

Spring 2016

Library Writers Reward Project

Saravana Kumar Gajendran
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_projects



Part of the [Databases and Information Systems Commons](#), [Information Security Commons](#), and the [Other Computer Sciences Commons](#)

Recommended Citation

Gajendran, Saravana Kumar, "Library Writers Reward Project" (2016). *Master's Projects*. 475.
DOI: <https://doi.org/10.31979/etd.cd4q-ce5y>
https://scholarworks.sjsu.edu/etd_projects/475

This Master's Project is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Projects by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

Library Writers Reward Project

A Project

Presented to

The Faculty of the Department of Computer Science San Jose State University

In Partial Fulfilment

of the Requirements for the Degree Master of Science

by

Saravana Kumar Gajendran

May 2016

© 2016
Saravana Kumar Gajendran
ALL RIGHTS RESERVED

The Designated Project Committee Approves the Project Titled

Library Writer Reward Project

by

Saravana Kumar Gajendran

APPROVED FOR THE DEPARTMENTS OF COMPUTER SCIENCE

SAN JOSE STATE UNIVERSITY

December 2015

Dr. Thomas Austin Department of Computer Science

Dr. Robert Chun Department of Computer Science

Mr. Praveen Veerath Tech Evangelist, Microsoft

Table of Contents

Abstract	5
Introduction	6
What is Bitcoin?	7
What Are Bitcoin Transactions?	8
What is Bitcoin Mining?	9
Integrate Bitcoin Miner Into Existing Libraries	10
What the Project Does	14
Related Work	15
Background	16
Bitcoin Mining	16
How Mining Works	17
Types of Mining	18
Hardware	19
Software	20
Why Mining?	21
Bitcoin are Money	22
Deflationary Currency [3]	23
Digital Wallet	23
Open Source	24
Implementation Details	27
SHA256 at Work	27
High Level Architecture	28
Mining Workflow	32
Web worker	34
Testing Results and Analysis	35
Test Mode (Without Integrating it to Any Library):	35
Test Mode (By Integrating it to jQuery Library)	38
Normal Mode With Integrating to jQuery	41
Analysis	42
Conclusion	45
Future Scope	46
References	47

Abstract

Open-source library development exploits the distributed intelligence of participants in Internet communities. Nowadays, contribution to the open-source community is fading [16] (Stackalytics, 2016) as there is not much recognition for library writers. They can start exploring ways to generate revenue as they actively contribute to the open-source community.

This project helps library writers to generate revenue in the form of bitcoins for their contribution. Our solution to generate revenue for library writers is to integrate bitcoin mining with existing JavaScript libraries, such as jQuery. More use of the library leads to more revenue for the library writers. It uses the visitor system's computational power to mine bitcoins.

As stated above, library writers can make sure that every visitor is contributing towards revenue generation. The amount of bitcoins that can be generated is directly related to the user's participation. When I tested this project for about a week on the single machine, it was able to make 0.021226 BTC, which in today's value is 8.96 U.S. dollars. The project also includes support for a digital wallet, which will keep track of a private key for currency balance.

Introduction

The motivation of the project is to generate revenue for library writers. The contribution of library writers to the open-source community is key to building more production model libraries like jQuery. According to W3Techs, jQuery is now used on half of all websites worldwide [19] (W3Techs, 2016). This project helps library writers to make revenue by generating Bitcoin. Bitcoin is a free software project originally introduced in a paper published in 2008 by Satoshi Nakamoto. In this project, the focus is on how library writers can benefit by using the end user's computational powers to generate Bitcoin for their contribution [21]. An alternate way that a library writer can make money is using end users' CPU cycles, which helps them to mine and generate bitcoins. The author needs to set up a bitcoin pool account and digital wallets so that whenever a visitor visits a webpage that uses the library writer's scripts, that will start sharing some small percentage of CPU power to generate bitcoins by mining. When compared to the amount of CPU utilization online that leads to overconsumption and mishandling of CPU, this option is safer because the author will have total control of how much CPU power can be used. This also allows for a stabilized system and no freezing or crashing of browsers. Library writers or website owners can set up what percentage of an end user's CPU cycle is utilized, and it can vary from 8% to 11%. When thousands of visitors are contributing a decent amount of CPU cycle, library writers can create large bitcoin pools, thereby increasing power to solve complex computational problems with limited time and power. Solving more complex problems increases the chances of generating bitcoins, which means more revenue for library writers.

What is Bitcoin?

