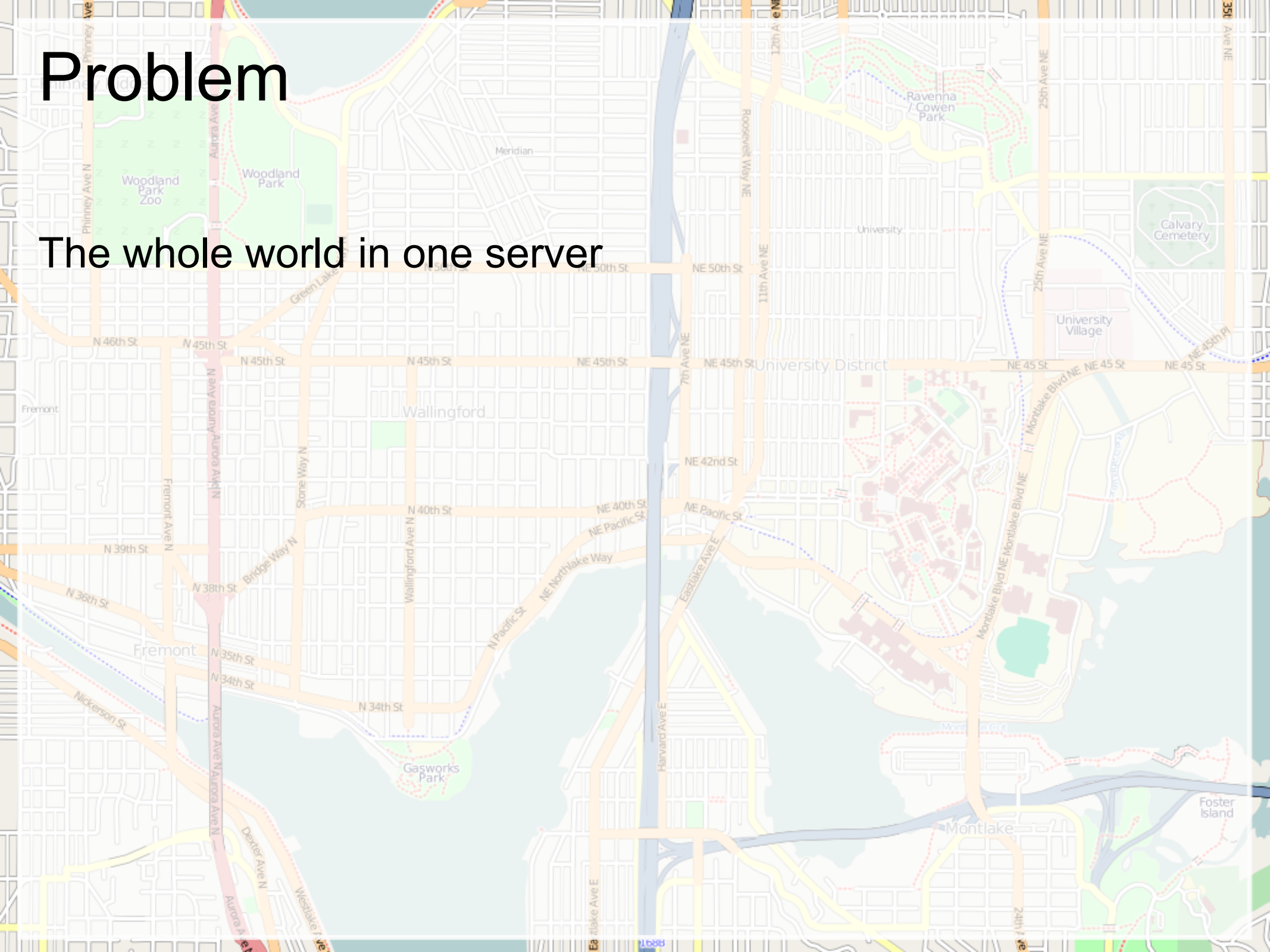# Menzies

Distributing the world.

Problem

The whole world in one server

API

GET **node/#id** *Returns the XML for that node.*
PUT **node/#id** *Updates the node, returns new version number.*
DELETE **node/#id** *Deletes the node, returns new version number(?).*
PUT **node/create** *Creates the node, returns new node number.*
GET **node/#id/history** *Returns all versions of the node.*
GET **node/#id/#version** *Returns the XML for that version of the node.*
GET **node/#id/ways** *Returns the XML for all ways that this node is part of.*
GET **node/#id/relations** *Returns the XML for all relations that this node is part of.*
GET **nodes?nodes=#id,#id,...** *Returns the XML for all given node numbers.*
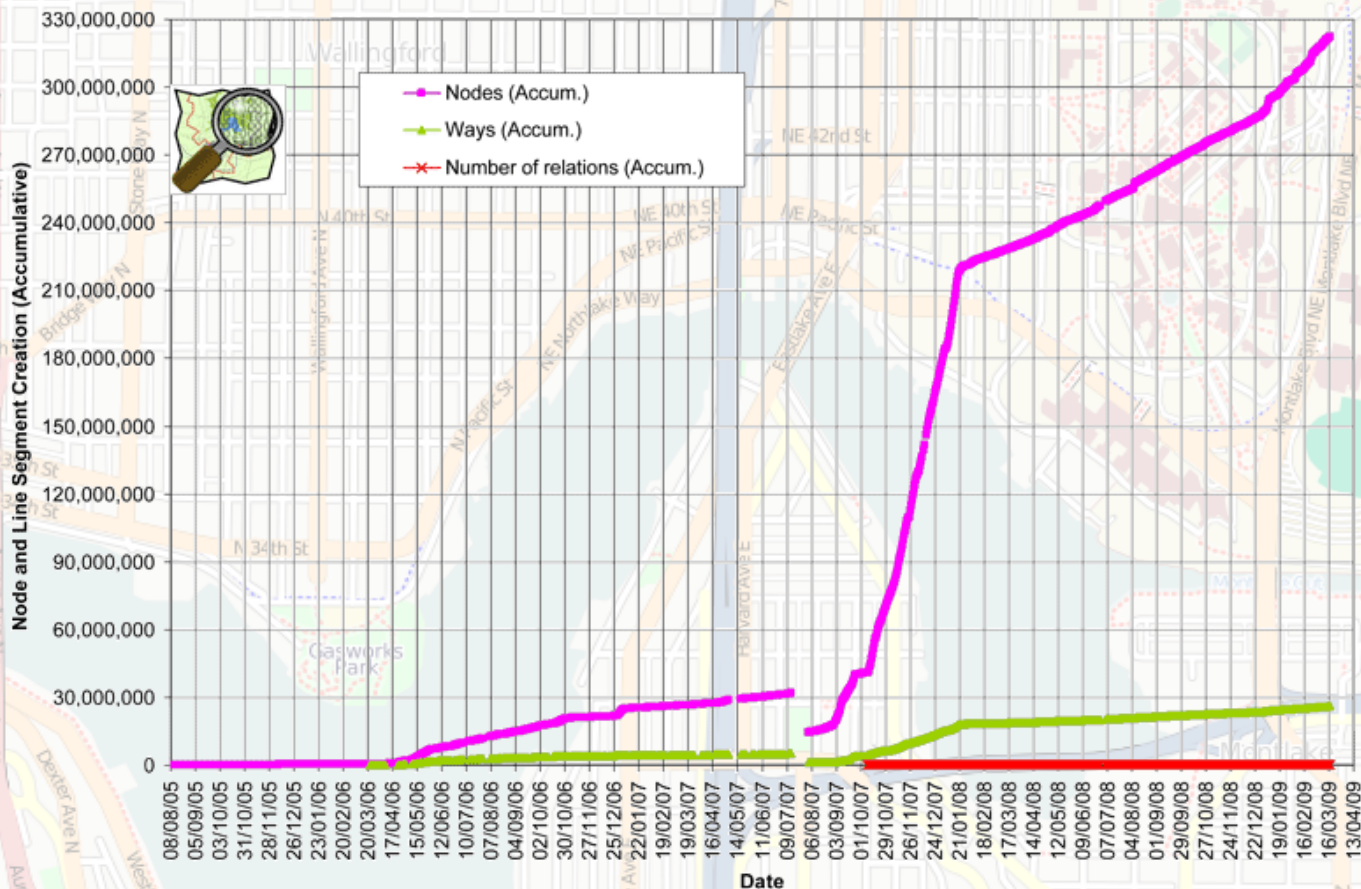
GET **map?bbox=min_lat,min_lon,max_lat,max_lon** *Gets all the way, nodes and relations inside a bounding box*

Challenges

Optimize bounding box.
Distribute data amongst servers.

OpenStreetMap Database Statistics
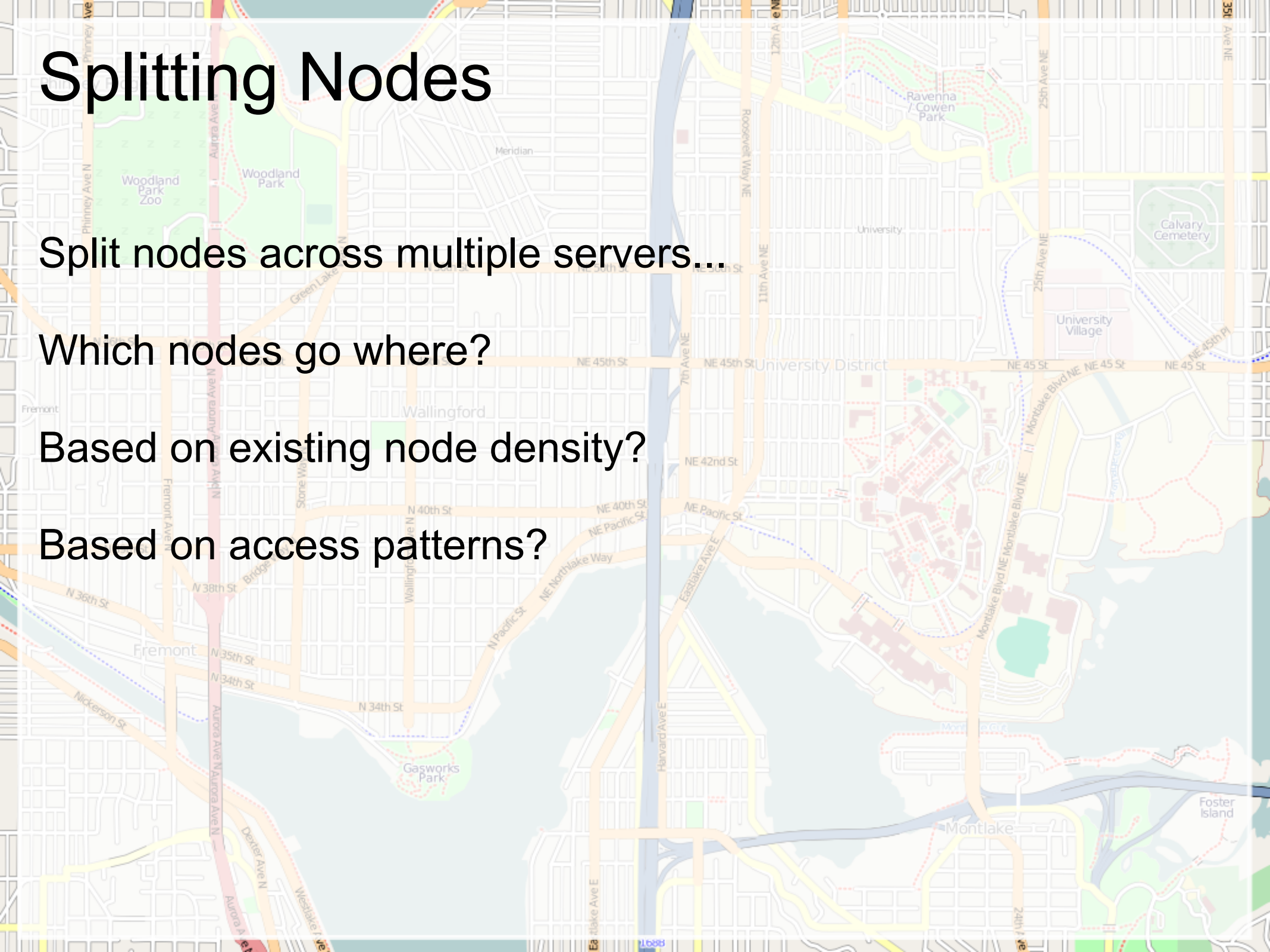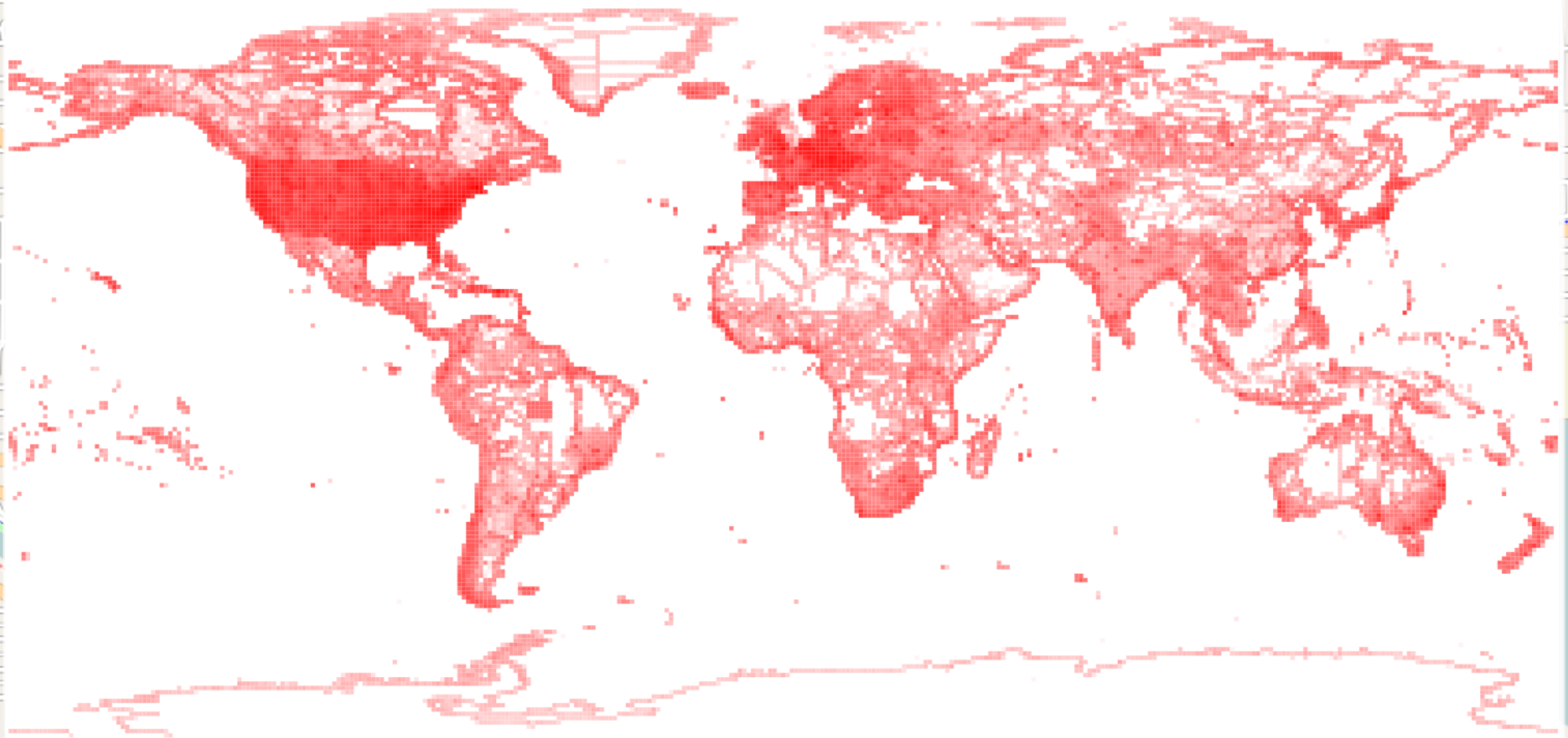Node, Way and Relation Creation

