Scalable Data Analysis for Public Bus Systems

Mayuri A. Morais[1], **Raphael Y. de Camargo**[1]

[1]Centro de Matemática, Computação e Cognição – Universidade Federal do ABC (UFABC)
Rua Arcturus, 03 – 09606-070 – São Bernardo do Campo – SP – Brazil

{mayuri.morais,raphael.camargo}@ufabc.edu.br

Abstract. *Urban mobility through quality public transportation is one of the major challenges for the consolidation of smart cities. Researchers developed different approaches for improving bus system reliability and information quality, including travel time prediction algorithms, network state evaluations, and bus bunching prevention strategies. The information provided by these approaches are complementary and could be aggregated for better predictions. In this work, we propose the architecture and a present a prototype implementation of a framework that enables the integration of several approaches, which we call models, into scalable and efficient composite models.*

1. Introduction

Providing efficient urban mobility is one of the major challenges facing large metropolitan centers today. An efficient way to reduce congestion is with the provision of quality public transport systems. Public transport systems using buses in Metropolis, such as São Paulo, are complex systems that continuously interact with the dynamics of the city. Buses are delayed due to congestion and overcrowding, as well as having their flow interrupted by traffic lights. Understanding the behavior of this system in different contexts, such as weekdays, hours of the day and holidays, is vital for better planning of these systems.

Older literature works proposes frameworks for bus GPS data collection and storage [Liying 2009, Liu et al. 2010], handling matters like how each bus will transmit the data and how this data will be stored and made available to the public. Recent works focus on how to use this now available data, but in general each work focus in a specific problem, such as bus network state [Zhang et al. 2016], bus position estimation [Adachi et al. 2015], travel time predictions [Mori et al. 2015, Choudhary et al. 2016] and bus headway evolution [Yu et al. 2017]. These studies evaluate each aspect separately, despite the clear interactions between them, such as the influence of headway evolution on travel time predictions. There are only a few efforts in this direction of combining aspects of bus system analysis in a single framework [Mazloumi et al. 2011]. Private companies (such as Google) have their own frameworks for this type of data processing and they make some of their results available but in a way which the data for further analysis is not openly available.

One way to improve this interaction is by providing an open framework for bus data collection, analysis, and visualization, which contains a set of models, each contemplating a specific aspect of the bus system. Such a system contain some lower level

This research is part of the INCT of the Future Internet for Smart Cities funded by CNPq proc. 465446/2014-0, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001, FAPESP proc. 14/50937-1, and FAPESP proc. 15/24485-9.

models, such as the graph representing the bus network and the estimation of the network link states (congestion level), bus positions and bus headways. On top of these, higher level predictions models, such as for travel time, bus headway evolution, and network link state evolution, can be built. We expect these models to interact with each other to generate combined analysis, estimations and predictions.

In this work we propose the architecture and a prototype implementation of this framework, containing a network model of two hundred bus routes from São Paulo city, an estimator for bus positions, and a travel time prediction models, which use historical and real-time data on bus positions. The presented results focus on the processing time of the models, since this is an important constraint for real-time processing and prediction.

2. Framework architecture

We propose a general framework for scalable and distributed processing of large amounts of historical and real-time data from thousands of bus lines. This framework would be useful for different applications, such as predicting bus travel times, evaluation of the flow along the bus routes, and preventing the occurrence of bus bunching.

2.1. Models

The core framework concept are `Models`, which can represent: (i) different views of the bus system state, pre-processed by some data analysis algorithm, such as estimated bus position, estimated links states, estimated bus headways; or (ii) predictions of future developments, such as travel time predictions in each link, evolution of link states, and likelihood of bus bunching occurrences.

3. Implementation

3.1. Distributed Execution using Dask

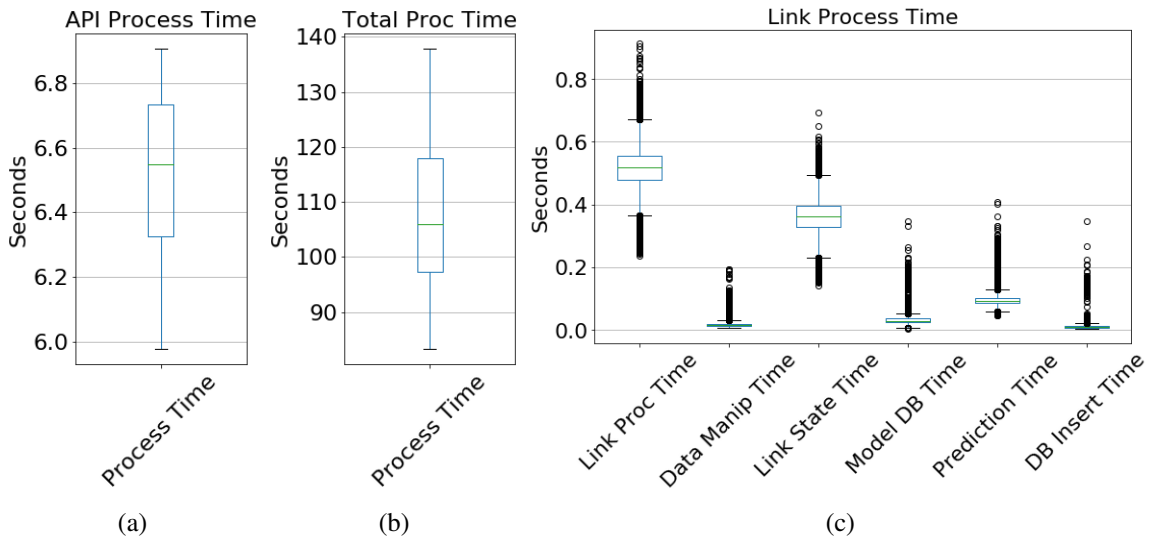
For the framework implementation we used Python with Dask.distributed, a lightweight library for distributed computing. This library consists of a centralized scheduler and distributed workers which can communicate with each other.

