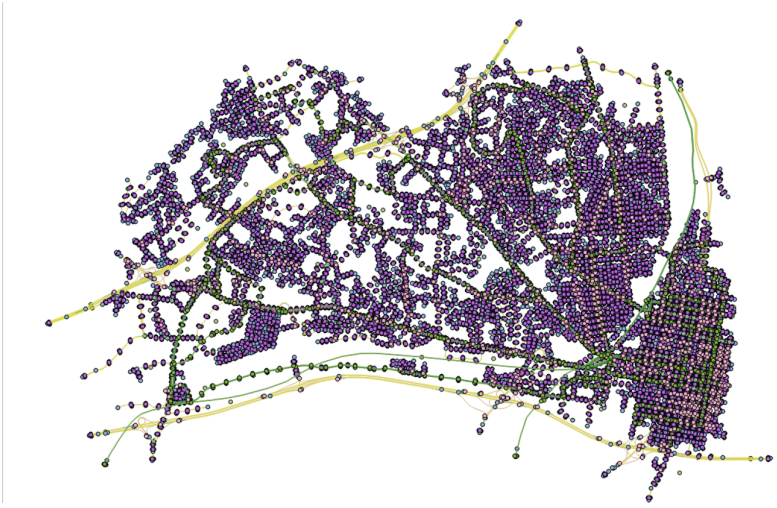

Figure 2: alexandria

Running the whole test case need around 45 mins. In fact, there are some ploblems in the test case. The method to fix them is recorded in the manual.

However, since the professor cannot make the HK geographic data ready by the end of this internship, we did not use it to simulate the traffic flow in HK. The phase one does not go well. The main reason is at the beginning of the project, we did not realize the difficulty and make a reasonable plan.

In fact, last update of TRANSIMS 4 is in 2014 and no one maintain it any more. As for TRANSIMS version 5, it looks like a beta version, the quality and quantity of its reference are quite limited. A lot work is done by trial and error. The TRANSIMS is really a profession and it is not possible for us to understand the whole code and debug during ten weeks.

After spending a round one month in TRANSIMS, we decided to stop trial in TRANSIMS. The phase two is also canceled since that meso-simulation is based on TRANSIMS. We decided to make every effort in Repast HPC.

4 Repast

Repast is an agent-based simulation platform, and it is designed for HPC platform. One important advantage of using it, instead of using Boost/MPI directly, is the Repast HPC will control the communication between the processes. Programmer only need to call the corresponding synchrony function and write the provide/receiver codes. Once an agent is moved to the boundary area of one process, it will be automatically shared by the nearby process. If agent is moved from two areas, which belong to different process, Repast HPC will automatically set the source and destination, send and receive the data. The developer can focus on programming in model. Repast HPC is not only for traffic simulation, but also for simulation in physics and biology.

After around 6 weeks work in Repast HPC, a prototype is completed. In this prototype, two main feature are implement, one is car following and other is lane changing. The typical test case used in developing consists of 200x300 grids, which is divided into 6 areas, controlled by 6 processes. There are two roads, each has 4 lanes. The vehicle will move from one side to another side. Some Octave scripts are also implements to help visualize the output data. If there is more time, more complex model we can use so that the simulation will much more close to reality.

Compared with TRANSIMS, developing on Repast HPC is much easier. Repast HPC also provides some tutorials and manuals, which gave us enough information for development, testing, and also debugging. Although some problems are encountered in these process, they are not as hard to solve as TRANSIMS, whose codes are much complex.

One thing we have to admit is that this prototype is really a prototype. TRANSIMS is developed by a professional team in Argonne National Laboratory, and it contains around 40 modules. Some of these modules are designed for networking preparation. Some are in charge of data formatting, like converting data from one software to the format that TRANSIMS prefers. Others are doing the work related to simulation. In TRANSIMS, each vehicle will have source, destination and preferred start time. Then one module called Router will assign the route for vehicle. After that, another module, Microsimulator, will do the simulation, step by step, second by second. The output will be used as feedback for Router. TRANSIMS is a not a single program, it is a set of programs. It is a professional software, and it is mentioned in a lot of academic papers in this transportation area.

The prototype we have, can only do the simulation, i.e., it is approximately equivalent to module Microsimulator. The meaning of this prototype is it is proved that it is possible to use Repast HPC do the traffic simulation on HPC platform. Since Repast HPC using parallel method while TRANSIMS does not, it is hopefully to reduce the simulation time

significantly. Furthermore, later we may use Repast HPC to do the simulation in other areas.

A manual of this prototype is finished and it explain the datail of this prototype.

5 Future work

We do not have enough time to do more in this internship, however, there are a lot future work can be done: 1. Add traffic light and traffic cross. A beta version is developed, where the traffic light is another kind of agent. 2. Add more features, like traffic flow mergence or separation. 3. Use the data format of TRANSIMS, so that we can use the module Router to route. 4. Compare the performance of Repast HPC vs TRANSIMS.

6 Conclusion

In conclusion, the main outcome of this internship is that we develop a prototype for traffic simulation using Repast HPC. At the same time, TRANSIMS is tested. Although it doesnt work well, it provides quite a lot useful information for developing the prototype. Hopefully, this prototype will be useful after more future is done.

Bibliography

[1] http://transims.googlecode.com,https://sourceforge.net/p/transims/code/HEAD/tree/.

[2] http://repast.sourceforge.net/hpc_tutorial/TOC.html.