Simulation-supported urban movement analysis

J. Borgdorff[1], W. van Hage[1], L.J. Dijkstra[2,3], E. Mancini[2], M.H. Lees[2,3]

[1]Netherlands eScience Center, Science Park 140, 1098 XG Amsterdam, The Netherlands
Email: {j.borgdorff, w.vanhage}@esciencecenter.nl

[2]Compuational Science Lab, University of Amsterdam, Postbus 94323, 1090 GE Amsterdam, The Netherlands
Email: {l.dijkstra, e.mancini, m.h.lees}@uva.nl

[3]Department of High Performance Computing, ITMO University, Saint Petersburg, 197101, Russia

1. Introduction

Movement dynamics are challenging to understand, especially in rapidly developing and urbanizing countries such as India and in massive crowd gatherings like the Kumbh Mela festival in India. Traffic is dynamic and heterogeneous and the crowds are large; data is diverse and data sizes range from minor to vast; so how can these phenomena be studied effectively?

In this paper we argue that this requires combining data analysis, simulations and geospatial visualization. That a software platform with this combination facilitates an iterative scientific process; from its early phases of planning/setting up the experiment to predicting behavior. We study the advantages of shortcomings of this approach based on three use cases. This work is a continuation of earlier work done in the SIM-CITY project (Borgdorff et al. 2015).

This paper is organized as follows: in section 2 we show uses of the platform, in section 3 its implementation, and in section 4 we show how it applies to three scenarios.

1.1 Related work

Large events where simulations and crowd analysis played a large role include the Love Parade disaster in 2010 (Helbing and Mukerji 2012) and the yearly Hajj (Helbing et al. 2007). Crowd simulation in general is reviewed by Duives et al. (2013), and they indicate that current pedestrian models are adequate only for either fast or for accurate processing. We do not aim to do real-time crowd modelling in this paper. Picornell et al. (2015) show that phone call data can improve urban travel simulations. In our platform, these kind of data cross-pollinations could be performed on an automatic basis.

For wildfires, a similar platform that integrates data, simulations and geospatial visualization is described by Altintas et al. (2015). Although the subject is different, the architecture is very similar. Systems like TrackLab (Spink et al. 2013) and KAVE (KPMG 2016) combine sensor data collection, visualization and analysis, but do not run simulations.

2. Simulation platform

The platform we propose combines three primary facets:

- data sets, constantly updated and supplemented with simulation results;
- visualization, from raw to aggregated to simulated data, both spatial and non-spatial; and
- a simulation infrastructure, to start simulations based on the data on the fly.

We see a role for this platform in several phases of research, especially when sensor data is involved. This includes planning/setting up a trial, monitoring data collection during the trial, analyzing the data afterwards, and constructing a set of possible scenarios and outcomes.

2.1 Planning

While setting up a trial with sensors/trackers, one needs to decide on 1) which type of sensors to use, 2) how many of each to employ, 3) their placement, and 4) their settings (e.g., with what frequency should they collect and transmit data). One wants to obtain as much information on the movement dynamics as possible, while adhering to practical restrictions such as budget and/or availability of data from a third party.

Simulation provides a basis for making such decisions. Typical input for such a simulation would be a specification of the various sensors under consideration (e.g. range, typical error margins), combined with the geography of the domain under study.

The platform can visualize the geography, proposed layout of the sensors, the output of the simulations, and it provides a statistical interface to the simulations. Ideally, this aids the researcher in coming up with a reasonable set-up and in finding potential weaknesses in a deployment strategy beforehand.

2.2 Data collection

The user imports data directly into the system. Due to the diversity of data sources, it is a challenge to automate this process. As data is entered into the platform, the user starts processes to clean and privatize the data. By using the data processing system Apache Spark, this scales very well with data size. After an initial clean-up, the platform can visualize the data and assess its technical validity. It is essential that this is done while intervention in data collection is still possible by, for example, replacing faulty devices, increasing/decreasing collection frequency, or increasing storage space.

2.3 Analysis

The collected data is processed with Apache Spark and analyzed within a Jupyter Notebook, allowing the user to perform statistical inference, e.g., parameter estimation, outlier detection and/or hypothesis testing. In addition, the system is equipped with a web-service for geospatial visualization, making it easier to assess the data and check for potential irregularities.

The data is also used to parameterize and validate a simulation. The data might be used for aggregate velocity and density (Wirz et al. 2013), and an associated simulation may then yield micro-statistics such as individual or group position, velocity, clustering and long-term goals, as well as likely parameter settings.

