Basic Information:

Title: A day in the life of an ethereum miner

Email: mlu18@dons.usfca.edu

Repo: https://github.com/Simonl07/eth-miner-viz

Background and Motivation: I am currently working with prof. EJ on Ethereum data analytics and I am able to access logs from a raw ethereum miner node (logs of raw broadcast of transactions and new blocks). I've always been interested in visualizing how blockchain works as a system and most existing visualizations visualize the network as a whole (http://ethviewer.live/ and https://txstreet.com/v/eth-btc). With this dataset at hand I want to visualize the system from the perspective of a single miner node, with concepts that are not present in network level visualizations such as a local/partial transaction pool, peers, and consensus. I am most excited about how I can convey the working of the consensus algorithm as blocks are coming in. This is an interactive dashboard with the goal of storytelling.

Project Objectives:

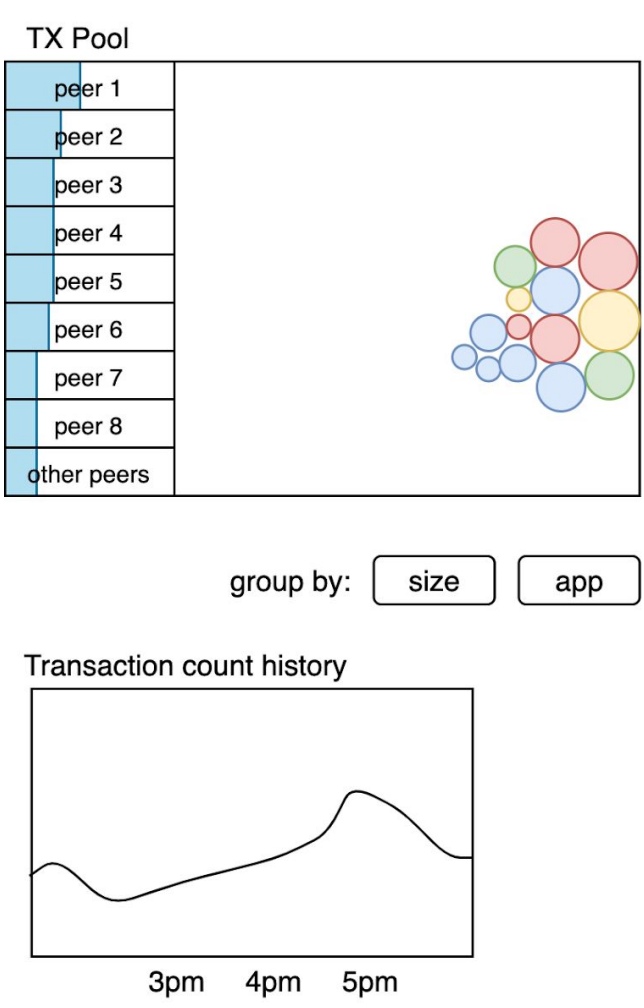
1. An interactive dashboard that provides an intuitive way to convey how blockchain works on a basic level through animations and interactions
2. Highlight, brush, and group the size and distribution of different types of transactions (smart contracts) and block producers (mining pool)
3. Visualize the peer nodes of the current miner, and how many messages they sent to the current node
4. Visualize how total number of transactions fluctuate over time during a day

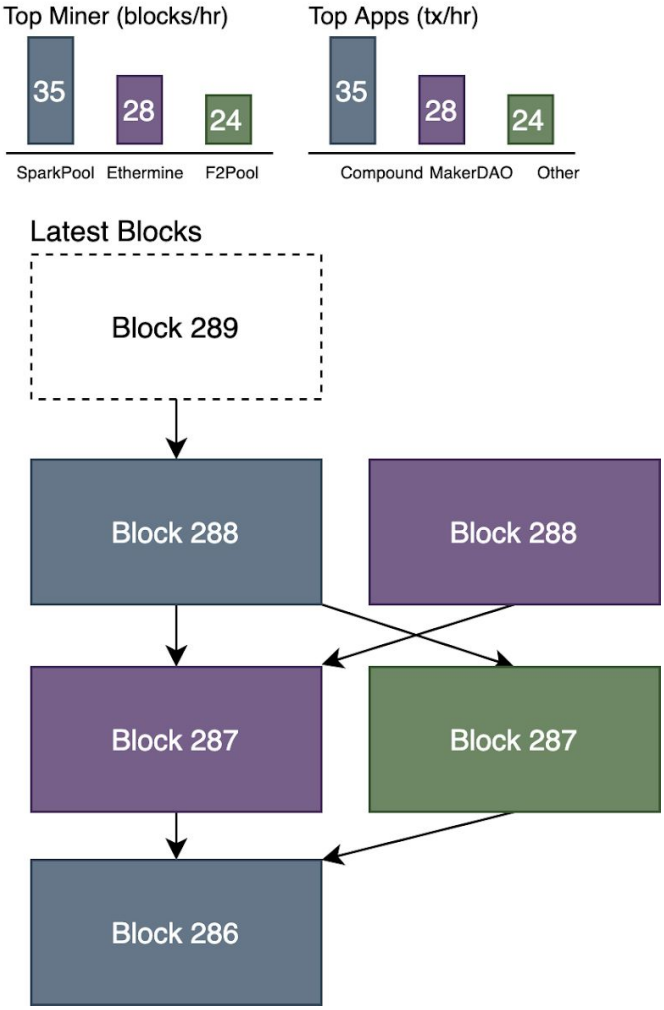
Data: Data is already collected from the node between 5/13/2020 and 5/27/2020. The full dataset is too big to be linked over the web so here's a sample of the raw transaction broadcast data set and the block data set that I'll work with.

