Minecraft Exploration via Neo4j

2021.05.04 Jason Snouffer, Ron Neely, Kevin Malone, Jonathan Vedro

Project Overview

Our project examines Minecraft, as a topic, from 4 different perspectives, each from a different community. The 4 perspectives from which we examine Minecraft from are:

- Reddit (Jason Snouffer)
- Ycombinator HackerNews (Ron Neely)
- Speedrun.com (Kevin Malone)
- YouTube (Jonathan Vedro).

Conclusions are drawn as to the depth of relationships between conversationalists, depth of topic comments, and related topics via Cypher queries made directly from Neo4j's web UI upon data gathered into a Neo4j graph database.

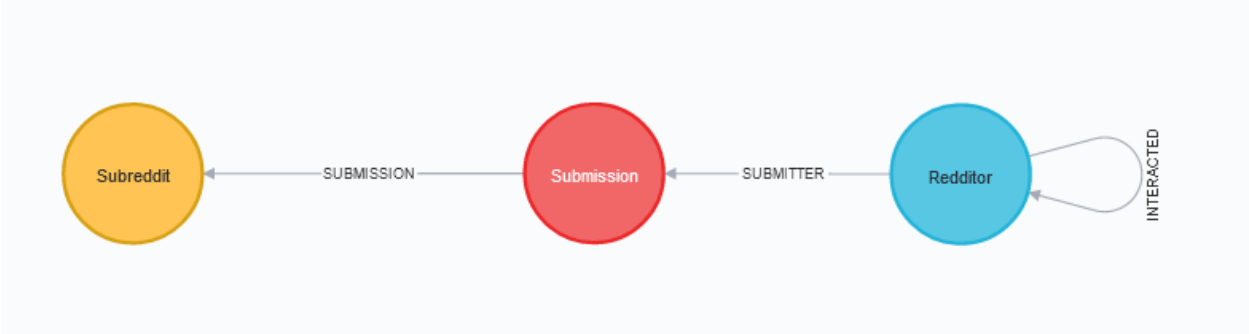
Scraping and Database Modeling

We used Python as our base language and NeoModel https://pypi.org/project/neomodel/ to define our schemas and populate our Neo4j graph database. APIs and schemas for each source are discussed below.

Reddit

Reddit is a network of communities based on people's interests. Subreddits are user-created areas of interest where discussions on Reddit are organized. The Minecraft Subreddit (https://www.reddit.com/r/Minecraft/) was used as the subject for our Reddit data collection. Reddit provides a full featured API through REST. We used a Python library, known as PRAW (The Python Reddit API Wrapper) to collect data from the Minecraft Subreddit.

Database Schema:



The Minecraft Subreddit data was persisted to Neo4j as nodes for each Subreddit, Submission, and Redditor, with directed relationships from Redditor to Submission and Submission to Subreddit, and a bi-directional "INTERACTED" relationship between Redditors.

This schema was chosen so that we could explore how strongly inter-connected subreddit submitters and commenters are.

Ycombinator Hacker News (HN)

Hacker news is a social news website focused on entrepreneurship and computer science. Stories on HN are intended to share intellectual curiosity about a topic. Authors can comment on stories and comment on comments.

HN provides an API that allows access to its raw json data which began to be posted in 2007. However, the HN API is not searchable. Algolia.com created a searchable API that runs on top of HN's API. By example, query results for 'minecraft' can be found in via

