## Introduction

E-Commerce is increasingly becoming a key part of the global economy and our daily lives. Prior to the invention of cryptocurrencies, e-commerce relied almost exclusively on financial institutions serving as trusted third parties to process electronic payments. This conventional e-commerce system suffers from the inherent weaknesses of the trust based model which include a lack of transparency, fraud and high operational costs. Bitcoin was invented to address these weaknesses and is the first decentralized digital currency as it works without a central bank or single administrator. This project focuses on Bitcoin, but there exist many cryptocurrencies now.

Bitcoin transaction are peer-to-peer - they are sent directly from one address / wallet to another. These transactions are verified by network nodes through cryptography and recorded in a public distributed ledger called a blockchain (proof-of-work) to prevent double-spending.

Mining is the process of adding transaction records to Bitcoin's public ledger of past transactions. This involves solving complex mathematical problems and was intentionally designed to be computationally-intensive. Mining is also the mechanism used to introduce Bitcoins into the system: Miners are paid any transaction fees as well as a "subsidy" of newly created coins. Hence mining serves the purpose of disseminating new coins in a decentralized manner as well as motivating people to provide security for the system.

In this project, we have used Spark and Scala. This report will detail the following: extracting transactions from the raw bitcoin files; creating the vertices and edges; and building a directed graph model to detect and extract triangles.

Our additional project goals are as follows:

1. Visualization of the bitcoin data (for 5 hours from 2013/01/01 00.00 to 04.59)
2. Analysis of Wikipedia top 30 donors, total donations and graph for the data period
3. Analysis of CryptoLocker ransomware bitcoin address

## Transforming the raw data into a directed graph and the triangle count implementation

**Introduction to the data set**

The source data from the project is raw bitcoin information, in the form of 5 csv datasets in HDFS. The following were identified as the most important, for extracting the transactions and creating the edges of the graph.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Vin | Transactions | Vout |
| Items | 132,406,270 | 53,870,404 | 147,820,700 |
| Description | Links to previous Tx | Tx, block and epoch | Tx, amount and pubKey |

Analysis also showed that a few lines had wrong number of fields and format errors in amount and epoch. Spark was chosen due to the dataset size and the flexibility and richness of the parallel operations. RDD was used rather than DataFrames and sparkSQL.

