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

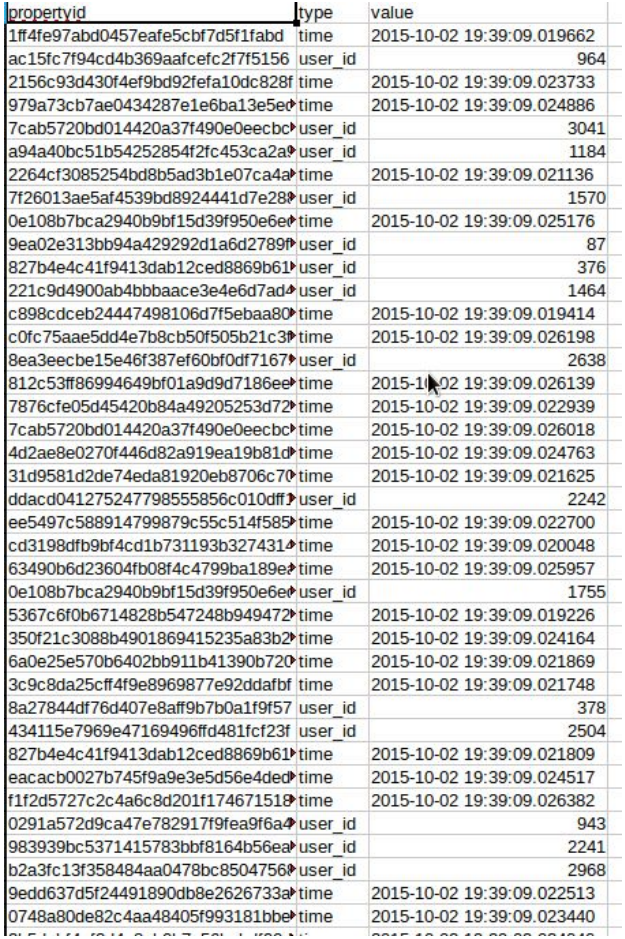
- Companies constantly dealing with Big Data
  - Social media
  - Employee directories
  - Documents on a server
  - Etc...
- How can we understand Big Data? What does it mean?

Solution - Graphs

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

Looking at data in tables

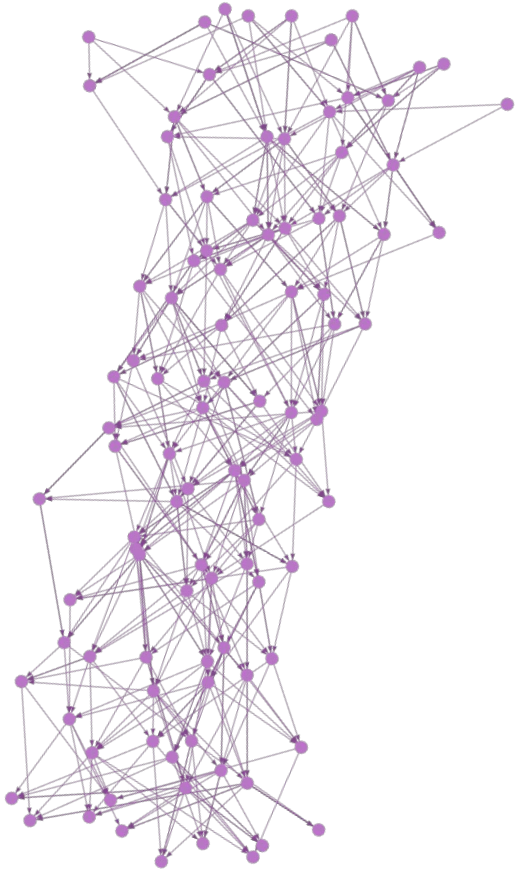
- Difficult to see or find patterns
- Requires you to join many tables
  - High performance cost
- Example: Sales
  - You have a tables for salesmen, customers, orders, and items
    - Not joined
      - Impossible to see any patterns
    - Joined
      - Hard to see patterns in giant lists
      - High performance cost