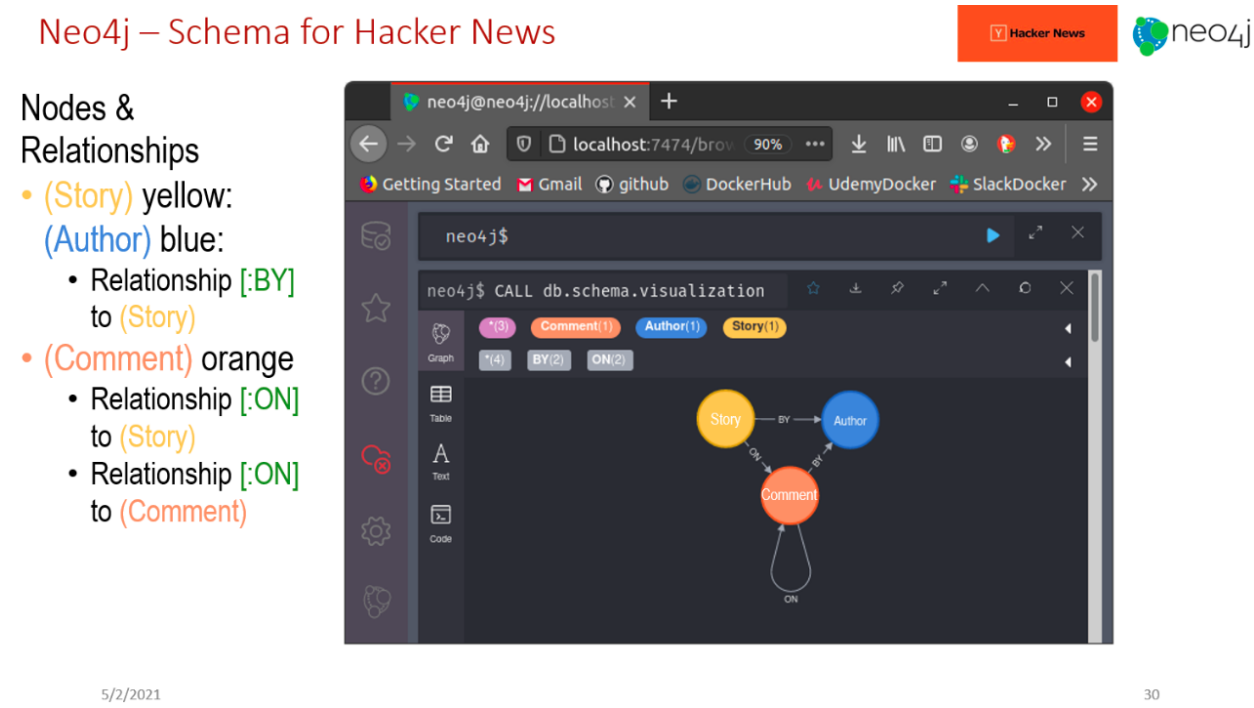
https://hn.algolia.com/api/v1/search?query=minecraft&page=0

Once a HN node_id is known, a story and its comment(s) can be retrieved directly from HN. For example, the following comment's parent is a story. The comment also has a kid comment.

https://hacker-news.firebaseio.com/v0/item/8203867.json?print=pretty


```
{
  "by" : "infogulch",
  "id" : 8203867,
  "kids" : [ 8204231 ],
  "parent" : 8202964,
  "text" : "As parent said, …",
  "time" : 1408560832,
  "type" : "comment"
}
```


The database schema chosen for our Neo4j instance has nodes for Authors, Stories, and Comments. A Story or Comment is [BY] an Author. A comment in [ON] a Story or Comment.



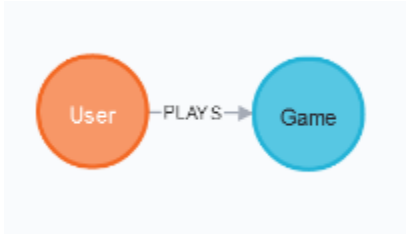
This schema was chosen so that we could maximally explore top commenters and dense relationships between comments on stories.

Speedrun.com

Speedrun.com is a website that allows users to submit the fastest times that it took to complete a videogame that they have played. Users attempt to beat games such as Minecraft in as little time as possible and submit their “runs” to the game leaderboards.