- Bitcoins are digital coins that can be sent through the Internet.
- Bitcoin is the 1st decentralized digital currency, with “digital” meaning that it cannot be printed or physically made and “decentralized” meaning that it is not controlled by the government or any banking sectors
- Bitcoins are transferred directly from person to person on the Internet without going through a bank or any clearance, which means transfer is instant and fees are much lower. Which can be used in any country.
- Bitcoins are mined all over the Internet by anybody running a free application called Bitcoin Miner.
- All transactions are verified by nodes in the bitcoin network, which are published on the public ledger.
- A bitcoin is the unit of the account on this ledger.
- The smallest unit of bitcoins is a satoshi, which is 0.00000001, or one hundred millionth, of a bitcoin. In other words, one bitcoin equal 100,000,000 satoshi

What Are Bitcoin Transactions?

When a user sends a bitcoin, a data structure is created that is known as a bitcoin transaction by the user's wallet client, and it is then broadcasted to the network. All the bitcoin nodes on the network will relay and rebroadcast it. If the transactions are valid, it will become a block in the blockchain. The main focus of the bitcoin transaction is to transfer ownership of a bitcoin owned by one bitcoin address to another bitcoin address. According to bitcoins.org, a transaction will take place in 10-20mins with other transactions in a block in the blockchain (bitcoins.org, 2016). The following image is an example of a bitcoins transaction.

txid 90b18aa54288ec610d83ff1abe90f10d8ca87fb6411a72b2e56a169fdc9b0219

```
{
  "hash": "90b18aa54288ec610d83ff1abe90f10d8ca87fb6411a72b2e56a169fdc9b0219",
  "ver": 1,
  "vin_sz": 1,
  "vout_sz": 2,
  "lock_time": 0,
  "size": 226,
  "in": [
    {
      "prev_out": {
        "prev_out": {
          "hash": "18798f8795ded46c3086f48d5bdabe10e1755524b43912320b81ef547b2f939a",
          "n": 0
        }
      },
      "scriptSig": "3045022100c1efcad5cdcc0dcf7c2a79d9e1566523af9c7229c78ef71ee8b6300ab...[snip]"
    }
  ],
  "out": [
    {
      "value": "5.93100000",
      "scriptPubKey": "OP_DUP OP_HASH160 4b358739fc7984b8101278988beba0cc00867adc OP_EQUALVERIFY OP_CHECKSIG"
    },
    {
      "value": "1678.06900000",
      "scriptPubKey": "OP_DUP OP_HASH160 55368b388ccfe22a3f837c9eee93d053460db339 OP_EQUALVERIFY OP_CHECKSIG"
    }
  ]
}
```

tx format version - currently at version 1

in-counter - number of input amounts

out-counter - number of output amounts

tx lock_time - should be 0 or in the past for the tx to be valid and included in a block

size - of the transaction in bytes

The main components of standard transactions are the following:

- Transaction ID (the line beginning with “hash” in the above image)
- Descriptors and meta-data (the text to the right of the blue bracket in the above image)
- Inputs (the text following “in” in the above image)
- Outputs (the text following “out” in the above image)

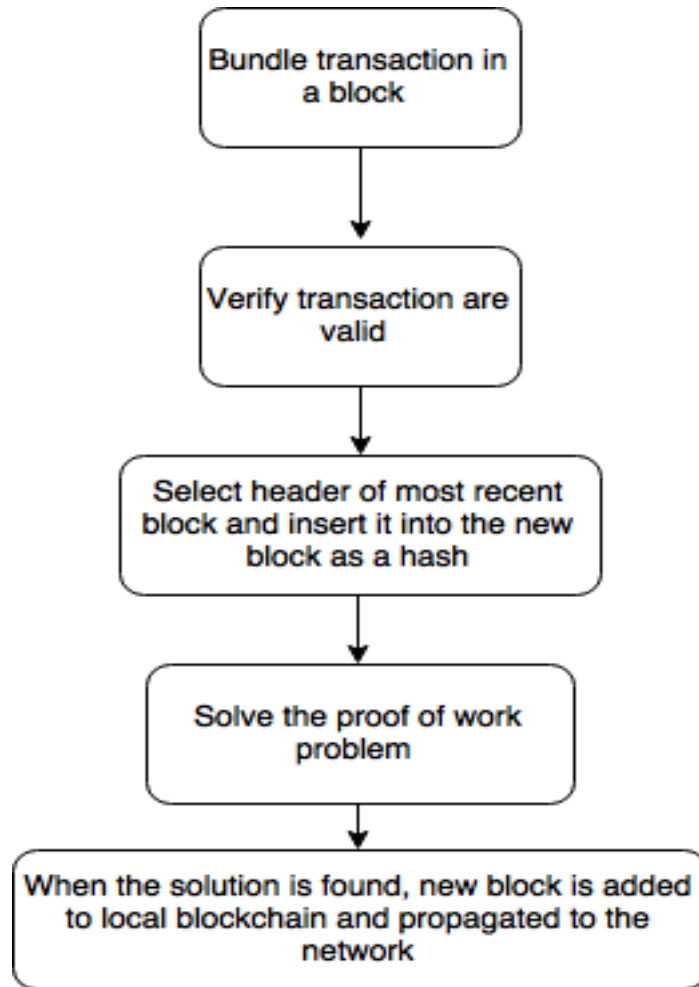
In bitcoin transactions, any bitcoin amount that one sends is always sent to an address, and any Bitcoin amount that is received is locked to the address of the receiver’s wallet.