2.4 Scenarios

When the models of the analysis phase are thoroughly validated, they may estimate the outcome of a number of possible scenarios, such as interventions, emergencies, or extreme crowding.

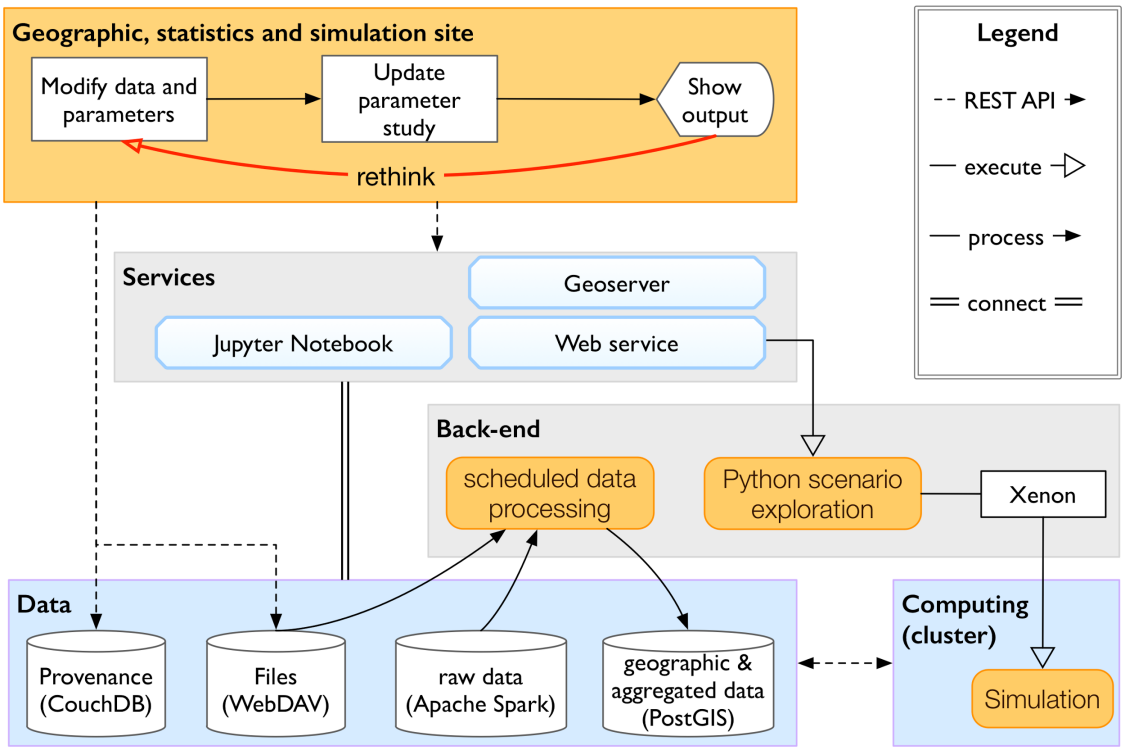

Figure 1. Platform architecture.

3. Software architecture

The architecture of our platform is shown in Figure 1. The easiest way for interdisciplinary and international researchers to analyze and collaborate on the same data is via a web interface. Its advantages are that everyone works with a very similar interface and that it can exploit new features in fast evolving JavaScript libraries. Each of the components is open source, so the platform can be installed and improved by any research group.

The web interface of the platform consists of two parts. The first part is an AngularJS website with an interactive OpenLayers map, to explore the (usually aggregated) geospatial data. The website supports data visualization (for some sorts of data) including, among others, interactive graphs and styling. It assumes that the data are clean and formatted, and it is adapted to the available data sources and simulations. The second part is a Jupyter notebooks service with interactive Python or R sessions, that allows for faster prototyping of data analyses. These sessions can assess raw data files, databases, and data repositories. Once a data analysis script is deemed suitable, it can be integrated into the main data processing workflow.

These web interfaces get their data from numerous web-services. Geographic and aggregated data are served by a GeoServer with a PostGIS database. Apache Spark is installed on a separate cluster for dynamic and repeated queries on large sensor and video datasets, which shifts a large part of the load of the analysis from the laptop or server to the cluster. A great advantage for international collaborations is that this drastically reduces data transfer sizes. A custom web service takes care of starting any simulations or standard analyses. The Xenon middleware library starts the simulation on a computing cluster; the Python sim-city-client package monitors its progress. A WebDAV file storage stores the output files. Finally, a CouchDB database keeps track of all simulations, and forms the central location for storing the configuration of the website and web service. Both the file storage and database were chosen to support HTTP for interoperability with the web interface and ease of access for researchers.

