
Graphalyzer

— A graph visualization and
analysis tool —

Team May1618/Workiva

May1618

Team Members:

- Andrew Bowler - Webmaster
- Alberto Gomez-Estrada - Communications Lead
- Michael Sgroi - Key Concept Holder
- Richard White - Key Concept Holder
- Taylor Welter - Project Lead
- Dr. Simanta Mitra - Advisor
- Ross Hendrickson - Client

Problem - Big Data

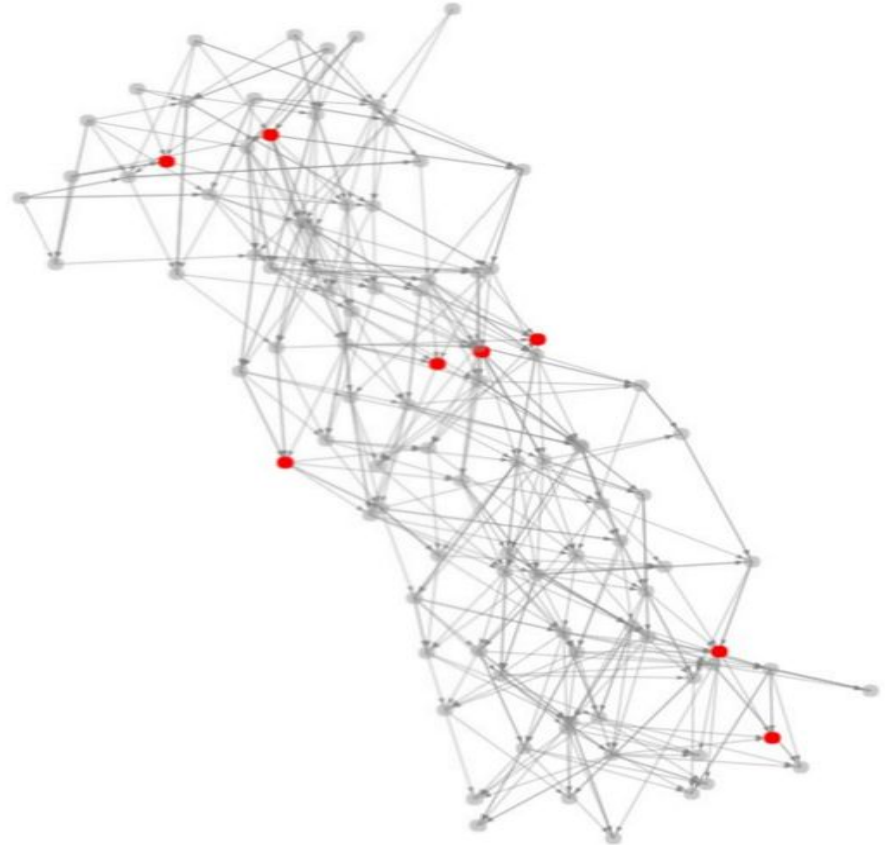
- Companies constantly dealing with Big Data
 - Social media
 - Payroll
 - Documents
 - Etc....
- How can we understand Big Data? What does it mean?

Solution - Graphs

- Graphs are a helpful way to represent Big Data
 - Structural behavior
 - Relationships
 - Key properties or members of high impact

Goals of Graphalyzer - Visualization

- Present a visualization of graphical data to the user through an Internet browser
 - Context of data is arbitrary, serve as a tool for any graph data
 - Display graph through intuitive interface - **use shapes and colors**



Goals of Graphalyzer - Analysis

- Allow user to specify parameters for visualization
 - Filtering and highlighting nodes by properties
 - Search for and focus in on nodes, display their properties

Goals of Graphalyzer - Performance

- Handle large data
 - Size of data can range from very small to many gigabytes
 - Hundreds of nodes or edges to millions of nodes or edges