What is Bitcoin Mining?

Bitcoin mining is a distributed computational process that serves two purposes:

- confirms transactions in a trustful manner
- creates new Bitcoin in each block.

Bitcoin mining in general is the process of adding transaction records to Bitcoin’s public ledger of previous transactions, known as a block chain. The purpose of block chain is to confirm to the rest of the network on what transactions have taken place.



Integrate Bitcoin Miner into Existing Libraries

Bitcoin Miner can be used with any other existing libraries, such as jQuery, in many ways. One way of integrating it with jQuery is by calling the mining API inside the library so that sourcing happens automatically and no dependence is needed when the API is directly called in the function. The mining API is tested with the well-known jQuery library and proved to be working without any conflicts while also making use of the jQuery DOM. The test we did provided the same result as when it is tested as a standalone script.

As the use of jQuery becomes more and more popular, we wanted to integrate jQuery scripting into the project. Below we discussed how basic configuration steps to get started on the path of integrating jQuery [9,10].

Background

The JavaScript library jQuery allows website designers and developers to use a relatively simple coding syntax when implementing animation, event handling, and Ajax interactions. The benefits of jQuery include its small file size, cross-browser compatibility, and strength in developing animations and CSS3 styling.

Requirements

jQuery library should be downloaded into project.

Folder Structure

Now that we have the files we need, we have to discuss where to place them. We have to place file correctly. Our directory structure for jQuery will look like below.

```
├── index.html
├── js
│   ├── miner.js
│   ├── jquery-ui
│   │   ├── accordion.js
│   │   ├── autocomplete.js
│   │   ├── button.js
│   │   ├── core.js
│   │   ├── datepicker.js
│   │   ├── dialog.js
│   │   └── ...
│   ├── jquery.js
│   └── require.js
```

Loading the Application

Now that we have the files in place, let's use them. Here are some of the contents of our miner index.html file.

```
1  <!doctype html>
2  <html lang="en">
3  <head>
4  ...
5  </head>
6  <body>
7
8  <script src="js/require.js" data-main="js/miner"></script>
9
10 </body>
11 </html>
```

the (dot)js script, helps to loads and executes the file specified in the main attribute.

We can use console log statement in app.js, you can verify that it loads appropriately

```
/* app.js */
console.log( "loaded" );
```

Some code snippet on how we used JQuery elements

```
function onSuccess(jsonresp) {  
    // console.log("I am hitting onSuccess");  
  
    var response = jsonresp.result;  
    var data = JSON.stringify(response);  
  
    $('#info').val(data);  
}
```

```
var total_time = ((new Date()).getTime() - job.start_date)  
var hashes_per_second = job.total_hashes / total_time;  
$('#total-hashes').val(job.total_hashes);  
$('#hashes-per-second').val(hashes_per_second);
```

New Bitcoin are created every 10 minutes (“Protocol Rules,” 2016) [15] with each worth close to \$300 according to en.bitcoin.it. Thousands of end users’ computers will be contributing for mining to unlock a batch and continue to a new batch every time. When the counter begins, the end user’s system, which runs the bitcoins mining software, tries to use a lot of computational power to eliminate the work that needs to be done by guessing the nonce, which will be discussed more later in this document. This script is not going to exhaust the client’s CPU cycles to the fullest; instead, it will use only a certain percentage of the CPU cycle, and when it runs on a tablet or mobile device, it will automatically end in 10 minutes to stop draining the battery. More information on integrating the miner into existing libraries will be discussed later in the document.

What the Project Does

Bitcoin are a limited resource so that batch of coins can be mined will be dropped into half every 4 year, because total number of Bitcoin can be mined is about 21 million and it all ends there (“Bitcoin,” 2016). Miners will compete with each other to approve a transaction by storing it in a ledger and sealing the work/transaction. By doing this, miners who complete this work first will be rewarded with a fair amount of Bitcoin. Miners can win this situation by having powerful CPU cycles at their disposal. This project helps to get the work from the pool server and use the visitor’s CPU power to solve the complex computation to access the block. Whoever finds the valid block will be rewarded. What exactly is happening is the systems are trying to verify the transaction and seal it as soon as possible, so computational power is the key here since it actually takes time for miners to find something quickly to win this situation and get rewards. The computational problem is to solve cryptographic problems, which involves the computer canceling the millions of combinations or guessing the solution with different nonce numbers at a very fast rate until it finds the answer to the next set of blocks. This process will continue for as long as some kind of transaction is happening in the bitcoin network. There are many stages in how this project is carried out, from worker connecting to servers for a job to process it and post back to server with assigned job id. When it gets a job, the worker will start with scaring the hash and performing the computation tasks, then posting it back to the server when a valid block is found and repeating the process [22].