Ideally, I would love to run a full miner node setup with real-time network data, but given limited time I will start by simulating 5/13/2020 in real-time using the data and provide the options to accelerate/decelerate time.

Data Processing: Data cleanup is minimal, but some custom transformation is needed in order to structure the data for visualization. The current data lacks names and pools associated with each address, so I will need to crawl the known names from the web in order to display the names in the visualization. The records must also be structured chronologically since the animation will simulate real-time. Data processing will be done using SQL in a cloud data warehouse.

Visualization Design:





The full dashboard is structured into four sections:

1. Top right barcharts show the top miners and apps. This aims to provide the distribution/proportion of the top miners and applications in the past hour.
2. Bottom Left shows a line chart showing the number of transactions over time. This aims to provide some context on how much transactions are changed over time. The reason I included this is because the transaction pool in the top left only shows transactions at this instance, therefore it is difficult to show the change.
3. Top left shows the transaction pool - a bubble chart with force-directed layout and collision with border. Force-directed (physics-based) layouts are chosen due to their ability to convey intent, especially for showing the priority of tx fee, making it very intuitive for those who are not familiar with Ethereum.
    a. Each bubble represents a single transaction
    b. Transaction size is radius and the app of the transaction is encoded with colors
    c. Upon receiving a transaction from a peer, the bubble spawn on top of the peer boxes on the left and animate towards its force-directed location
    d. The peer box shows a mini bar chart race that ranks the peers based on the number of transactions sent from them (their proportion of contribution to the transaction pool).
    e. By default, transaction with higher gas price (tx fee) are pulled stronger towards the right side (suggests miner’s priority and intent to include it in the next block as soon as possible for maximum profit, and shows how some transactions with low tx fee are less attractive and likely to remain on the left)
    f. When one of the “group by” is toggled on, the force center changes into multiple clusters that are evenly distributed across the pool area and the bubbles naturally attract towards their cluster (app names, and transaction sizes).
    g. On click, bubbles of similar apps are highlighted, and others are dimmed.

4. The bottom right shows a node-link diagram that represents the blockchain as the current node sees.
    a. A dotted block will always be shown on the top indicating that this is the block that miners are working towards.
    b. Upon receiving a block, the block is colored by the miner (pool), linked to the previous block, and if it's a block at a new level, shift the entire list of blocks down. Once the block is added, transactions that exists in both the current transaction pool and the new block will be attracted to the block that includes them and disappear indicating their inclusion/finalization in the new block
    c. If received two blocks of the same height. They will render on the same level with related transactions flying and disappearing towards them. If these competing blocks are 6 levels behind the current height (reaching consensus on canonical chain), competing blocks' transactions will be re-released into the transaction pool (simulates miner's re-inclusion of transactions from forked blocks)
    d. On click of a block, the transactions within the block is populated in the transaction pool for visualizing distribution and grouping. User can return to the live transactions by clicking the dotted block

Must-Have Features:

1. Force-directed bubble chart: group by size, peers barchart, bubble highlight (brushing)
2. Top Miners
3. Transactions count history
4. Blockchain node-link without competing blocks (consensus visualization)

Optional Features:

1. Group by apps
2. Fancy animations (animations of transaction moving into/out of transaction pool)

3. Competing blocks visualization (release transactions from forked blocks back to tx pool)
4. Accelerate/decelerate simulation time

Project Schedule:

Week 1 (10/25): Data processing and minimal TX Pool functionality with basic force directed layout, continuous update, and no grouping

Week 2 (11/02) & Alpha: TX Pool completes with peer bars, highlight, and grouping, and transaction count over time

Week 3 (11/09): Minimal block node-link visualization (shifting, coloring)

Week 4 (11/16) & Beta: Complete block node-link with click behavior, top miner, and top apps bar chart

Week 5 (11/23): Competing blocks, transaction release back into pool after 6 levels, simulation speed