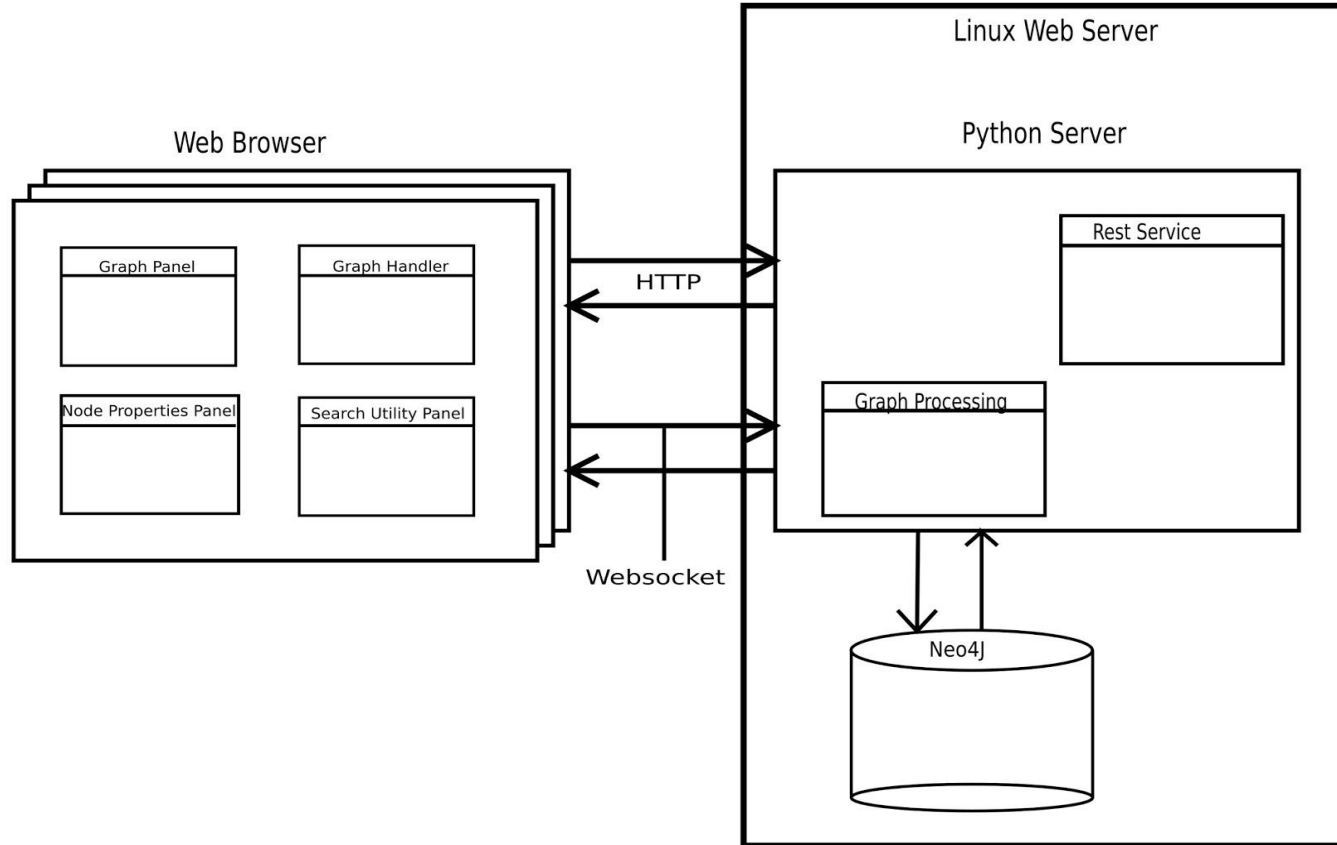
Technical Challenge #1 - Uncharted Waters

- Working with new technologies
 - Angular and D
 - Server administration
- **Solution:** Don't reinvent the wheel.
 - Use familiar tools to get the job done
 - Follow Workiva's advice

Technology Stack

- Graph Libraries
 - Vis.js
 - neo4j
- Workiva Stack
 - React.js
 - Python
- REST service
- Ubuntu Web Server
- This project is expected to continue under Workiva.
- Keep code organized and familiar with style guidelines, so work can continue.

Design



Test Plan

Using Jest:

- Validate rendering of Javascript objects.
- Verify that the React components maintain a consistent state and manipulate their data as expected.
- Validate requests and response to Python server.

Goal:

- Ensure maintainability of code, keeping in mind that our software may be used by Workiva in the future.

Technological Challenge #2 - Implementing a test plan

- We planned on using Jest, the testing framework bundled React.js and based on JUnit.
 - Our implementation encountered issues when running Jest.
 - Further research revealed that our issue might be the result of a recent update to Jest; other users were encountering the same problem.
- Testing front-end and back-end simultaneously presented issues on TravisCI.
 - Our Python server filters all IP Addresses except those within a range determined by Workiva.
- **Solutions:**
 - After very difficult configuration, we are using Jest and PyUnit to test both the front and back-ends.
 - For the sake of the project, we only unit tested, since the server is not reachable.