A RESTful API is available for Speedrun.com which allows you to gather website information in the form of raw json data. An example of this API can be seen when finding the “personal-best” times that user has submitted to Speedrun.com:
https://www.speedrun.com/api/v1/users/qjn1mw8m/personal-bests
The Beautiful Soup web scraper can also be used to search through the raw html of the site to acquire specific information. When using the RESTful API in tandem with the Beautiful Soup web scraper, precise user and game information relevant to that user can be acquired.

User and Game nodes were defined in this schema where a User PLAYS a Game.



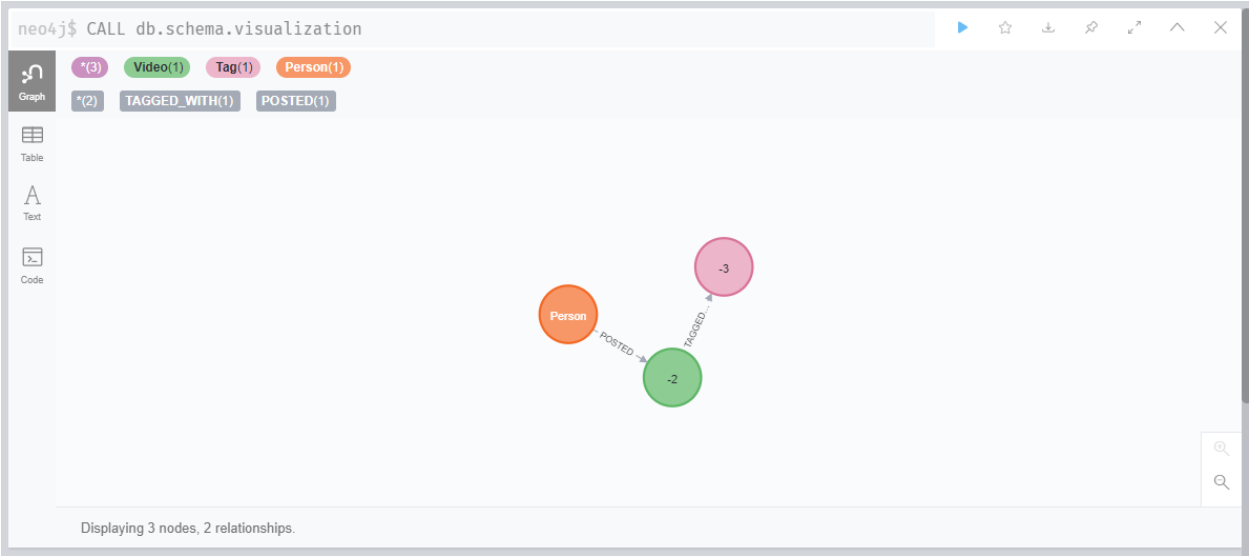
This schema was chosen on all Users that PLAY Minecraft to identify if other Users have overlap on the Games that they PLAY in addition to the unique Games they also PLAY.

YouTube

YouTube is a popular site for posting videos. Thousands of videos are uploaded onto it every day. There are videos on a wide variety of topics, including popular video games like Minecraft.

YouTube has a built-in public API for the general public to access various information about its videos. This information can be retrieved easily as long as the desired channel or video is known. For the purposes of this project, many of the same users from Speedrun.com were used here, as well. It is common for “speedrunners” to post their “runs” on streaming sites, such as YouTube.

The schema used for this project includes three different types of nodes: channel/user, video, and tag. Each video has a relationship with the user that posted it and all of the tags it is associated with. A user [POSTED] a video, and a video is [TAGGED_WITH] tags.



This schema was chosen so that we could further explore some of the top speedrunners and identify some of the relationships between their content, by looking at how different users used different tags.

System Description

Hardware Configuration

The tech stack was deployed to a cloud-hosted virtual machine from DigitalOcean. DigitalOcean droplets are Linux-based virtual machines (VMs) that run on top of virtualized hardware. A droplet was used because of its ease-of-use, speed, and flexibility. The droplet was configured with 8 shared Premium Intel virtual CPUs, 16 GB RAM, 50 GB solid-state drive (SSD), and a 25 GB detachable HDD volume.