Miners can be described as people who maintain and centralize the public ledgers. They maintain and build large public ledgers, which contain the records of every bitcoin

transfer that has happened over a given period of time. What exactly can be found in ledgers is transaction history dating back to the beginning of these transfers, so every time someone wants to make a transaction, that transaction needs to be verified by this miner. If more than five miners verify the transaction, then it can be called a valid transfer. Miners need to verify the transaction so that someone cannot spend the money they already sent to someone else, which is called double spending. There are many factors to avoiding the double spending issue. Once the miners store the transaction into the ledger, they add many layers of computational work to keep it away from hackers, so the chances of accessing the ledger are very remote. Miners get rewards for keeping the ledger safe. More discussion on implementation details can be found in part four of this document, which is split in various stages.

Related Work

There has been considerable research being done with bitcoin mining. Alex Heid, gave a detailed view on cryptocurrency technologies and protocols [27] required to familiarize with principles behind open source economic ecosystem. David R. Sterry also worked on running multiple miners [2]. Meni Rosenfeld suggested various scoring system used to calculate [26] rewards for participants in Bitcoin pooled mining and also explained the problems each were designed to solve and analyze respective advantage and disadvantage [28]. Yoad Lewenberg and Yoram also examined mining pool in Bitcoin, using a game theoretic model for reward sharing [29]. They showed that the Bitcoin protocol results in a pool's reward being a non-linear function of the pool's computational power

Background

Bitcoin Mining

Bitcoin mining consists of three key components:

- a set of rules that defines how mining networks should be and how they operate,
- scripts or software that help to define the set of rules, and
- pool users with a powerful system to help and contribute to the mining network.

In simple words, miners help to verify transactions and prevent double spending, for which they will be fairly rewarded with Bitcoin. They also prevent hackers from accessing past transactions by piling large amounts of processing power so that it is impossible to get into the transaction logs. Miners find a valid block by building a recent transaction and trying to calculate information about proposed blocks. Then, the block is added with a random number called nonce[11]. Miners will try to see if they can win the current difficulty level, or else they will choose a new nonce to try a new hash to calculate and test by brute force search. Brute force is used to guess and find valid block. As this whole process is very unpredictable, there is no backtracking, so the only way to find a valid block is to increase computational speed and get help from others. Therefore, the more computational power a miner has, the more likely the miner is to win a mining pool. When a miner finds a valid block, it will be immediately broadcasted to bitcoins mining network and verified with other miners in the network. The difficulty of finding the valid block is adjusted according

to the time it takes to find 2,016 valid blocks at a rate of finding one valid block per ten minutes.

How Mining Works

Mining is all about getting a good hash rate. All miners are trying to get a valid block with the amount of computational power they have. If that is not sufficient, they try to form a pool with the CPU power they have. Once mining is started, there is not much to be taken from the miner side. Only when a piece of software gets a glitch do miners need to reset the hash and repeat the task on the other side. Any hardware failure needs to be taken care of, so when the process is started, it will run smoothly without any hardware or software glitch. Nowadays, if any software glitch happens, it was programmed to restart itself so that not much attention is required from miner taking care of it. A few other factors have to be taken into account to reach max hash rates, such as maintaining the temperature of hardware and electricity. Whenever a valid block is found, the winner will be rewarded with Bitcoin in addition to the small transaction fee. If a miner does not want to invest a lot of money in hardware components, since they are updated every year anyway, they form a pool of miners to help with networking that requires additional routers and other networking components [1]. Each user will be assigned a username and password according to their contribution to winning the valid block in the network, and they will be rewarded and shared accordingly. One factor that needs to be carefully looked at it is the amount of heat generated from the hardware components. Proper cooling systems should be installed with these hardware components to eliminate fire hazards. Mining software is open source and developed by a group of volunteers. A key point in the success of mining is how much computational power a miner contributes to eliminate wrong guesses, and

guessing the right nonce will decide who wins the golden ticket. Mining is competitive because all miners are trying to solve the same block simultaneously while miners who find the solution to the hash function are the only ones who are rewarded, but all other miners who contribute computational power are very much needed too even though they might not get rewarded.

Two key contributions from miners are preventing people from double spending and distributing new Bitcoin in a fair way for winners who solved the hash problem. Therefore, the miners who spend a lot of electricity and computational CPU cycles are the ones who are rewarded with Bitcoin. As this business is very competitive, miners must pool the available resources.

