An Approach Inspired by Simulation Points to Accelerate Smart Cities Simulations

Francisco Wallison Rocha[1], **Emilio Francesquini**[2], **Daniel Cordeiro**[1]

[1] Escola de Artes, Ciências e Humanidades – Universidade de São Paulo (USP)
São Paulo – SP – Brasil

{wallison.rocha, daniel.cordeiro}@usp.br

[2]Centro de Matemática, Computação e Cognição – Universidade Federal do ABC (UFABC)
Santo André – SP – Brasil

e.francesquini@ufabc.edu.br

Abstract. *Approaches using simulations are of great value for smart cities research. However, city-scale simulators can be both processing and memory-intensive, and hard to scale. To speed up these simulations and to allow executing larger scenarios, this work presents an approach based on an technique named Simpoint to estimate the result of new simulations using previous simulations. This technique aims to identify and cluster recurring patterns during a simulation. Then, unique representatives of each cluster are selected and their simulation is used to estimate the simulation results of the remaining cluster elements. The experimental results for our estimates are promising. On a dataset with 16,993 time series, our technique was able to estimate the original series with an average error of 1.60979e-11 and standard deviation of 9.18228e-11.*

1. Introduction

Urban mobility is an aspect of great interest of citizens and government. Urban mobility is key to ensure quality of life in large cities [Martins et al. 2020]. Nevertheless, cities like São Paulo Tokyo, Paris and New York face issues that are consequence of mobility problems, such as pollution, insufficient public transportation and heavy urban traffic. Solving these problems requires innovative solutions. Generally, the implementation of these solutions is complex. So, there is a need to test and validate before deploying them. [Santana et al. 2017].

Computer simulations have been used by city planners and smart city researchers to investigate and validate solutions for mobility problems. Their use on large-scale in conurbation scenarios like São Paulo, however, can be challenging due to the amount of computational resources required for running larger simulations.

The InterSCSimulator is a simulator written in *Erlang*, an actor-based language that simplifies parallel and distributed computing. The use of Erlang alone, however, is not enough to allow the execution of such large scenarios. The simulator has some limitations such as excessive memory use, does not have good scalability with speedup far from ideal, and limited ability to employ all available resources, e.g. using at most 50% of the available processing power available [Santana et al. 2017].

Computer architecture designers also suffer from limitations on simulation of program behaviours when designing novel computer architectures. Simulations of the execution of complex programs on these architectures could take weeks to be executed. Thus, Simpoints [Hamerly et al. 2005] was proposed to accelerate the simulation in the context of computer architecture.

The goal of the Simpoint technique is to find and exploit the large scale behavior of programs. After a full execution of the simulation, representative of patterns of the execution (called *simulation points*) are identified and reused in future executions. Clustering algorithms like K-means are used to identify the patterns and which parts of the simulation are represented by them. In this work, we investigate the use of this technique to accelerate traffic simulations in InterSCSimulator.

2. Simpoints

We adapted the Simpoints technique to the traffic simulation problem. The use of Simpoints requires the definition of metrics that can summarize the behaviour of the simulation. Computer architecture designers used metrics such as data cache misses, performance (IPC), branch mispredictions, among others. For traffic simulations, we can use different metrics such as average vehicle speed, distance driven, percentage of occupation of roads by vehicles, etc.

Originally, Simpoints [Hamerly et al. 2005] were based on the behavioral signatures of Basic Block Vectors (BBVs). The BBV is a vector where each element corresponds to the number of times that basic block basic executed in an execution interval. In this paper, behaviors are represented by time series. Time series are formed by the events of entry and exit of vehicles of a road in a certain time interval.

To compare the behaviors (BBVs) to cluster them, [Hamerly et al. 2005] uses the Manhattan distance and clustering algorithm K-means. On the other hand, due to the use of time series to represent the behaviors of the simulation. For greater accuracy, this paper using algorithms to compare time series like Dynamic Time Warping (DTW) [Berndt and Clifford 1994], and Shape-Based Distance (SBD) [Paparrizos and Gravano 2015] with K-means and K-shape.

We use the K-shape algorithm with SBD because its asymptotic complexity $O(mlog(m))$ is less than that of K-means with DTW which is $O(m^2)$ [Paparrizos and Gravano 2015]. Where m is the size of the time series. However, time series can behave the same, but with a different dimension, for example, a series of one street with capacity for 80 vehicles and another with 20 capacity. Both can have a traffic jam, but with different number of vehicles. So, before clustering, it is necessary to apply a scale transformation in time series. These scale transformations will allow comparing the data in the same amplitude. Equation $w' = \frac{w-\overline{x}}{\sigma}$ shows how the transformation is done, where w' is normalized series, w is the series to normalize, $\overline{x}$ is the average of w values, and σ is the standard deviation of w values.

Then, after performing clustering using K-shape and SBD, the next step is to determine the simulation points. In this work, the simulation points are represented by the centroids of each simulation cluster, just as it was done by [Hamerly et al. 2005]. The centroid in the cluster is the element (time series) with the shortest distance to the other elements of the cluster.

Finally, to estimate a time series from the simulation point (the cluster centroid), we used the warping path calculated using DTW. The warping path is the element-by-element association with the shortest distances between two series. The warping path is used to map simulation points into series in the cluster. After, this same warping path is used to warp the simulation point into each series. Equation 1 shows how the simulation point is warp to estimate each series in the cluster using the simulation point-series warping path of each one. Where u_j is the element at position j in the series to be estimated, d_{ij}^2 is the distance between element c_i in the cluster and the element u_j in the series. Furthermore, it is necessary to apply scale reversal to return the series to its original amplitude. Equation $w = w' \cdot \sigma + \overline{x}$ is used to calculation of the scale reversal.