Software Configuration

The DigitalOcean droplet was configured with Ubuntu 20.04 (LTS) 64-bit Linux OS. The following packages were also installed: Docker 20.10.5, docker-compose 1.28.5, Python 3.8.5. Neo4j Community Edition 4.1.6 was deployed as a Docker container, orchestrated by docker-compose. The container image and docker-compose spec are from bitnami, a library of cloud-native software stacks and virtual appliances.

Project Monitoring

Neo4j Enterprise Edition can be monitored using Prometheus and Grafana. Prometheus is an open source system and service monitoring system, which collects metrics from via a pull model. Grafana is a popular and powerful open source time series analytics and interactive visualization platform. Grafana is used to visualize the data stored in Prometheus (and other sources). Neo4j Community Edition can be monitored using Halin, a cluster-enabled monitoring tool for Neo4j, that provides insights into live metrics, queries, configuration and more.

Issues and Challenges

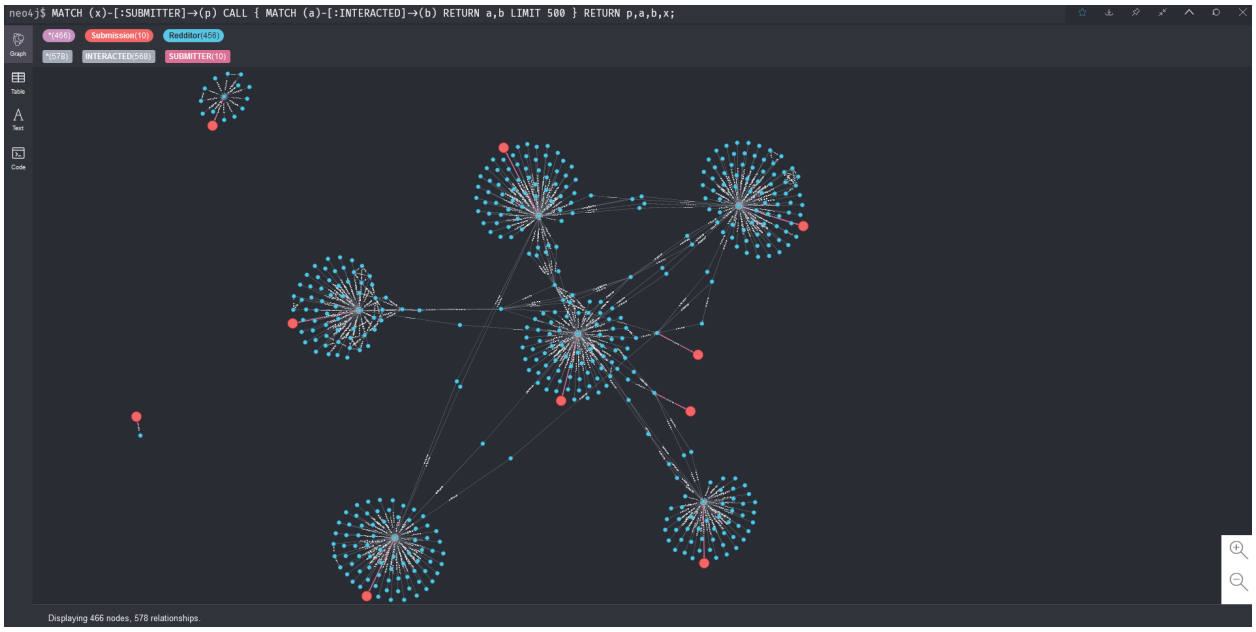
Ingesting a lot of data into Neo4j is a time consuming operation. Each node and relationship needs to be created one at a time. We used Redis Queue to parallelize this operation across all 8 cores as a background operation to greatly speed up the ingress. Redis is an in-memory data structure store, used as a distributed, in-memory key–value database, cache and message broker. Redis Queue is a Python library for queueing jobs and processing them in the background with workers. It is backed by Redis and is designed to have a low barrier to entry.