Types of Mining

CPU mining

Computer processor speeds are measured in clock speed, and nowadays it is very common that most systems run in high gigabytes of clock speed. By the time of manufacture, the speed of the processor will be tested. CPU mining performance is purely based off the clock speed of the CPU [2].

Pool Mining

Nowadays, it is very unfortunate that there can be no standalone miner who finds a winning block because of the difficulty level. It could take many years for a solo contributor to find a single block considering today's difficulty level. However, a solo contributor with several machines running at very high hashes per second has a better chance of finding a winning block and proceeding with the next one, but if a solo contributor does not have much horsepower at his or her disposal, the best way to earn decent payouts today is to join

the mining pool and contribute CPU cycle to the pool to increase the chances of winning a valid block and make better payouts. That is because winning a block as an individual contributor is very infrequent, so mining pools make a lot of sense because if one user finds a valid block, the reward will be shared with the rest of the miners depending on their computational contribution to the mining pool. The pool operators also take a small percentage of profit for their contributions. Many pools are available today, and miners can switch between pools.

Key Contribution for Solo System

Each machine in the pool is not exactly trying to solve the block itself. Instead, the machines are reshaping the computational load and helping to collectively eliminate wrong guesses and make guesses with different nonce. It guesses the key by trying with different nonce, so trying to solve the problem in this way is a lot easier than trying to solve the block without it. When the process is finished, the pool pays everyone who contributed to eliminating wrong guesses or making a guess that made the pool win the golden ticket.

Hardware

Miners start with hardware since that needs to be set up first because computer hardware is what decides in what speed and how efficiently the mining performs. Software is as important as hardware because software is important for executing mining logic. When it comes to systems [17], some are faster than others depending on amount of RAM the process uses and the type of hard drive it has. However, some particular types of processor do better jobs than others; that is because mining is about brute force, so processors that do that task better work best for bitcoins mining even though the CPU does the job. GPU will have upper hand when it comes to mining because of GPU's internal

arrangements. The difference can be working hundreds of time more efficiently than CPU. Miners have quite a few options that they can use to mine. The market has FPGAs, which are very expensive but use electricity efficiently. Now, miners are moving towards FPGA since it will help them to save electricity, which was taking a margin of their profit.

Software

Bitcoin mining software is available for most platforms, such as Mac, Linux, and Windows, and all this software is open source and free to use. The following are some of the best mining software available for these platforms:

- MinePeon – Open source
- EasyMiner – GUI based for Windows and Linux
- BFGMiner – A modular CPU, GPU, ASIC miner
- CGMiner – Multi-threaded, multi-pool GPU miner

When a hardware and software setup is done, miners can start mining the Bitcoin. Usually, speed is determined in mega hashes or giga hashes, but nowadays, pooled users can easily use tera hash speed. By having this amount of computational speed, miners try to use as many hashes as possible using the best combination of hardware and software [25]. Software is a key component since it maintains the stability of the overall system. That is important because there should not be any software glitch that will lose progress and reset the mining process, so the software aspect should be taken care of properly.

Bitcoin Wallet Software

This software is used keep Bitcoin safe and secure. For a bitcoin wallet, the strong recommendation is to hold your private keys in contrast to hosted wallets, such as Circle.

Armory is a highly secured desktop bitcoin wallet, and Copay is an easy-to-use mobile bitcoin wallet.

Why Mining?