Splitting Nodes

Split nodes across multiple servers...
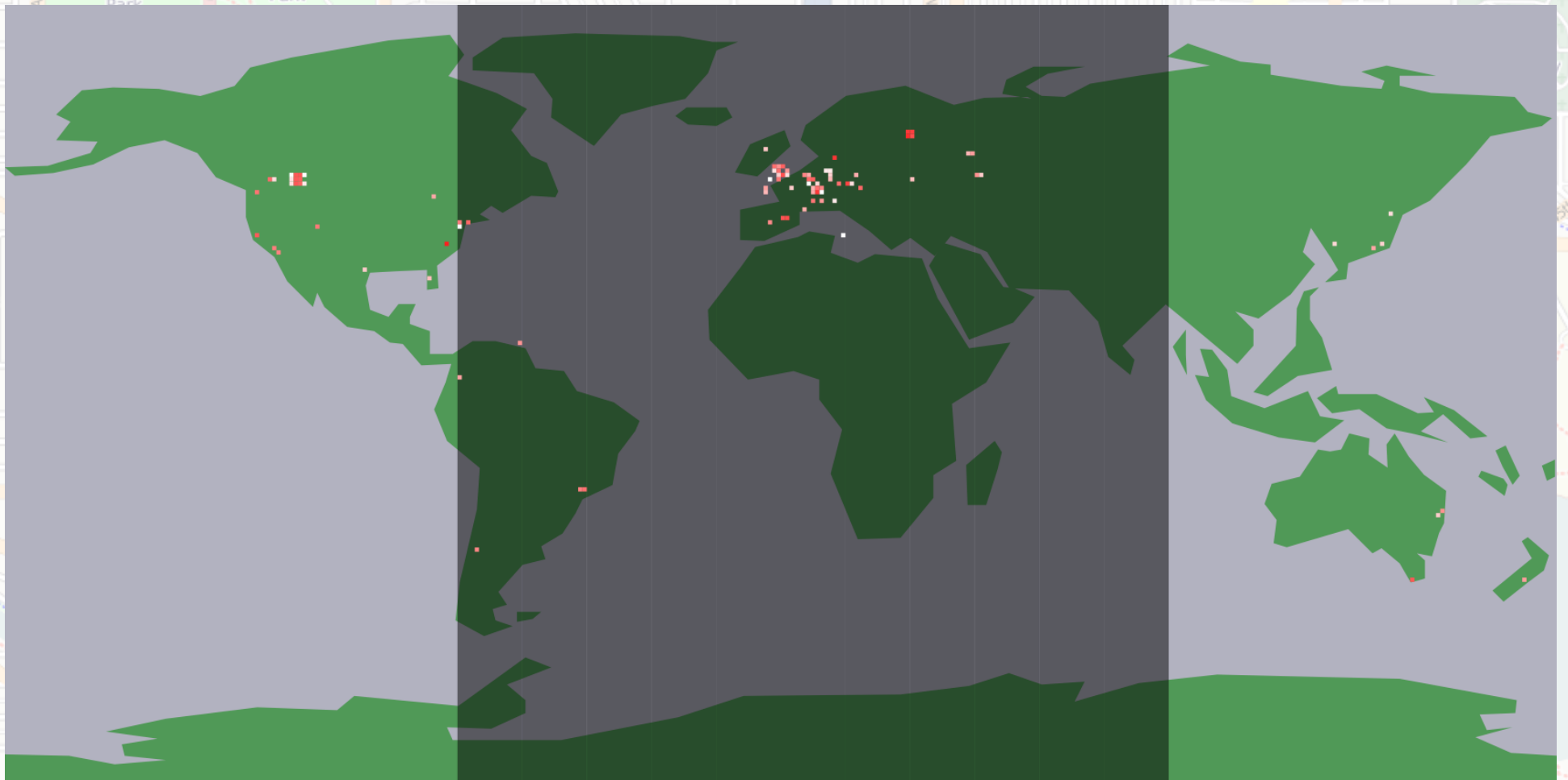
Which nodes go where?

Based on existing node density?
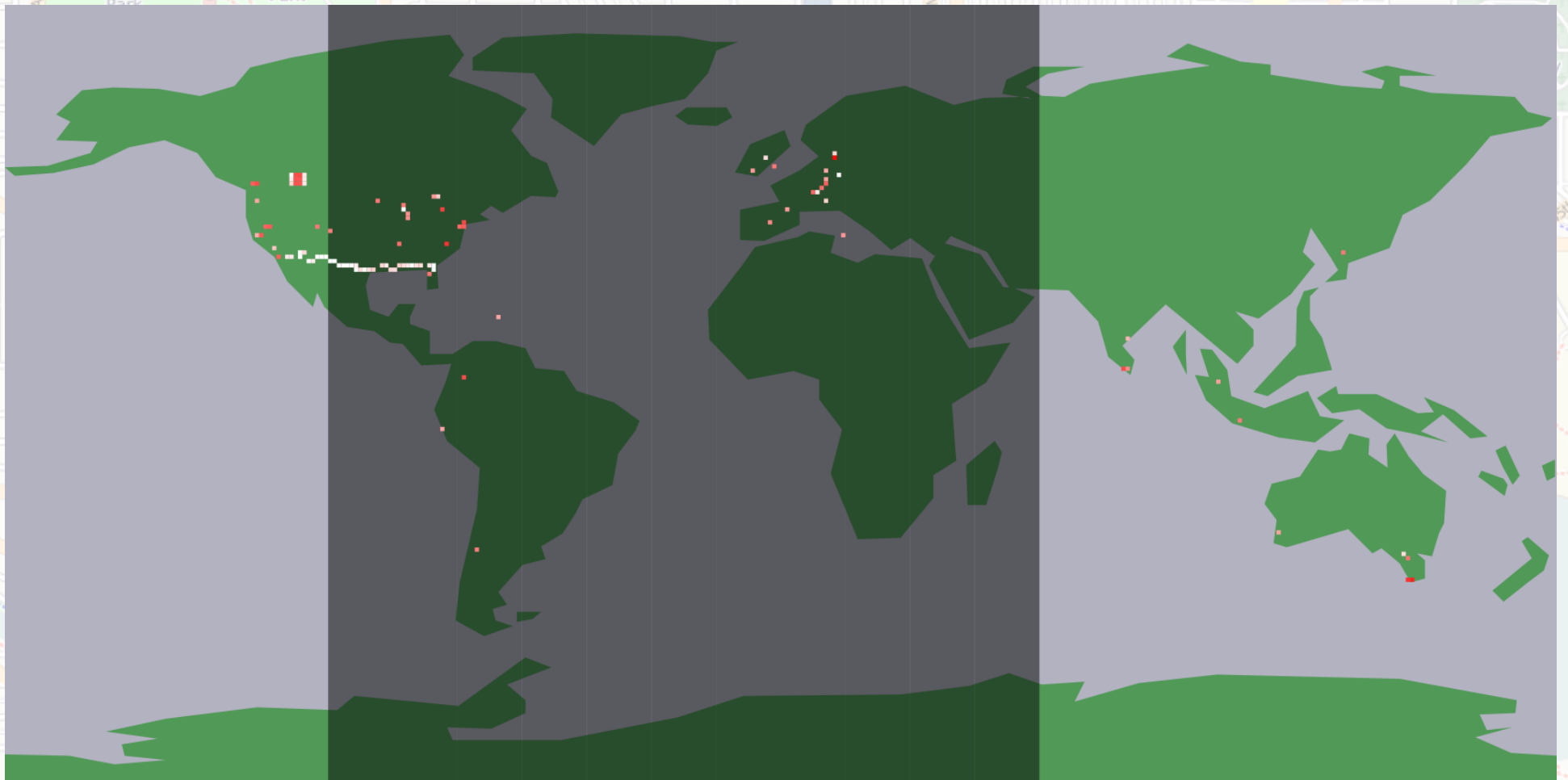
Based on access patterns?
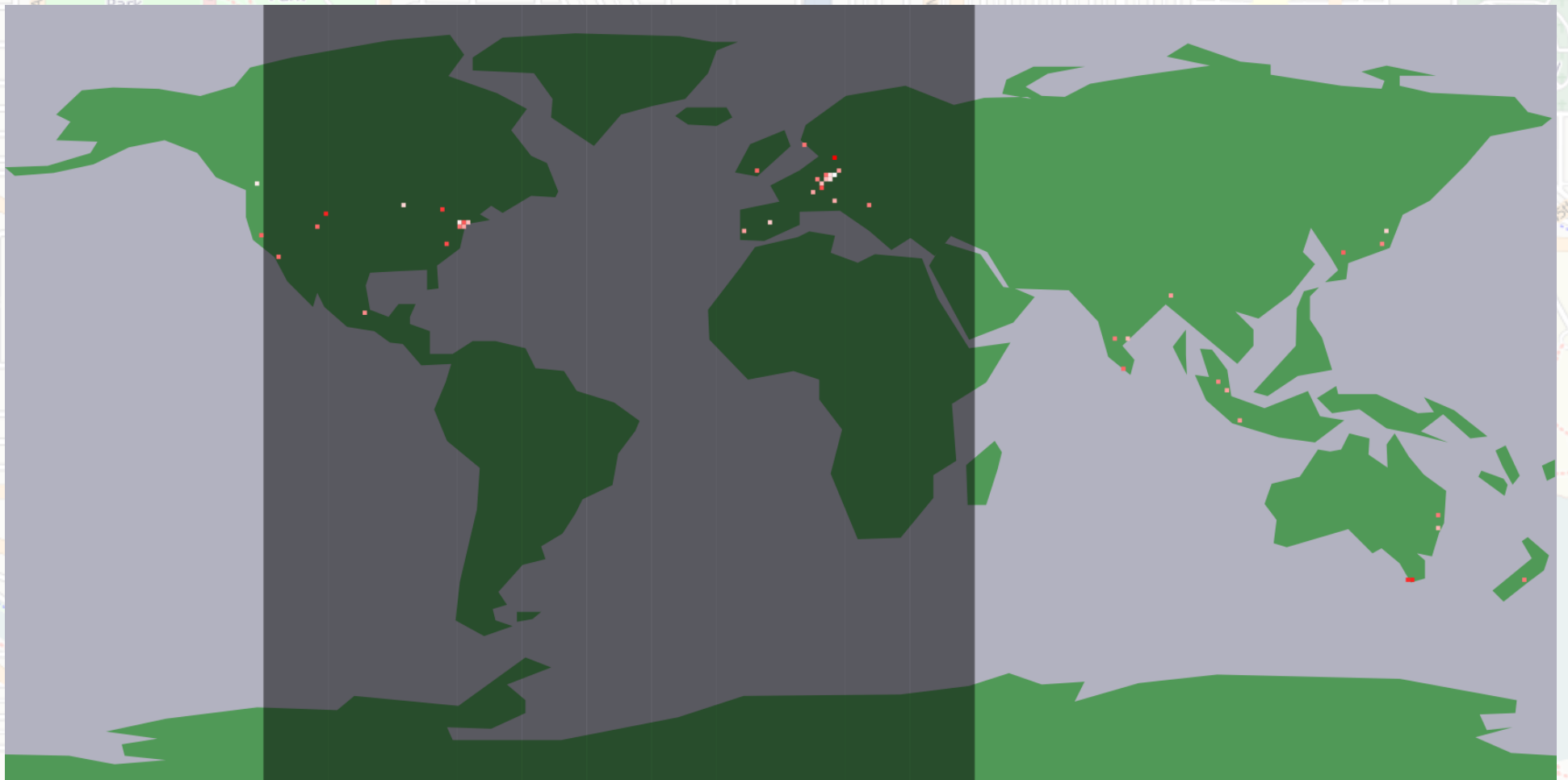
Node Density

