Partitioning of Urban Transportation Networks Utilizing Real-World Traffic Parameters for Distributed Simulation in SUMO

Md Salman Ahmed, Mohammad A. Hoque
Department of Computing
East Tennessee State University
{ahmedm, hoquem}@etsu.edu

***Abstract*—This paper describes a partitioning algorithm for real-world transportation networks incorporating previously unaccounted parameters like signalized traffic intersection, road segment length, traffic density, number of lanes and inter-partition communication overhead due to the migration of vehicles from one partition to another. We also describe our hypothetical framework for distributed simulation of the partitioned road network on SUMO, where a master controller is currently under development using TraCI APIs and MPI library to coordinate the parallel simulation and synchronization between the sub-networks generated by our proposed algorithm.**

Keywords—OSM, Network Partition, METIS, SUMO, TraCI, MPI, Parallel Simulation

I. Introduction

For parallel network simulation, network partitioning is an effective method for speeding up the simulation process as well as maintaining the compatibility with machines with low resources that can run each partition. Since the simulation time and memory usage exponentially increase with the network size (number of vehicles and traffic volume), efficient network partitioning can greatly improve the scalability of parallel simulation. However, network partitioning is proven to be an NP-hard problem. Hence, an optimal partitioning may not be feasible. Practical partitioning heuristics are required that account for road networks and vehicular density and mobility to ensure an even division of the workload while minimizing communication between the partitioned elements.

While effective partitioning is crucial for speeding up the simulation of large-scale transportation network, this partitioning is very challenging due to many reasons. First, with connected vehicles emerging on the roads, the partitions could not be fully separated. In fact, due to communication and high mobility, partitions may have high level of interdependency and interactivity (i.e. a message or a vehicle moves from one partition to another) that demands communication between partitions to achieve consistency and accuracy. Second, inefficient partitioning of such networks can produce high communication volume between the different partitions, and imbalanced computing load in each partition, consequently results in low simulation speeds. So, it is necessary to create partitions in such a way that reduces the interactivity and interdependence between them. Thirdly, the synchronization and partitioning equity. Due to the interdependency between events in different partitions, simulation should be synchronized in all the partitions, i.e. low load (high speed) partitions must wait for high load (low speed) ones to finish. This means that the maximum overall simulation speed is limited to the minimum speed among all the partitions. Thus, the best speed is achieved when partitions have approximately equal loads. In fact, solutions for these three reasons may contradict one another i.e. creating independent partitions may result in huge load differences that can eventually degrade the speed. It is an optimization problem between a set of tradeoffs such as number of partitions, result accuracy, simulation speed, memory requirements etc.

In this context, the transportation network information such as road network (road links, road nodes), vehicle density on each link, vehicle speeds and distribution can be effectively utilized to optimize the partitioning techniques. For example, the vehicle density and the length of each links can be employed as link weight in partitioning techniques (such as minimum cut or minimum k-cut algorithms) to partition the network and minimize the interactivity between different portions. The lower the density and the longer length for a link, the higher the probability of being a cut link in the network. The rationale is that the density and length represents the continuity of the communication route on this link. Therefore, the lower this ratio (density/length) the lower the communication between the ends of the link.

Our current research contributions include the development of a novel partitioning algorithm for large scale urban transportation network incorporating previously unaccounted parameters like traffic volumes, signalized intersections, number of lanes, length of links etc. to balance the load for distributed simulation using SUMO. This would allow the large-scale evaluation of any innovative connected vehicle application or algorithm in a cluster-computing environment.

The rest of the paper is organized as follows. Section II describes the existing work on partitioning of transportation network. In section III, we identify the important parameters needed for partitioning. Section IV describes the actual steps involved in our proposed partitioning scheme with some preliminary results followed by a high-level overview of the work-in-progress distributed simulation platform on SUMO that can simulate individual partitions in parallel. Finally, we conclude with our future work leading to the development of distributed simulation platform enabling the simulation of large scale urban transportation network with connected vehicles.

II. Related Work

Many researchers have attempted to develop efficient partitioning schemes for large-scale transportation networks to simulate the scenarios in distributed environments using clusters. A well-designed partitioning scheme can greatly reduce the number of inter-process communication because vehicles frequently move from one partition to another in which case all the information corresponding to the mobility of those migrating vehicles need to be transferred to the new partition. Johnson *et. al.* [1] generated partitions using the shortest distance domain decomposition algorithm utilizing the standard label correcting technique with the objective of minimization of system boundary nodes to reduce inter partition communication cost. A significant amount of research effort has been dedicated for load-balancing among the partitions. For example, Meshkat *et. al.* [2] used genetic algorithm to divide a road network into two equally balanced partitions and repeated the process recursively to further divide the two generated partitions. Hyper-graph based partitioning algorithms using hmetis [5] have been discussed in [3] and [4], considering two-heuristics based hypothetical partitioning techniques. However, all the above partitioning techniques lack of the context of real transportation road networks—traffic density, number of lanes.