Neo4j Community Edition only provides a web-based graph visualization tool, which performs poorly when trying to visualize several hundred nodes. Neo4j Enterprise Edition offers Neo4j Bloom, powered by WebGL, which offers significant performance improvements when visualizing large graphs.

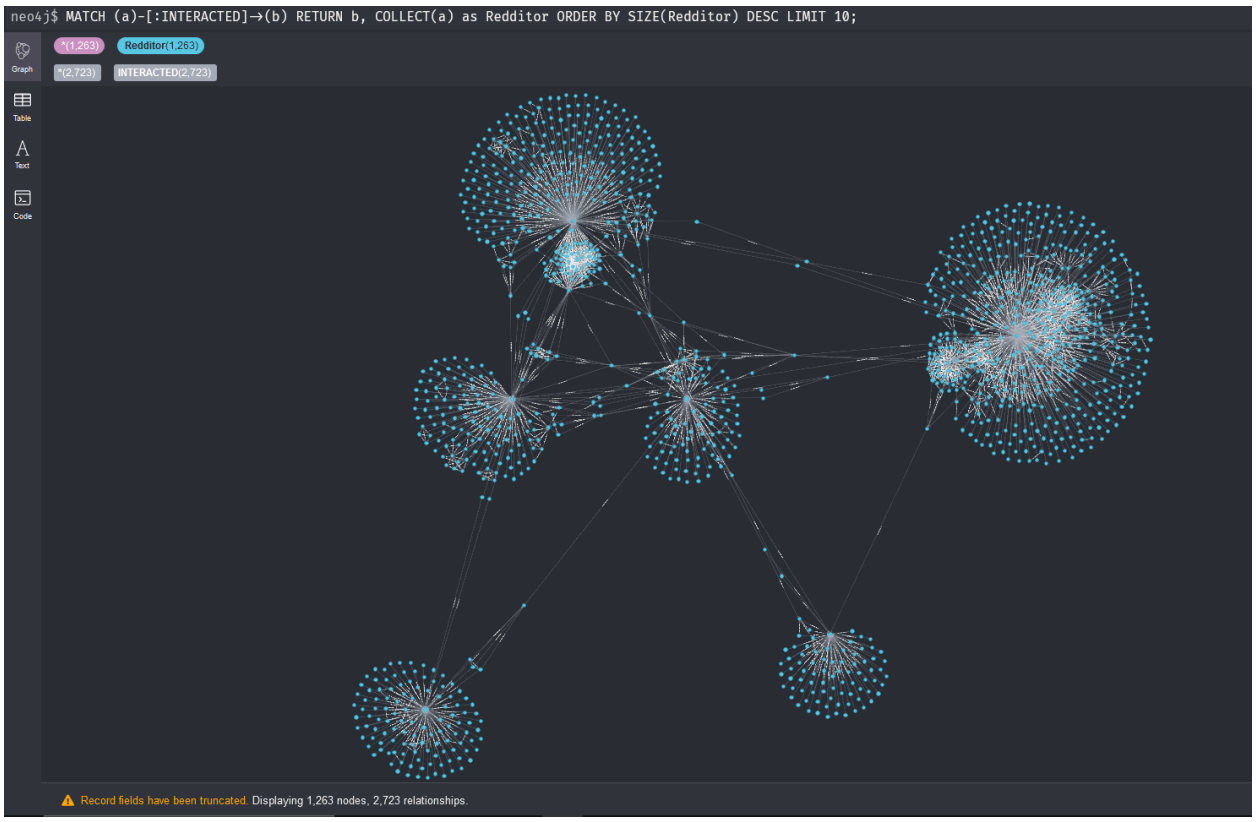
Conclusions

Reddit

Sample subgraph containing Redditors and Submissions nodes.