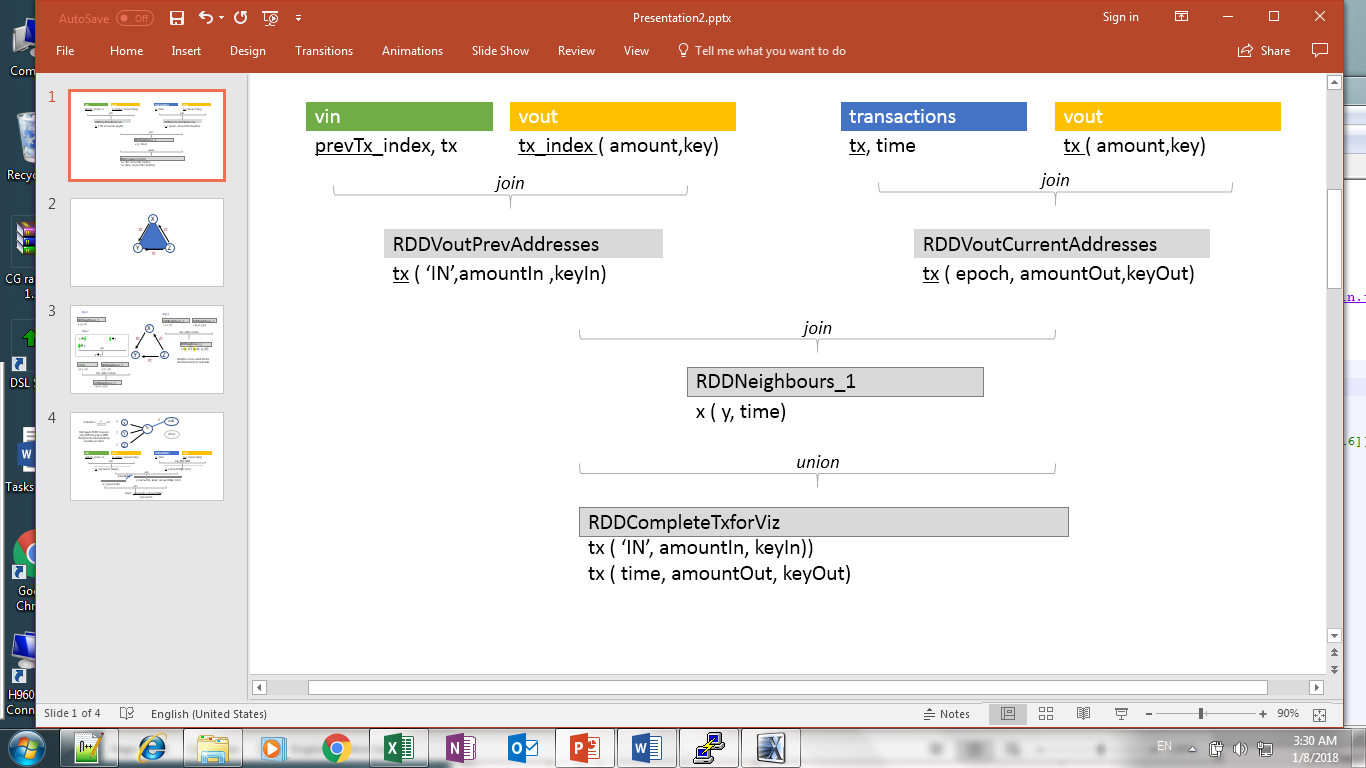
**Data filtering and clean up**

The vin, vout and transaction data was filtered for correct number of fields and checked for formatting exceptions, when creating the RDDs.

RDD = .filter(x => x.length == N && Try(x(m).toFloat).isSuccess)

**Creation of the graph edges**

A directed graph consists of directed edges. Vin was used to do a lookup with vout to get the inputs. Vout was combined with the transaction to get the time and outputs. A tag ‘IN’ was added to RDDVoutPrevAddresses to make the fields match with RDDVoutCurrentAddresses. This allowed creation of the edges using a join and creation of complete transaction for visualization using a union.



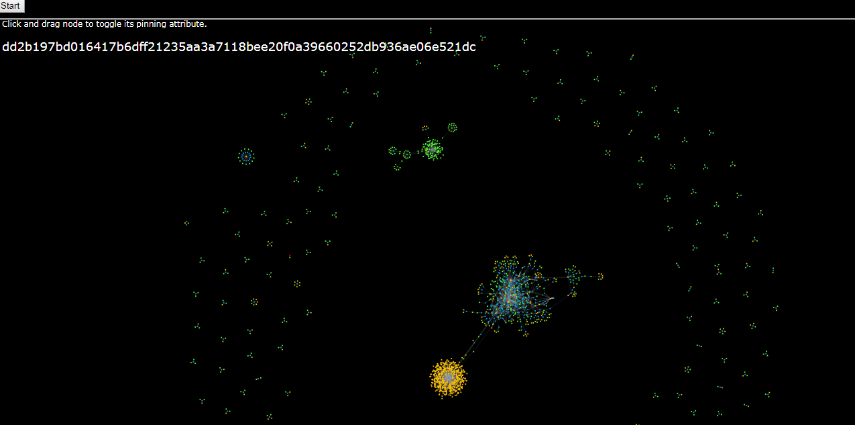
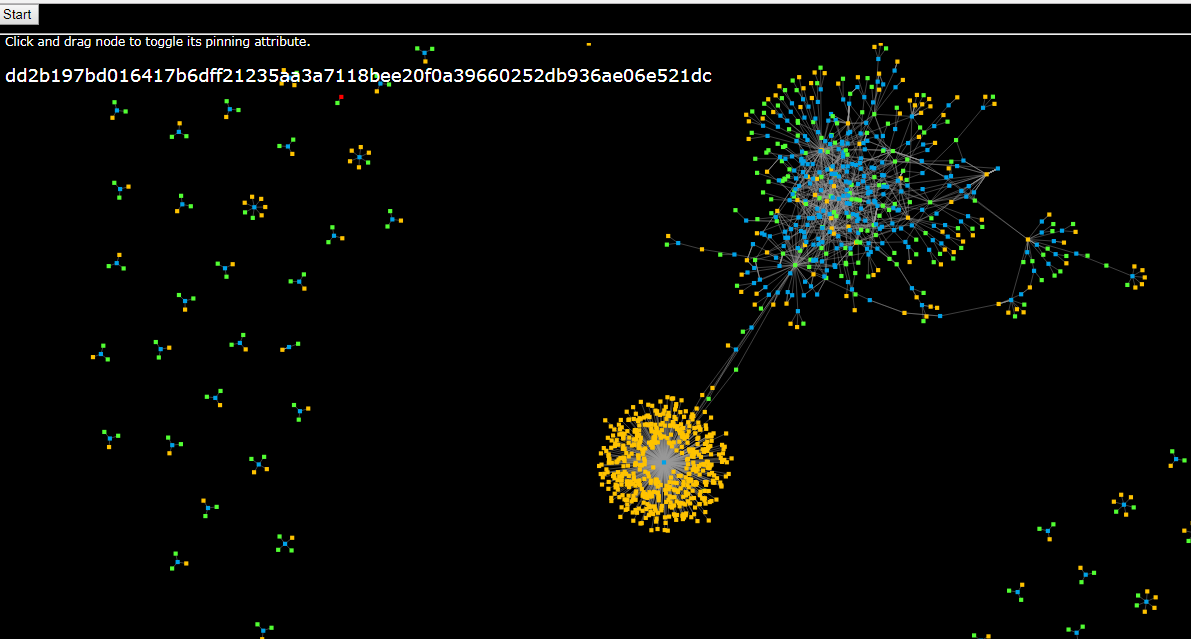
The process creates the edges. Vertices are the bitcoin addresses. A directed edge has an additional time parameter.