A complete road map for parallel road traffic simulator is discussed in [6] and [7]. In [6], the authors provide their own road network partitioning scheme and distributed version of SUMO. In [7], the transportation networks are partitioned by spatial decomposition [8] and simulated using JUTS, TRANSIMS, and AIMUSN. However, the former lacks of the parameters of actual road networks that affect the partition significantly. The later one considers only grid like road network. MOVES [9] also provides a distributed simulation platform on top of ARTIS simulation software. MOVES focuses on mainly the modularity and integrity of its layered software architecture, but does not focus on the real-world transportation networks and partitioning techniques. The distributed versions of SUMO are also discussed in [10] and [11]. The authors discuss about border edge management of a partition in [10] whereas in [11], the authors focus on the implementation of the distributed version in clusters. In both cases, the authors assumed that the network is already partitioned.

III. Issues for Road Network Partitioning

We have identified the following issues and parameters that are crucial for consideration while designing a heuristic for the partitioning of urban transportation network for parallel simulation.

A. System boundary nodes of each partition

System boundary nodes of a partition are responsible for communicating with other partitions to transfer and receive data and control of vehicles. The system boundary nodes pack several transfer requests and transfer the packed request to other partitions. So, the minimum number of system boundary nodes in a partition ensures the separation of responsibility and low communication cost.

B. The number of partitions

Deciding the number of partitions in the transportation network is a crucial factor for balancing the loads and minimizing the communication cost. Almost every graph partitioning algorithm determines the partitioning based on a pre-specified number of partitions which may not always generate the optimal solution in practice. Instead of specifying an exact number of partitions, an upper bound and lower bound can be provided as input to the algorithm to determine the best partitioning solution within the specified range.

C. Load balancing

As mentioned before, load balancing issue has been studied extensively for network partitioning since this directly impacts the overall simulation time. However, the metrics considered for load-balancing are not sufficient from the context of real transportation networks involving variable traffic densities and lane distributions. Hence, the weights for the nodes and links should be carefully assigned to address this issue.

D. Intersection cut

If an intersection is kept in a partition for the sake of one high density road and left all the links incident to the intersection in other partitions, it introduces a huge communication overhead. In other words, if an intersection is considered as a boundary node for a partition, then a significant amount of vehicle mobility data must be communicated to and from each partition that contains the intersection as a boundary node due to large number of vehicles migrating from one partition to another. In this context, an important factor—whether to prioritize signalized intersection over un-signalized intersection as a candidate for boundary node—remains open for further research.

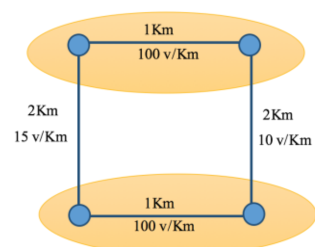

Fig. 1: Sample partitioning illustrating link cut minimizing inter-partition information exchange

E. Link/Edge cut

When a link or edge is selected to be cut then the traffic volume along the cut link is directly proportional to the amount of information exchanged between the two partitions along the link. In this case a good strategy would be to choose the links with minimum traffic for cut to reduce the communication overhead between partitions. For example, the road network in Fig. 1 shows four links with the lengths and average car densities. These two partitions have the minimal interaction between them due to the lower traffic densities (10 vehicles/km and 15 vehicles/km), thus their discrete simulation events can safely run in parallel.

IV. Proposed Partitioning Approach

Below we describe the steps involved in our partitioning scheme along with some preliminary results obtained for the road network of Johnson City, TN.

A. Creating graph

To create the graph, the OSM file of Johnson City, TN is downloaded from the openstreet.org website. A python script was written for extracting the intersections, road segments, traffic signals, and number of lanes. Since a road segment or a road is a combination of two or more nodes in OSM file, the degree of all nodes is calculated to find out the intersections. To keep the graph clean, many road types such as living street, service path, foot way, cycle-way, motorway and unclassified roads are excluded from the graph. The nodes that have only one degree (e.g. dead end) is also excluded from the graph. The Fig. 2 depicts the generated graph of Johnson City, TN where Google map API is used to overlay the graph vertices and edges.