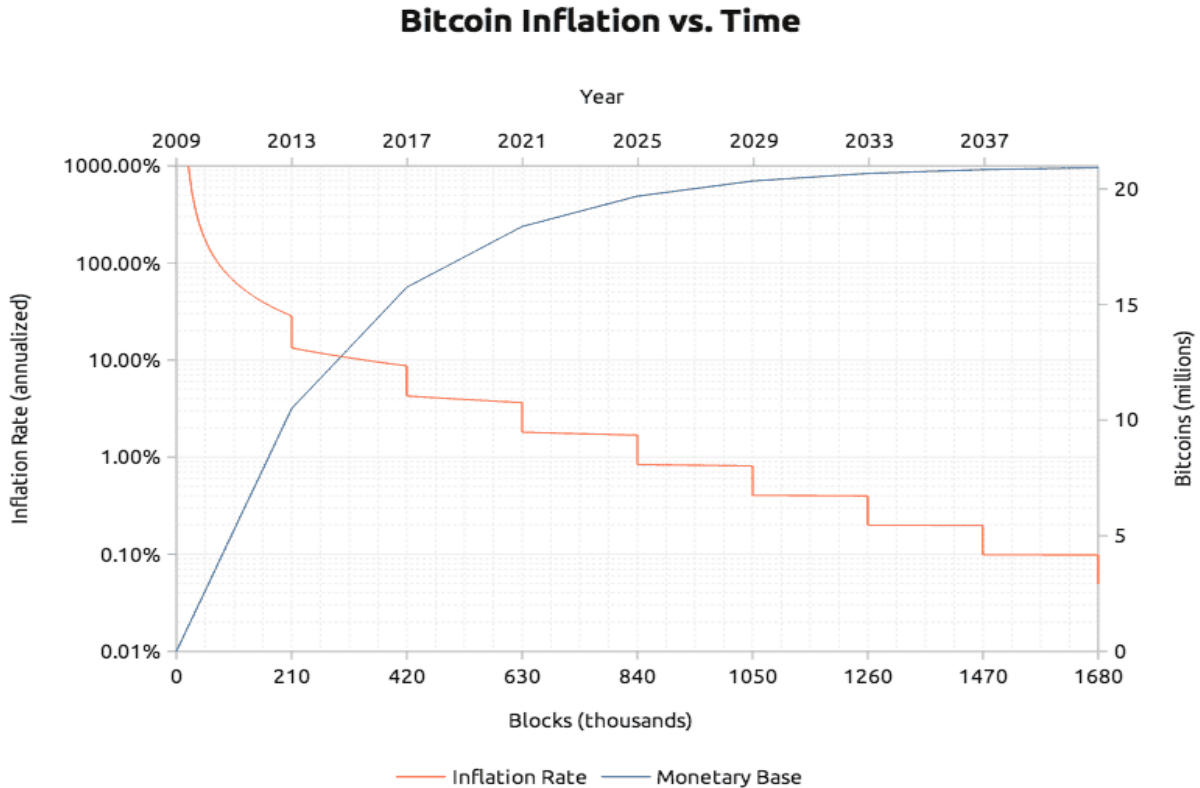
The motivation of this project is to start mining by a natural extension of the people already contributing to the open-source community. This project is very similar to grid computing that has reached its limits since it provides an opportunity to form a pool of users to solve any complex computational problem. In other words, distributed computing project users share the computational horses they have to support the open-source community to increase the speed of the overall processing power for payment methods and to create currency that is not handled by a particular company or third party. Bitcoin mining is the latest technology that includes cryptography, peer-to-peer networking, and distributed computing. Even though the process can be fun, the majority of miners do it to make profit. As they have invested many thousands of dollars in building up their whole system, they can make steady profits efficiently as they make big purchases of hardware for this system. For miners, mining is like a steady business since it does not require them to sell or exchange anything. Also, miners can earn Bitcoin anonymously since they can contribute to solve a block and be rewarded completely anonymously. One of the best uses of bitcoins is as international commerce since transfer can be immediate and no centralized authorities are required.

Bitcoin are Money

Bitcoin are money and can be used as such. They can be transferred to someone else using that person's public address, or one can just keep them safe over a period of time so that their value goes up [18]. Taking a look at bitcoins value charts over a period of time show that their value went all the way to 700 dollars but now is close to 300 dollars. Nowadays, many third -party consumers accept Bitcoin, and many bitcoins exchanges are available around the world, so they can be exchanged not only with U.S. dollars but also with many other currencies, such as yen, euros and so on. However, to do so, one needs an exchange account. With that, one can buy online products, and these exchange services make sure that money is transferred properly before a product is shipped to clients. Bitcoin transactions cannot be reversed once transferred because they are one way, so even a big credit card company or big banks cannot get them back because of the complex mathematics of Bitcoin. This leads Bitcoin to be incapable for financial services that cannot allow payments to be disrupted.

Bitcoin is a deflationary currency because more cannot be added, but still people are motivated to move them since they will be rewarded with transaction fees, so they keep mining to build distributed trust so that the transaction fee may go up in the future.