Top 10 Most Connected nodes

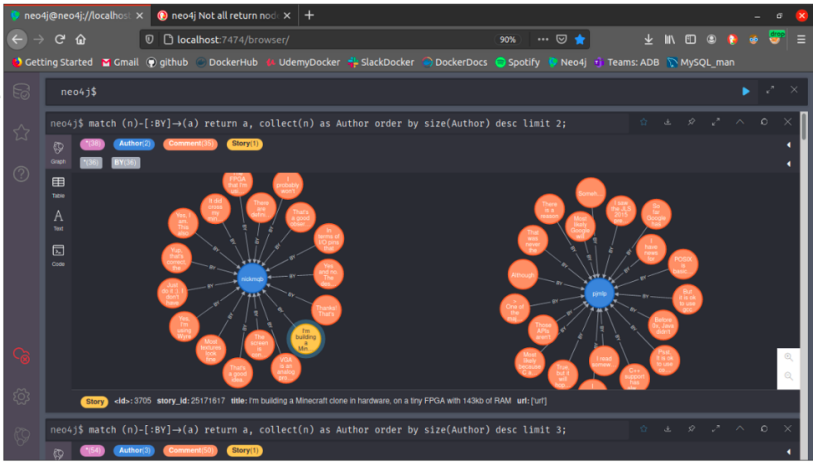


Ycombinator Hacker News

Using cypher we were able to query the top 2 commenters.

Neo4j – Top 2 commenters for top 20 Minecraft HN stories

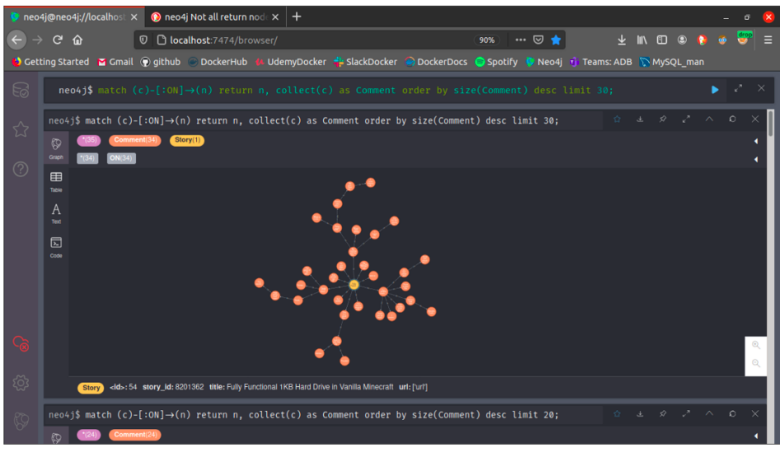
- Cypher: `match (n)-[:BY]->(a) return a, collect(n) as Author order by size(Author) desc limit 2;`
  - nickmqb
    - 16 comments
    - 1 story: Minecraft in FPGA
  - Pjmip
    - 19 comments
    - 0 stories



We were also able to query the densest comment tree.

