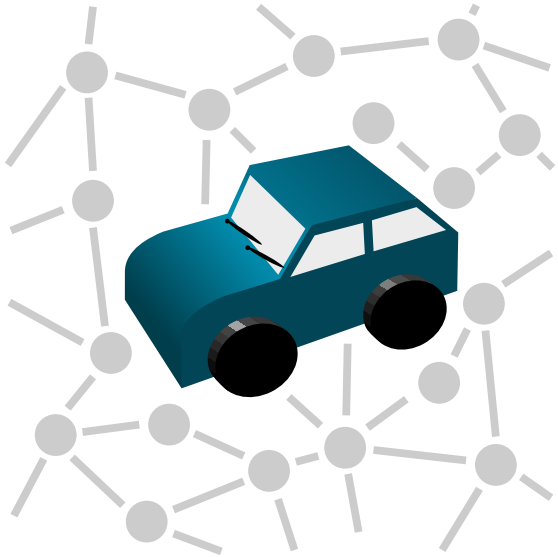
Parallel Programming Project Plan



Julian Fietkau
Joachim Nitschke

University of Hamburg

February 22nd, 2012

Agenda

Introduction

Concept and Model
- Overview
- Computation

Technologies and Data Sources

Plans for Parallelization

Summary

Project Task

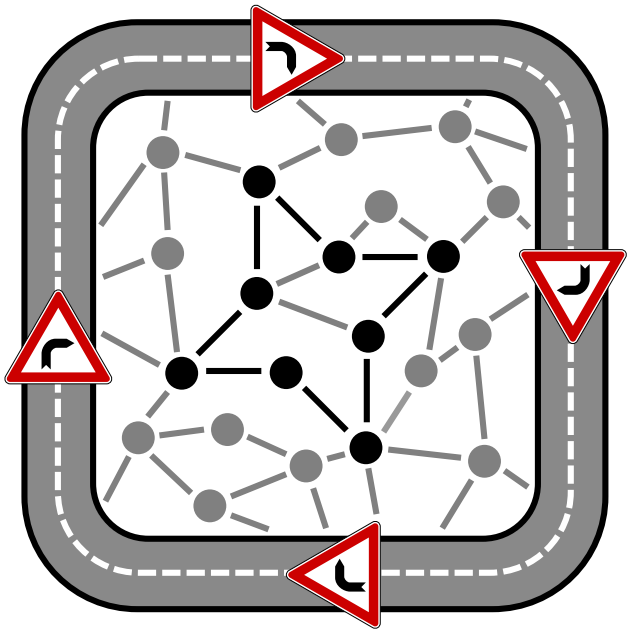
Decide on a problem that may be solved using parallel processing, and implement a solution. → **Street traffic simulation**

Main Caveat

Realistic traffic predictions can only be made using an exceedingly detailed model. **This makes things prohibitively complicated.**

Project Path

1. Formulate the problem in such a way that it can be solved
   - ...using a suitably abstract model
2. Solve it programmatically
   - ...on a distributed architecture



Streets 4 MPI

Project Goal

1. Simulate thousands or millions of cars/drivers in a city
2. Watch for congested and unused roads
3. Optimize the road system step by step
4. Visualize this process as a changing map