propertyid	type	value
1ff4fe97abd0457eafe5cbf7d5f1fabd	time	2015-10-02 19:39:09.019662
ac15fc7f94cd4b369aafcefc2f7f5156	user_id	964
2156c93d430f4ef9bd92fefa10dc828f	time	2015-10-02 19:39:09.023733
979a73cb7ae0434287e1e6ba13e5ed	time	2015-10-02 19:39:09.024886
7cab5720bd014420a37f490e0eecbc	user_id	3041
a94a40bc51b54252854f2fc453ca2a9	user_id	1184
2264cf3085254bd8b5ad3b1e07ca4a	time	2015-10-02 19:39:09.021136
7f26013ae5af4539bd8924441d7e28	user_id	1570
0e108b7bca2940b9bf15d39f950e6e	time	2015-10-02 19:39:09.025176
9ea02e313bb94a429292d1a6d2789f	user_id	87
827b4e4c41f9413dab12ced8869b61	user_id	376
221c9d4900ab4bbbaace3e4e6d7ad	user_id	1464
c898cdceb24447498106d7f5ebaa80	time	2015-10-02 19:39:09.019414
c0fc75aae5dd4e7b8cb50f505b21c3	time	2015-10-02 19:39:09.026198
8ea3eecbe15e46f387ef60bf0df7167	user_id	2638
812c53ff86994649bf01a9d9d7186ee	time	2015-10-02 19:39:09.026139
7876cfe05d45420b84a49205253d72	time	2015-10-02 19:39:09.022939
7cab5720bd014420a37f490e0eecbc	time	2015-10-02 19:39:09.026018
4d2ae8e0270f446d82a919ea19b81d	time	2015-10-02 19:39:09.024763
31d9581d2de74eda81920eb8706c70	time	2015-10-02 19:39:09.021625
ddacd041275247798555856c010dff	user_id	2242
ee5497c588914799879c55c514f585	time	2015-10-02 19:39:09.022700
cd3198dfb9bf4cd1b731193b327431	time	2015-10-02 19:39:09.020048
63490b6d23604fb08f4c4799ba189e	time	2015-10-02 19:39:09.025957
0e108b7bca2940b9bf15d39f950e6e	user_id	1755
5367c6f0b6714828b547248b949472	time	2015-10-02 19:39:09.019226
350f21c3088b4901869415235a83b2	time	2015-10-02 19:39:09.024164
6a0e25e570b6402bb911b41390b720	time	2015-10-02 19:39:09.021869
3c9c8da25cff4f9e8969877e92ddafbf	time	2015-10-02 19:39:09.021748
8a27844df76d407e8aff9b7b0a1f9f57	user_id	378
434115e7969e47169496ffd481fcf23f	user_id	2504
827b4e4c41f9413dab12ced8869b61	time	2015-10-02 19:39:09.021809
eacacb0027b745f9a9e3e5d56e4ded	time	2015-10-02 19:39:09.024517
f1f2d5727c2c4a6c8d201f174671518	time	2015-10-02 19:39:09.026382
0291a572d9ca47e782917f9fea9f6a4	user_id	943
983939bc5371415783bbf8164b56ea	user_id	2241
b2a3fc13f358484aa0478bc8504756	user_id	2968
9edd637d5f24491890db8e2626733a	time	2015-10-02 19:39:09.022513
0748a80de82c4aa48405f993181bbe	time	2015-10-02 19:39:09.023440