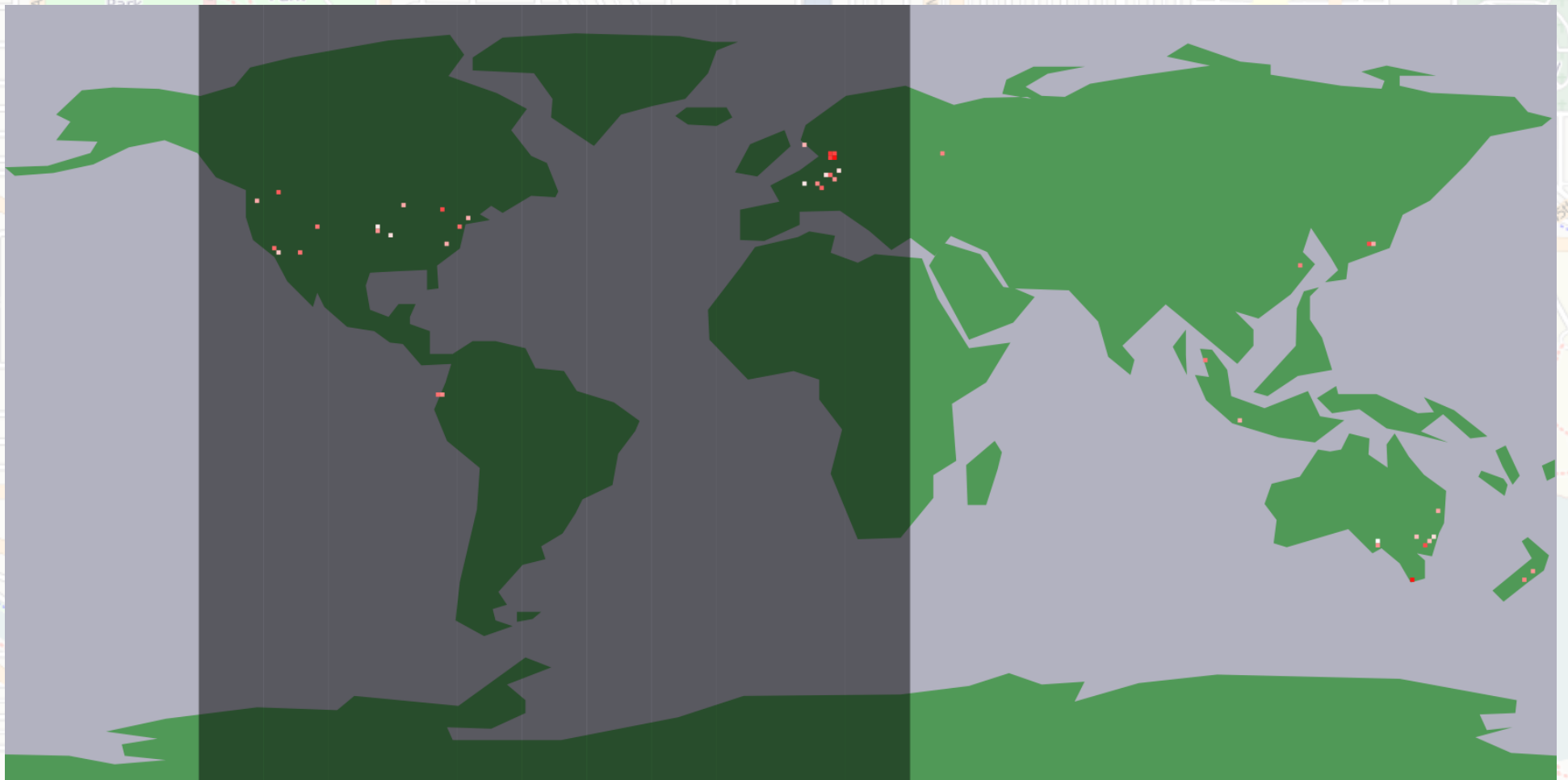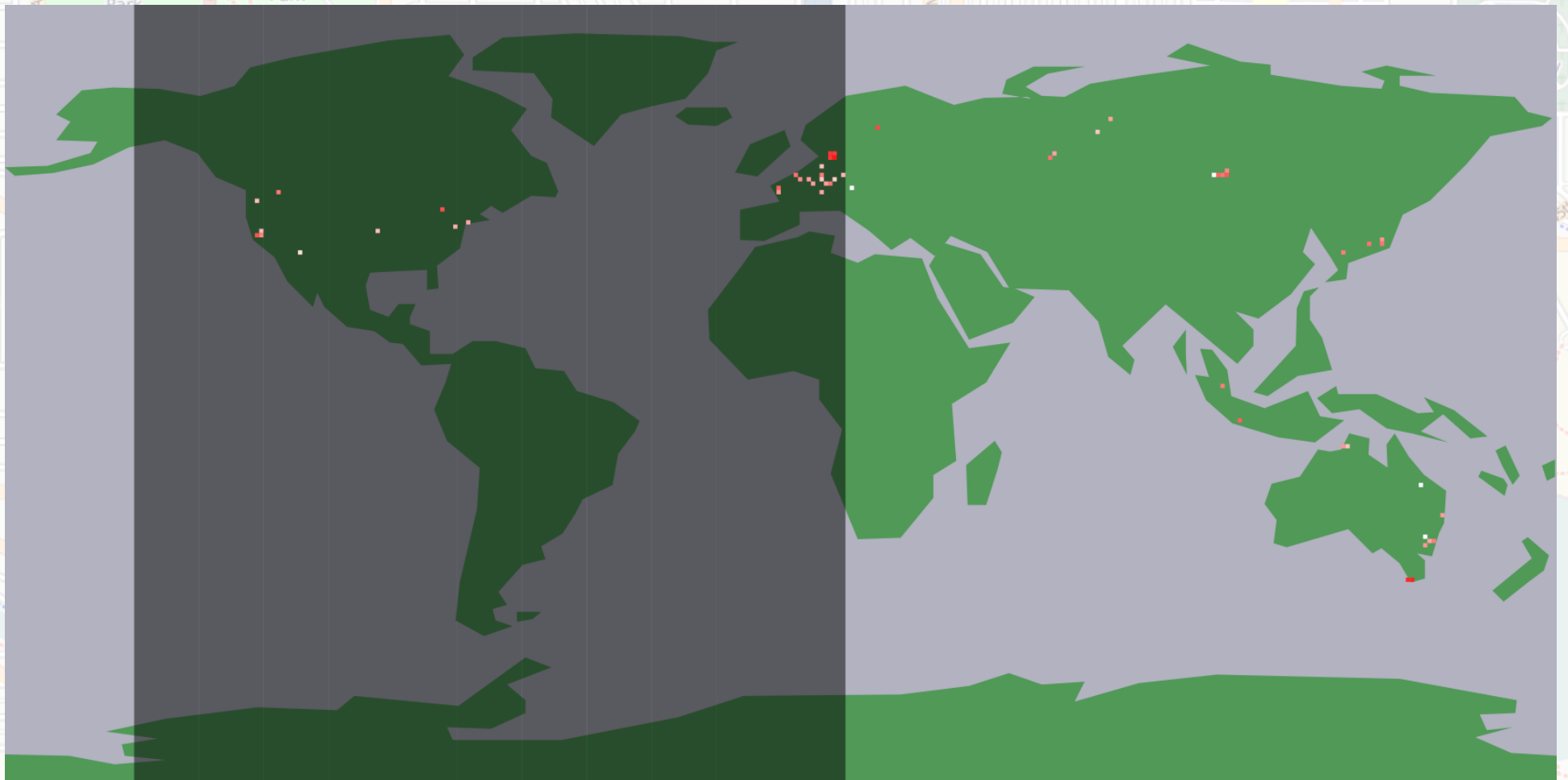Editing @ Midnight UTC
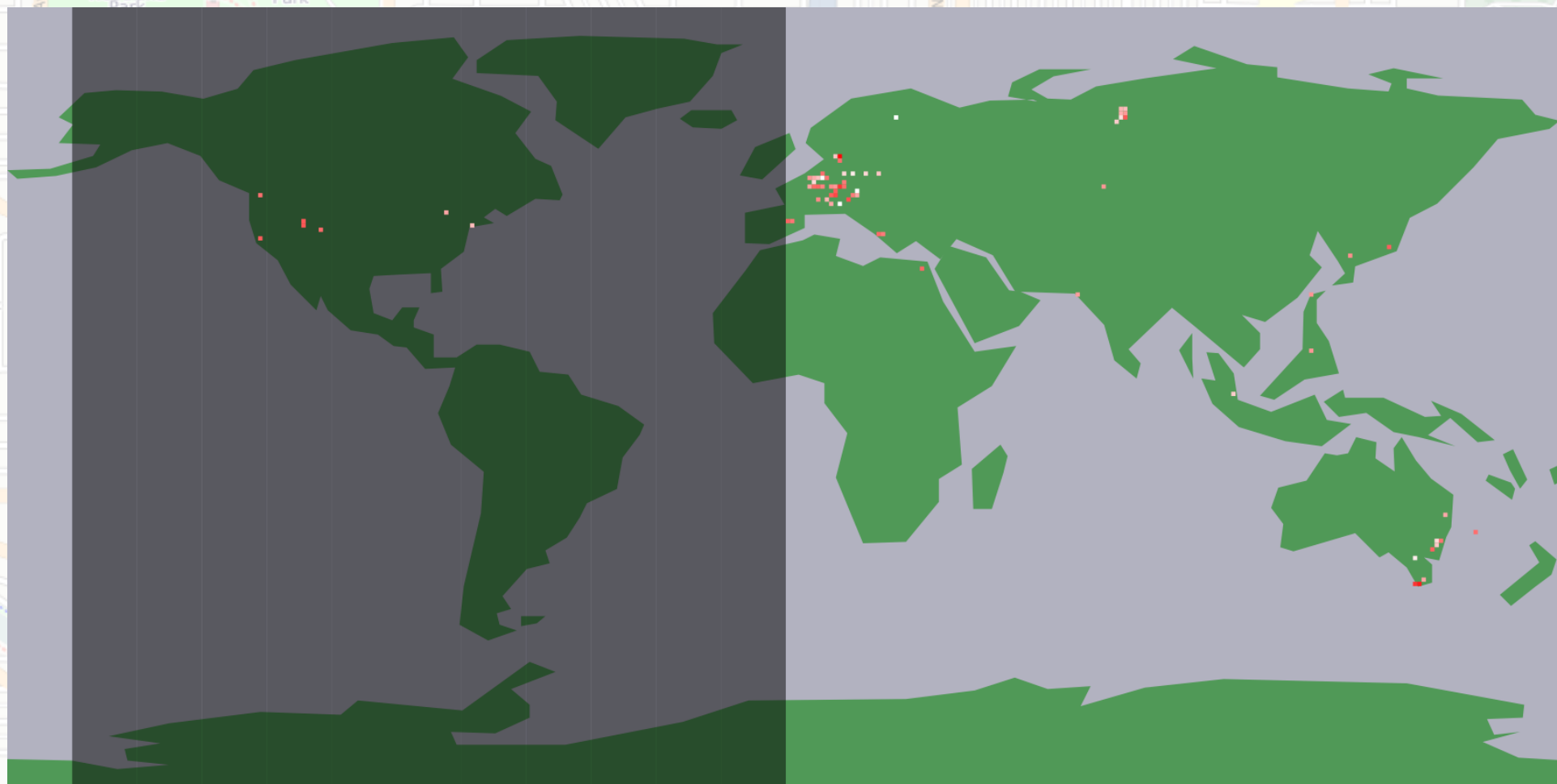
Editing @ 1AM UTC
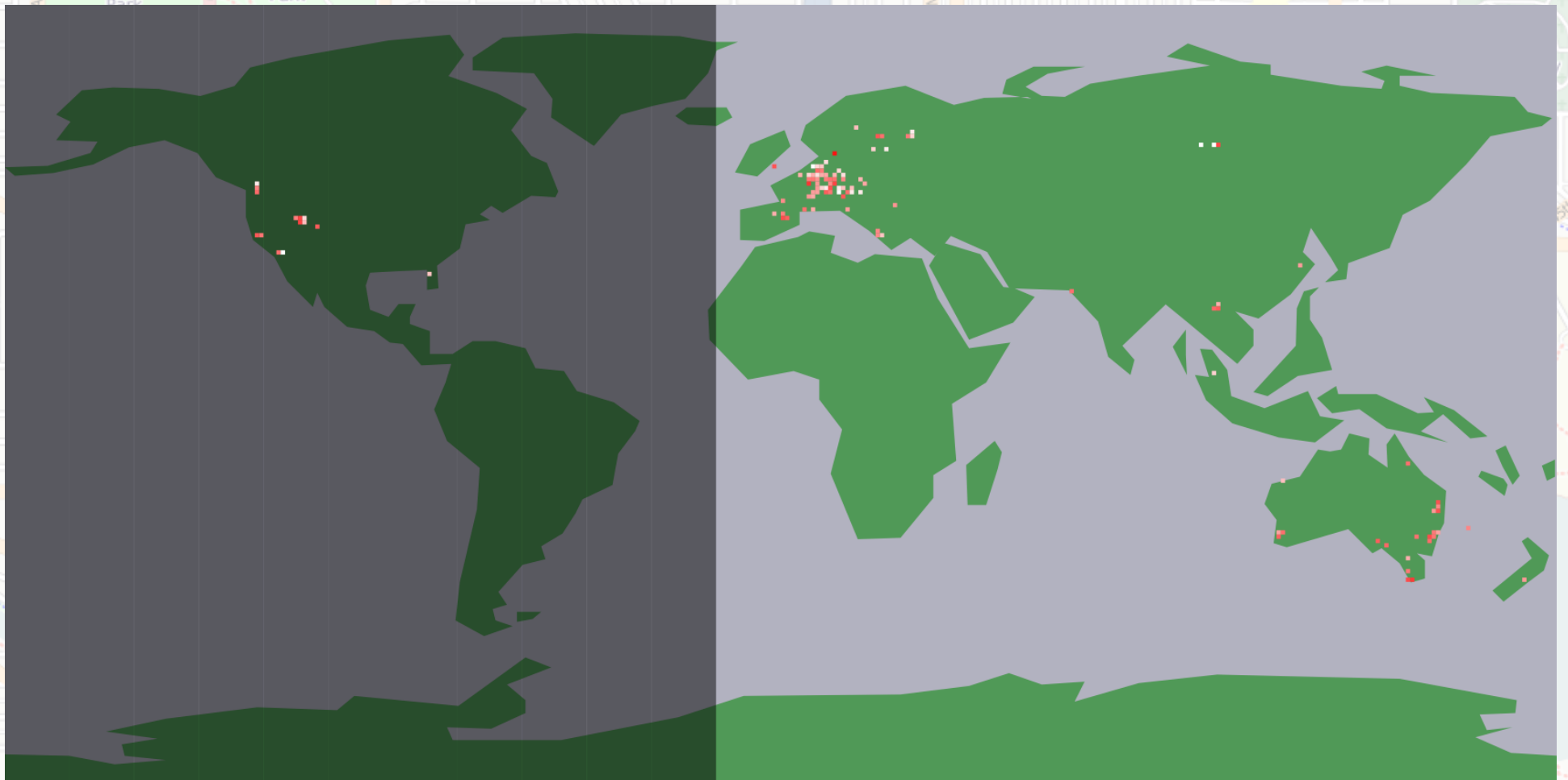
Editing @ 2AM UTC