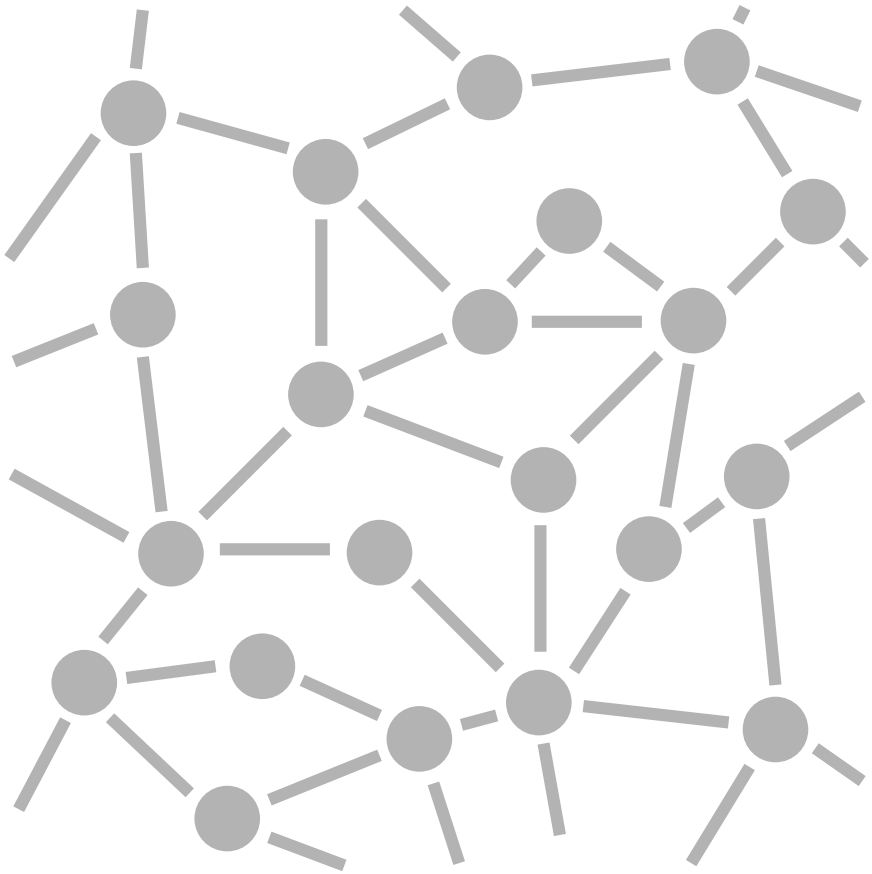
Macroscopic Simulation

- Abstract from single cars, traffic lights etc. to daily traffic
- Display traffic development over longer time periods and influences on street network

Discrete Simulation

- 1 simulation step $\triangleq$ 1 day
- Traffic changes every day
- Changes to street network after longer time periods

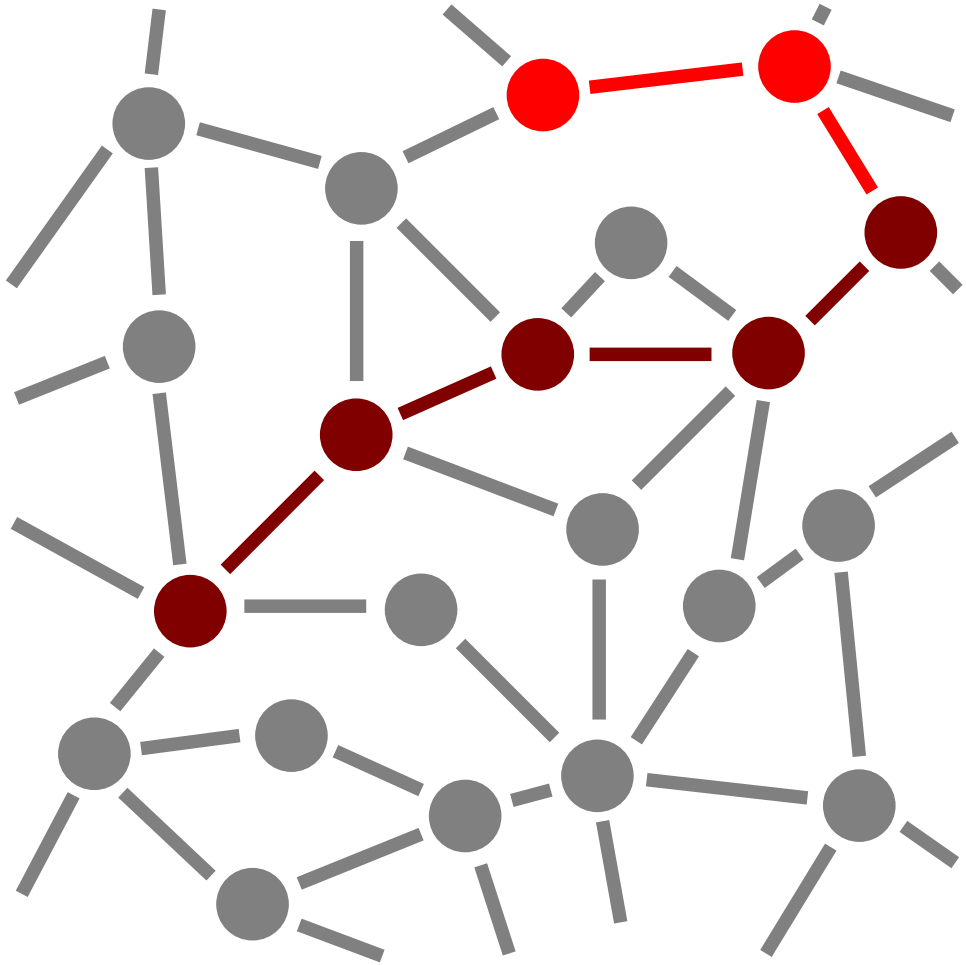
Street Network



Trips

- Representation for a resident's daily traffic
- Simplification:
  - No car locomotion
  - One route per day