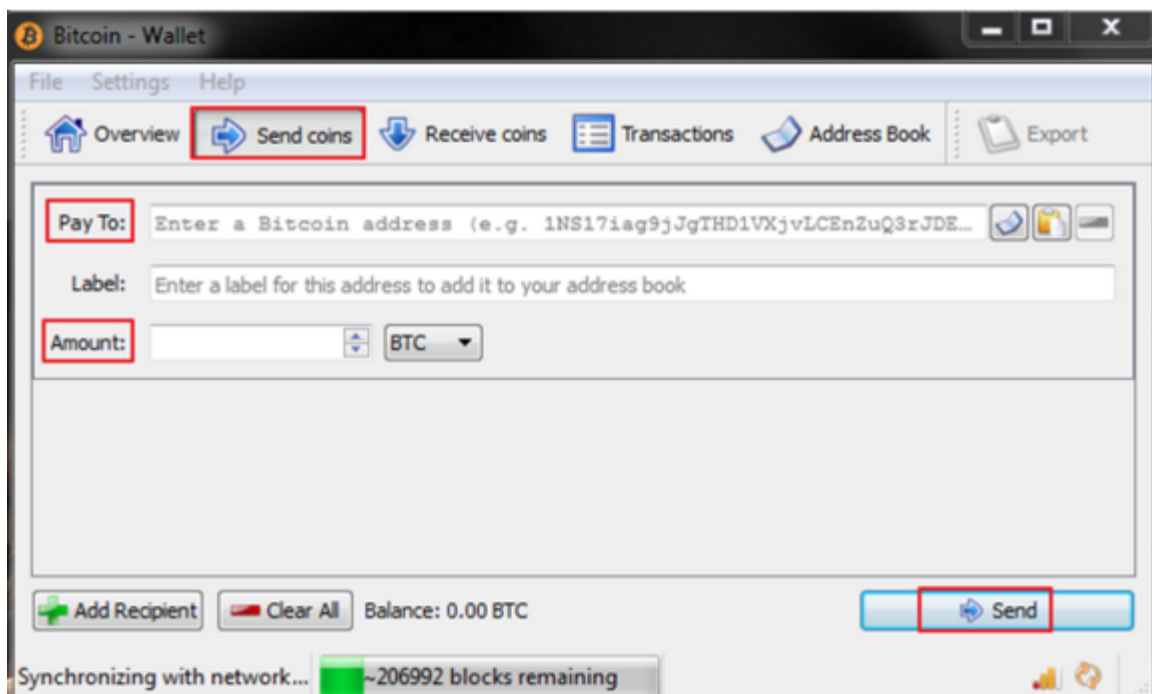
Deflationary Currency [3]



Digital Wallet

Regardless of whether a miner is either an individual contributor or part of a pool, once the miner gets his or her share over a period of time for his contribution, he or she needs a place to store it. One effective way to store it is using a digital wallet. Many bitcoin clients help miners to create a bitcoins digital wallet, in which miners keep the wallet safe with a private key, a combination of characters and numeric that help miners to send bitcoins to others in the network. Every private key has a public key and bitcoins address. When transactions happen, that private key accesses the amount of bitcoins that is associated with a public address that it can be sent to, and then this transfer is broadcasted to the bitcoin network. Taking a closer look at the transfer shows that Bitcoin are not sent

anywhere, just re-assigned to addresses where a user wants to send them. A private key is required so that those who have access to a private key literally own it. Once a user receives Bitcoin, this private key must be kept in case of hardware failure or a similar issue. One effective approach to keeping it safe is to store it in a USB drive or DVD, and another way is to keep it away from online attacks or hackers. If a user puts an entire system online, then hackers can easily access it. If a user thinks someone can access it, then he or she can encrypt it to store in some place safe so that even if it is stored offline, no one can access it[12].



Open Source

Open-source software is software that is provided for free with access to the source code so that one can see the insides of how it works. In the 1960s, companies such as IBM and Honeycomb sold hardware and provided software for free. However, after a few years,

people realized that software is more important since it is the software that gives the hardware more a competitive advantage over somebody else's hardware[4]. Therefore, software began being viewed as a valuable intellectual property, and to protect it, people came up with software licenses, patents, nondisclosure employment contracts, etc. However, software developers felt that if the source code was freely redistributed, it could lead to more and more development, thereby resulting in social betterment.

Open-source software works in the following way: a software developer writes the source code, compiles it, and makes it available to other users/developers through the World Wide Web. These users/developers can run the program for any purpose, modify the program to suit their purpose, redistribute copies for free, and distribute modified versions with changes and improvements. In this way, users/developers have access to a vastly improved source code [4].

According to W3Techs, library writers the contribution to open source community is fading, as there is not much recognition for their contribution. To maintain the contribution to open source community actively they should be able to generate some form of revenue. This project will help the library writer to generate the revenue in the form of bitcoins. More utilization of their library leads to more amount of bitcoins they can generate.

APPLICATION SOFTWARE- 7-zip, Eclipse, Chromium

OPERATING SYSTEM- Android, Linux, FreeBSD

PROGRAMMING LANGUAGES- Python, Perl, PHP

Implementation Details

This chapter discusses more about how the mining getting the work form pool server and how it computes the hashing. It includes both design and implementation of the approach. The discussion is mainly about what are the process involved in getting the job from the pool server and how is it processed and feed into miner finally how is submitted back to pool server for rewards.