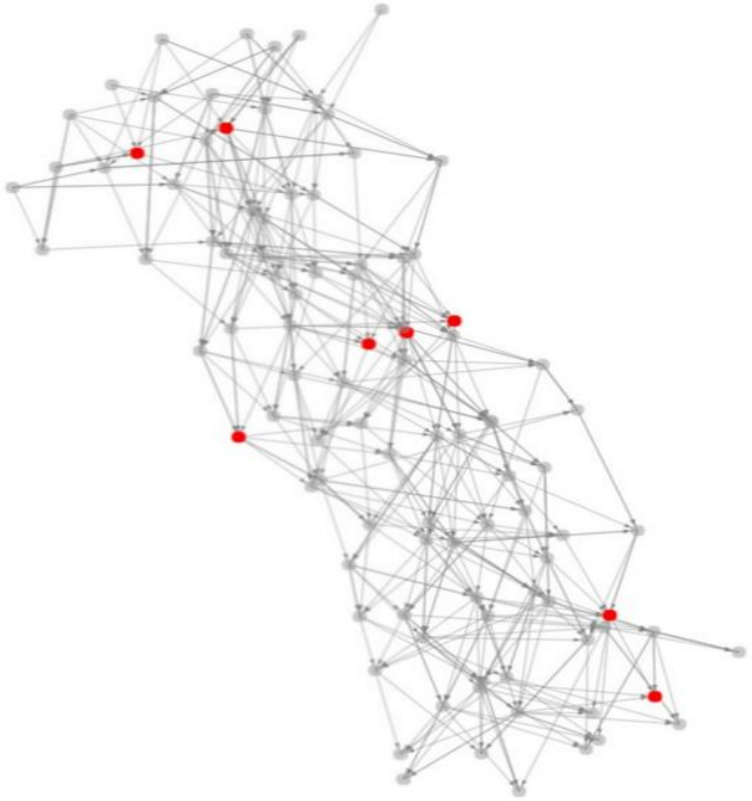
Looking at data in graphs

- Easy to see patterns
- Minimal performance cost
- Example: Sales
    - Each salesmen, customer, order, item is a node
    - Relationships are expressed as edges



