Workload Estimation Algorithm in Parallel Traffic Simulation

Yan Xu and Gary Tan
School of Computing
National University of Singapore
Singapore 117417
xuyan@comp.nus.edu.sg, gtan@comp.nus.edu.sg

Abstract—Parallel traffic simulation is a critical component in large-scale traffic simulations and real-time traffic simulations. Dividing workloads evenly to multi-cores and multi-machines is a challenge in parallel traffic simulations. Current researches focus on map decomposition algorithms. However, without effective workload estimation algorithms, map decomposition algorithms tend to output imbalanced partitions. This paper proposes an elliptical-shaped workload estimation algorithm. The main idea of the algorithm is to assign the computational cost of a vehicle to links in an ellipse, whose centers (or foci points) are the vehicle’s origin node and the destination node. The algorithm is evaluated on a test-bed using a mesoscopic traffic simulator on the Lower Westchester County network. Case studies show that the new algorithm reduces 36% of the estimation errors in current length of links based workload estimation algorithms.

Keywords—Parallel traffic simulation; Load balancing; Workload estimation; Algorithm;

I. INTRODUCTION

Parallel traffic simulation is a critical component in large-scale traffic simulations and real-time traffic simulations. Load balancing is one of the most challenging problems in parallel traffic simulations. The aim of load balancing is to divide the workload of traffic simulations evenly to partitions. Workload estimation algorithms and map decomposition algorithms are two key components of a solution for a good load balancing.

Given estimated workloads on links in a traffic scenario, map decomposition algorithms divide a road network into k disjoint partitions and each partition has a similar estimated workload. METIS [2] and hMETIS [3, 6] are two of the most efficient algorithms. Besides these two, orthogonal recursive bisection is also used in some parallel traffic simulations [5]. Compared with map decomposition algorithms, workload estimation algorithms have not received enough research attention. Workload estimation algorithms estimate the computational workload on links. The estimated workload is an input to map decomposition algorithms. If the estimated workloads do not match the real workloads, map decomposition algorithms tend to give an imbalanced partition, which reduces the performance of parallel traffic simulation.

Currently, there are two types of methodologies to estimate workloads in a traffic scenario. The first methodology is based on the number of vehicles. Researches in [4, 5] give evidences that the number of vehicles on a link is a good indicator of the computational cost of the link. However, the number of vehicles on a link is not available when the scenario is simulated for the first time. The second methodology is based on the length of links (the LL algorithm) [5]. This methodology is effective only if the computational cost is evenly distributed on all links. Motivated by the desire of an effective workload estimation algorithm for parallel traffic simulations, when the traffic scenario is simulated for the first time, this paper proposes an elliptically-shaped workload estimation algorithm. The main idea of the algorithm is to assign the computational cost of a vehicle to links in an ellipse, whose centers (or foci points) are the vehicle’s origin node and the destination node. The algorithm is evaluated on a test-bed using a mesoscopic traffic simulator on the Lower Westchester County network. Case studies show that the new algorithm reduces 36% of the estimation errors in current length of links based workload estimation algorithms.

II. PROBLEM FORMULATION

Let the time period of interest be divided into intervals h=1,2,...,H. The road network consists of a number of links L= {l}. W = {w_l}, ∀h is the workload of links at each time interval. However, in most cases W is unknown before parallel traffic simulation. The workload estimation algorithm is to give an estimated W = {ŵ_l}, in order to minimize the difference between W and Ŵ.

For parallel traffic simulations, it is difficult to get precise measurements of workloads. In most cases, even executing the same program in the same configuration twice will get two different measurements of workloads. One practical strategy is to estimate the percentage of workload each link takes within all links. It is noted as P = {p^l}. It means that there is no need to estimate the workloads of each link at each time interval quantitatively. This simplification will not affect the efficiency of map decomposition algorithms. The problem can be further simplified by making the assumption that p^l does not vary with respect to time interval h. Basically, the assumption says that it is only necessary to estimate the percentage of workload each link takes within all links, during the whole traffic scenario.