|  |
| --- |
| Sample of the edges created (total: 507,410,803) |
| ({139KWcy5ZUkAZgAa1tYqb6fgX28M7kRu7U},({1MkMxL46tb5zV9LdRE9qi4RnRhTaLMnhMn},1415100777))  ({1dice7W2AicHosf5EL3GFDUVga7TgtPFn},({1J17uib67vYNVcvaoVxitPnsf76CvccSRG},1360273173))  ({1dice7W2AicHosf5EL3GFDUVga7TgtPFn},({1PeohaRGaTF8cSzDqP1yYfzDah66xiriEQ},1360273173))  ({1vAipePmrGxZgwv57jwU8PX1zUSVZi8a3},({1GbEZcvqj4BYsgw1D6Z62gZS9zt32FtrQQ},1387171557))  ({1vAipePmrGxZgwv57jwU8PX1zUSVZi8a3},({12pGg6GZABaq4hfsNPCjLUPHBQUST9tL2p},1387171557)) |

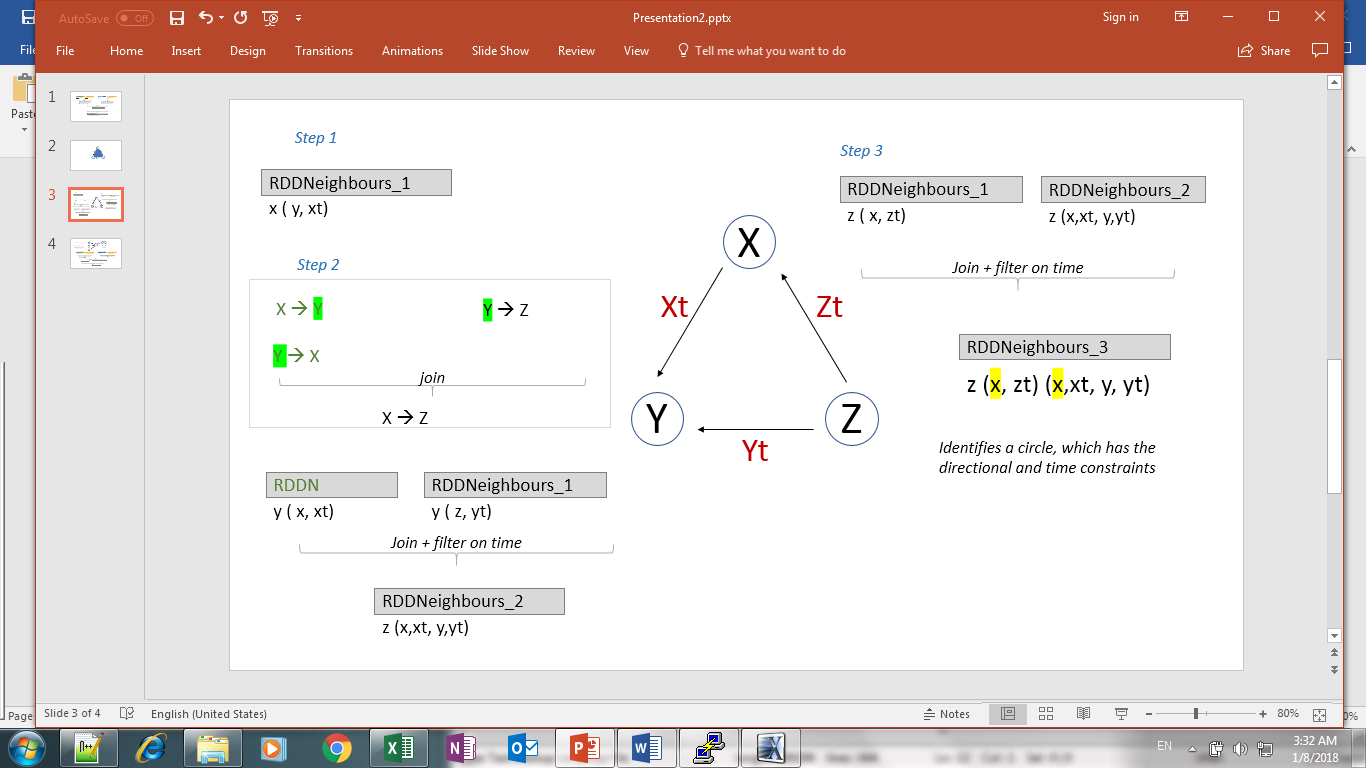
The edges are used to create the directed and time constrained the bitcoin payment triangle.