Dask provides scalability by permitting the inclusion of more workers as demanded by the framework and supports complex workflows dependency graphs beyond those of map/filter/reduce. The scheduler sends each task to a different worker which can be running on one or more machines. Dask manages data transfer from the coordinator process to each worker. Each model holds the data it generates in memory or persist it in a database. One can include new models by writing new Dask tasks that perform the required model computations and access data from other models.

3.2. Implemented Models

Here we present three implemented models, which we used as a prototype implementation to test the framework architecture.

Bus Position Model: This model collects online real-time data, calculates the distance traveled by each bus in its predetermined route, estimates the bus passing time at the vertices of the graph model and finally estimates the Link Travel Time for each link of the graph.



(a) (b) (c)

Figure 1. Execution time for predicting the link states of 200 bus lines, using one machine (20 workers). Figure (a) shows the online data retrieve and processing time; Figure (b) shows the total processing time; Figure (c) shows the processing time of a single link state prediction (first column), as well as the processing time of its internal steps.

Graph Model: This model transform general transit feed specification (GTFS) files into a citywide graph containing all bus routes from a city. It defines the midpoints between bus stops of a bus route as vertices and the paths between consecutive vertices as links. The model first defines sub-graphs for individual bus routes and then concatenates them into a citywide graph, identifying common links to multiple bus routes.

Link State Prediction Model: This model predicts future link states (i.e, the travel time on each link) based on current state. The prediction Model uses Linear Regression, Lasso, Ridge, Random Forest Regressor and Support Vector Regression. They are trained on Historical data and evaluated with real-time data. Each call to this models predicts to a single link of the graph model.

4. Experimental Setup

We performed an experimental evaluation over the implemented prototype. We measured the execution time for the Bus Position Model, Link State Prediction Models and the total processing time of both models. Graph Model was processed beforehand. We executed these models 20 times using 200 Bus lines on a Monday in peak time (7:00AM-8:00AM). The processing of each graph link on Link State Prediction Models was distributed over 20 Dask workers running in a single machine with 2 CPUs model *Intel(R) Xeon(R) CPU E5-2620v2@2.10GHz* with 6 (12) physical (virtual) cores per CPU and 128GB of RAM.

5. Experimental Results

From the collected data, the Bus Position Model processing time variate between 6 and 7 seconds (Figure 1a). This is a reasonable time, since each access to the online API is made in a minimum interval of 40 seconds. On the other hand, the total processing time (online data collection, process and link state prediction) of 200 bus lines variate between

80 and 140 seconds (Figure 1b). This time must be reduced, since 200 bus lines is far less than the total number of bus lines in Sao Paulo (around 2000 bus lines), which is the processing goal.

Since the online data collection is at max 7 seconds, the slow processing time is due to the Link State Prediction Model. Thus, we analyzed the total processing time of each link as well as the internal processing time of different parts of the code, to identify where the code is slower. From Figure 1c, we can see that each link is taking around 0.5 seconds on average to be processed (column *Link Proc Time*). From that, almost 0.4 seconds on average is due to the *Link State Time* step, which is an access to a database to retrieve previous link states. Our goal now is to reduce this link prediction and total processing time. One alternative is to change the database use by a memory structure, which hopefully will allow faster retrieval of previous link states and reduced total processing time.

References

Adachi, H., Suzuki, H., Asahi, K., Matsumoto, Y., and Watanabe, A. (2015). Estimation of bus traveling section using wireless sensor network. In *2015 Eighth International Conference on Mobile Computing and Ubiquitous Networking (ICMU)*, pages 120–125. IEEE.

Choudhary, R., Khamparia, A., and Gahier, A. K. (2016). Real time prediction of bus arrival time: A review. *Proceedings on 2016 2nd International Conference on Next Generation Computing Technologies, NGCT 2016*, (October):25–29.

Liu, Z., Zhao, T., and Yan, W. (2010). A novel architecture and open issues towards infrastructure-less city-wide traffic information systems using vehicular ad hoc networks. In *2010 International Conference on Communications and Mobile Computing*, volume 3, pages 522–527.

Liying, W. (2009). Design of the distributed framework and dataflow process for urban road traffic information systems. In *2009 Second International Conference on Intelligent Computation Technology and Automation*, volume 3, pages 558–561.

Mazloumi, E., Rose, G., Currie, G., and Sarvi, M. (2011). An integrated framework to predict bus travel time and its variability using traffic flow data. *Journal of Intelligent Transportation Systems*, 15(2):75–90.

Mori, U., Mendiburu, A., Álvarez, M., and Lozano, J. A. (2015). A review of travel time estimation and forecasting for Advanced Traveller Information Systems. *Transportmetrica A: Transport Science*, 11(2):119–157.

Yu, H., Wu, Z., Chen, D., and Ma, X. (2017). Probabilistic prediction of bus headway using relevance vector machine regression. *IEEE Transactions on Intelligent Transportation Systems*, 18(7):1772–1781.

Zhang, X., Chen, G., Han, Y., and Gao, M. (2016). Modeling and analysis of bus weighted complex network in qingdao city based on dynamic travel time. *Multimedia Tools and Applications*, 75(24):17553–17572.