The efficiency of workload estimation is measured by Sum Error (SE), which calculates the total absolute difference between the P and P̂.

$$SE = \sum_{l \in L} \left| p^l - \hat{p}^l \right|$$

(1)

where l is one link, L is all links, p^l is the workload percentage of link l during the whole traffic scenario. \hat{p}^l is the estimated workload percentage of link l during the whole traffic scenario. Sum Error (SE) is a variable between 0 and 1. If SE = 0, it means that the estimated workload percentages are exactly the same with the real workload percentages. Generally, a smaller Sum Error (SE) means a higher reliability to get a better load balancing.
III. ELLIPSE-SHAPING WORKLOAD ESTIMATION ALGORITHM

The philosophy in “Elliptically-shaped Workload Estimation Algorithm (ES)” is that a vehicle, in most cases, only affects a local area within the trip, and the local area should be related to the origin and the destination. In this algorithm, the local area is represented as an ellipse, whose centers (or foci points in mathematics area) are the origin and the destination. The vehicle is assumed to drive in the ellipse.

There are two major reasons to choose an ellipse. Firstly and the most importantly, an ellipse is a natural view of a vehicle when choosing one route from the origin to the destination. Secondly, it is easy to calculate whether a node or a link is inside an ellipse or not. Given one node \( <x,y> \), if the sum distance from the node to the origin and the destination is smaller than a value \( B \), then the node is inside the ellipse. Otherwise it is not inside the ellipse.

\[
B = D_{od} \times (1 + 2 \times d)
\]  

(2)

where \( D_{od} \) is the distance between the origin node and the destination node. If only one node of a link is inside an ellipse, but the other node is not, the link is considered un-contained by the ellipse. As shown in Figure 1, although the origin node and the destination node are fixed, the shape of an ellipse varies. The variable affecting the shape of an ellipse is the “looking back ratio \( (d) \)”. Looking back ratio is defined to be the ratio of the looking back distance over the distance between the origin and destination. As looking back distance \( (d) \) increases, the area of the ellipse will be larger and the shape of the ellipse will be more like a circle. The figure shows two shapes of an ellipse when \( d \) equals to 10% and 25%.

Figure 1: The different shapes of an ellipse when \( d = 10\% \) and 20%

The value of looking back ratio \( (d) \) affects the performance of this algorithm. The choice of looking back ratio \( (d) \) depends on the length of the shortest path. To be exact,

\[
d = d_0 \times \frac{D_{sp}}{D_{od}}
\]  

(3)

where \( d_0 \) is a constant. \( D_{sp} \) is the length of the shortest path. \( D_{od} \) is the straight distance from the origin to the destination. For example, if there is a straight expressway between the origin and the destination, \( d \) tends to be small, because most vehicles can find a route between the origin node and the destination node within a flat ellipse. However, if the origin and the destination are separated by a river and the bridge of the river is far away, \( d \) tends to be large, because vehicles have to choose a curve over the bridge. After figuring out the candidate routes in the ellipse, the computational workload of vehicles is evenly assigned to links of the routes.

IV. EXPERIMENTS

The test-bed is to estimate the workload in a traffic scenario simulating the Lower Westchester County (LWC) Network for 6 hours using a mesoscopic traffic simulator DynaMIT-P. The details of the test-bed can be found in the report [1]. The observed density on links is used to calculate the true workload. Two algorithms are tested to estimate the workload on links in this traffic scenario: (1) the LL algorithm and (2) the ES algorithm. The estimation error is measured by Sum Error (SE). The result is shown below. By using the OD Matrix, the ES algorithm can reduce the Sum Error by 36%, compared with the LL algorithm. The improved workload estimation can be important for large-scale parallel traffic simulations.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Data Requirement</th>
<th>Sum Error (SE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL</td>
<td>No Extra Data</td>
<td>33.6% (+40%)</td>
</tr>
<tr>
<td>ES</td>
<td>OD Matrix</td>
<td>21.5% (+36%)</td>
</tr>
</tbody>
</table>

V. CONCLUSION

Motivated by the desire of an effective workload estimation algorithm for parallel traffic simulations, when the traffic scenario is simulated for the first time, this paper proposes an elliptically-shaped workload estimation algorithm. The algorithm is evaluated on a test-bed using a mesoscopic traffic simulator on the Lower Westchester County network. Case studies show that the new algorithm reduces 36% of the estimation errors in current length of links based workload estimation algorithms.

REFERENCES