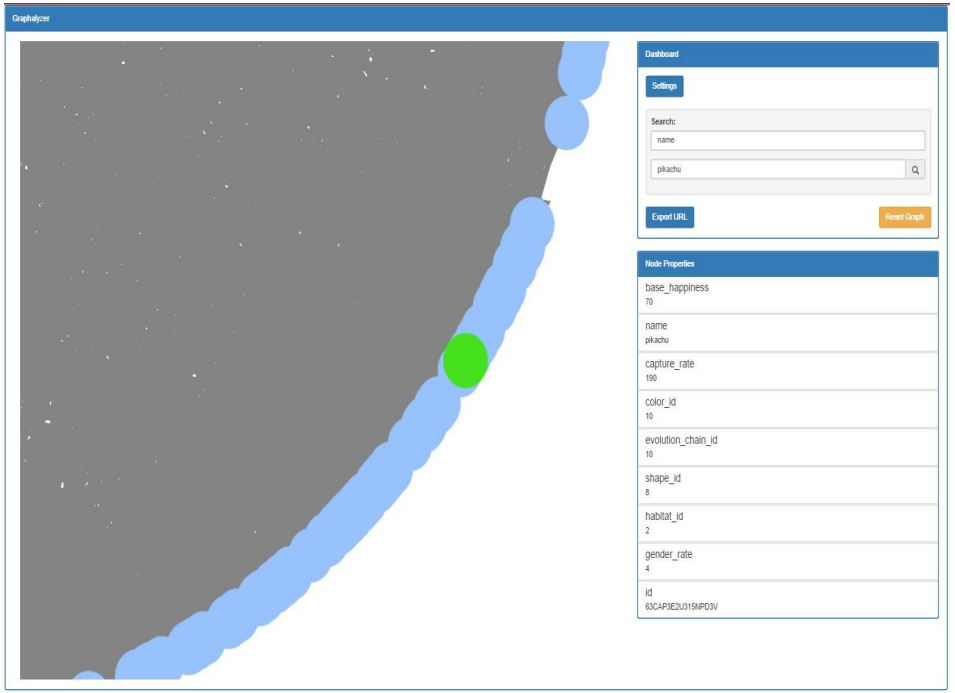
Goals of Graphalyzer - Visualization

- 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

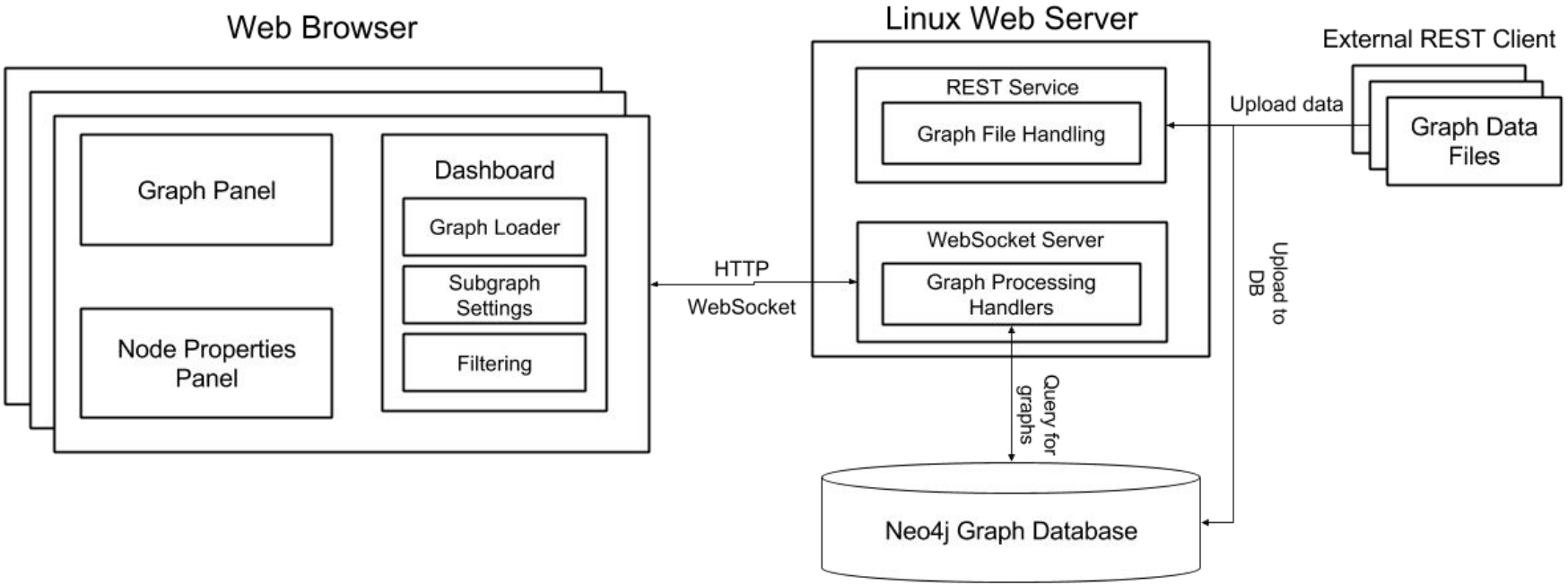
Technology Stack

- 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


Web Browser
Graph Panel
Node Properties Panel
Dashboard
Graph Loader
Subgraph Settings
Filtering
HTTP
WebSocket
Linux Web Server
REST Service
Graph File Handling
WebSocket Server
Graph Processing Handlers
Query for graphs
Upload data
Upload to DB
External REST Client
Graph Data Files
Neo4j Graph Database

Technological Challenge #2 - Implementing a test plan

- We planned on using Jest, the testing framework bundled with React.js and based on JUnit
  - Our implementation encountered issues when running Jest.
- 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:**
  - 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.