Editing @ 3AM UTC

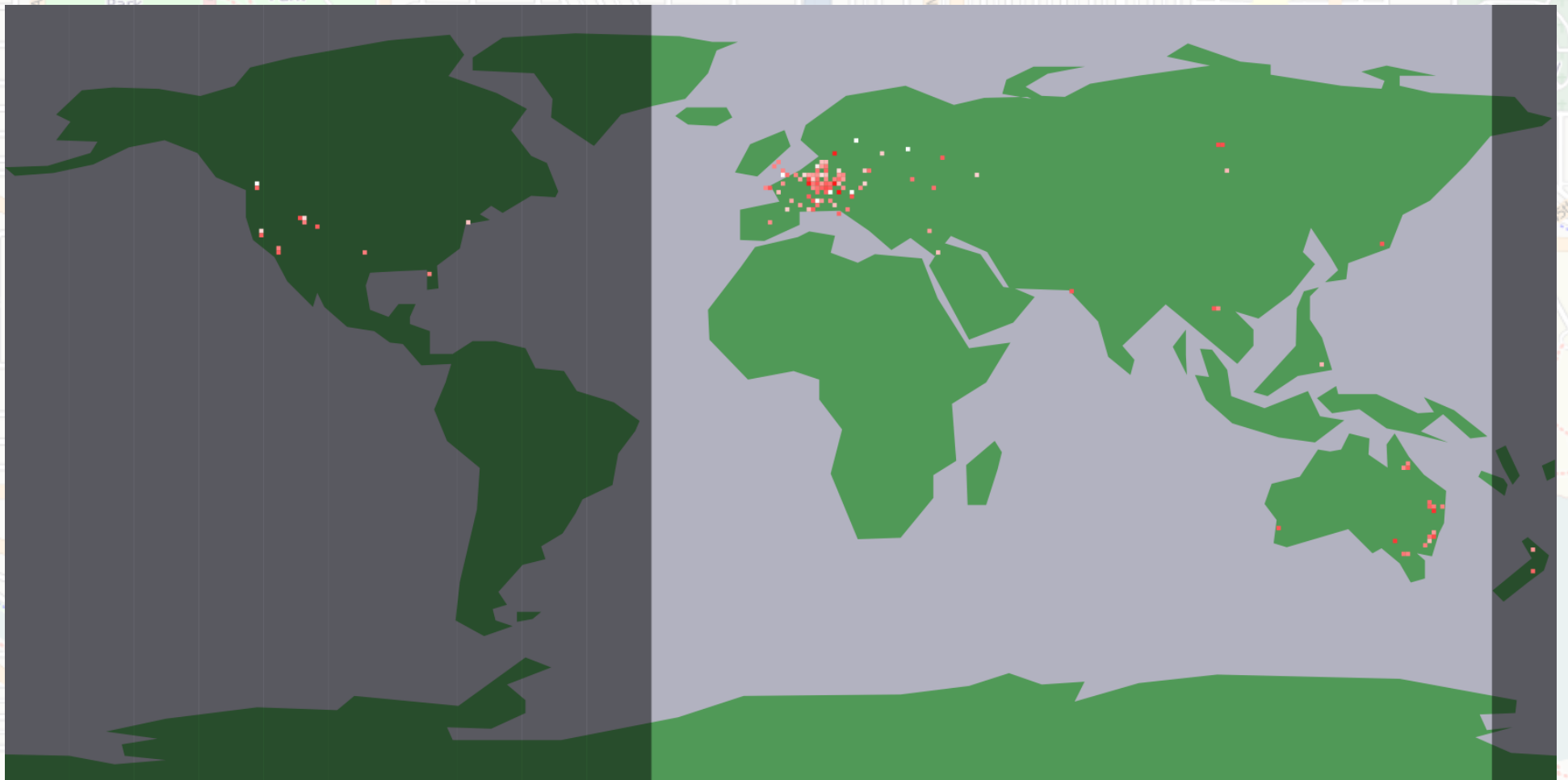Editing @ 4AM UTC

Editing @ 5AM UTC
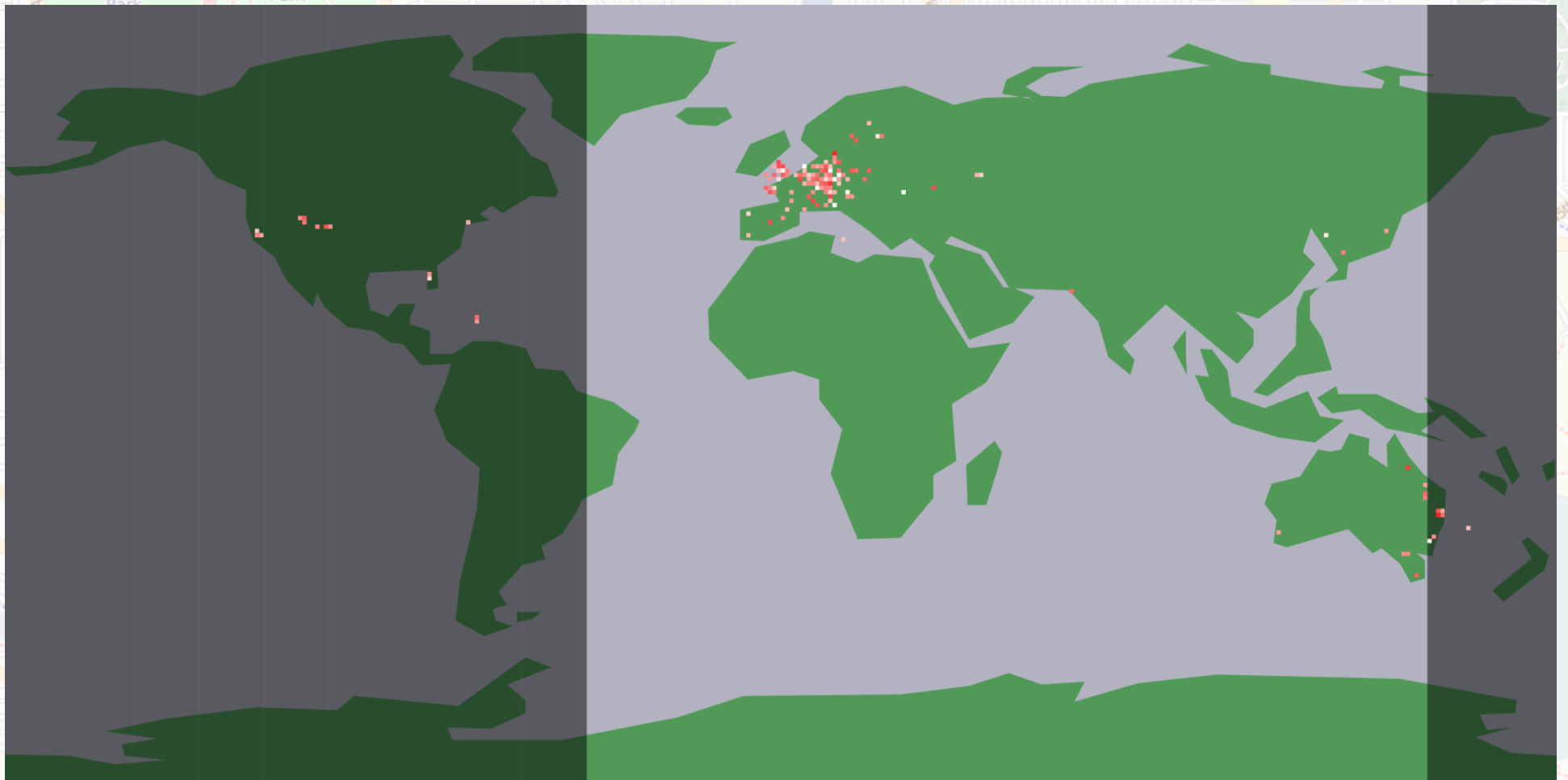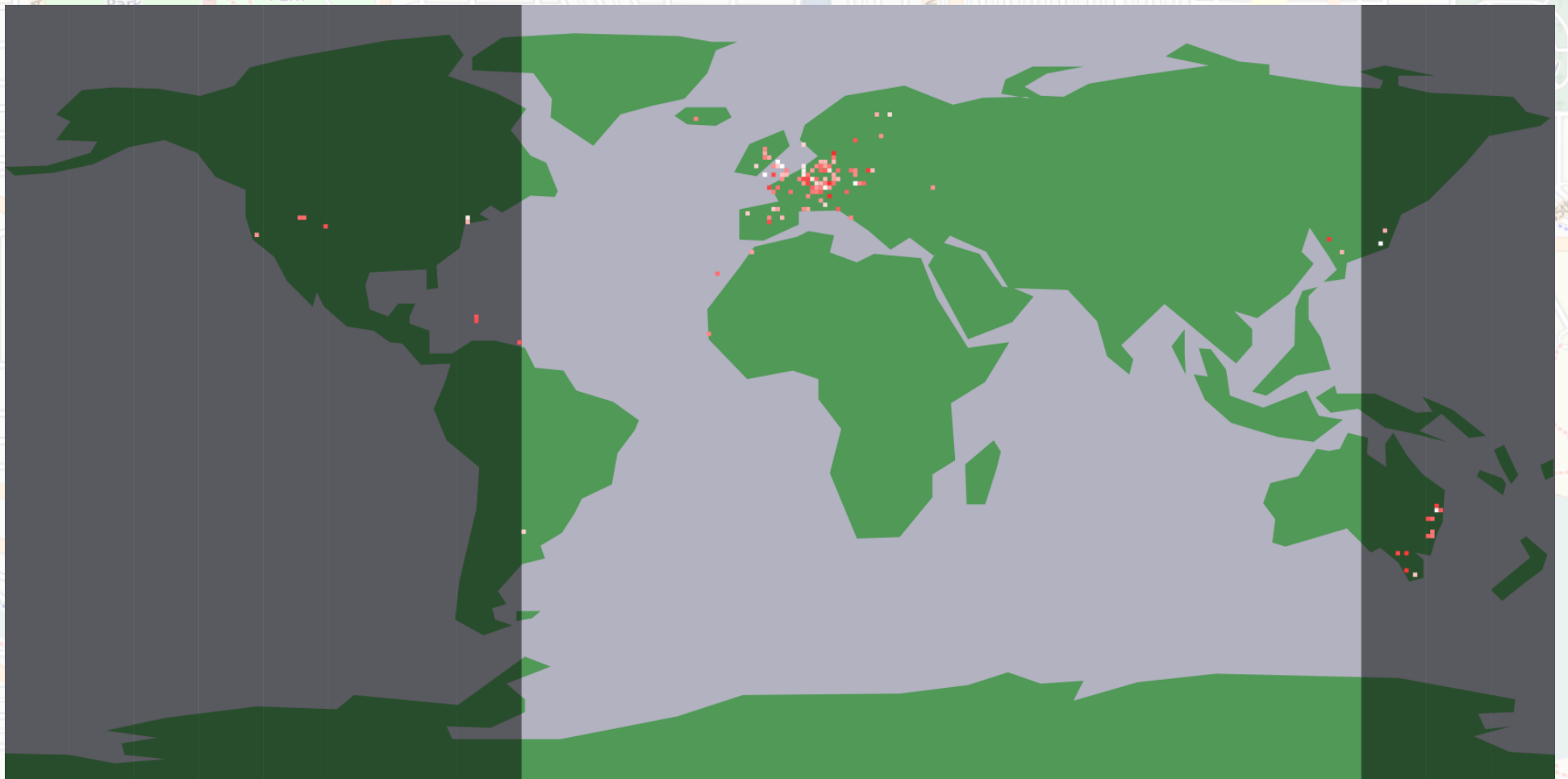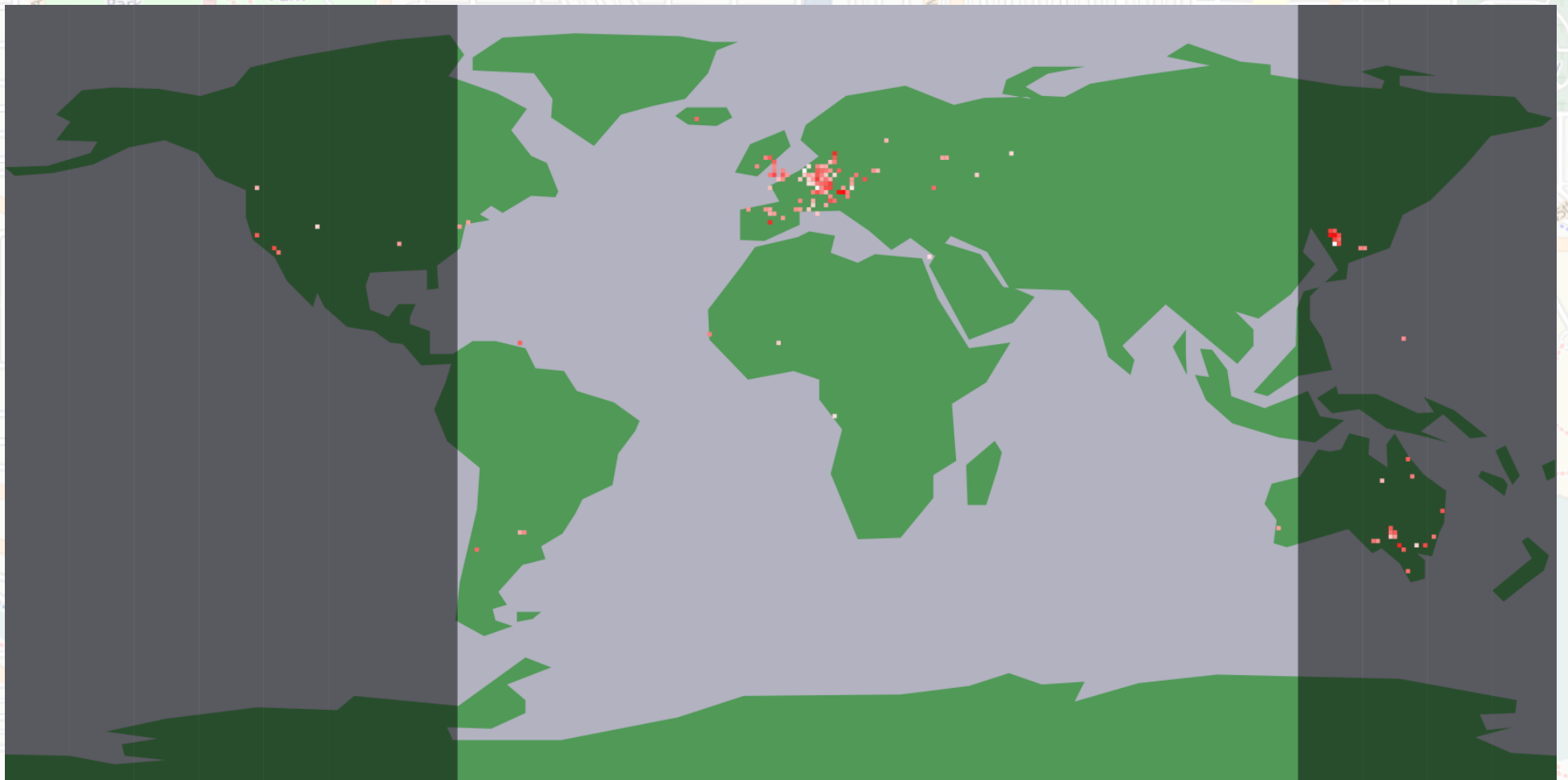
Editing @ 6AM UTC