**Visualization of the bitcoin data**

As part of the same process, RDDCompleteTxforViz gives the complete transaction item data in the form of (tx Array[ input amounts, pubkeys ]. This is used to generate visualizations using the vivaGraph (<https://github.com/anvaka/VivaGraphJS>) JavaScript libraries. The working model in included in the attachments.



**Triangle detection and extraction algorithm**



The edges RDD gives X🡪Y collection, which is the 1st step, with no further processing required. Step 2 is where X 🡪 Y and Y 🡪 Z is linked up to form X 🡪 Z. Since RDD (K,V) tuple operations joins with K, RDDN creates a (V,K) tuple and does a join, effectively giving X 🡪 Z. In step 3 what remains is only to find an edge that matches with the X 🡪 Z. This is accomplished via a join. The triangle detection uses 2 joins and at each, eliminates the links based on the time constraint.

Following shows the sample edges and output at each step, for a hypothetical sample

|  |  |  |
| --- | --- | --- |
| Step 1 | Step 2 | Step 3 |
| (w1,(w2,100))  (w1,(w12,205))  (w2,(w4,101))  (w2,(w10,201))  (w3,(w10,200))  (w4,(w1,120)) | (w1,((w12,205),(w4,120)))  (w2,((w4,101),(w1,100)))  (w2,((w10,201),(w1,100)))  (w4,((w1,120),(w2,101))) | (w4,((w1,120),(w1,100,w2,101))) |

**Outcome of the implementation**

The edge detection and triangle generation code was written in Scala, and run on the cluster. The DAG shows the progress. However, the last stage (triangle generation – step 3) did not complete for the full data set.

**Evaluation of the algorithm**

The algorithm used is based on an adjacency-lists representation. The triangles are built from this information, subject the transaction time constraints. There is no cost in generating step 1, which is the direct result of the edge generation.

The join in the second step is costly. The 53,870,404 transactions result in 507,410,803 edges in RDDNeighbours\_1.

For example, the transaction 16caa8aea01154d1b6db067337395329725dd95c10bfe7de96810f7111f4cf57 has 235 Inputs and 1653 Outputs resulting in 388,455 edges.

In the join, this can result in a size of 507,410,803 to a max of (507,410,803/2)2 = 6.4x1016 combinations.

This makes traditional approaches impossible. Even in a spark environment, it is not possible to use a parallelized join, due to the shuffle and repartition overhead. The full dataset did not complete. GraphX, with its Pregel based implementation was also not able to complete the task. The team also tried DataFrames and GraphFrames which failed.

The main issue is with the graph density. Unlike other graphs, the bitcoin graph has

1. Almost always a next edge to any given edge, unless it’s an unspent transaction.
2. An edge going in to a payment hub would have a massive explosion in edge counts

The complexity arises mainly from the size as well as the high connection density. A less dense adjacency list would have yielded less 2 edge terms, making the computation feasible.

