論文の内容の要旨

Thesis Summary

論文題目

HPC and feature enhancements of micro- and macroscopic traffic simulators for disaster management applications

(災害対応における HPC 活用のための微視的・巨視的な交通シミュレータの機能向上)

氏 名

ペットプラッコ　ワスワット (Wasuwat Petprakob)

本文

In earthquake disaster mitigation, traffic flow simulations have two major applications of significance. The first is to find near-optimal traffic assignment such that the functioning portion of a road network, which is damaged by a major earthquake, can be optimally utilized to minimize economic losses and even plan the recovery sequence of the damaged road segments. The second is high fidelity traffic simulations for virtually mimicking mass evacuation scenarios with the aim of quantitatively evaluating various strategies to accelerate the evacuation processes.

Though both these applications are of significant importance, there lacks HPC capable software to solve real-life problems involving large regions. According to my literature survey, one iteration of optimal traffic assignment such as day-to-day traffic assignment took 1 hour and 30 minutes for New York's road network. At this speed, it could take months to find an acceptable traffic assignment of a large-scale road network, since the optimal traffic assignment may take several thousand iterations to converge to an acceptable traffic assignment. In earthquake disaster mitigation, traffic assignment plans have to be updated at least weekly to best utilize the latest repaired roads. This large gap between the required and the available times-to-solutions emphasizes the need of the development HPC enhanced software to find near-optimal solutions within at least several days period. When it comes to microscopic modeling of emergency evacuations, there are a number of commercial products targetting small-scale applications like fire evacuation of buildings, etc. These cannot be utilized to simulate emergency tsunami evacuations which involve hundreds of thousands of people, on foot and vehicles, over several hundreds of square kilometers. Though a Multi-Agent Simulator (MAS) developed by a group at ERI is capable of such large-scale simulations utilizing HPC, its traffic simulation module lack of essential features like controlling vehicles on multi-lanes and at junctions, and requires further HPC enhancements to scale to larger regions like those affected by Tokai, Tonankai, and Nanaki mega-thrust earthquakes.

The objectives of this research are to develop HPC enhanced systems to address the lack of traffic simulators for the above-mentioned applications by:

1. Developing a parallel computing extension for a macroscopic traffic simulator with the aim of finding near-optimal traffic assignments within a reasonably short time.

2. Implementing lightweight junction model and enhancing the performance of a microscopic agent-based simulator for applications in mass evacuations.

Major earthquakes, like the impending Tokai, Tonankai and Nankai earthquakes, are predicted to inflict serious damages to the infrastructures of the industrial heartland of Japan and inundate the coasts with destructive tsunamis. Damages to the lifeline networks during a major earthquake can bring long-lasting disruptions to manufacturing and other economic activities leading to a secondary disaster. Especially when a commercial center like Tokyo is affected by a major earthquake, the resulting secondary disaster can bring serious risk to the nation's economy and even send ripples in the global economy. It is hard to say which lifeline network plays the most critical role since the industries and other economic activities depend on these in a rather complicated manner. Road network is one of the vital elements with a significant influence on the economic activities.

According to historic earthquakes, the repair time for road networks after major earthquakes can be more than a year; repair time after the 1995 Kobe earthquakes was 21 months. By optimally utilizing the functioning portions of the road network to meet the traffic demand during this long period, the degree of secondary economic disasters can be minimized or even eliminated. Finding optimal allocation of traffic is a challenging problem, which is well known to be an NP-hard (i.e. algorithms involve non-deterministic polynomial hardness).

Hence it is essential to choose a suitable algorithm and develop efficient high-performance computing extensions to find near-optimal solutions for this post-disaster traffic assignment problem. There exists a number of methods to find near-optimal traffic assignment. However, according to our literature survey, none of these methods have been applied to solve large-scale problems like Tokyo probably due to the extensive computational demand. This emphasizes the need of choosing a computationally light algorithm and high-performance computing. The use of HPC is especially necessary after a major earthquake disaster since the traffic allocation has to be updated, at least weekly, to best utilize the latest repaired roads.

Out of the many methods for near-optimal traffic assignment, in this study, we use the method of day-to-day traffic assignment, which mimics how people find shorter routes by changing their routes according to experiences on previous days. As one would easily guess, this method replans the route of a random subset of vehicles according to travel time information of the previous simulation, and estimate the resulting travel time by simulating the traffic after the route replanning. The above process is repeated until the total delay time reaches a certain convergence criterion. The advantage of this method is that the total computation time can be reduced by using a lightweight macroscopic traffic simulator.

Day-to-day traffic assignment involves traffic flow simulations, and it usually requires a large number of traffic flow simulations to obtain a converged solution by mimicking drivers' behavior. Therefore, a lightweight link transmission model based traffic simulator is used in this study as a strategy to lower the total computation time. In link transmission simulators, each road segment stores information of every passing through vehicles. Each element in the links' vehicle list stores information pertaining to a vehicle such as a vehicle id, entry time to the link and departure time from the link.