Neo4j – Densest comment tree for top Minecraft HN stories

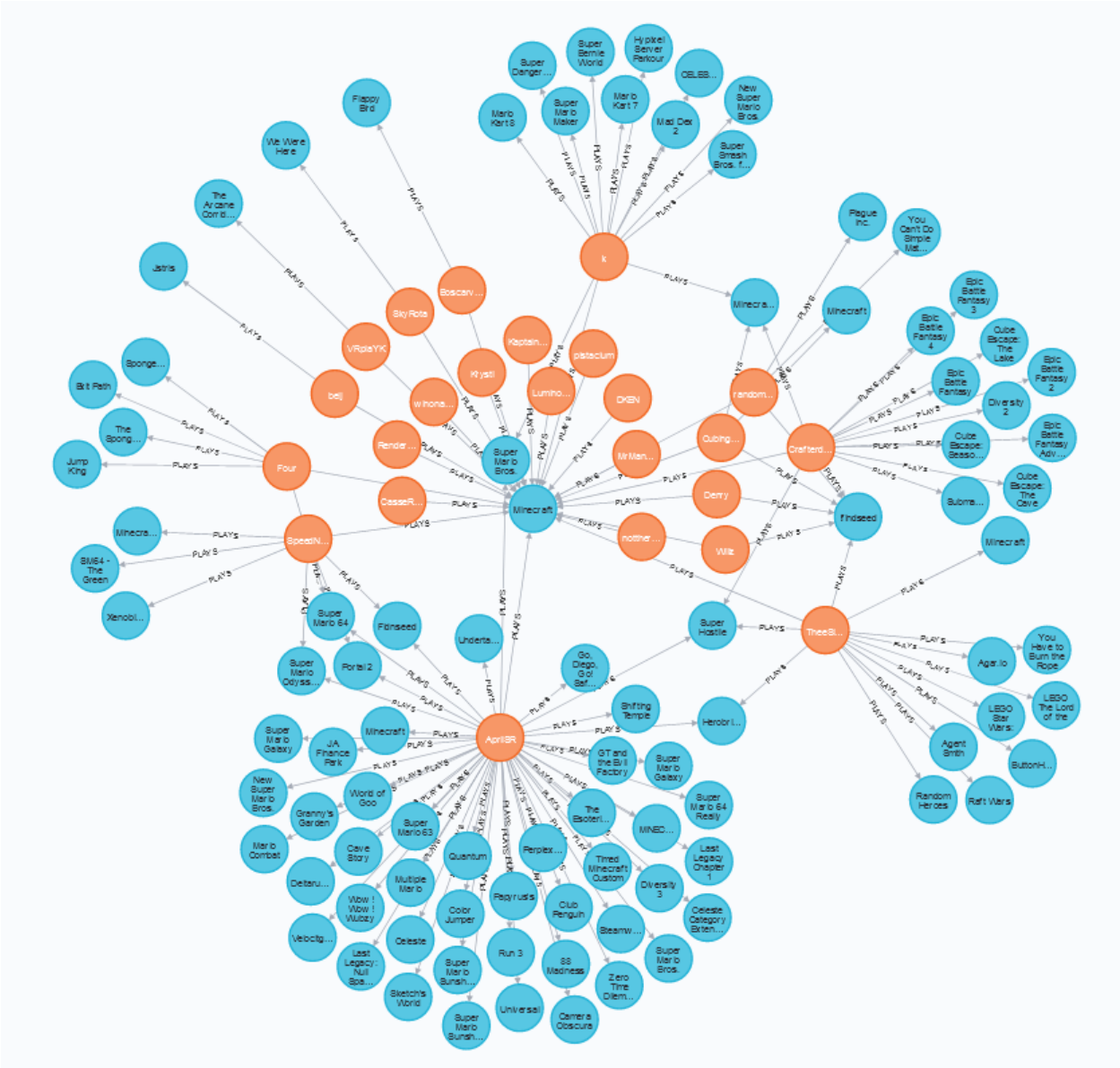
- Cypher: `match (c)-[:ON]->(n) return n, collect(c) as Comment order by size(Comment) desc limit 30;`
  - Our top story: 1k hard drive in Minecraft (no surprise)
  - Comments on comments go 5 deep with limit



Conclusions reached about the above queries are that Neo4j provides a very interesting way to query nodes and relationships between nodes. These queries would be more difficult using a non-graph database such as MySQL Server. Additionally we found Neo4j great for visualizing small results, and not so great at returning results with lots of relationships and nodes.