Editing @ 7AM UTC

Editing @ 8AM UTC

Editing @ 9AM UTC

Editing @ 10AM UTC

Editing @ 11AM UTC
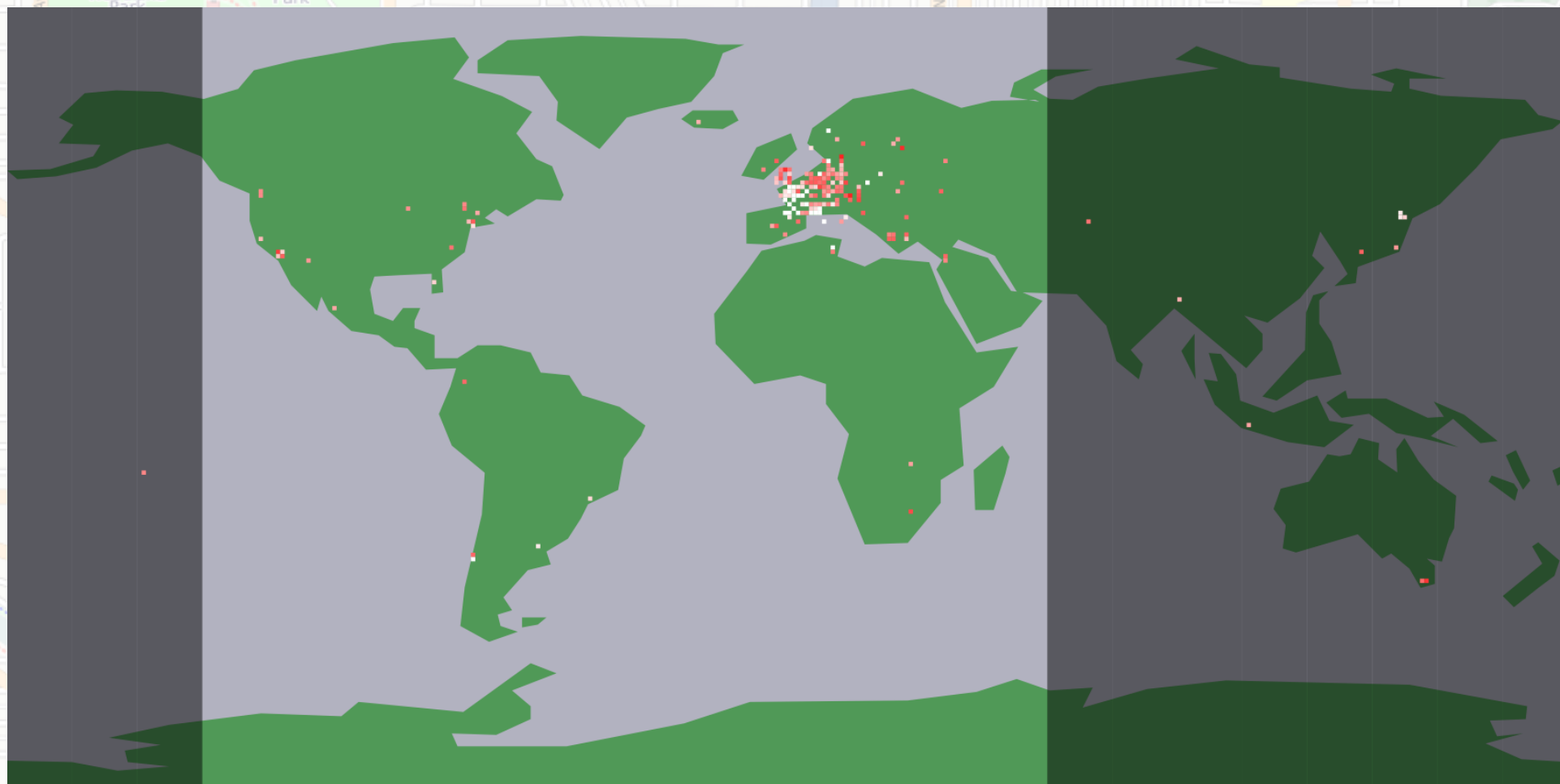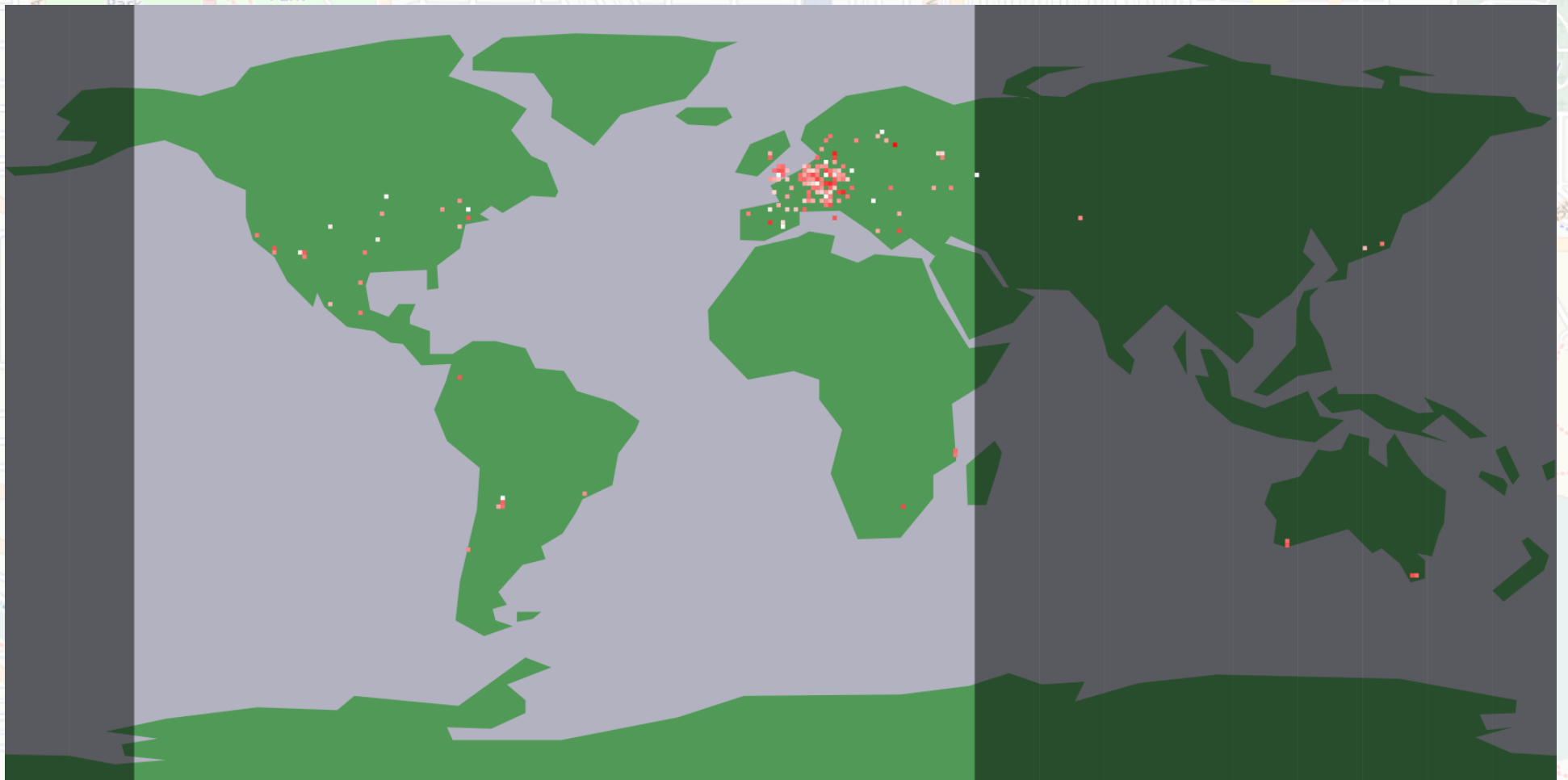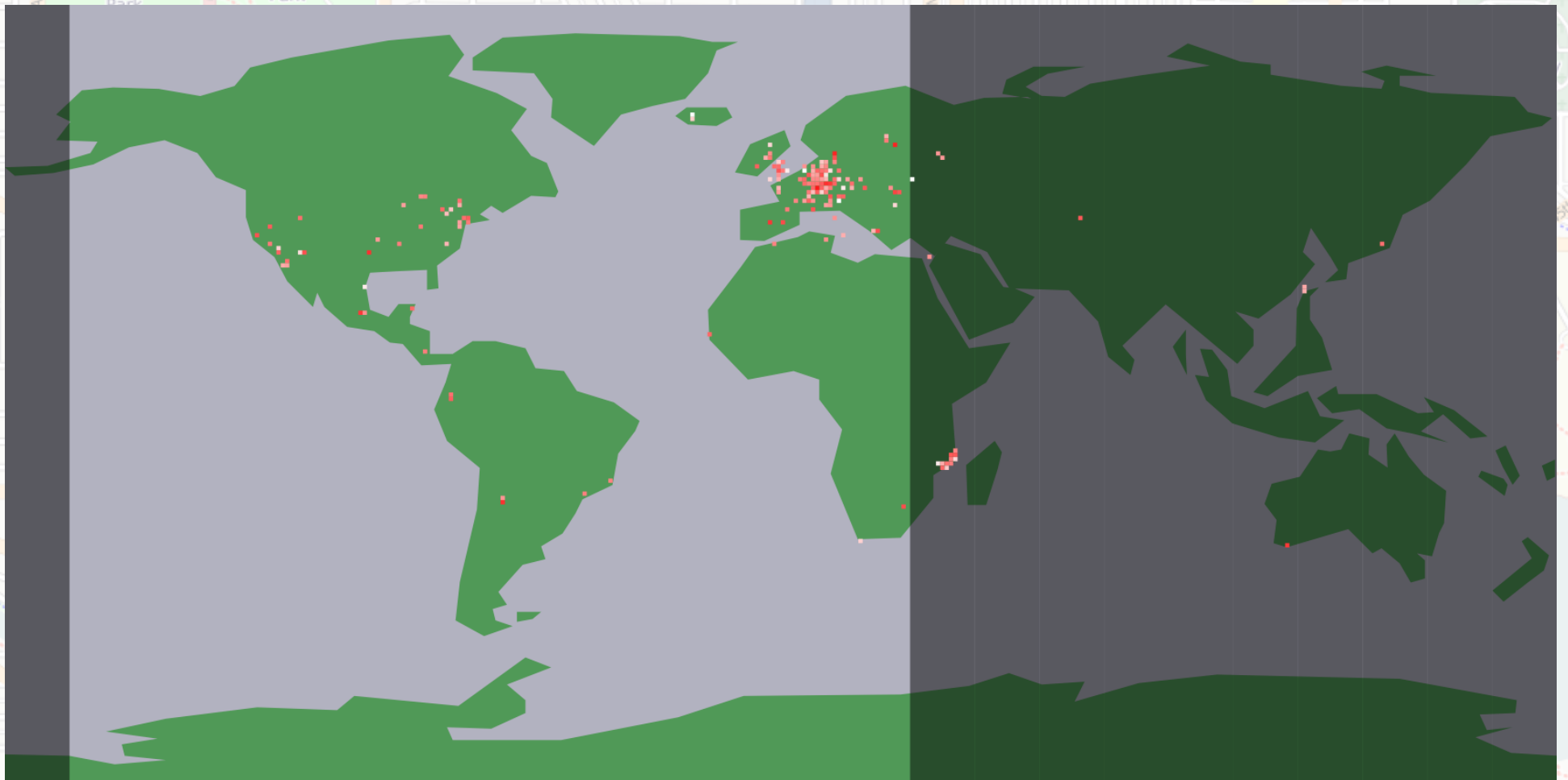
Editing @ Noon UTC