**Taking Advantage of External Libraries: GraphX vs Graphframes**

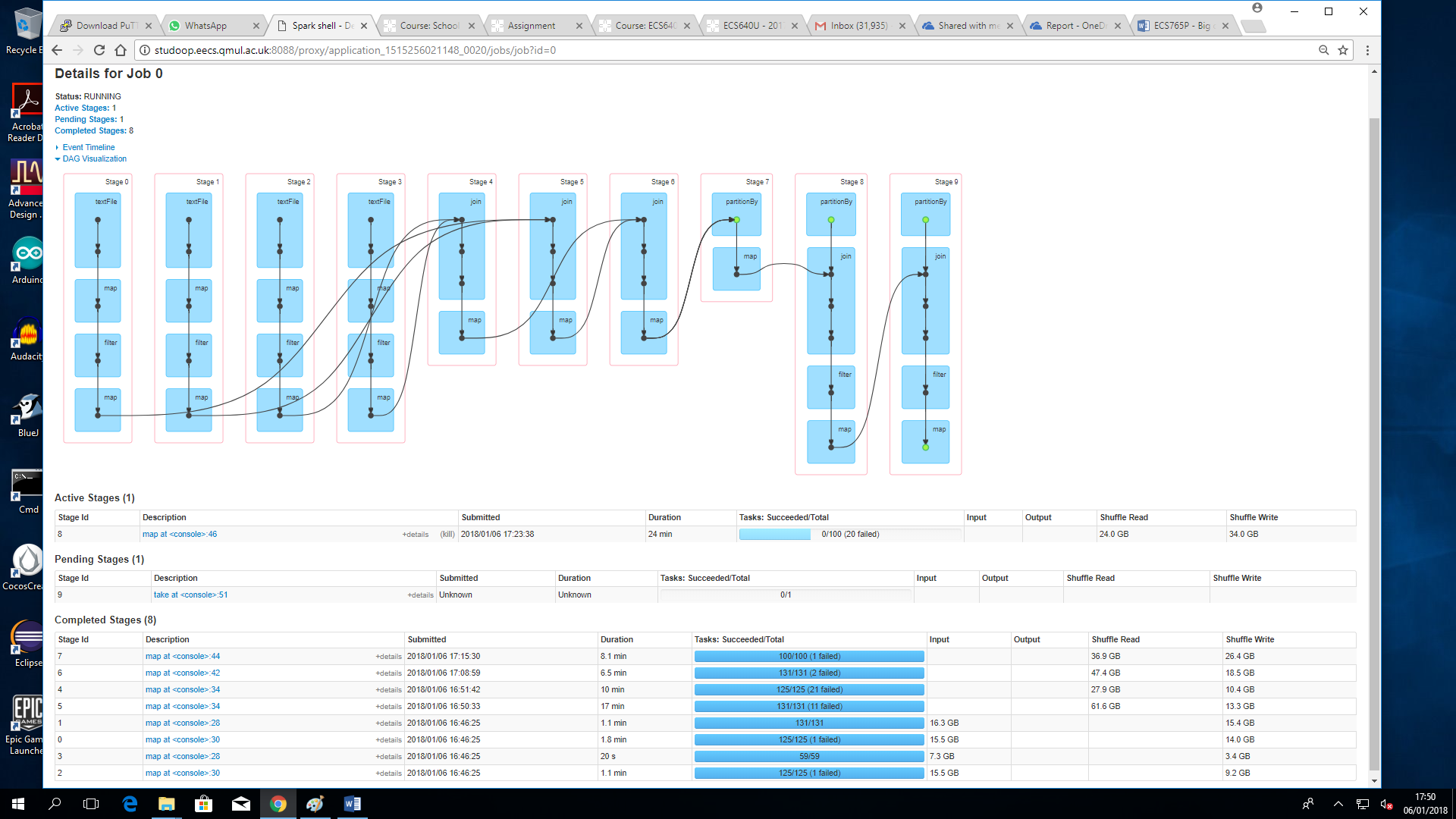
Besides working on our own original algorithm for triangle count, part of our group worked on using the ready libraries in order to increase our understanding of graph algorithms. Firstly, we started with GraphX as it’s the one that has been talked about in the lectures. GraphX extends the Spark RDD to introduce RDD Graph. It’s available in Scala and also, you can use a variant of the original Pregel API as well.

However, upon getting familiar with Spark and Spark Dataframes, part of our group started using DataFrames to build up our tables and graphs. Using SQL to do the joins, we even wrote another original algorithm on triangle count with the help of SQL and DataFrames. After getting efficient at DataFrames, we have read about Graphframes which is a package for Spark that provides dataframe based graphs. Graphframes has been reviewed as an improved version of GraphX, more user friendly and efficient.[[1]](#footnote-1)

Building the structure on DataFrames, using Graphframes came naturally. However, two problems emerged. Firstly, ready-to-use triangle count algorithm did not serve our purpose. The algorithm works for undirected graphs such as social network graphs and does not give a complete cycle. For that reason, we could use the motif finding feature where we could define our own motifs to look for. Yet, this time the algorithm has become computationally expensive and we couldn’t manage to complete our task. We could get all of the algorithms to work with a really small data sample. However, when we tried it with a monthly slice of bitcoin data, none of the algorithms completed their tasks.

Graphframes is claimed to perform optimizations under the hood including using the Graphx Pregel API[[2]](#footnote-2) for Pagerank algorithm. Yet, Pagerank algorithm failed to complete its run on our 3 monthly bitcoin data. Then we turned to Graphx, hoping our failures were resulting from Graphframes. We converted our dataframe tables into RDDs and tried the same algorithms with GraphX. However, we ended up with the same result.

Underlying reasons could be stemming from our approach in joining the different bitcoin tables in order to end up with a final table. It could also be that we are skipping some optimization steps when running the algorithms in the libraries. Either way, it was puzzling not being able to run already optimized algorithms on data that is 1/10ths of the complete bitcoin data.