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

Technological Challenge #3 - Scalability and Performance

- Be able to visualize data that could be gigabytes in size

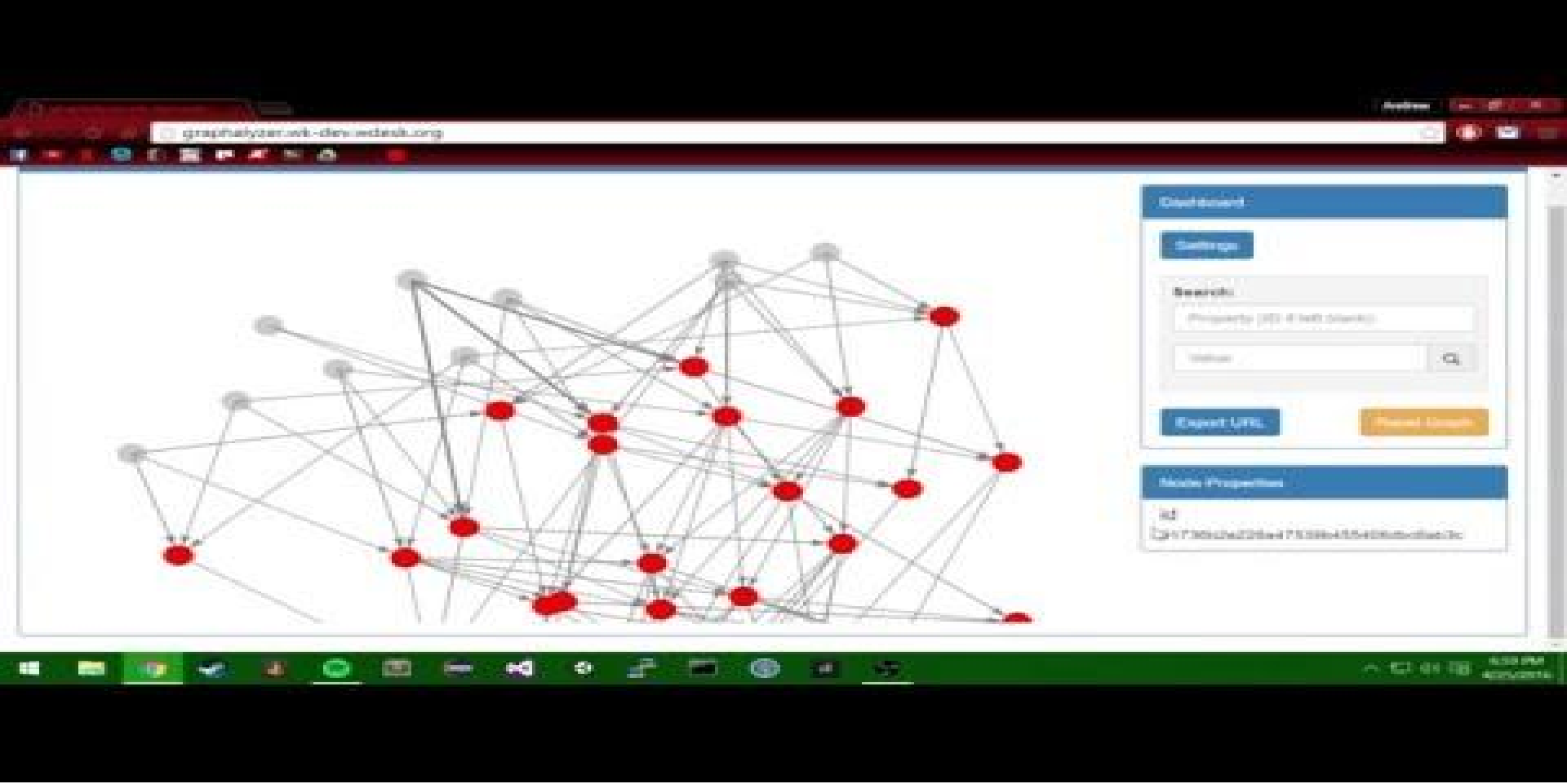
- Maximize scalability and performance, minimize impact to user interactivity

Technological Challenge #3 - Solutions

- **Solutions:**
  - Stream the data - query data from neo4j and send previously received data to the client simultaneously in chunks.
  - Do graph processing server-side as much as possible
    - Reduces loads on browser, CPU, and GPU
  - Visualize only what the user wants
    - Draw subgraphs - up to the user
- The provided server has very limited disk space and RAM, so we can only store a few hundred megabyte sized graphs

