Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher, eds

TOWARDS PERFORMANCE OPTIMIZATION OF X10-BASED AGENT SIMULATION PLATFORM WITH ADAPTIVE SYNCHRONIZATION METHOD

Toyotaro Suzumura[1,2,3] and Hiroki Kanezashi[2,3]
IBM Research – Tokyo[1]
Tokyo Institute of Technology[2]
JST CREST[3]

ABSTRACT

This paper describes highly scalable X10-based agent simulation platform called XAXIS. XAXIS is designed to handle millions or billions of agents on recent highly distributed and parallel computing environments with more than hundreds of CPU cores. To make the runtime scalable on such environments, we need to redesign and implement the simulation middleware. In this paper, we propose the software design, implementation on X10, one of the state-of-the-art PGAS language, and then application to large-scale traffic simulation. By using 192 CPU cores in distributed memory computing environment, the performance scalability is achieved with a traffic simulation. We also propose performance optimization approach that accelerates the simulation time by adaptively changing the simulation synchronization frequency while maintaining the simulation result accuracy. With the Tokyo road network data, from 3 to 4 fold speeds up are obtained with our proposed performance optimization.

1. X10 AND XAXIS– HIGHLY PRODUCTIVE PARALLEL AND DISTRIBUTED LANGUAGE AND SCALABLE X10-BASED AGENT SIMULATION MIDDLEWARE

We used X10 as the underlying programming language of our propose simulation platform, XAXIS. X10 is a novel parallel distributed programming language that IBM Research is developing as an open source project. It offers a global address space that is partitioned into multiple places, called Partitioned Global Address Space (PGAS). A place is the abstraction of the locality of memory, and it typically corresponds to one compute node with more than one CPU cores.

XAXIS (X10-based Agent eXecutive Infrastructure for Simulation) is a highly scalable multi-agent simulation middleware that enables application developers to make their application runnable on large-scale environment in a productive manner.

2. LARGE-SCALE TRAFFIC SIMULATION ON XAXIS AND PERFORMANCE EVALUATION

In the city, there is a demand to optimize the resources of human, cars, and energies. We need a large infrastructure that can meet the demand. The system that reduces wastes and predicts disasters or traffic jam can make a contribution to the whole humanity. In this meaning, IBM Research developed an agent-based traffic simulation modeling system called IBM Mega Traffic Simulator (Megaffic) to optimize a city traffic system as an important research in recent years.

A traffic simulation was conducted on XAXIS using the single node at TSUBAME 2.0 super-computer. The promising speed-ups are achieved with comparatively smaller steps. By making use of 12 threads in one node, around 5 fold speed-ups is achieved compared to using single thread. However with larger steps long simulations do not show such speed ups. That is because most of the CPU usage is used in the kernel level for waiting for incoming messages with busy wait, and not used for the computation if the number of trips per each time step is not sufficient.

978-1-4673-4781-5/12/$31.00 ©2012 IEEE

3. PERFORMANCE OPTIMIZATION WITH ADAPTIVE SYNCHRONIZATION CONTROL METHOD

The linear performance speedups are not obtained if the number of trips per 1 simulation step is relatively smaller since the CPU usage becomes quite low and most of the time is spent on the synchronization with other computing activities. We devised how synchronization at high frequency hinders the performance scalability in multi-node environment. Along other experiments, we understood that the less synchronization frequency we set at this simulations, the more speed up performance the result showed, and how synchronization hinders the performance scalability.

However, it is not still clear that the simulation precision is affected by its synchronization. It is highly important to achieve high performance without sacrificing the simulation precision as much as possible. Our overall experiment show that most of the road is not affected but precision loss is done by relatively few set of cross points as is followed by the power law. By computing the difference of vehicle density, the strong correlation between road congestion represented in absolute vehicle density and congestion in difference of vehicle density. Using this finding, we applied adaptive synchronization control method for optimization.

The optimization is done with the following two phases, congestion monitoring and computing adaptive synchronization frequency. First, to calculate vehicle density at this step, the congestion rate is calculated as the division of # of vehicles by each road length. The calculation is done for all the roads at each simulation step. Next, we sort all of congestions and find out the top 100 cross points per 1 place. Calculate density at this step as the total sum of the congestion rate of cross points within Top K cross points. Calculated values at each node are sent to the central server at Place 0, and it computes the appropriate synchronization frequency. This frequency is adjusted if it is smaller than 1 or larger than 100. The system can make the synchronization frequency longer in the case that a road network is not congested or occupied since we can not sacrifice the simulation precision that represents the number of vehicles on each road for each simulation time step. However when it is congested, the frequency should be high. The frequency is controlled by steps for synchronization so that the step is 1 when it is strongly needed when the road network is greatly congested.

After applying this adjusting system, we determine the appropriate synchronization frequency with one of the following three types. First method is only to adopt calculated value simply, even interval changes suddenly. Second one is to use calculated value and stored recent 20 values, calculate the average of values. Even the result of calculation about interval changes suddenly at synchronization, this method adjusts interval to the near recent values, swift changes will be suppressed Third one is an extension of the second type, controlling the result of road congestion. If the calculated congestion increases, sometimes the interval may decrease very much because interval was calculated dividing by square of congestion. We thought that is not effective to run simulation quickly, so this type is to calculate interval using suppressed congestion that is limited to the sum of previously obtained value and a constant number determined in each simulation.

We made some experiments about traffic simulation by changing synchronization interval (1, 10, 50, and 100 steps) and three types of methods to adjust interval. To observe changes of road congestion and synchronization interval, we used three patterns of how many vehicles are running at each steps. We used the road network of Tokyo and 16 nodes of TSUBAME. At the three adaptive methods, the value of interval is more stable at second and third methods than at first. The reason is these two methods use previous values. Simulation time at each step swiftly increased and decreased five times at two types. The reason is that simulations which change sync interval adaptive have to calculate and collect vehicle density from hundreds of cross points.

REFERENCES

Toyotaro Suzumura and Hiroki Kanezashi, "Highly Scalable X10-based Agent Simulation Platform and its Application to Large-scale Traffic Simulation", 2012 IEEE/ACM 16th International Symposium on Distributed Simulation and Real Time Applications, Dublin, Ireland, 2012/10