Fig. 2: Graph of Johnson City, TN is generated using OSM file and overlaid on the Google Map

B. Generating graph matrices

Each node or vertex of the generated graph has the latitude and longitude values along with a unique number identifier assigned by OSM file data structure. Along with the latitude, longitude, and the node identifier—an index ranged from 1 through $|V|$, where $|V|$ is the number of vertices in the graph—is assigned to each vertex. The length of the links between nodes are calculated using the Haversine formula that takes the latitude and longitude of two nodes and returns the distance between them. The following equation was used for calculation of link lengths based on Haversine formula.

$$d = 2r \sin^{-1}\left(\sqrt{\sin^2\left(\frac{\varphi_2-\varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)}\right)$$

where,
d=Distance between two points/nodes
r=Radius of Earth (6367 km)
φ_1=Latitude of point 1
φ_2=Latitude of point 2
λ_1= Longitude of point 1
λ_2= Longitude of point 2

Table 1: List of parameters considered for partitioning

Parameter Name	***Extraction Technique***
Node weight	All signalized and un-signalized intersections in the OSM data are identified using the above-mentioned python program. An un-signalized intersection is assigned a weight by multiplying its degree with the average of incoming and outgoing link densities. A signalized intersection is assigned a higher weight than un-signalized intersections.
Link length	The length between two nodes is calculated using the Haversine method.
Number of lanes	The number of lanes of a road segment or link is extracted from the OSM data.
Link density	The density of a road segment or link is extracted from the Google Map Application's newly introduced traffic layer [13]. The traffic volume is sampled in each of the 24 hours in a day and calculated the average density. For simplicity, the density is expressed in three categories: low, medium, and high.
Link priority	The road segment is assigned the summation of link length, the number of lanes, and link density as the priority.

The above table (Table 1) shows all the parameters that have been extracted from the OSM data to generate a weighted graph.

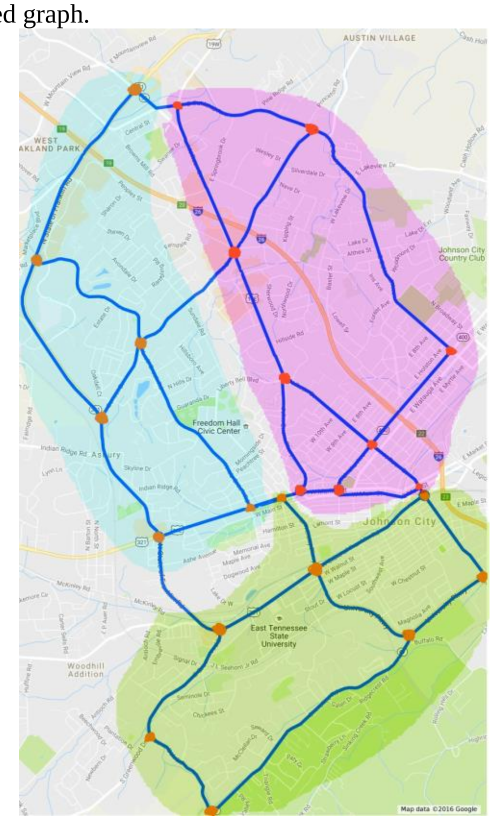

Fig. 3: Road network partitioning of Johnson City, TN

C. Partitioning the graph by METIS

Since METIS [5] is the most stable unstructured graph partitioning package, we partition the generated graph using METIS. The input data for METIS is provided using the generated graph and weight parameters. Since METIS only supports the node and link weight, a node and link weighted graph is generated. METIS performs the partition of a graph in three phases: the coarsening, partitioning, and uncoarsening phase. In coarsening phase, the heavy edge matching scheme is used, whereas in the uncoarsening phase, the Kernighan-Lin graph refinement algorithm is used. The coarsest graph is bisected using graph growing followed by boundary Kernighan-Lin algorithm with graph partitioning using recursive bisection technique.

Fig. 3 shows a sample partitioning of the road network of Johnson City, TN considering the parameters as described in the previous section. For simplicity, here we have only provided the multi-lane signalized corridors as the input road network to the METIS-based partitioning algorithm.