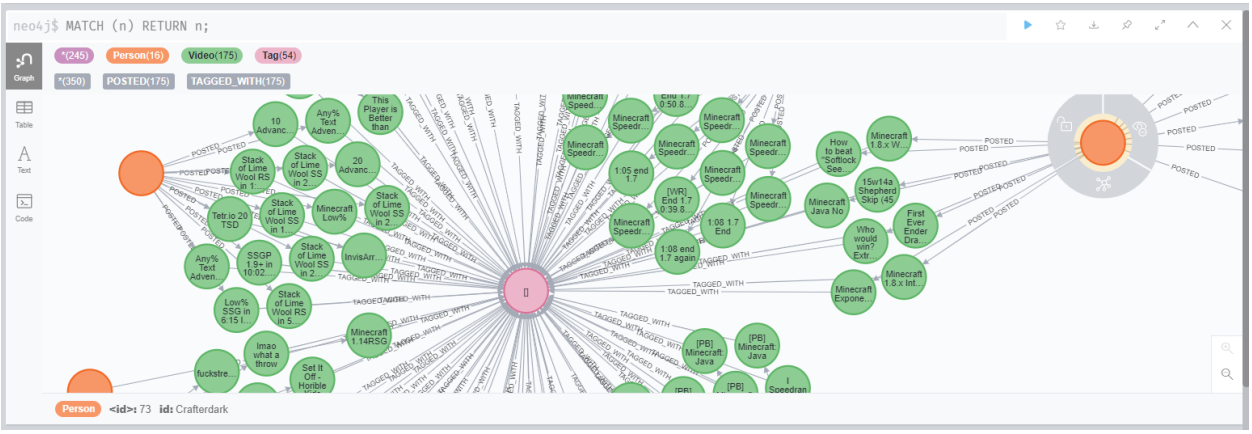
Speedrun.com

Through the use of Cypher queries, we were able to combine an assortment of users who have submitted runs to the Minecraft leaderboards and see which games they played in addition to Minecraft. Neo4j demonstrates the advantage to using graph databases by clearly displaying the games that multiple users have played vs games that only one user has played. However, had I continued with pulling every single user from the Minecraft leaderboards (2,172 Users vs the 24 Users I used), the number of nodes displayed would create substantial work for the processor and the graph would become very busy.

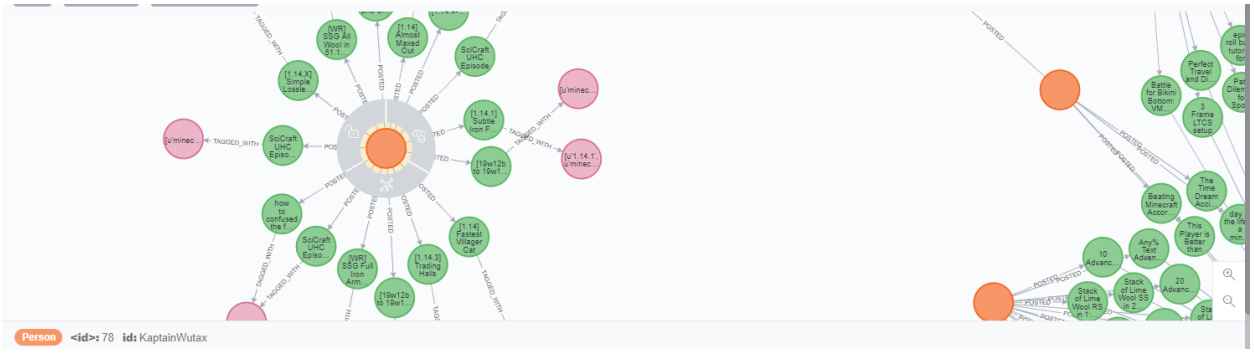


YouTube

Using cypher, we were able to bring all of the users and their most recent videos together, into one graph. Below, you can see a sampling of these videos, many of which are untagged. The node representing the user “Crafterdark” is highlighted.



These nodes are clustered by user, and each video's tags are also visible. For example, here are all of the nodes for the user "KaptainWutax"'s videos:



(Note: the tag '1.14.1' indicates the version of Minecraft that is being played for that video.)
In conclusion, Neo4j was able to display all of these relationships in a concise and visually interesting manner. However, the limitations of Neo4j also meant that the analyzed data had to be limited.

Future Recommendations

We built a robust scraper that scrapes 4 major social websites about 'minecraft' and stores the results in a Neo4j. Given more time on the project we could add the ability to search for and scrape other topics. Additionally we could explore other top topics that are related to the 'minecraft' topic we chose for this paper.