Technological Challenge #3 - Scalability and Performance

- Be able to visualize data that could be gigabytes in size
- Maximize scalability and performance, minimize impact to user interactivity

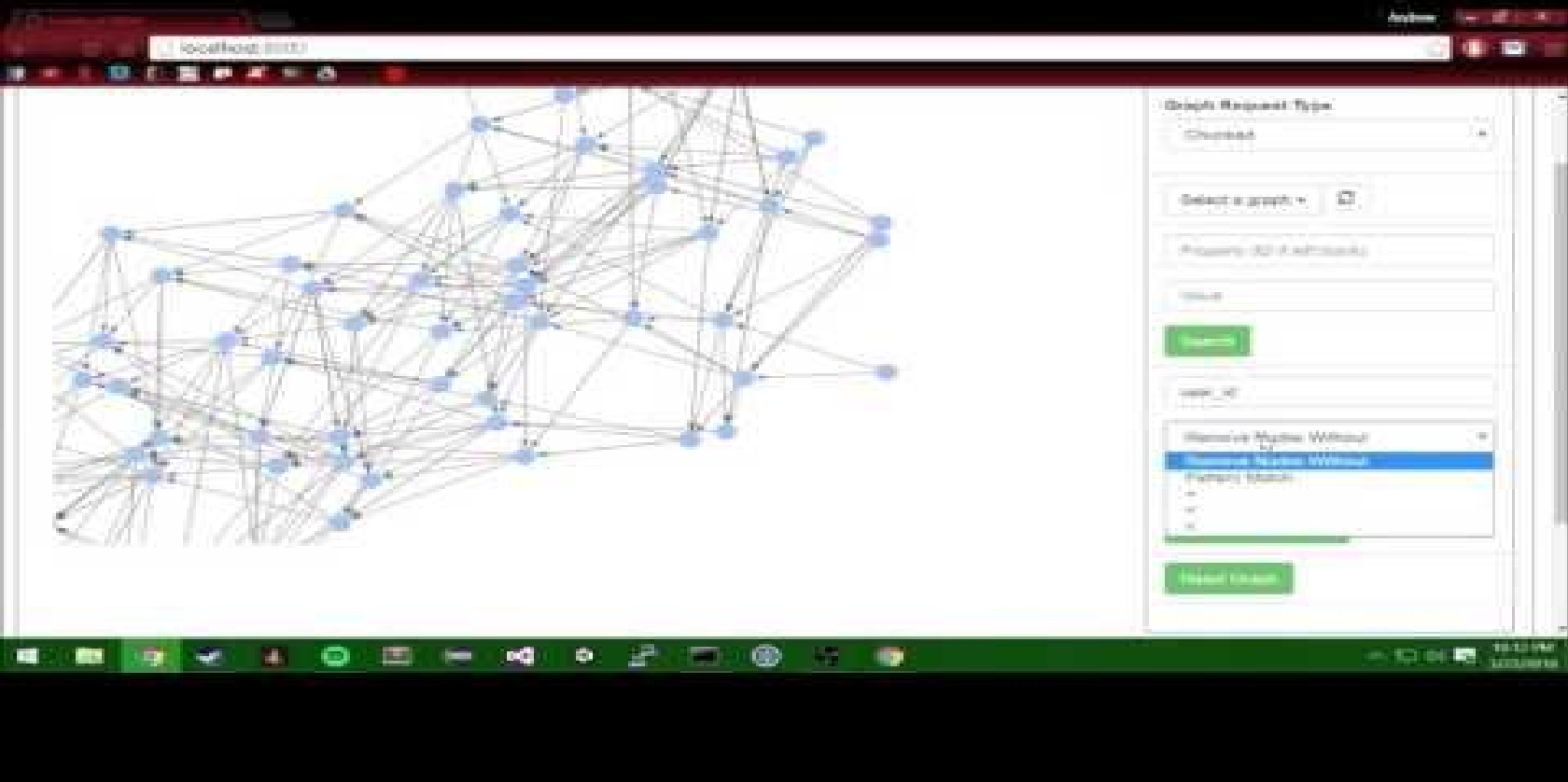
Technological Challenge #3 - Solution

- **Solutions:**

- Stream the data - query data from neo4j and send previously received data to the client simultaneously in chunks.
 - Running time: $O(n)$, constant is significantly cut down by streaming
- Do graph processing server-side as much as possible
 - Reduces loads on browser, CPU, and GPU
- Visualize only what the user wants
 - Graph draws in $O(n \log n)$
 - Save time and performance - draw only what is necessary

