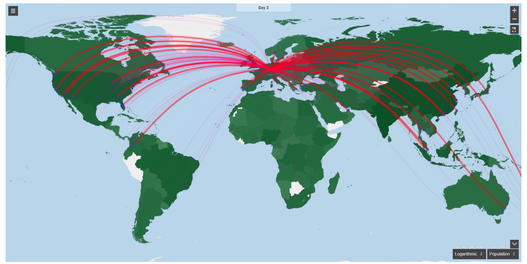
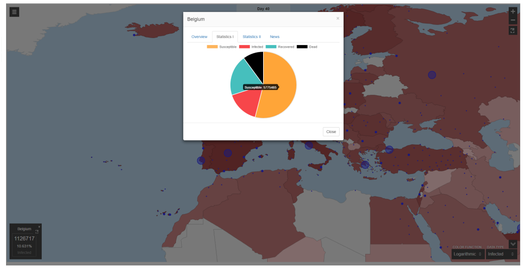
Going Viral — A Plague Simulation

Group11

Introduction

Inspired by the Plague Inc., Going Viral is a free, open-sourced simulation which lets you simulate a world-wide plague, propagated by airplanes. The simulation uses a well-established algorithm. Combined with real life flight data, you can have the state of the art experience of plague simulation.





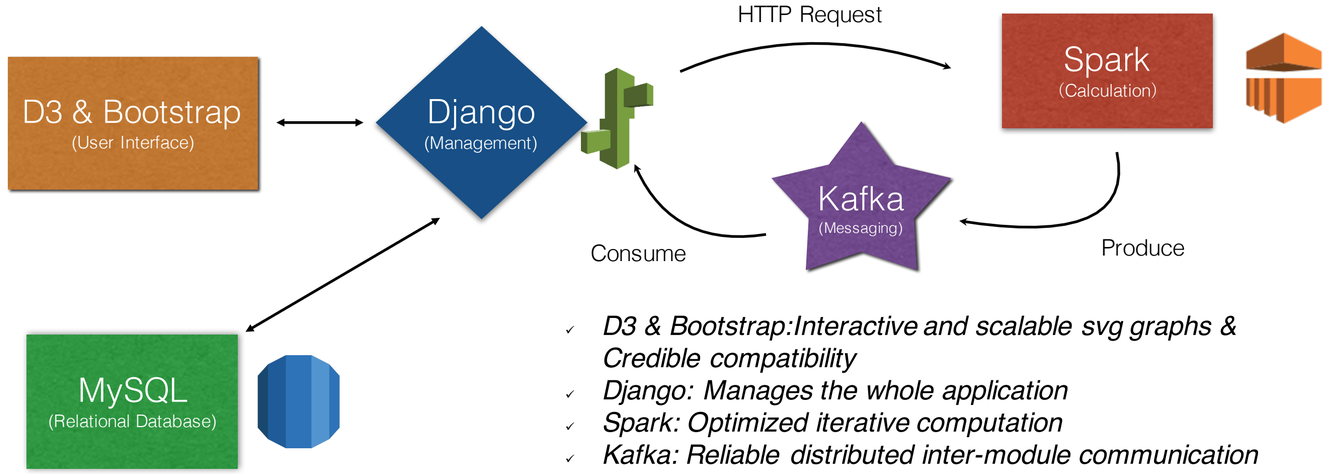
Model

$$\frac{dS}{dt} = -\frac{\alpha * S * I}{S+I+R+D}, \quad \frac{dR}{dt} = \beta * I, \quad \frac{dD}{dt} = \gamma * I, \quad \frac{dI}{dt} = \frac{\alpha * S * I}{S+I+R+D} - \beta * I - \gamma * I$$

S: No. of susceptible I: No. of infectious R: No. of recover D: No. of death

- α: transmission rate represents the encounter rate between susceptible and infectious individuals together with the probability of transmission
- β: recovery rate represents the proportion of people recover from disease every day
- γ: death rate represents the proportion of people dead from disease every day

Architecture



- ✓ *D3 & Bootstrap:Interactive and scalable svg graphs & Credible compatibility*
- ✓ *Django: Manages the whole application*
- ✓ *Spark: Optimized iterative computation*
- ✓ *Kafka: Reliable distributed inter-module communication*
- ✓ *Elastic Beanstalk: Hosting the application*
- ✓ *EMR: Hosting Spark*
- ✓ *RDS: Hosting MySQL database*

Data

- Real life flight data collected from flightaware.com
- Covering over 190 countries and regions, 1200 major airports, and 250000 airline records from all over the world
- Collected by web crawling

Optimization

- **Front-end:** Using asynchronous AJAX calls to provide smooth user experience
- **Back-end:** Caching static data in local files to minimize remote database queries
- **Spark:** Using broadcast variables and cache, tuning hyper parameters such as the number of workers and partitions to optimize performance, omitted empty rdds