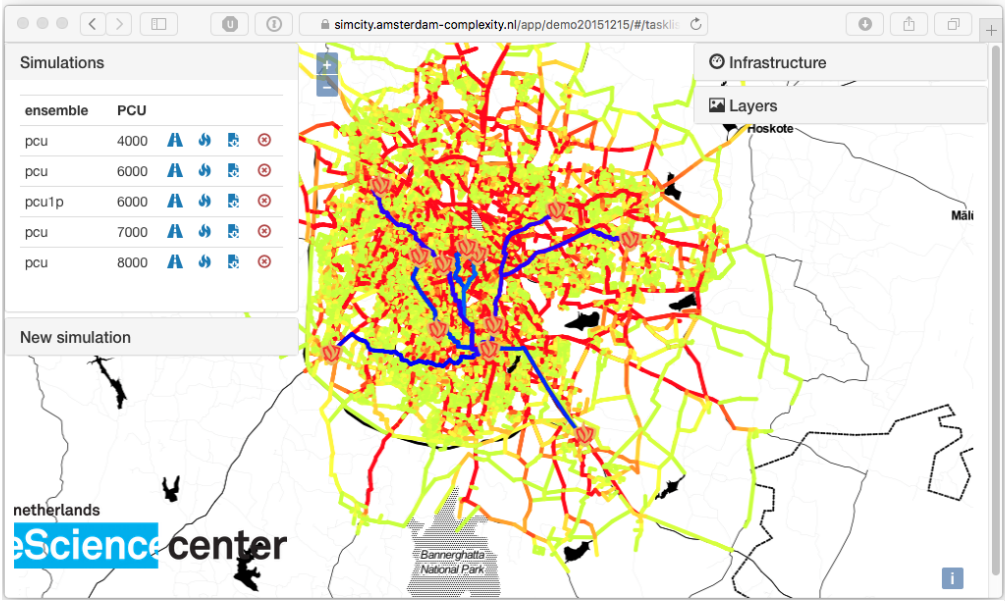

Figure 2. Screenshot of the web interface for the SIM-CITY project, showing traffic density and fire response routes. Simulations, geospatial layers, and infrastructure are controlled from this web interface.

Repeated background processes on the server can evaluate a number of scenarios simulations depending on the current dataset and update the aggregated data in the PostGIS database. Most of these services can be run with Docker, to reduce the amount of configuration needed.

The components in the architecture are loosely coupled so they can be used in isolation. This allows for more flexibility and faster prototyping, for example, to test a model with synthetic data or to perform data analysis with the researcher's favorite software package.

4. Use cases

In the SIM-CITY project, a major use case is a fire response decision support system. The main objective is to estimate fire response times in Bangalore, India, based on traffic conditions and the fire station placement. To get reliable traffic estimates, OpenStreetMap road network is combined with census data, traffic counts, city planning data, and GPS traces of buses. A web interface, in Figure 2, can be used to start simulations with alternative fire station placement and global traffic parameters. The web interface combines visualization and scenario exploration, and it is largely customized for this use case. In this project, an analysis notebook is not available.

The aim of the Kumbh Mela project is to study the dynamics of mankind's largest crowd, by combining data sources like wireless tracking and video feeds. The Kumbh Mela religious festival will be celebrated in Ujjain, India in 2016, with roughly 100 million visitors expected over a one-month period. Where wireless tracking can be used to get approximate movement of selected individuals, video feeds are suitable for determining the density and the general direction of crowds. In confined areas, where entrances and exits are controlled, a larger fraction of visitors can be tracked. Data gathering is confined to a single month, so during this period the collected data will be meticulously monitored. For fast prototyping, Python scripts will be used for this. After the event, which will have very limited network connectivity, the data will be fed into an Apache Spark cluster for remote analysis. So only after the event the full platform will be deployed for analyzing the data.

Finally, the Indo-Dutch MobiLab project will combine bus GPS traces, police response statistics and select mobile phone data to form an interactive traffic monitoring and simulation system of Bangalore, India. Here data will be integrated into the data stores as it is collected, since the goal is to provide near real time feeds and simulations. The platform can be used during all research phases.