V. DISTRIBUTED SIMULATION USING SUMO

A distributed simulation platform on SUMO is currently under development that simulates each partition of the graph in a separate processor node. A master program is responsible for starting the simulation in all partitions and synchronizing the simulation results. The master program is written in C++ using the Traffic Control Interface (TraCI) [12] and Message Passing Interface (MPI) libraries. The communication and synchronization between processors are done using MPI. Each processor node has also the information of the complete graph information along with its own partition information. The SUMO input and configuration files are generated dynamically for each partition. The master program communicates and starts the SUMO simulator using TraCI which is packaged along with the SUMO source tree. SUMO simulator can be operated as a server. TraCI is performed as a middle-ware between the master program and SUMO where the TraCI is connected with SUMO as a client. Vehicles and routes are dynamically created by the master program and added to the SUMO simulator using TraCI. The routes are calculated from a source node to a destination node using the Dijkstra's algorithm. When a vehicle leaves a boundary node of a partition, the master program determines the next partition the vehicle will enter, removes the vehicle from current partition, and passes the whole vehicular dynamics of the vehicle to the entering partition. The master program also tracks the time needed to transfer the vehicle and its dynamics to the new partition.

VI. CONCLUSION

In this paper, we proposed our network partitioning approach for large-scale transportation network considering some important parameters like signalized traffic intersection, road segment length, traffic density, number of lanes and inter-partition communication overhead. Most of these factors were not accounted for in earlier work. We also discussed the critical issues involved in partitioning of a typical road network. Finally, we described our hypothetical framework for distributed simulation of the partitioned road network on SUMO, where a master controller is being developed using TraCI APIs and MPI library to coordinate the parallel simulation and synchronization between the partitions generated by our current algorithm. OUR FUTURE WORK INCLUDES INCORPORATING all the identified weight parameters in tHE GRAPH PARTITIONING TECHNIQUE BY MODIFYING THE FOUR ALGORITHMS) used in METIS (heavy edge matching, Kernighan-Lin graph refinement, graph growing followed by boundary Kernighan-Lin, and recursive bisection) to meet the needs of real-world transportation network.

REFERENCES

[1] Johnson, Paul, Duc Nguyen, and ManWo Ng. "Large-scale network partitioning for decentralized traffic management and other transportation applications." Journal of Intelligent Transportation Systems (2016): 1-13.

[2] Meshkat, Amir, and J. L. M. Vrancken. "Multi-Objective Road Network Partitioning." Procedia-Social and Behavioral Science (2014).

[3] Xu, Yan, and Gary Tan. "An offline road network partitioning solution in distributed transportation simulation." In Distributed Simulation and Real Time Applications (DS-RT), 2012 IEEE/ACM 16th International Symposium on, pp. 210-217. IEEE, 2012.

[4] Etemadnia, Hamideh, and Khaled Abdelghany. "On the Network Partitioning of Large Urban Transportation Networks."

[5] Karypis, George, and Vipin Kumar. "METIS--unstructured graph partitioning and sparse matrix ordering system, version 2.0." (1995).

[6] Ventresque, Anthony, Quentin Bragard, Elvis S. Liu, Dawid Nowak, Liam Murphy, Georgios Theodoropoulos, and Qi Liu. "SParTSim: a space partitioning guided by road network for distributed traffic simulations." In Proceedings of the 2012 IEEE/ACM 16th international symposium on distributed simulation and real time applications, pp. 202-209. IEEE Computer Society, 2012.

[7] Potuzak, Tomas. "Distributed-parallel road traffic simulator for clusters of multi-core computers." In Proceedings of the 2012 IEEE/ACM 16th International Symposium on Distributed Simulation and Real Time Applications, pp. 195-201. IEEE Computer Society, 2012.

[8] Nagel, Kai, and Marcus Rickert. "Parallel implementation of the TRANSIMS micro-simulation." Parallel Computing 27, no. 12 (2001): 1611-1639.

[9] Bononi, Luciano, Marco Di Felice, Gabriele D'Angelo, Michele Bracuto, and Lorenzo Donatiello. "MoVES: A framework for parallel and distributed simulation of wireless vehicular ad hoc networks." Computer Networks 52, no. 1 (2008): 155-179.

[10] Acosta, Andrés, Jairo Espinosa, and Jorge Espinosa. "1 Distributed Simulation in SUMO Revisited: Strategies for Network Partitioning and Border Edges Management."

[11] Loolaid, Indrek. "Using SUMO in a distributed manner to generate road traffic data."

[12] Wegener, Axel, Michał Piórkowski, Maxim Raya, Horst Hellbrück, Stefan Fischer, and Jean-Pierre Hubaux. "TraCI: an interface for coupling road traffic and network simulators." In Proceedings of the 11th communications and networking simulation symposium, pp. 155-163. ACM, 2008.

[13] https://productforums.google.com/forum/#!topic/maps/byNxu_lT0do