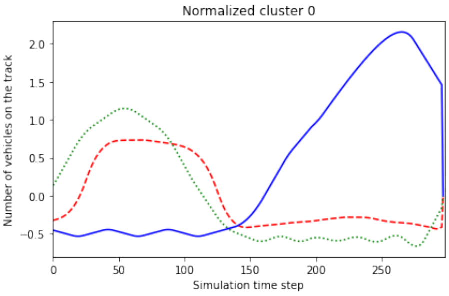
$$u'_j = \begin{cases} c_i - \sqrt{d_{ij}^2 - d_{ij-1}^2}, & \text{if } u_j \geq c_i \\ \sqrt{d_{ij}^2 - d_{ij-1}^2} + c_i, & \text{if } u_j < c_i \end{cases} \qquad (1)$$

3. Experimental Results

We cluster the patterns of a simulation. We select a representative from each cluster and use them to estimate the other series. This experiment was done using a dataset with $16,993$ time series of size 298, distributed into 8 clusters. This dataset was obtained from a simulation of 1500 trips in a region of 292 roads. The machine used to run the experiment has 8 GB RAM and a Intel Core it-7500U CPU 2.70 GHz quad-core processor.

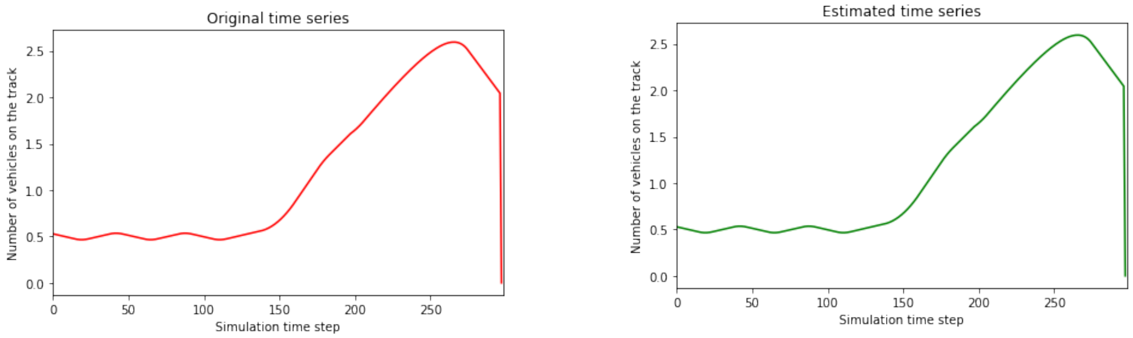
Figure 1 shows Cluster C_0 with a sample of two series with greater and lesser distance to the centroid. The series were selected according to the calculation of the DTW distance [Berndt and Clifford 1994]. The centroid c_0 of the cluster is depicted in red, in green the closest s_{249} series and in blue the most distant series s_{1385} from the centroid. On the x axis it represents the simulation time steps and the y axis represents the number of vehicles on the road at that moment. All series are normalized on a scale from -4 to 4.

Figure 1. In the dashed series the centroid is shown, in the dotted series the series closest to the centroid and the continuous series the most distant



Figures 2a and 2b show the result of the reverse process. Given a centroid in red (Figure 1) and the signature of a series, we compute an estimate for the original series (Figure 2a). Figure 2b shows the estimate from the centroid. In this example was used the Equation 1 and after Equation $w = w' \cdot \sigma + \overline{x}$. The estimation error on a series of n values is given by $E = \sum_{i=1}^{n} |s_i - s'_i|$, where s_i is a value of the original series and s'_i its estimation. The error of the estimated series (Figure 2b) in relation to the original series (Figure 2a) was $6.34732e - 11$. For the complete simulation, the biggest error was $9.58240e - 09$, the mean error was $1.60979e - 11$ with a standard deviation of $9.18228e - 11$.

Figure 2. Estimate of the longest series to the centroid.



(a) Original time series s_{1385} of cluster c_0

(b) Estimated time series s_{1385} from the centroid of cluster c_0

4. Conclusions

This work presented a preliminary study on the application of the Simpoints technique for traffic simulations. We proposed an adaptation of Simpoints for time series instead of Basic Block Vectors and K-shape with Shape Based Distance instead of K-means with the Manhattan distance. Due to its velocity and smaller asymptotic complexity, the K-shape was used to cluster time series instead of K-means with DTW. However, it is still necessary to evaluate the accuracy of the clustering process. Preliminary results obtained for this estimation are very encouraging, showed that the biggest error for the complete simulation was $9.58240e-09$, the mean error was $1.60979e-11$ with a standard deviation of $9.18228e-11$.

The next steps are to estimate the time series of other simulations. Furthermore, we want to study the results obtained with different metrics, like the average speed of a region, and the average road occupation for a given region.

References

Berndt, D. J. and Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, AAAIWS'94, page 359–370, Seattle, WA. AAAI Press.

Hamerly, G., Perelman, E., Lau, J., and Calder, B. (2005). Simpoint 3.0: Faster and more flexible program phase analysis. *Journal of Instruction Level Parallelism*, 7(4):1–28.

Martins, T. G., Lago, N., de Souza, H. A., Santana, E. F. Z., Telea, A., and Kon, F. (2020). Visualizing the structure of urban mobility with bundling: A case study of the city of são paulo. In *Anais do IV Workshop de Computação Urbana*, pages 178–191. SBC.

Paparrizos, J. and Gravano, L. (2015). K-shape: Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, page 1855–1870, New York, NY, USA. Association for Computing Machinery.

Santana, E. F. Z., Lago, N., Kon, F., and Milojicic, D. S. (2017). InterSCSimulator: Large-scale traffic simulation in smart cities using erlang. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 211–227. Springer.