Initially, the platform architecture had fewer components, which is easier to install and maintain, but does not always scale. For example, the CouchDB database was initially used to store all input and output of simulations. The advantage is that all provenance is available in a single location. As the number of simulations increased, it became apparent that CouchDB was not ideal for storing large files, so a WebDAV file storage was introduced. However, due to loose coupling between components, this introduces few changes to the code.

Another issue with a component-based architecture is that it complicates the configuration for new users. We found this to be alleviated in two ways: by storing configuration centrally in a database instead of scattered over configuration files and by using Docker and Docker Compose to assemble the platform.

Finally, a web interface is easy to use but not very flexible: to be effective, they must be customized based on the data and visualizations. Web interfaces that are generated based on configuration are more flexible, but never become as flexible as directly manipulating a data set. During the projects, it became apparent that technical users need direct access to the simulation and data analysis infrastructures and that they need assistance in using the full power of the platform.

5. Conclusions

The platform described in this paper integrates data analysis, geospatial visualization and simulations. It was first tested in the SIM-CITY project, and is being prepared for the Kumbh Mela and Indo-Dutch Mobilab projects. The main usage so far is to gain insight in the data and the simulations for the validation of those simulations. As we progress, we intend to emphasize the data processing and interactive data analysis aspects of the platform, since these allow for faster iterations between data and hypotheses.

The experience in the projects gave two main conclusions:

1. Well-tested simulations and data analyses make for an appealing and easy to use web interface, but for model and analysis development, direct access to the infrastructure is needed.
2. A component-based architecture allows for both these modes of operation, but the number of components and technologies have to be limited for maintenance and re-use.

In future work, we intend to expand on real-time data collection in such a platform.

Acknowledgements

We would like to thank in particular Dr. Ashish Verma of the Indian Institute of Science for his excellent coordination of data collection. The technical infrastructure was facilitated by SURFsara, the Netherlands. The work in this paper is funded by the Netherlands eScience Center as the SIM-CITY project, by NWO in the Netherlands as Indo Dutch MobiLab under grant agreement 629.002.101 and by NWO as the Kumbh Mela project under grant agreement 629.002.202.

References

Altintas, Ilkay, Jessica Block, Raymond De Callafon, Daniel Crawl, Charles Cowart, Amarnath Gupta, Mai Nguyen, Hans-Werner Braun, Jurgen Schulze, Michael Gollner, Arnaud Trouve, and Larry Smarr. "Towards an Integrated Cyberinfrastructure for Scalable Data-driven Monitoring, Dynamic Prediction and Resilience of Wildfires." *Procedia Computer Science* 51 (2015): 1633-642.

Borgdorff, Joris, Harsha Krishna, and Michael H. Lees. "SIM-CITY: An E-Science Framework for Urban Assisted Decision Support." *Procedia Computer Science* 51 (2015): 2327-336.

Duives, Dorine C., Winnie Daamen, and Serge P. Hoogendoorn. "State-of-the-art Crowd Motion Simulation Models." *Transportation Research Part C: Emerging Technologies* 37 (2013): 193-209.

Helbing, Dirk, and Pratik Mukerji. "Crowd Disasters as Systemic Failures: Analysis of the Love Parade Disaster." *EPJ Data Sci. EPJ Data Science* 1.1 (2012). Print.

Helbing, Dirk, Anders Johansson, and Habib Zein Al-Abideen. "Dynamics of Crowd Disasters: An Empirical Study." *Physical Review E Phys. Rev. E* 75.4 (2007).

"KPMG Analytics and Visualization Environment (KAVE)." *KPMG, the Netherlands*. URL: http://beta.kave.io. Accessed 10 Mar. 2016.

Picornell, Miguel, Tomás Ruiz, Maxime Lenormand, José J. Ramasco, Thibaut Dubernet, and Enrique Frías-Martínez. "Exploring the Potential of Phone Call Data to Characterize the Relationship between Social Network and Travel Behavior." *Transportation* 42.4 (2015): 647-68.

Spink, Andrew, Ben Locke, Nico Van Der Aa, and Lucas Noldus. "TrackLab." Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication - UbiComp '13 Adjunct (2013).

Wirz, Martin, Tobias Franke, Daniel Roggen, Eve Mitleton-Kelly, Paul Lukowicz, and Gerhard Tröster. "Probing Crowd Density through Smartphones in City-scale Mass Gatherings." *EPJ Data Sci. EPJ Data Science* 2.1 (2013).