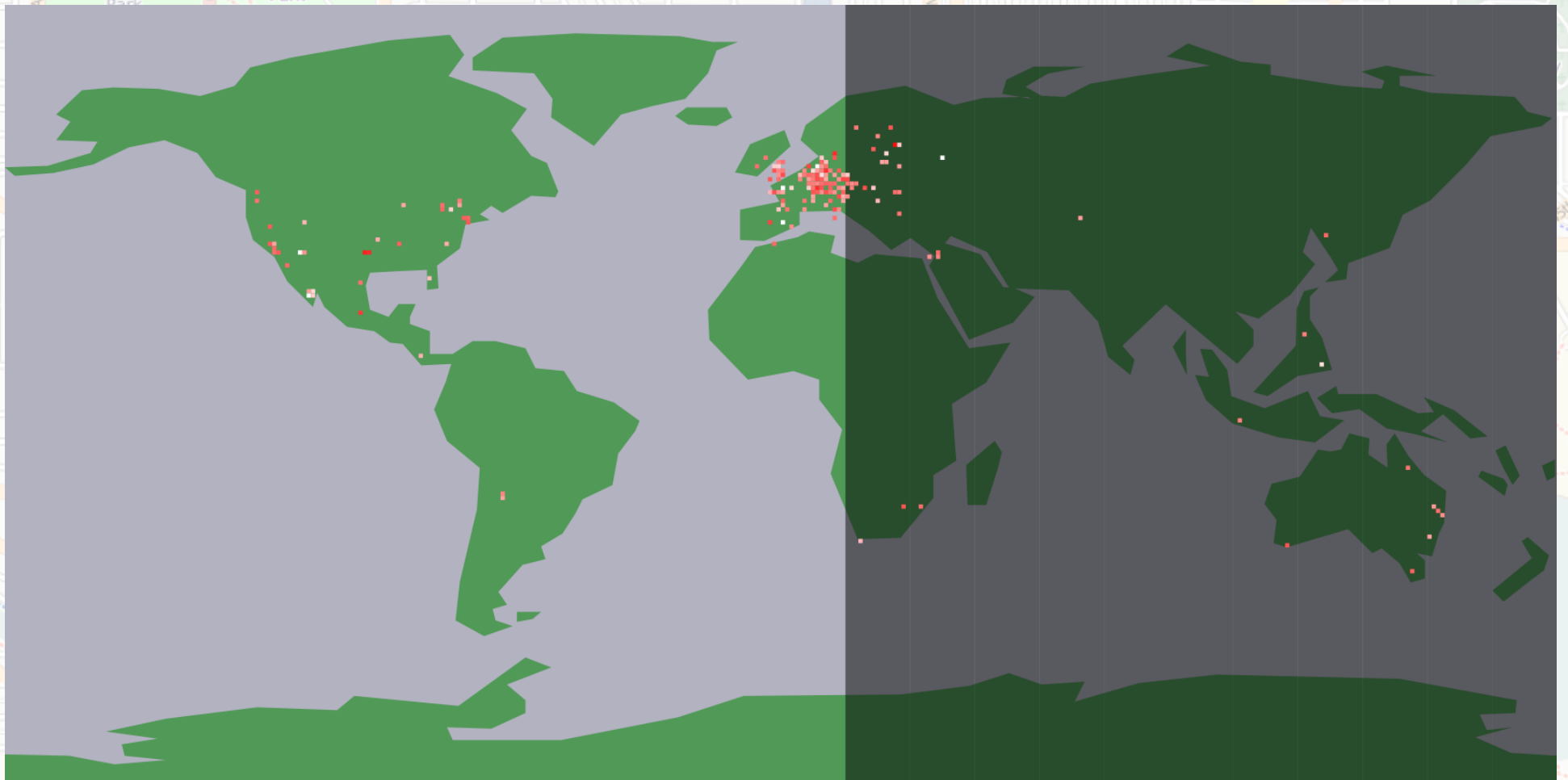Editing @ 1PM UTC

Editing @ 2PM UTC
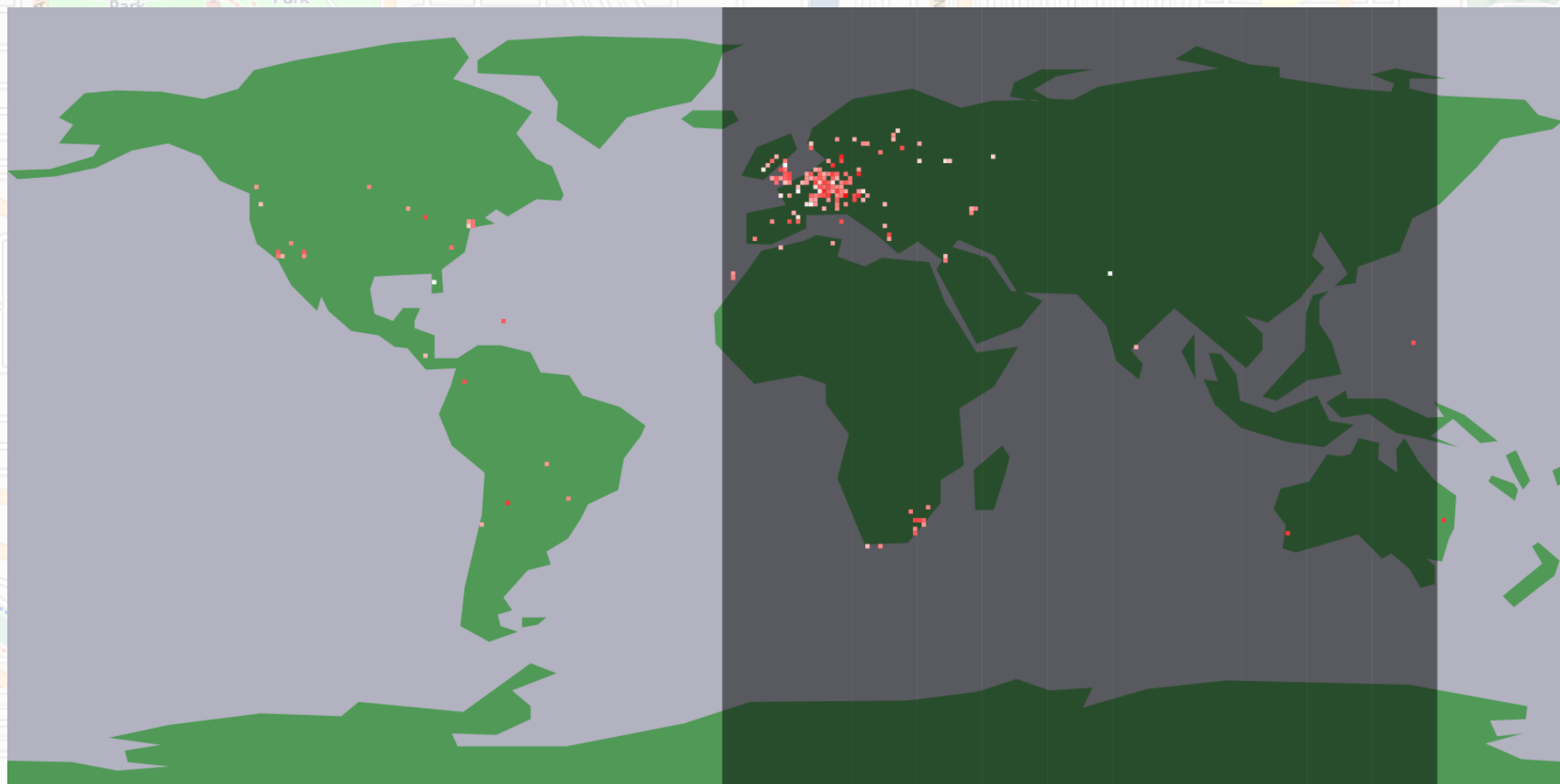
Editing @ 3PM UTC

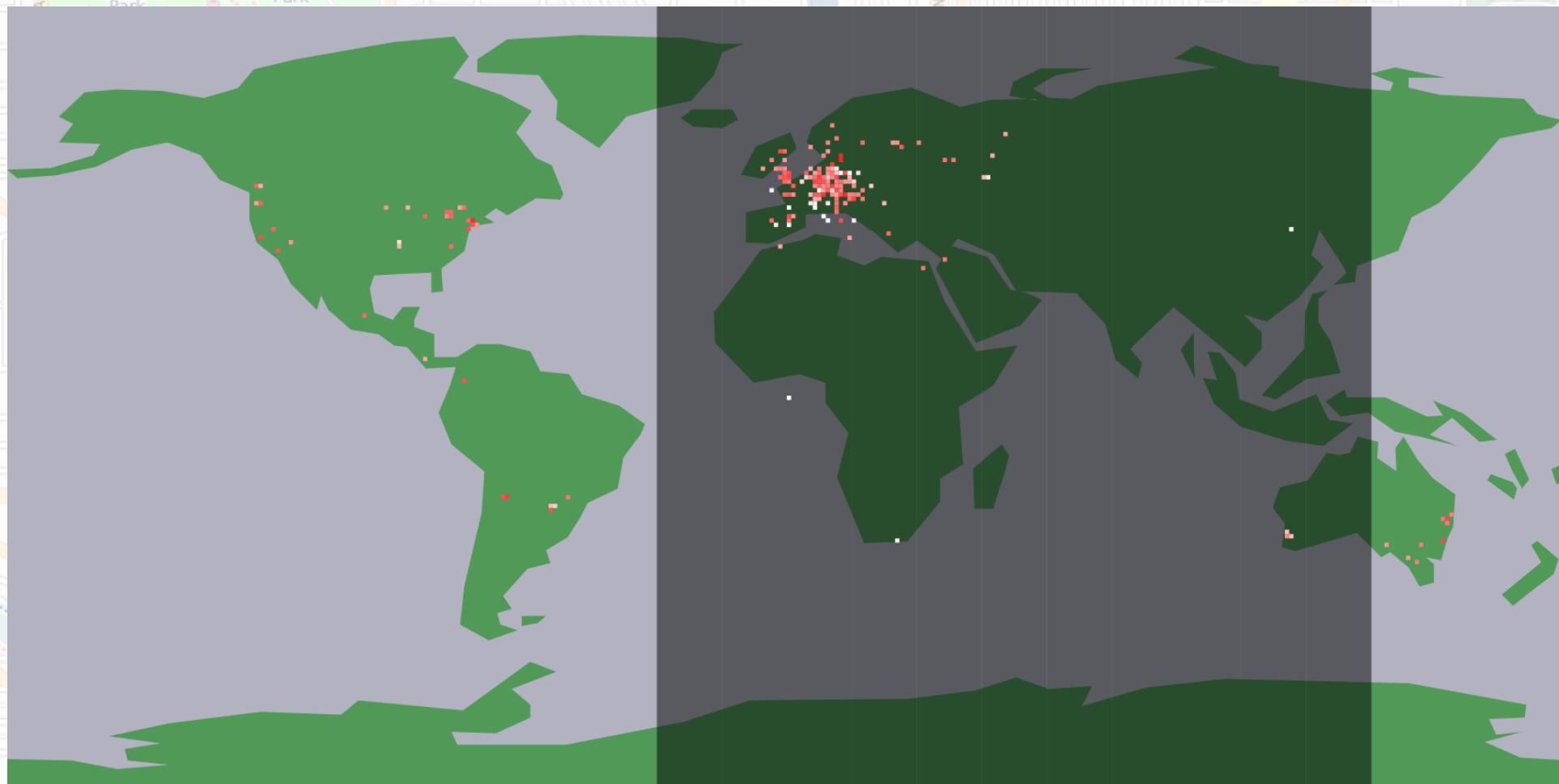Editing @ 4PM UTC

Editing @ 5PM UTC
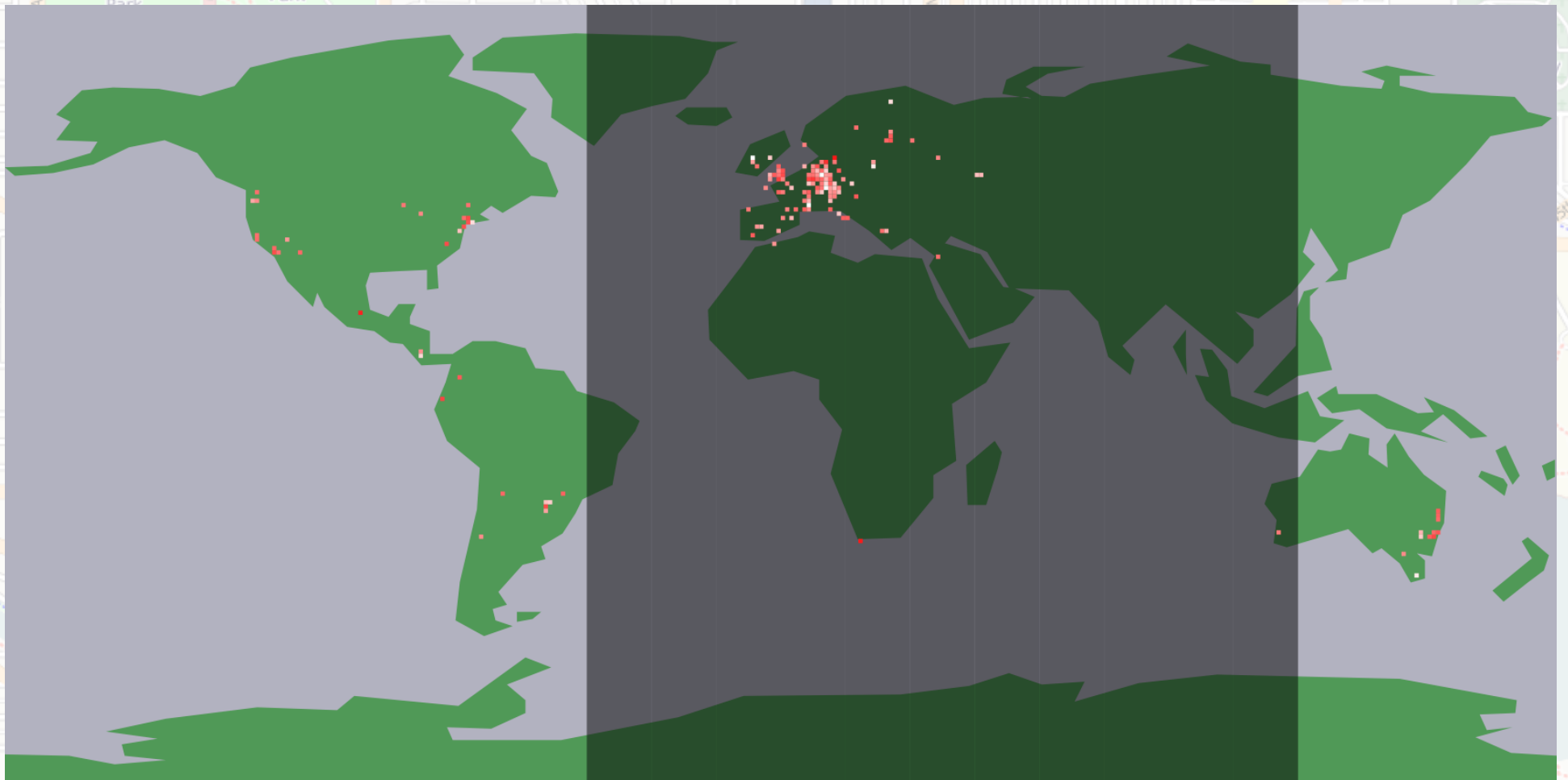
Editing @ 6PM UTC

Editing @ 7PM UTC
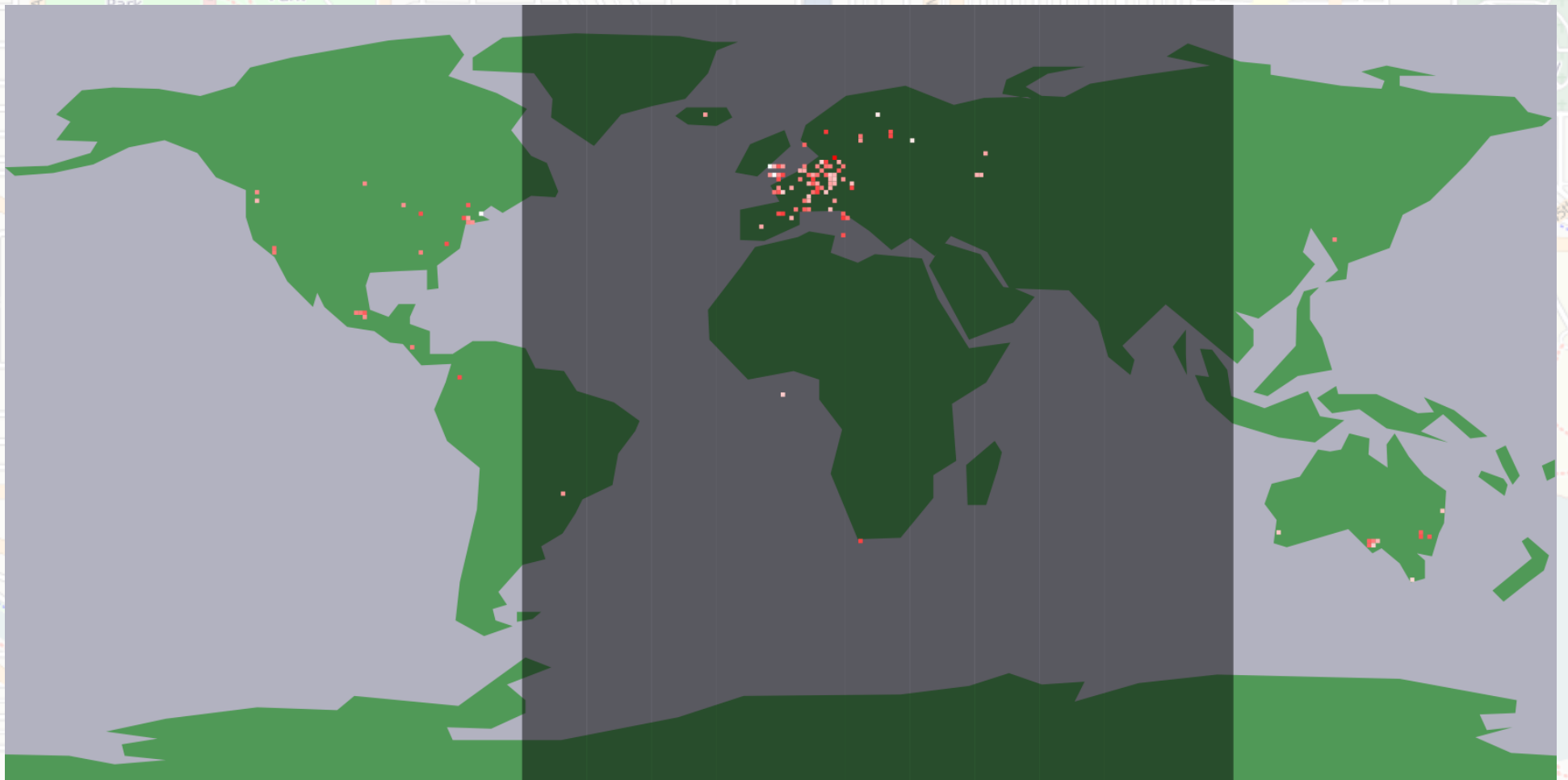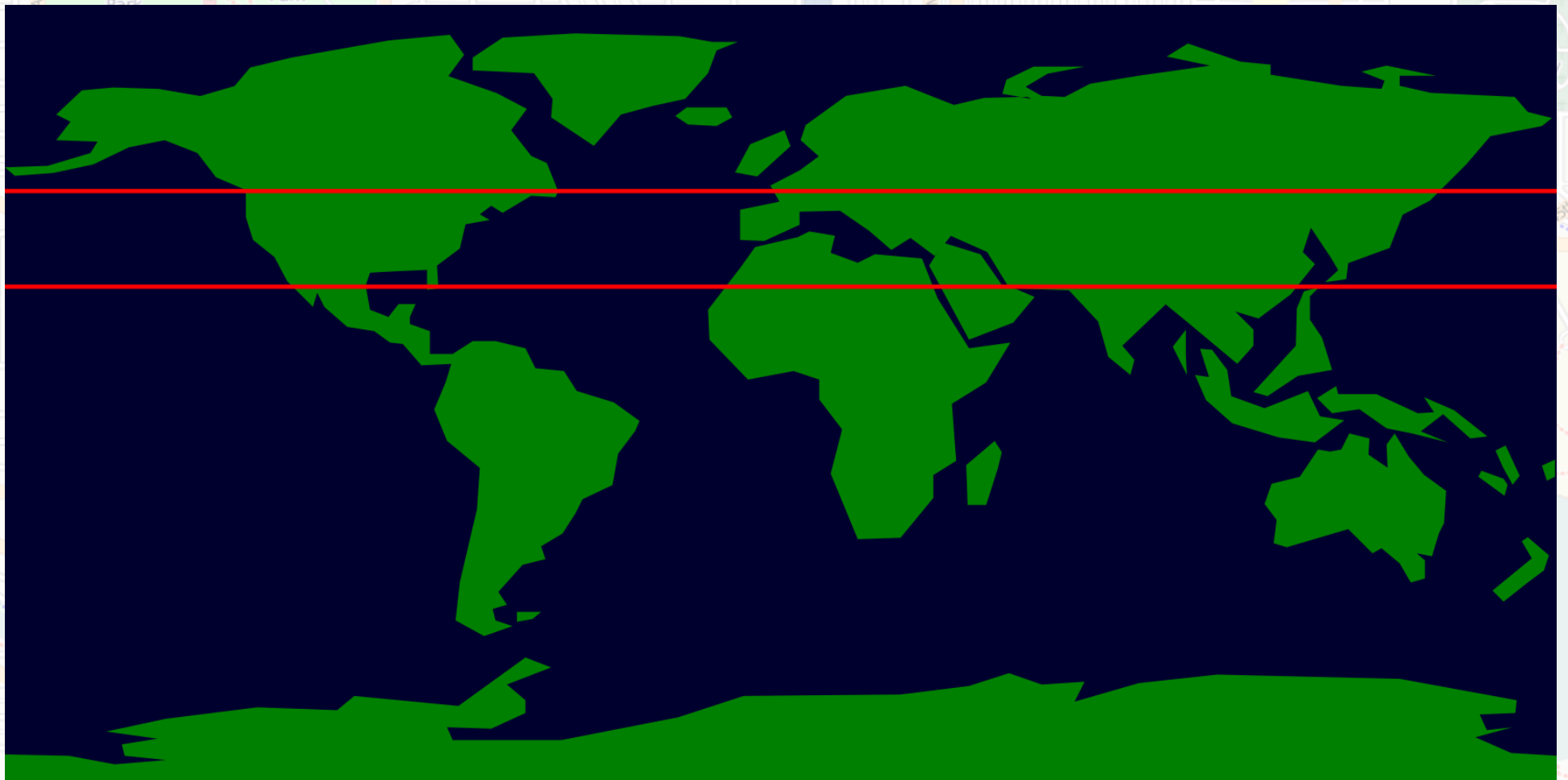
Editing @ 8PM UTC