**DAG visualization for the complete triangle count process**

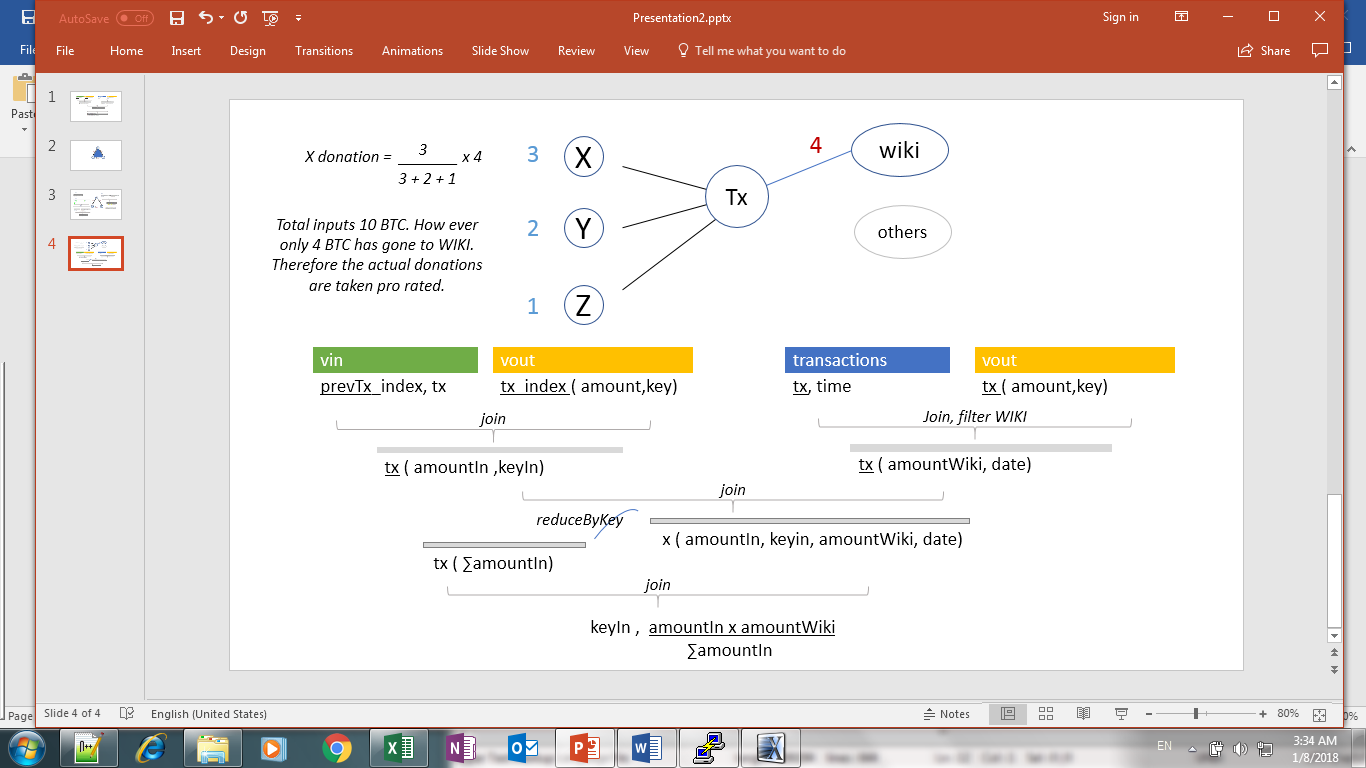
## WikiLeaks Donations Analysis

WikiLeaks is an organization that promotes publishing of, and itself publishes, secret information provided by anonymous sources. On June 15th 2011, WikiLeaks has announced that it will be taking Bitcoin donations. Since the organization has a publicly owned bitcoin public key[[3]](#footnote-3), it is possible to track all the donations made to WikiLeaks until the latest data available in our cluster; late 2014.

****In this analysis, we will present three outcomes:

* Evaluation of the top 30 donors
* Total donation sum
* Evaluation of donations over time

**Methodology and Assumptions used**



1. **Evaluation of the top 30 donors**

Out of the 30 top donors, only one of them is tagged with a real identity on blockchain.info and that is the public key of Mt.Gox, the bitcoin market in Japan that bankrupted in February 2014. 24 public keys out 30 are used only for the WikiLeaks donations transaction. They don’t have any other outgoing transactions.

