

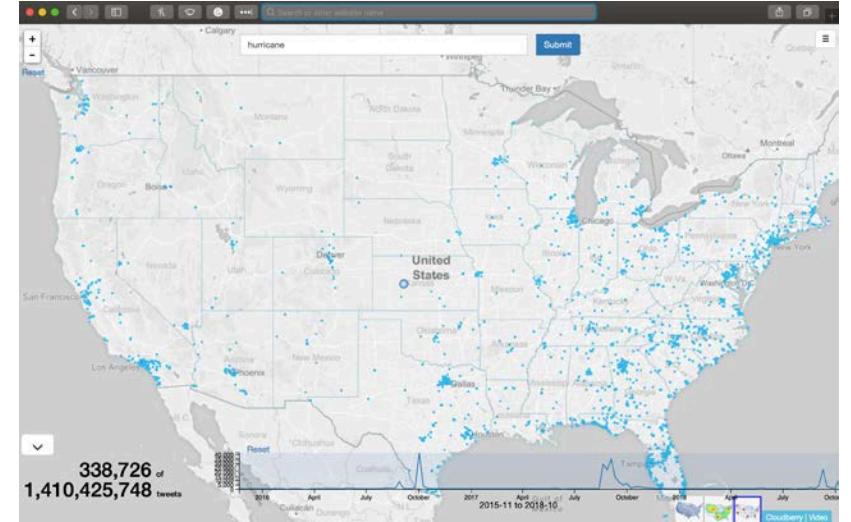
Visualizing Big Data in Your Browser Interactively

Shuang Zhao

Assistant Professor
Computer Science Department

Problems to Explore

- Point data
 - Collections of points in 2D or 3D
 - Examples:
 - Geo-tagged data (e.g., tweets)
 - Low-dimensional embeddings of complex data (e.g., images)
- Graph data
 - Relational graphs
 - Example:
 - Movie database



Challenges

- System scalability
 - We want to visualize BIG data (i.e., with millions and billions of records)
- Dynamic data
 - Data being visualized can be generated dynamically
- User interaction

Our Approach

- System oriented
- Three-tier architecture
 - Backend database
 - Middleware
 - Frontend rendering & interaction engine
- *Jointly designed* middleware and frontend

Backend Database

- We plan to use over-the-shelf open source databases
 - Relational database
 - MySQL, PostgreSQL
 - Graph database
 - Neo4j
- Databases will be treated as blackboxes
- Our middleware and frontend will be database-independent

Middleware

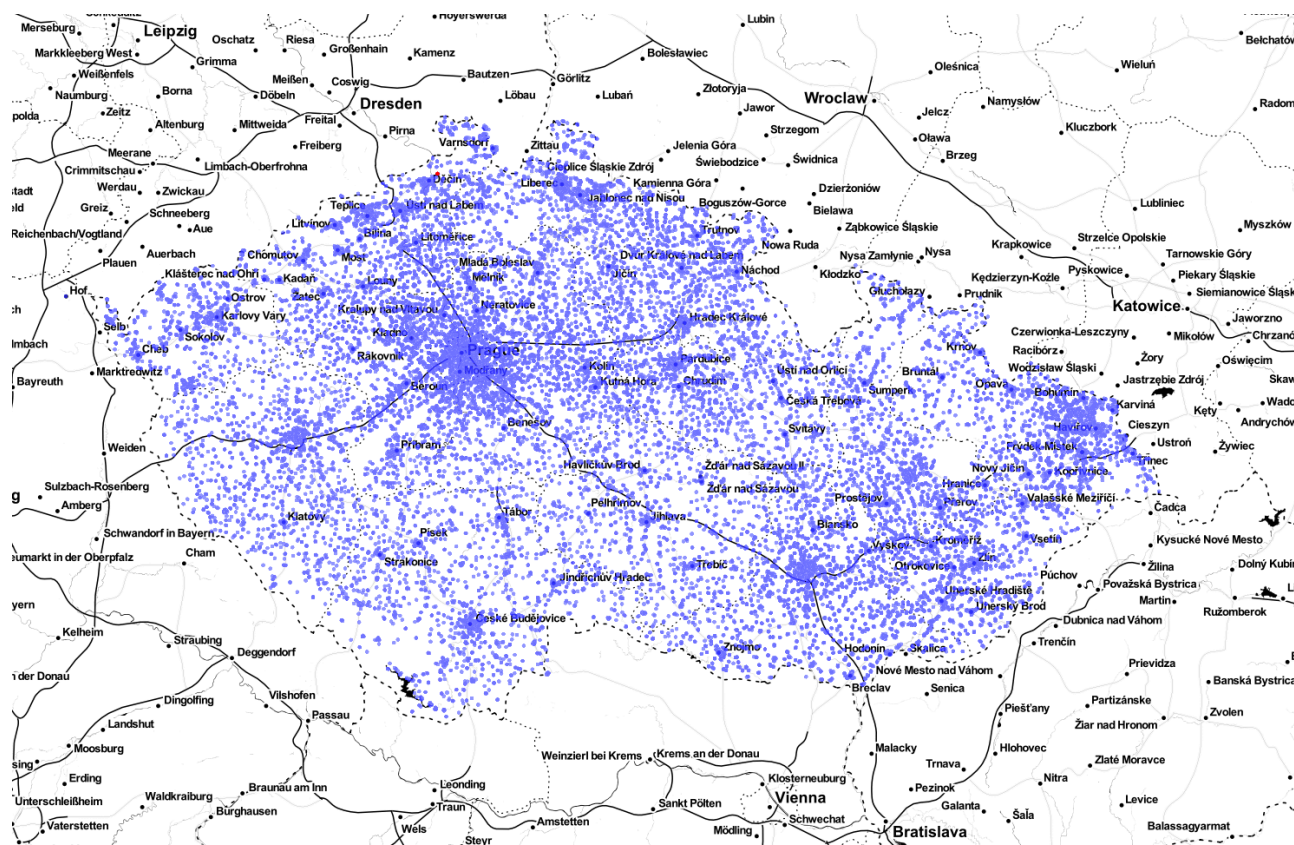
- A key component that talks to the database and the frontend
- Responsible for
 - Database analysis
 - Data approximation and compression
 - Progressive computation
 - ...
- Prof. Chen Li's group

Frontend (Our Focus)

- Rendering
 - Efficiently drawing points, lines, and text
- Animation
 - Allowing the visual contents to move smoothly
- User interaction
 - Handling different types of events (e.g., mouseover, drag)
- Data management
 - Make sure

Frontend (Our Focus)

- Live demo: pinmap



How You Can Get Involved

- Design & implement visualization-related algorithms
 - E.g., graph simplification, graph layout computation
- WebGL-based development
 - Using the GPU to enable interactive rendering
- System integration
 - Making the frontend and the backend to work together seamlessly