Filtering

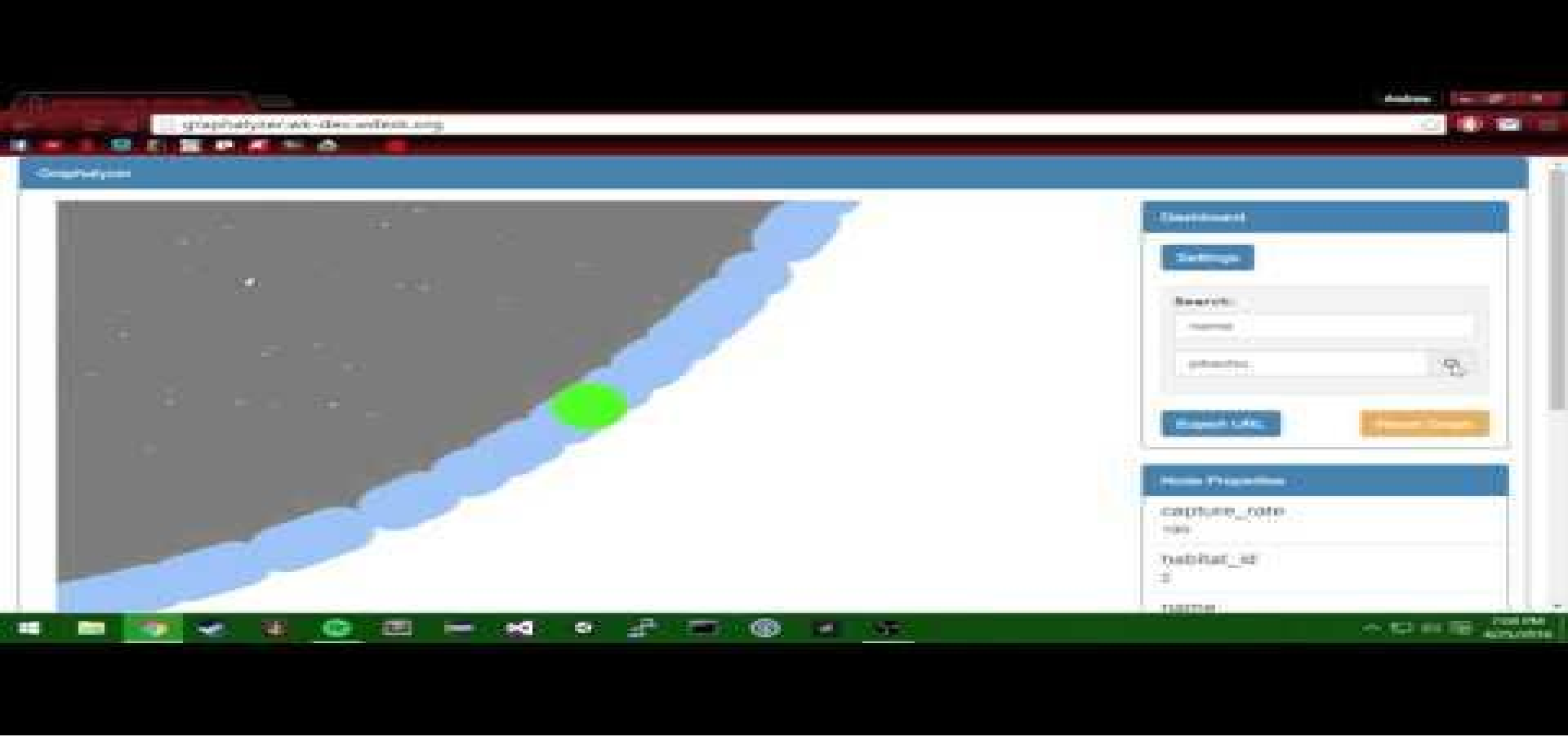
- Lots of data, need to find certain entries
  - May have properties with values
  - **Example:** find all people on payroll with a salary greater than a certain value
- Graphalyzer provides users with customizable options
- Highlight nodes that pass filter test, grey out all others