|  |  |  |  |
| --- | --- | --- | --- |
| Public Key | Total Donation | No of Payments | Notes |
| 1LNWw6yCxkUmkhArb2Nf2MPw6vG7u5WG7q | 96.19 | 7 | Mt. Gox |
| 1BiZSHyPuVLiPNV7GTg5okStXKm1FTNSb7 | 74.30 | 1 | Used for 1 tx |
| 1FZVyaDzeQD2N85Ea6kbcW2LW3FdUxrfdP | 63.33 | 1 | Used for 1 tx |
| 1MvwZ6e2PA9QxNimjvq5VuzXpicW2rbzf3 | 55.36 | 5 |  |
| 17SC6Ps71YMtexXjejdcV7HFJDYXKYDrKY | 50.00 | 1 |  |
| 19TCgtx62HQmaaGy8WNhLvoLXLr7LvaDYn | 49.98 | 1 | Used for 1 tx |
| 1GJx75qFbevZpeJ2giYbnrBjXbpnDdQZUq | 46.45 | 1 | Used for 1 tx |
| 17zeTMh8xXeXXjZnbULXV3g3t3f7pftnEh | 45.00 | 1 | Used for 1 tx |
| 13vFf3MZKxSA3Q9e14c8xUXbMpHQn1wCgq | 45.00 | 1 | Used for 1 tx |
| 1C2UNnYzEzmJ5qpyXDrcA3k6MLru6yiJF3 | 42.05 | 1 | Used for 1 tx |
| 12fLFY2cMoUstdQESwu7U8qNP9BpDHWwUC | 41.00 | 1 | Used for 1 tx |
| 1Cj24RN6apbAV36ErYbEM7xKud11ozTVHS | 39.05 | 1 |  |
| 1Huk5VnjvJY6dVyGb3w72wMrqSNZvga1Nz | 32.27 | 1 | Used for 1 tx |
| 19y1ENhaiswqpT7dwZgDciBEicQJNGu41d | 32.24 | 1 | Used for 1 tx |
| 1GYMagx3YrWr9C8a2itabnw7zftP7suCtW | 32.07 | 1 | Used for 1 tx |
| 1jLofSmwEBz7Vmo9R7mNcjMcgBZKvi1Ss | 31.66 | 1 | Used for 1 tx |
| 1CP4bsMWuqCfVFgU7FPTS2Jmi5jrA8SA9W | 30.04 | 1 | Used for 1 tx |
| 1Pp6VksaHVwEzHM43CLk4KXwLQdtAXQdPP | 29.09 | 1 | Used for 1 tx |
| 1K7Cc6wANHnfsXMT43LBqEn7tMVeW38uqX | 28.98 | 1 | Used for 1 tx |
| 18pcznb96bbVE1mR7Di3hK7oWKsA1fDqhJ | 28.81 | 1 | Used for 1 tx |
| 1NLzUGGsFt68Y4qZ4KFmbdP7mKueZfqocq | 27.62 | 1 | Used for 1 tx |
| 1Foa8SZCn9cmXJUf4sa3GnhxfkAHYSbZaG | 25.00 | 2 | Used for 2 tx, both to WikiLeaks |
| 18GYce4UbZgQa8HLm9pACtVCWkM1iRAnxN | 25.00 | 1 | Used for 1 tx |
| 168FmNwh5wZ6MQgp93WpxiNcMkvzWUmvCB | 24.98 | 1 |  |
| 1DRSNwA1Nw2Mw4DdYU966muEYQ145XEoEH | 24.28 | 1 |  |
| 1LcKW8givtCqZeQXb7T1APy3TjoW4BMiKj | 24.13 | 1 | Used for 1 tx |
| 142JZ7CkMVkinTBe4dXeHs5J865b3vZPre | 23.58 | 1 | Used for 1 tx |
| 12oxboyGq2q2bx5JwtMU9JYvWgXEx8LWF8 | 23.23 | 1 | Used for 1 tx |
| 161SkPXMWuMvekXVY329i7sBNJ9oyTPuDr | 21.66 | 1 | Used for 1 tx |
| 17Mad42WnwrTs52buhjhSuQj71BwHjcz4x | 20.68 | 1 | Used for 1 tx |

1. **Total amount of donations received for the period between 05.06.2011 and 12.12.2014**

The total payments gone to Wikipedia for the period, which was **3885.77** BTC

1. **WikiLeaks Donations Over Time**

In the graph below, donations are aggregated monthly. Donations seems to have climaxed around November 2011 and decreased from then on. However, BTC showed near exponential appreciation in value, which would have resulted in increasing real term value over time.

The graph shows the donations, aggregated per month.

## Ransomware Analysis

In this section, we trail the footprints of a cyberattack that forced the victims to make a bitcoin transaction to the attacker. CyptoLocker was the name of the ransomware attack that occurred from 5 September 2013 to late-May 2014. The attack was basically a Trojan that encrypted infected computer’s files. Decryption was only possible with a private key that is released only after the victim pays the stated amount.

Among the payment options to the attacker was a bitcoin payment. Victims are expected make bitcoin payment that is an equivalent of 300 USD to a public key that is shared with the victim. Our analysis is based on getting the known public keys of the ransomware and tracing the transactions that are directed at these public keys. We have acquired the public keys through scanning the web, bleepingcomputer.com’s detailed CryptoLocker information guide[[4]](#footnote-4), tracking comments in a popular reddit thread[[5]](#footnote-5) and from the thesis of Mr. Michele Spagnuolo of Politecnico di Milano that examined the same issue with a software in his paper[[6]](#footnote-6). We have managed to get 650 distinct bitcoin public keys that are said to collect the ransoms.

Early stages of the ransomware had static bitcoin addresses namely “18iEz617DoDp8CNQUyyrjCcC7XCGDf5SVb”[[7]](#footnote-7) and “1KP72fBmh3XBRfuJDMn53APaqM6iMRspCh”[[8]](#footnote-8). However, after a while CryptoLocker started generating new bitcoin public keys for every instance of infection. This made tracking the payment harder. On top of that, attackers seemed to use just-dice.com to mix and launder the ransom sums[[9]](#footnote-9). Therefore, it’s not possible to track the money all the way to the end; yet for our purposes tracking 650 public keys will yield a few key insights.