Editing @ 9PM UTC

Editing @ 10PM UTC

Editing @ 11PM UTC
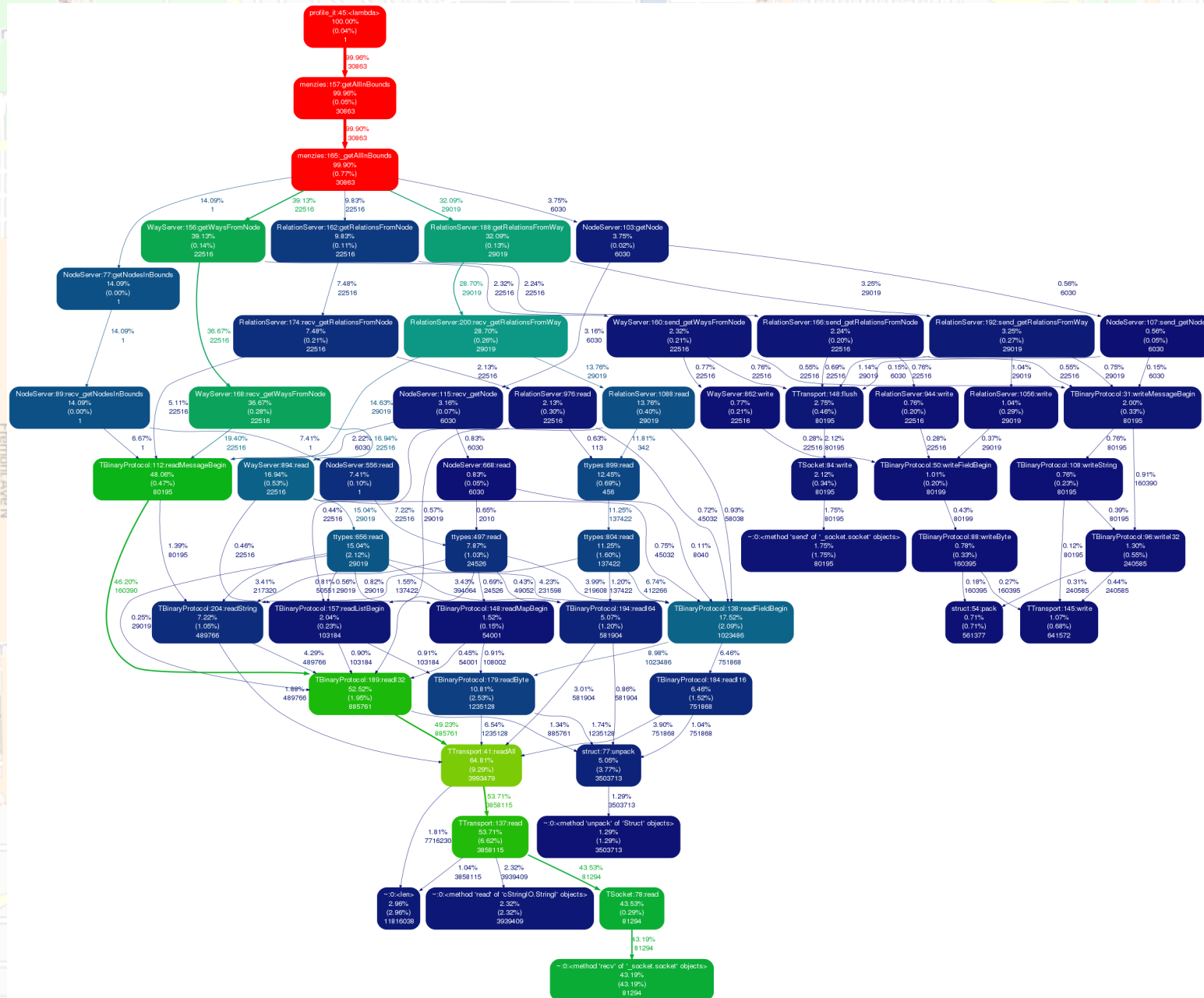
Static Latitude (3 servers)

Finding Nodes

R-tree: Spatial index implementation

Three levels of the R-tree containing Africa
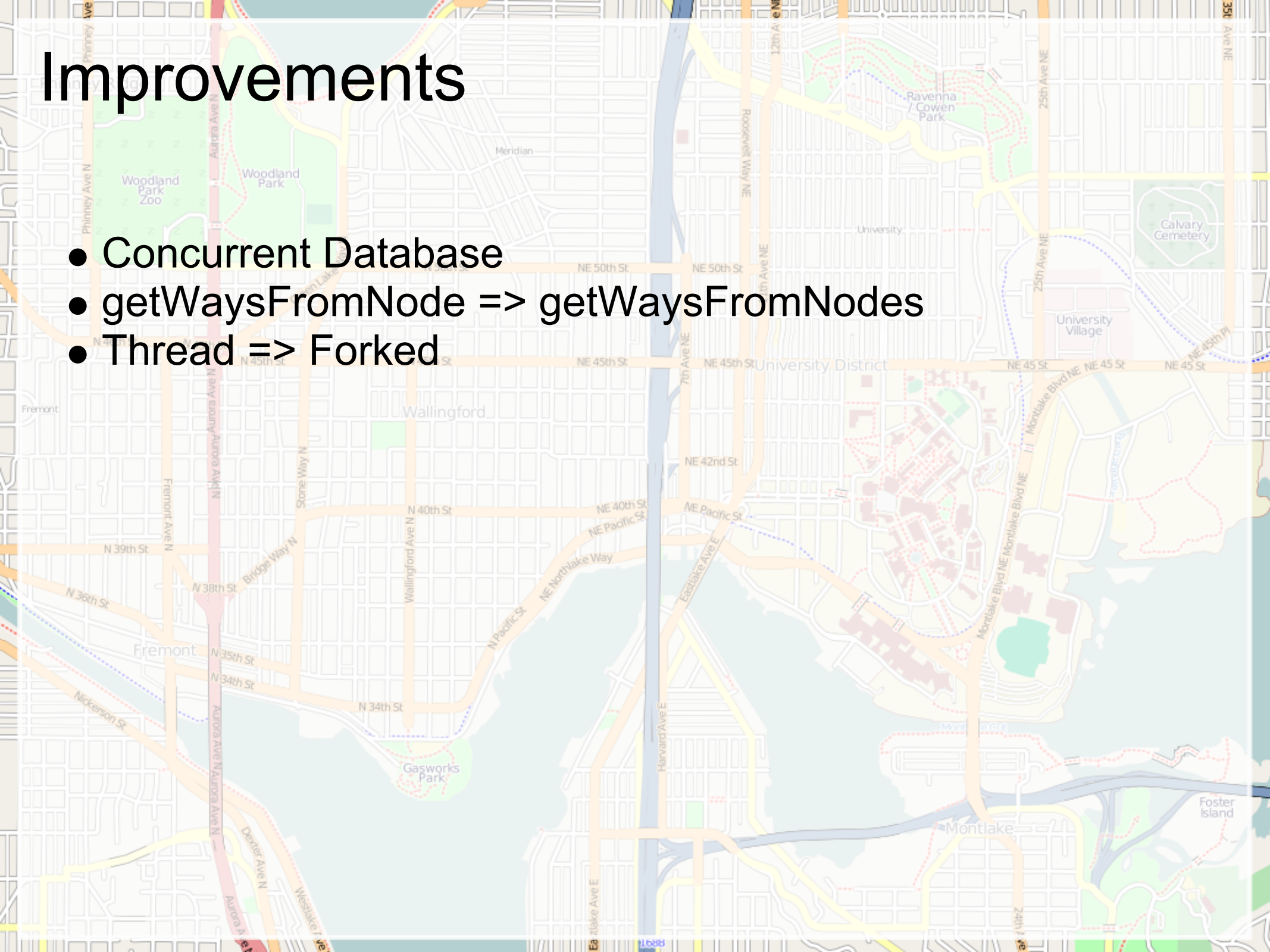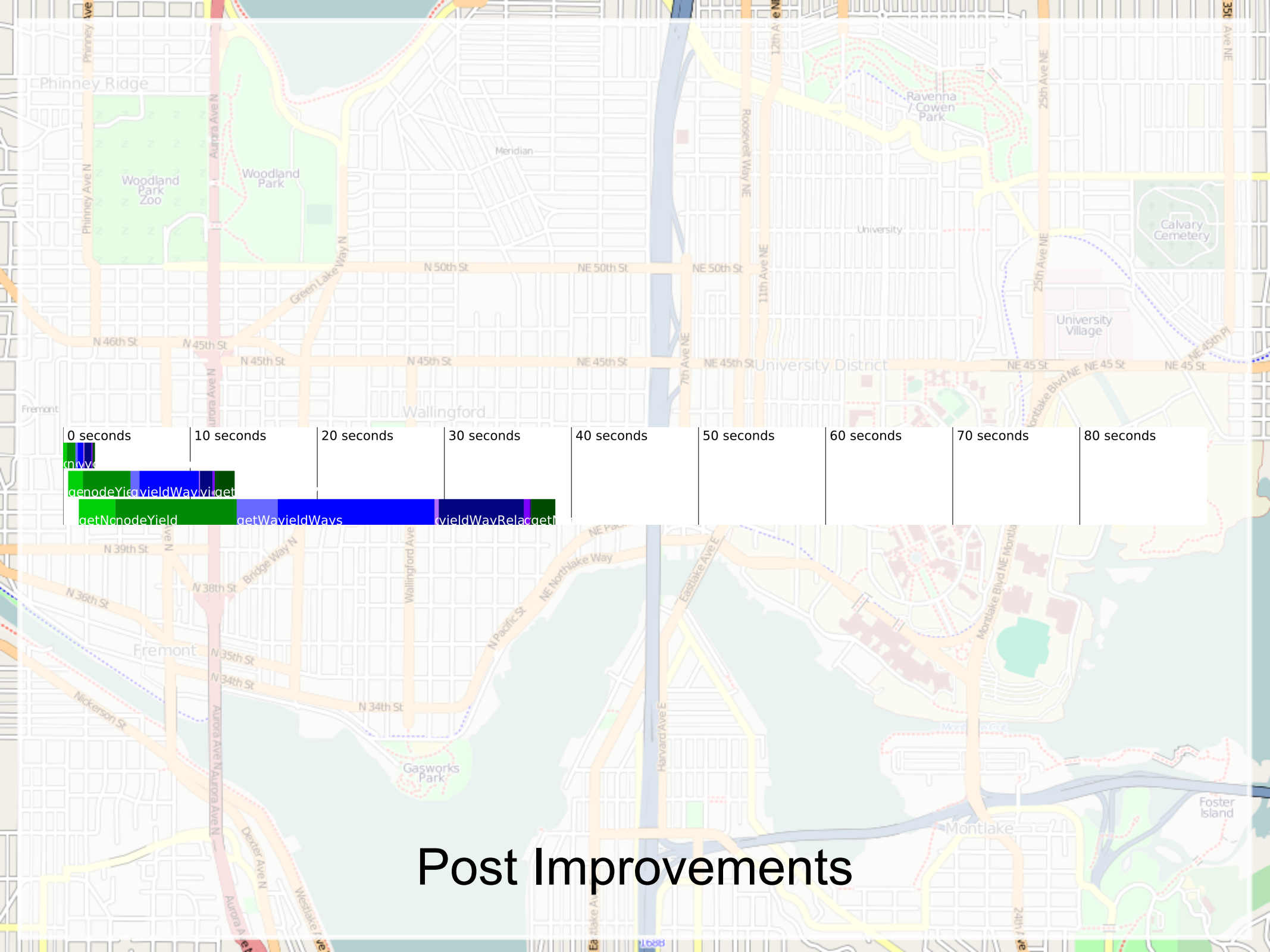
Before Profiling