https://youtu.be/P5A3UxZwodU

Searching

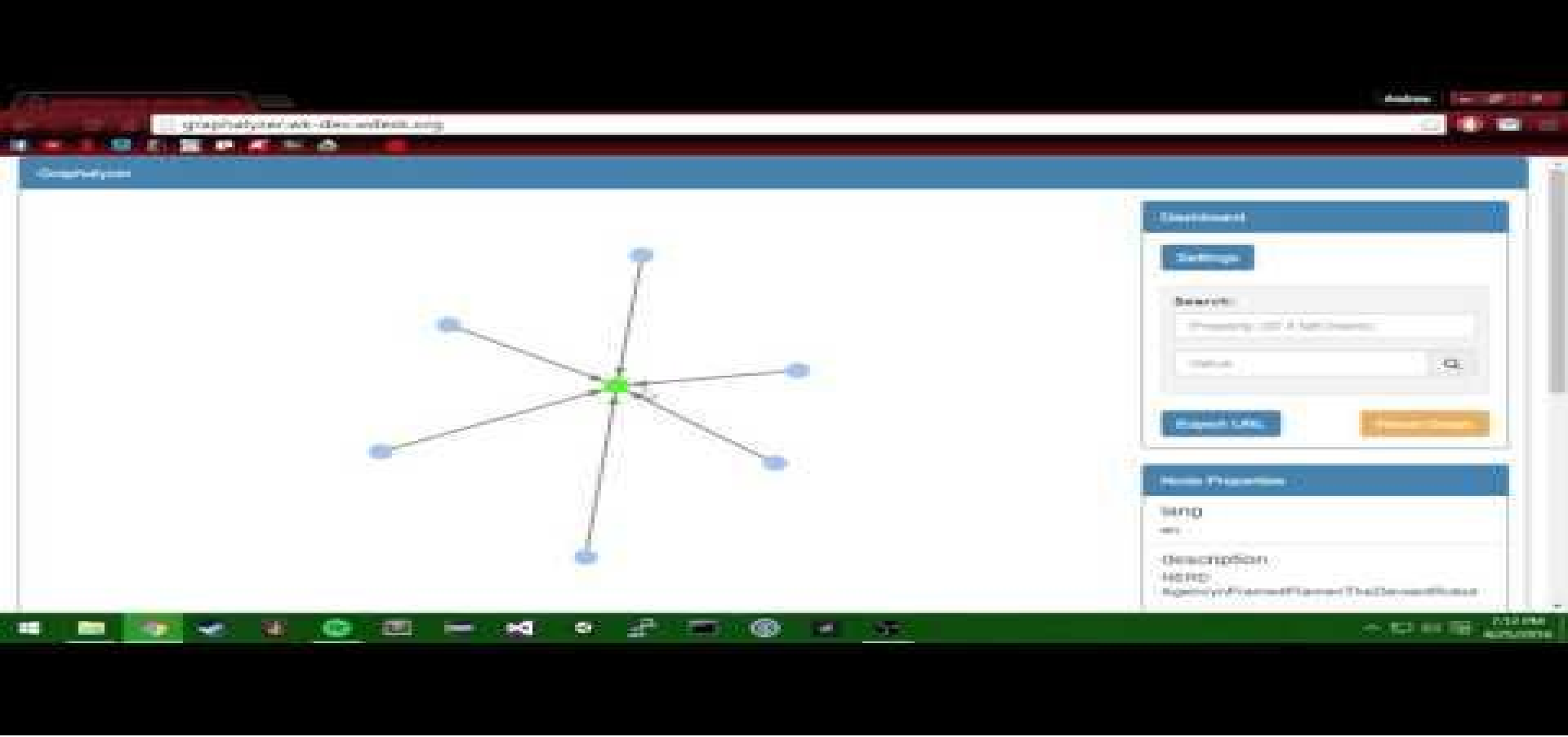
- Important node exists somewhere within giant graph
  - Find it, and display all of its properties to the user
  - **Example:** Find a CEO of a company with more than 10,000 employees and display all of that person's information in the graph's data
- Graphalyzer makes this easy by zooming into the node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



https://www.youtube.com/watch?v=ArWq8q4BD04&feature=youtu.be

Subgraphs

- Often times the user does not need to see the entire graph, but only a part of it
- The graph could have thousands of nodes and edges the user doesn't care about
    - **Example: Given a name of a person on Twitter, display all of their followers and people they are following**
- Graphalyzer only requires a source node (person in this case), and a depth of connectivity (incoming and outgoing)



https://www.youtube.com/watch?v=hqfuVaon-54&feature=youtu.be

Sharing Your Graph Analysis Results

- Graph analysis can often be done collaboratively
- Graphalyzer allows users to export a custom web URL and share it with others
  - Copied to clipboard, paste it in an email, group chat, etc.
  - Paste the URL in the browser on another computer
  - Watch it go
- In addition, you can save your visualization as a PNG image
  - HTML5 Canvas makes this trivial

Achievements

- **Visualization and Analysis:**
  - Subgraphs
  - Filtering
  - Searching
  - Listing properties
- **File Handling and Performance:**
  - Folder uploads
  - REST service
  - Tested to handle many thousands of nodes and edges
- **Other Achievements:**
  - Dedicated Amazon Web Server with deployed code
  - Unit testing suite with continuous integration
  - URL Exporting for sharing analysis
  - Open source GitHub repository

Questions