At first, the ransom fee was chosen as 2 BTC. In time, due to the increase in BTC value against the dollar, attackers decreased the ransom value to 1 BTC, then 0.5 BTC and lastly to 0.3 BTC. So in the analysis we will only include transactions valued more than or equal to 0.3 BTC and less or equal to 2 BTC.

Looking at the dates between September 2013 and June 2014, we have found 1240 transactions meeting our criteria, totaling a sum of 1570.30 BTC.

**Top 20 public keys that received the highest number of payments**

|  |  |  |
| --- | --- | --- |
| no | pubkey | payment count |
| 1 | {1KP72fBmh3XBRfuJDMn53APaqM6iMRspCh} | 54 |
| 2 | {1K8Mz2g1FzNEUrZPDe4ffXJ5kqKnUYqG4L} | 47 |
| 3 | {19h9GRcx1ABHPEzVpmWqqA9TTgug6JNcBc} | 30 |
| 4 | {177HgELfgYkwNoXn3B4tMfA9X6bECwdpde} | 29 |
| 5 | {18iEz617DoDp8CNQUyyrjCcC7XCGDf5SVb} | 23 |
| 6 | {14bh9uqKzgqV6yqTqHbU2bdbpY3a8pktzR} | 21 |
| 7 | {18ixwb5si3AQBpjkUA91vGE3ESoynXd95g} | 21 |
| 8 | {12dGtJF5dM4QPZfYwZk7nsY8ABRZo4LBJh} | 18 |
| 9 | {132ckcKGxofNhZfG7bm9dv8kZGtVpNHwpr} | 17 |
| 10 | {1AEoiHY23fbBn8QiJ5y6oAjrhRY1Fb85uc} | 16 |
| 11 | {1CRfmhUmjQRTuP4jnPPtUAffbkL6Dvfn73} | 13 |
| 12 | {1Hx2Mz59JgTELeAgwpyMKpc35MqY5Rm5B4} | 13 |
| 13 | {1HyqQaTQmUT3FpnNMCBcpjBFv9BS7dnbZL} | 12 |
| 14 | {1J6D7xqYBUj49atDRPVBCA3L22NL6pviZJ} | 12 |
| 15 | {1GnBAQXrjsJPtLD41p4nMKWGiJvhtGDwST} | 11 |
| 16 | {1EeYeuEPWv3xWKGZ41QeLX515TkmQv5iA3} | 10 |
| 17 | {1HRFyimD3gw1cZeujv16qbDdkVZzvWUHJ7} | 10 |
| 18 | {19MNqUeDa78NfuSzJttaWbXVavc7mRZqj2} | 9 |
| 19 | {1Haq892MR5i9tHymDBAa13yFaGqsorMjYK} | 9 |
| 20 | {12mPaFLrUsqr18eh5LxJ2vjda4Tj79YNzB} | 8 |

**Ransom Payments Over Time**

Ransomware seems to have made the most damage in the last quarter if 2013. Number of ransom transactions seems to have fallen considerably in 2014. Since the ransom fee changes over time reading the table with help of transactions count would be helpful.

Price: 1 BTC

Price: 2 BTC

To sum up, with what is available to track, we have reached the number of ransom fee transactions and total sums. Moreover, we were able to list the top public keys in terms of received ransoms and lastly a trend analysis of the ransom transactions over time. Due to the nature of bitcoin it’s easy for attackers to cover their tracks as they have done so. Further analysis could be done find the strongly connected public keys to the ones already uncovered. However, since marginal cost of having another public key is virtually zero, there is no reason why attackers to continue using the already used public keys long enough to leave a meaningful trace.

1. https://www.quora.com/What-is-the-difference-between-GraphFrames-and-GraphX-in-Spark [↑](#footnote-ref-1)
2. <https://graphframes.github.io/user-guide.html#pagerank> [↑](#footnote-ref-2)
3. https://shop.wikileaks.org/donate [↑](#footnote-ref-3)
4. <https://www.bleepingcomputer.com/virus-removal/cryptolocker-ransomware-information#cryptolocker> [↑](#footnote-ref-4)
5. <https://www.reddit.com/r/Bitcoin/comments/1o53hl/disturbing_bitcoin_virus_encrypts_instead_of/> [↑](#footnote-ref-5)
6. <https://static.miki.it/pdf/thesis.pdf> [↑](#footnote-ref-6)
7. <https://blockchain.info/address/18iEz617DoDp8CNQUyyrjCcC7XCGDf5SVb> [↑](#footnote-ref-7)
8. <https://blockchain.info/address/1KP72fBmh3XBRfuJDMn53APaqM6iMRspCh> [↑](#footnote-ref-8)
9. <http://www.zdnet.com/article/cryptolockers-crimewave-a-trail-of-millions-in-laundered-bitcoin/> [↑](#footnote-ref-9)