Trips



Weights

Residents want to choose the fastest route
$\Rightarrow$ Weights based on driving time

Expected driving time

$$t_{expected} = \frac{s}{v_{max}}$$

Weights

- Cars are slowed down in case of increased traffic
- Decrease of velocity based on number of trips using a street
- Idea: Consider distribution of cars along the street and throughout the day

Weights

As a result the weights are increased and the route becomes less attractive

Actual driving time

$$t_{actual} = \frac{s}{v_{actual}}$$

Weights

In reality not all drivers change their route since driving time delay is percieved differently ⇒ Perceived driving time is influenced by random “traffic jam sensibility”

Traffic jam sensibility

$$t_{perceived} = t_{expected} + t_{delay} \cdot r_{sensibility}$$
$$t_{delay} = t_{actual} - t_{expected}$$

Runtime Technologies

- Python
- MPI (→**bonus slide**)
- mpi4py

Map Data from OpenStreetMap

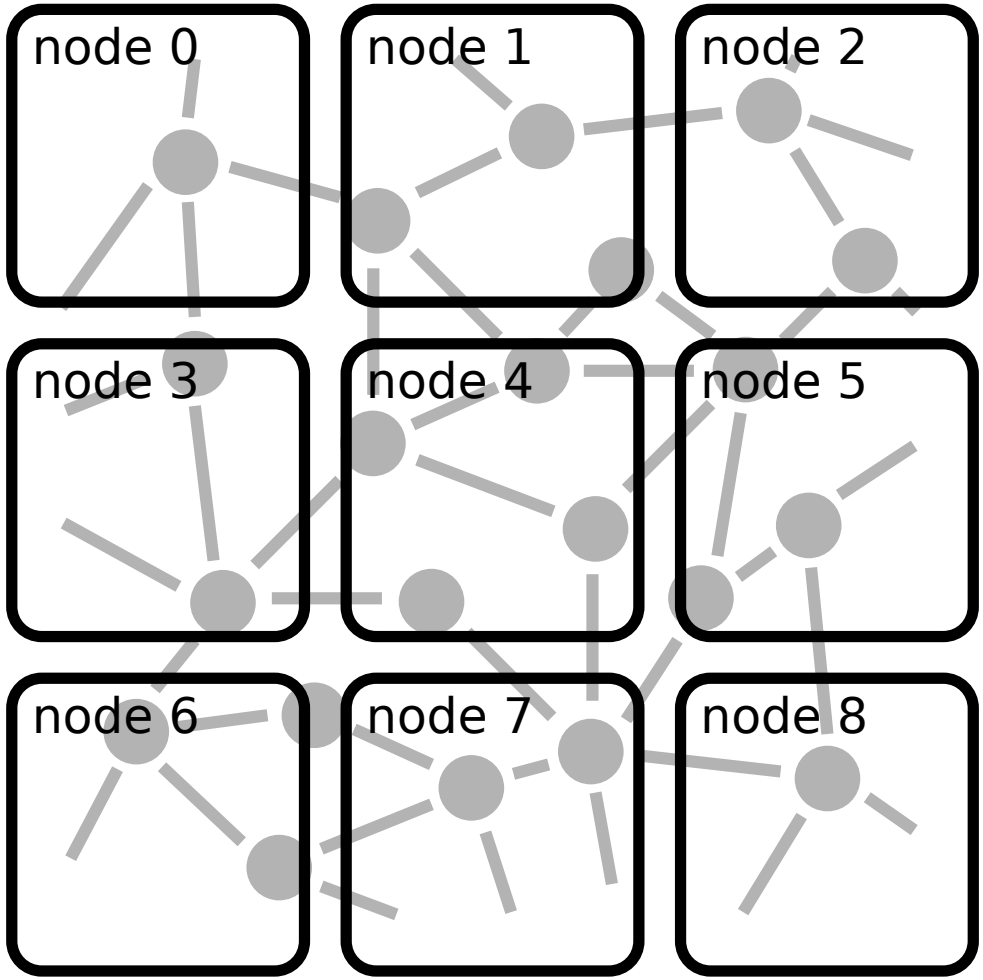
- Python and GIS
- OSM has a nice API
  - XML that includes nodes and ways

```
<?xml version="1.0" encoding="UTF-8"?>
<osm version="0.6" generator="CGImap 0.0.2">
 <bounds minlat="54.0889580" minlon="12.2487570" maxlat="54.0913900"
    maxlon="12.2524800"/>
 <node id="298884269" lat="54.0901746" lon="12.2482632" user="SvenHRO"
    uid="46882" visible="true" version="1" changeset="676636"
    timestamp="2008-09-21T21:37:45Z"/>
 <node id="261728686" lat="54.0906309" lon="12.2441924" user="PikoWinter"
    uid="36744" visible="true" version="1" changeset="323878"
    timestamp="2008-05-03T13:39:23Z"/>
 ...
 <node id="298884272" lat="54.0901447" lon="12.2516513" user="SvenHRO"
   uid="46882" visible="true" version="1" changeset="676636"
   timestamp="2008-09-21T21:37:45Z"/>
 <way id="26659127" user="Masch" uid="55988" visible="true"
   version="5" changeset="4142606" timestamp="2010-03-16T11:47:08Z">
  <nd ref="292403538"/>
  <nd ref="298884289"/>
  ...
  <nd ref="261728686"/>
  <tag k="highway" v="unclassified"/>
  <tag k="name" v="Pastower Straße"/>
 </way>
 ...
</osm>
```