To develop distributed memory parallel day-to-day algorithm with link transmission model (LTM), the workload is distributed among CPU cores of a cluster of computers connected via a dedicated high-speed network. The necessary information to make the CPUs to collectively solve the problem are exchanged among the CPU cores using Message Passing Interface (MPI). The basic processing unit of our current implementation is a CPU core since we use only MPI.

In order to distribute the computational workload of Day-To-Day Traffic Assignment (D2DTA), the bi-directional graph of the traffic network is decomposed into non-overlapping and continuous sub-networks using the graph partitioning software METIS. The time asynchronous nature of LTM (i.e. clocks of vehicles are not synchronized) makes it difficult to make a reasonable estimation of the amount of computations in each partition, within next several iterations. Devoid of better solutions, we use link-based partitioning scheme with a number of passing vehicles as weight.

Each of the partitions is assigned to a separate MPI process (i.e. CPU core) so that computation power of multiple MPI processes can be utilized to solve the problem in a shorter time. In this particular MPI-only setting we use, even if some of the CPU cores are located on the same motherboard sharing all the resources, each CPU core behaves as an independent MPI process with its own private memory space and resources. Therefore, to maintain the continuity of the original problem, each MPI process should exchange state of the upstream links located along partition boundary, which deliver vehicles to neighbor partitions, sending explicit messages to the owner MPI processes of its neighbor partitions. This extra overhead of communication between partitions, which does not exist in serial version, significantly increases with the number of MPI processes involved, and hence has to be minimized to efficiently utilize a larger number of CPUs to solve the problem in shorter time. Communication hiding (i.e. doing some useful computation while communication is progressing) is the main strategy we utilize to reduce this extra communication overhead.

The computation time of a perfect scalable parallel program halves when the number of CPUs doubles. Attaining high scalability is a challenging task and requires significant modifications to the data structures and serial algorithms, or it may even require the development of completely new algorithms.

Three main improvements are implemented to reduce the computation time. The Distributed memory pathfinding algorithm in the original code was a major bottleneck since pathfinding is an inherently serial algorithm. Replacing it with an embarrassingly parallel model, near perfect scalability could be attained for pathfinding, and reduce the computation time for pathfinding by 200 folds. An additional advantage of this embarrassingly parallel pathfinding eliminated the time-consuming parallel vehicle loading step. The time asynchronous nature and presence of links without any traffic found to waste CPU cycles in fetching unnecessary link data from main memory. Executing only the active links, not only this large waste of CPU cycles was eliminated but also improved load balance among CPUs. Compared to the small amount of time required to prepare the list of active links, this was found to reduce the time for link transmission model by 20 times. The third improvement is the development of an algorithm for finding a weak ordering for execution to nearly maximizing the vehicle flow rate of the links in each partition, at a given iteration step. Finding optimal execution order of links to maximize vehicle flow rate is an NP-hard problem and impossible for very large networks. However, it is found that certain ordering of links in each partition significantly increases the flow rate, reducing the total execution time up to 40%. According to our literature survey, this is a novel contribution which is applicable to distributed parallel computation of link transmission models in traffic engineering, etc.

Although tsunamis also destroy infrastructures, the large losses of lives are the major concern for coastal cities located in close proximity to subduction zones. Tsunami is expected to arrive at some coastal cities within a few tens of minutes, which is too short, especially for elderly coastal communities, to evacuate to a safe high ground. Allowing to use cars for evacuation is one of the cost-effective means of accelerating evacuation. According to the lessons from the past major tsunamis, the unconstrained use of cars can produce worse outcomes. Extensive studies are necessary to answer questions like, what percentage of people can be allowed to use cars, what are the safe time windows for each region to safely use cars, how to cope with unexpected road damages, how stable is car usage, etc. Such studies will contribute to finding safe strategies to make the evacuation process faster, saving many lives. Considering the number of lives involved and the presence of an unusually large number of pedestrians, which is the recommended mode of evacuation, a mass evacuation simulator with microscopic models of vehicles, pedestrians, and their interactions are required to find answers to questions like the above. Most of the large-scale traffic simulators does not include microscopic models of vehicles and pedestrians, while the existing large-scale agent-based microscopic simulators do not provide junction and lane changing models with accurate speed profiles to model interactions among pedestrians and vehicles.

Therefore, we improved the domain generator to improve the quality of a topological graph which improves the result of the simulation since the agents can make their decision more precisely. Because of enhanced environments, it enables us to apply high fidelity intersection model. Hence, we implement the traffic light control at the intersection and we formulate the high-fidelity intersection behavior of vehicle agents. The formulated intersection behavior does not depend on an optimization scheme. Therefore, it is theoretically faster than optimization-based collision avoidance algorithm. The travel time of free flow vehicle is almost matched with observation result. Further, we can add more details. For example, aggressive drivers may be included in the simulation so that we can simulate the realistic situation.