Streets4MPI Documentation

Release 0.1

Julian Fietkau, Joachim Nitschke

April 30, 2012

CONTENTS

CHAPTER

ONE

INTRODUCTION

Streets4MPI is a software that can simulate simple street traffic patterns in street networks imported from OpenStreetMap. It is written in Python and supports MPI (through mpi4py) for parallel computation.

The basic idea of *Streets4MPI* is to pick a fixed amount of *(origin, goal)* pairs in the street network. For each of these pairs, the shortest path is calculated, taking local speed limits into account. All calculated shortest paths are then traversed and the *traffic load* is recorded for each street, the cumulated results constituting a simulated *day*. From day two onwards, drivers are influenced to a semi-random extent by street congestions from the previous day. After a configurable number of days has passed, the street network is adapted: seldomly used roads are dismantled, heavily used ones are augmented to better cope with their usage.

Apart from the simulation core, *Streets4MPI* also contains a visualization component that can output the simulation results as traffic load heatmaps.

CHAPTER

TWO

INSTALLATION

At this point in time, we do not offer prepackaged versions of *Streets4MPI* for download. However, you can get the most recent version from our GitHub repository:

```
git clone git@github.com:jfietkau/Streets4MPI.git
```

If you are not familiar with *git*, refer to GitHub's Introduction to Git or any other of the free tutorials out there.

2.1 Requirements

Streets4MPI has been tested on Python 2.6 and 2.7. It is only known to work on Linux. No other platforms have been tested, but to the best of our knowledge it does not contain any platform-specific code. It is limited by the availability of the external packages on which it relies:

- mpi4py for MPI-based parallelization
- imposm.parser for *OpenStreetMap* imports
- python-graph for large-scale graph structures
- Python Imaging Library for visualization output

Recommended ways to install these packages will vary across systems and distributions. Please refer to the individual projects' documentation and your operating system's package management.

2.2 Street Network Data

To simulate traffic, *Streets4MPI* needs a street network to use. It must be available as a local file in a format that imposm.parser can read, e.g. OSM XML.

We supply a small street network for testing, approximately encompassing the Hamburg-Stellingen district, as *res/test.osm*.

OpenStreetMap data based around political borders can be downloaded at GEOFABRIK. Of course, you can also export your own area.

CHAPTER

THREE

SIMULATION

To get started, you should be able to run the *Streets4MPI* main script, *Streets4MPI.py*, from the command line. As long as all necessary packages are installed, it should be able to run even if you do not have a properly configured MPI available:

```
python Streets4MPI.py
```

The simulation is configured in *settings.py*. Here are some parameters which you might want to adjust:

osm_file This must be the location of your OpenStreetMap data file.

number_of_residents This is the total number of trips calculated per simulated day.

max_simulation_steps After this many days have been simulated, the program automatically exits.

persist_traffic_load If *True*, traffic load data is written to **.s4mpi* files for later visualization. If *False*, simulation results are discarded.

use_residential_origins If *True*, only use nodes tagged as belonging to a residential area as origins for trips to simulate drivers going from their home to work. Depending on the quality of your *OpenStreetMap* data, these might not be tagged sufficiently, severely limiting potential origins. That is why this parameter defaults to *False* (any node can be an origin).

logging If *"stdout"*, detailed logging to *stdout* occurs. If *None*, all logging is suppressed. Directly logging to a file is not supported yet.

If you want to run *Streets4MPI* in parallel on top of MPI, simply use *mpiexec* like you always would. Example:

```
mpiexec -n 16 python streets4mpi.py
```

Streets4MPI will then divide the total number of residents across all MPI nodes and manage the communication automatically.

CHAPTER

FOUR

VISUALIZATION

The visualization component of *Streets4MPI* is rund independantly of the main simulation. It checks its working directory for files with the *.s4mpi* extension and attempts to visualize them.

Several data modes are supported, default and probably the most interesting one being *TRAFFIC_LOAD*. Furthermore, *Streets4MPI* supports two color modes: one for heatmap-style colorful maps and one for grayscale images where brightness denotes intensity.

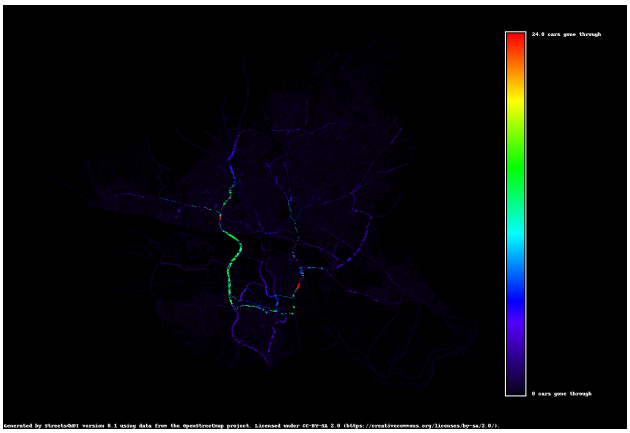

Figure 4.1: This visualization output shows the city of Hamburg, Germany, after some simulated traffic load. It uses the heatmap color mode.

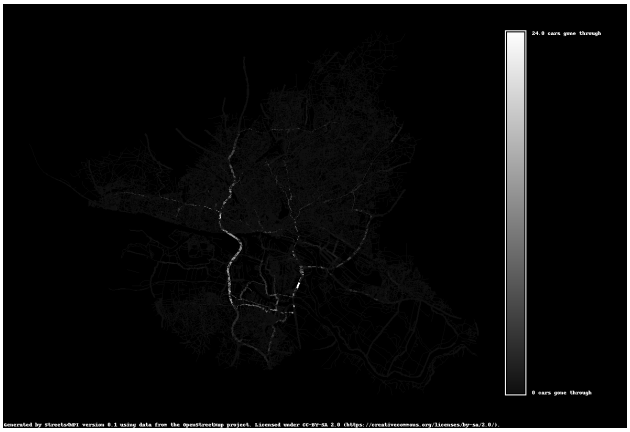

Figure 4.2: Using the same traffic load data, this visualization output uses the grayscale color mode.

CHAPTER

FIVE

KNOWN LIMITATIONS

When dealing with very large numbers of residents and/or street networks, *Streets4MPI* tends to use very large amounts of RAM and CPU time. We have performed tests up to 100,000 residents, and in some cases the simulation slowed down dramatically.

Unfortunately, *Streets4MPI* is not sufficiently tested for international map data. The simulation should be completely location-agnostic, but the visualization contains several *"Hamburg-isms"*, both explicit and implicit. We welcome any contributions to make our map projection more versatile.