Development Infrastructure

- Git
- MediaWiki

Idea for Data Decomposition



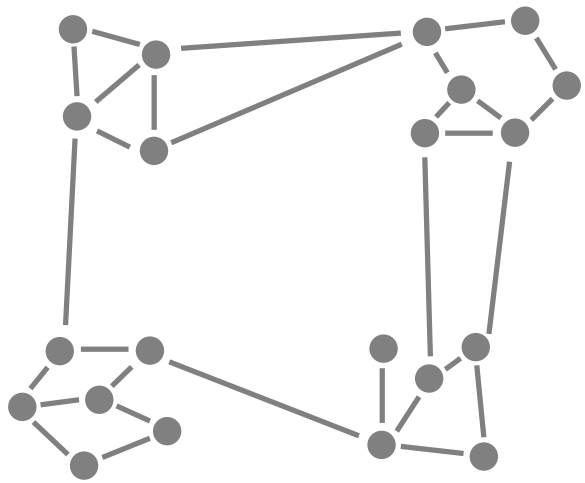
Computational Tasks

- Excluding: primary preparations, visualization
- Shuffling with times and velocities $\rightarrow$ mostly trivial
- Finding the shortest paths $\rightarrow$ **bulk of calculation**

Distributed Shortest Path

- Cost-effectiveness of specialized algorithms?
- Balanced hierarchical networks (Antonio, Huang, Tsai)
- Other approaches

Balanced hierarchical networks



- Networks with hierarchically organized clusters
- Strategy: calculate paths along “gate nodes”
- Potential reduction in complexity ($O(\log n)$ possible for single path)
- Data distribution along cluster boundaries?

Most Important Points

- Simple traffic simulation
- Macro level with congestion analysis, street development
- MPI on Python
- Distributed shortest path

Literature

Weber, B.; Müller, P.; Wonka, P.; Gross, M.:
Interactive Geometric Simulation of 4D Cities
In: EUROGRAPHICS 28 (2009), Nr. 2

Antonio, J. K.; Huang, G. M.; Tsai, W. K.:
A fast distributed shortest path algorithm for a class of hierarchically clustered data networks
In: IEEE Trans. Comput., vol. 41, pp. 710–724, June 1992

Weblinks

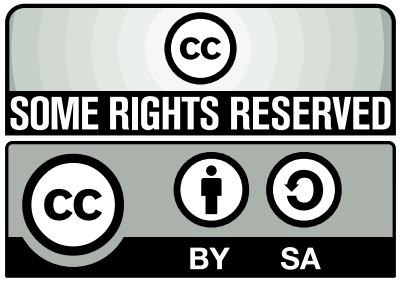
Project wiki

http://pwiki.julian-fietkau.de/

GitHub repository

http://github.com/jfietkau/Streets4MPI

Download and Usage



Download these slides and give feedback:

http://www.julian-fietkau.de/parallel_programming_plan