UW, London and Brazil at Once

Improvements

- Concurrent Database
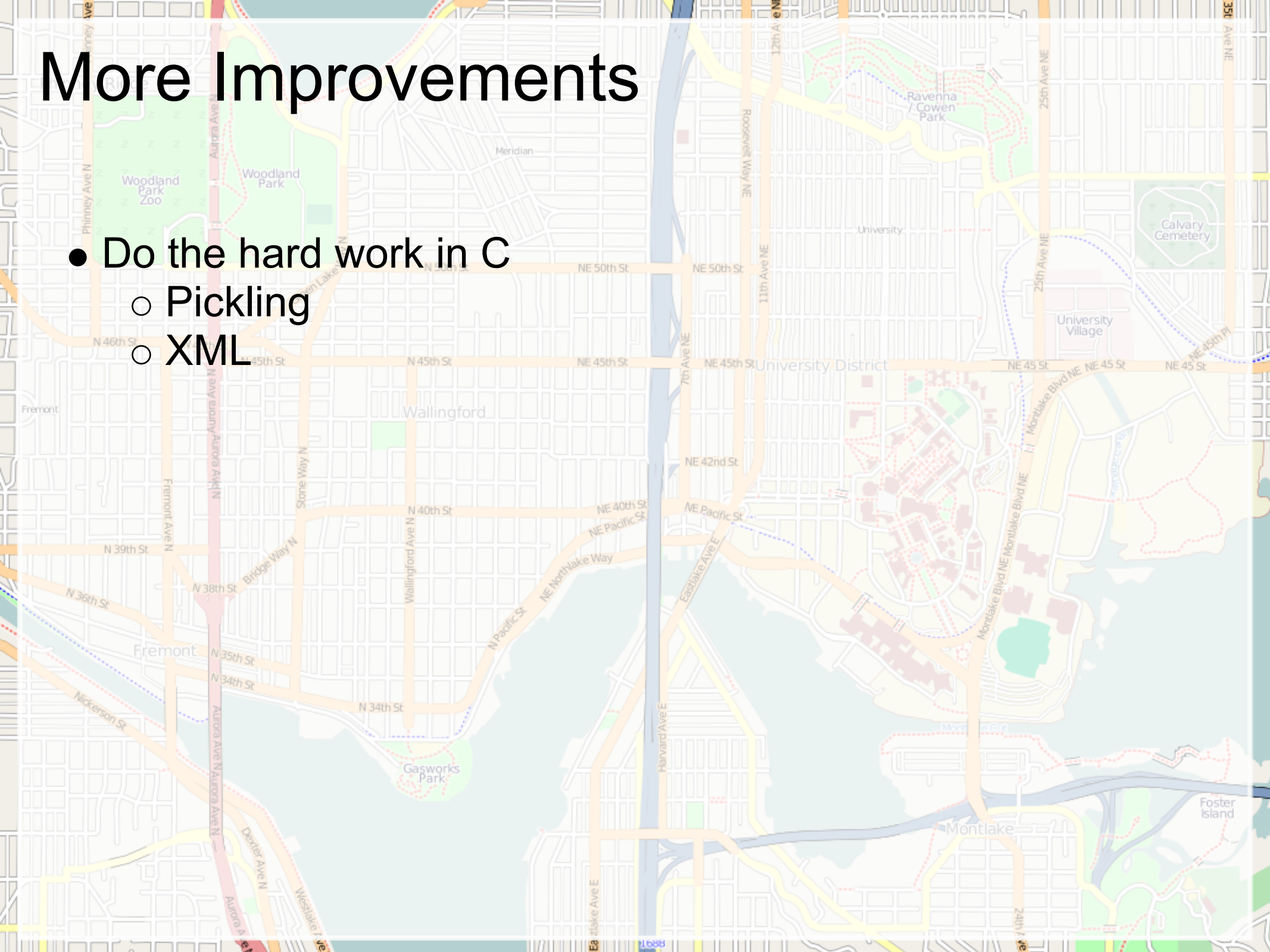- getWaysFromNode => getWaysFromNodes
- Thread => Forked

0 seconds	10 seconds	20 seconds	30 seconds	40 seconds	50 seconds	60 seconds	70 seconds	80 seconds

getNonodeYield getWavieldWays vieldWayRelacget
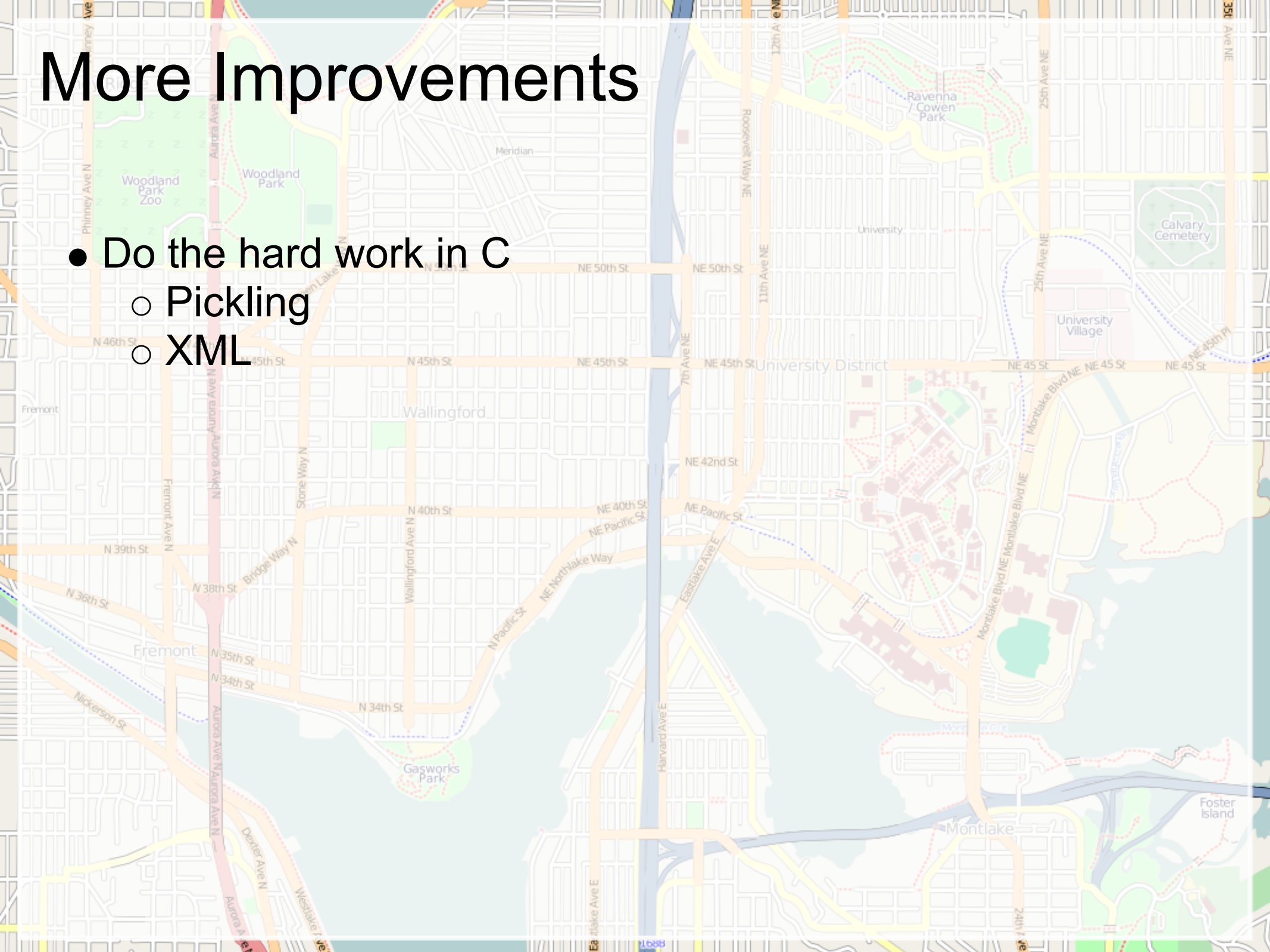
Post Improvements

More Improvements

- Do the hard work in C
  - Pickling
  - XML

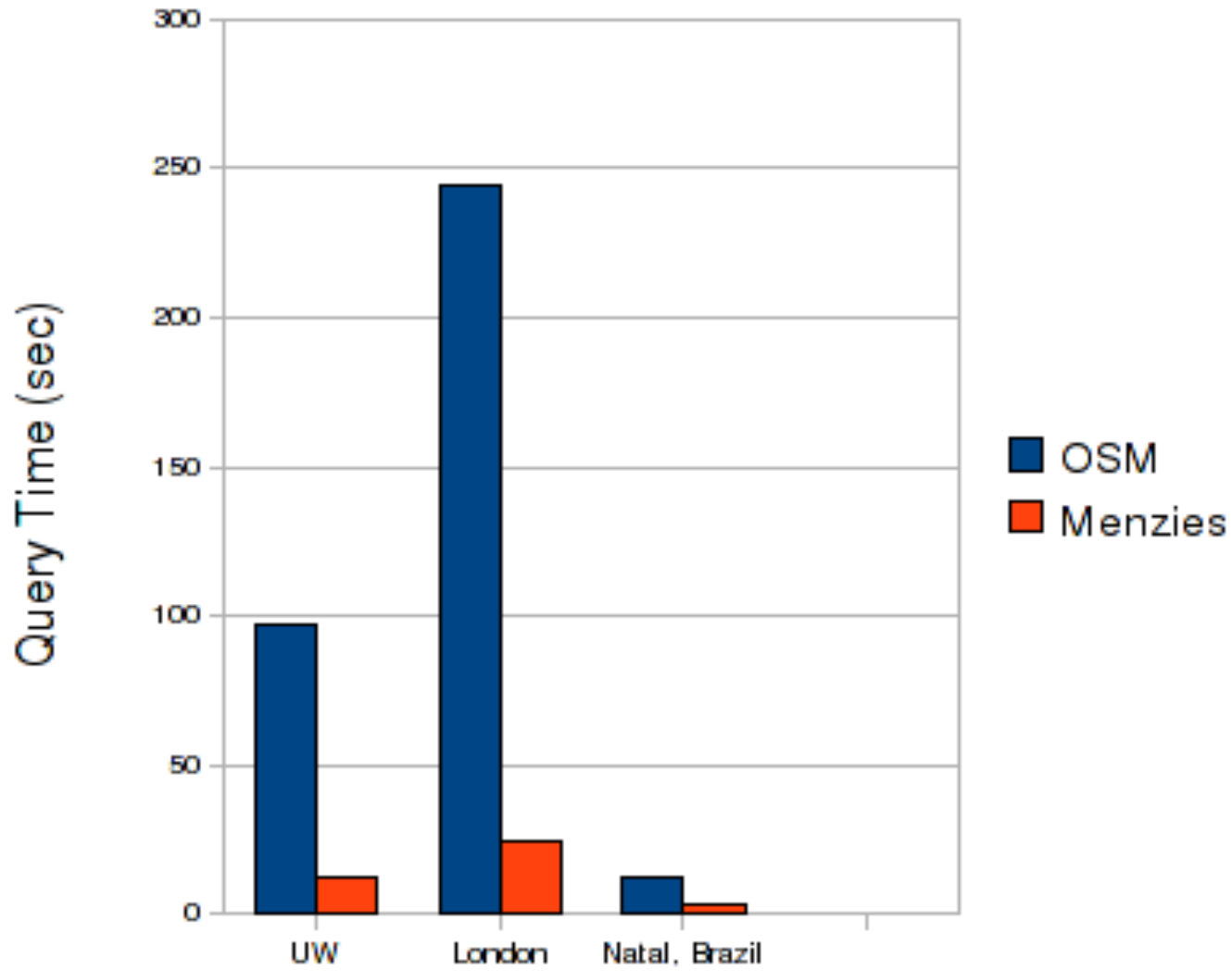| 0 seconds	10 seconds	20 seconds	30 seconds	40 seconds	50 seconds	60 seconds	70 seconds	80 seconds

genodeyield

getNonodeYield getYieldWaysyieldg

Final Run

After Profiling
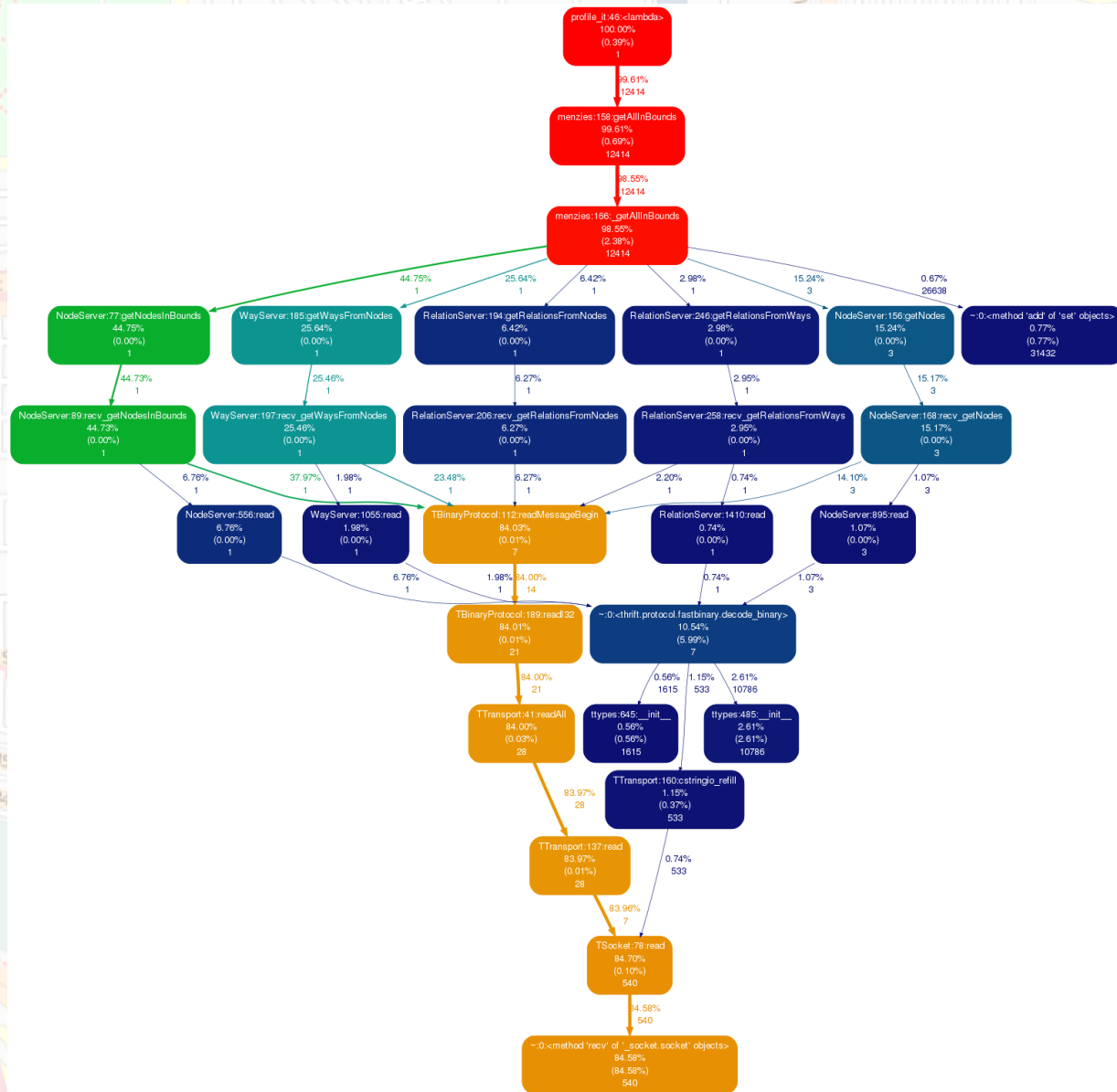
Comparison

Bounding Box Queries