Use Case - Filtering

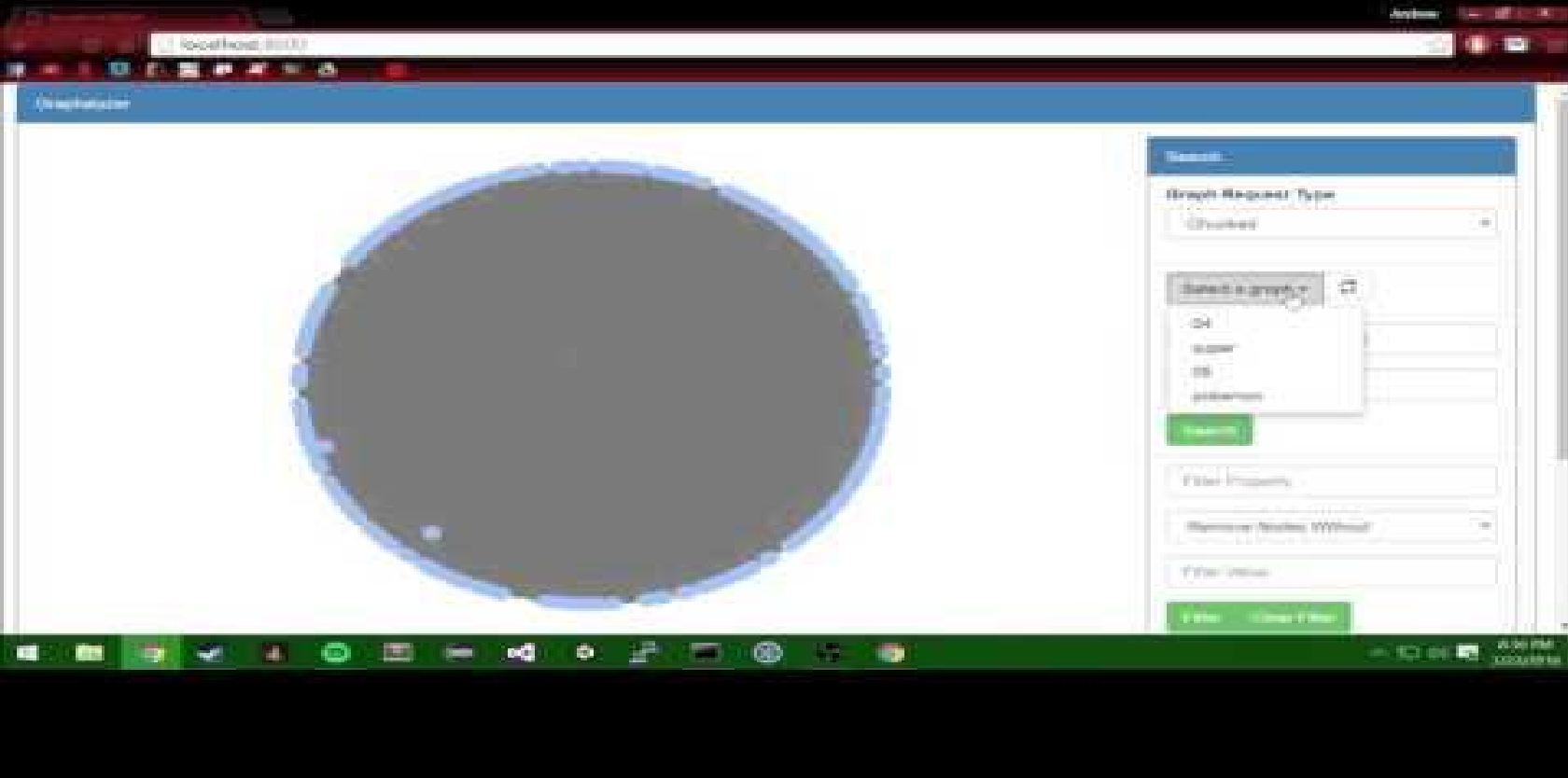
- Lots of data, need to find certain entries
 - May have properties with values
 - **Example:** find all people on payroll with a salary greater than a certain value
- Graphalyzer provides users with customizable options
- Highlight nodes that pass filter test, grey out all others



http://www.youtube.com/watch?v=n_gA81Q_ICM

Use Case - Searching

- Important node exists somewhere within giant graph
 - Find it, and display all of its properties to the user
 - **Example:** Find a CEO of a company with more than 10,000 employees and display all of that person's information in the graph's data
- Graphalyzer makes this easy by zooming into the node and listing all of its properties



<http://www.youtube.com/watch?v=JRYCwPoh0qc>

Current Progress

- **Complete:**

- Visualization of very large graphs, thousands of nodes and edges
- Filtering for properties, their values, or lack thereof
- Searching for nodes
- REST upload for Workiva's graph data
- Project deployed on an Amazon Web Server

- **Nearing Completion:**

- Revamp Search Panel into more user-friendly Dashboard
 - Process filtering parameters *before* drawing
 - Add ability to draw only parts of a graph
- Handle server-side data updates

Questions