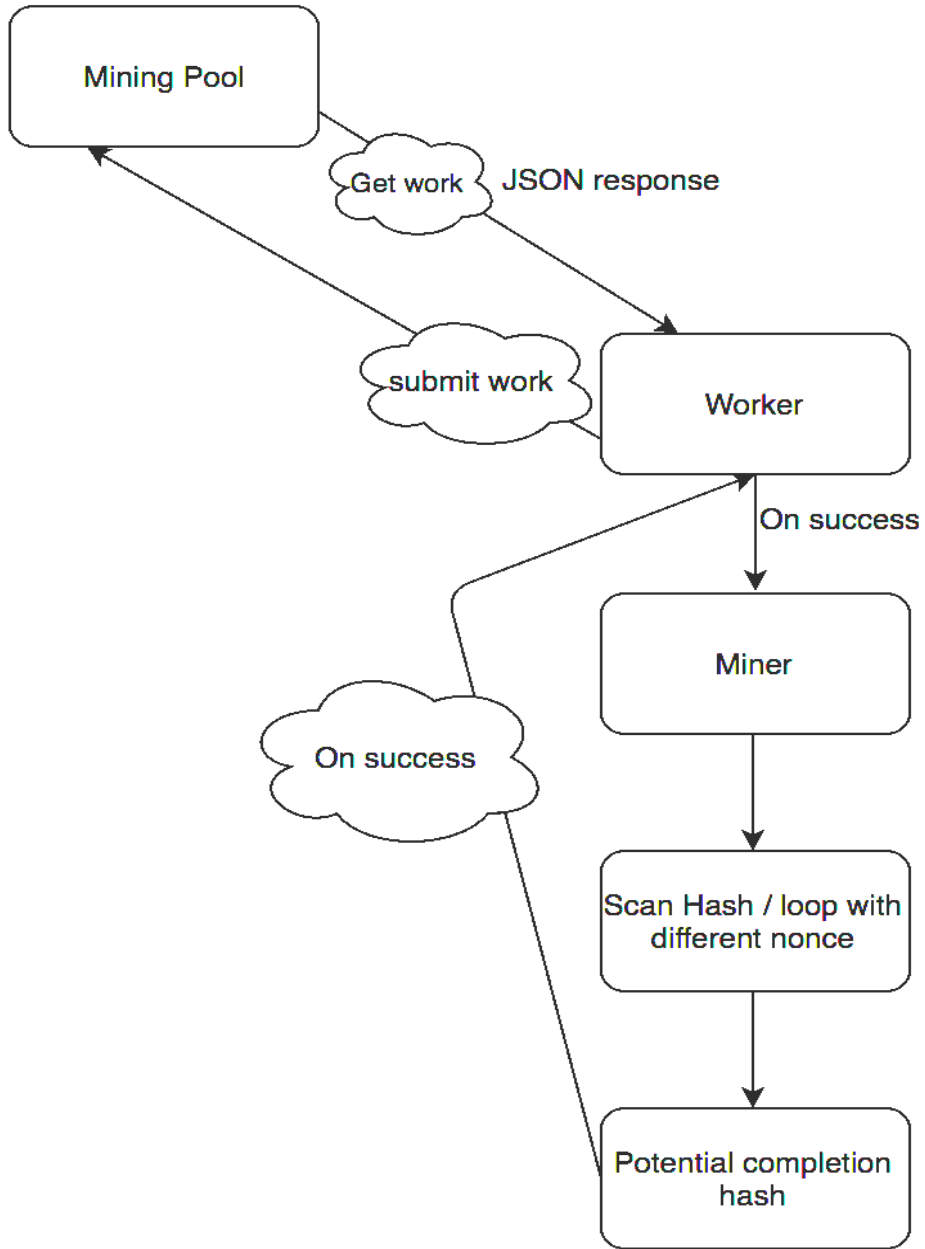
The cryptography hash function is an easy-to-compute hash value for any given message, but it is impossible to generate a original message from a hash. Moreover, it is impossible to modify the original message without changing the hash.

SHA256 at Work

I implemented SHA256 in JavaScript by following a pseudocode from the Wikipedia page “SHA-2” (“SHA-2,” 2016). Hash functions take a message string of any size, but when the hash function applies the mathematical transformation to this input and produces a single output, commonly known as digesting, the message can be very small in size or very large in size, but the size must always be fixed in length—for example, no matter what the input string is, the output will be a fixed 256bits in length. Another point about this hash function is deterministic function [5] MD5 also another way [20]. That is, output always gives some input, but minors changing the input will drastically change the output in a big way, so a given input value will always produce the exact same output. This hash function is also used for security purposes, privacy or confidentiality, and other areas of interest. The hash function is computationally efficient, which means it should not take a lot of time to compute the output from a given input if given a message to which the

mathematically transformation can be applied to digest. The pseudocode of my implementation of the SHA256 algorithm from the Wikipedia article is described in more detail below for better understanding.

High Level Architecture



Communication

Pool servers know very much about when a client needs a mining job and what the client needs for it. However, HTTP design working in a client's browser will request specific content, but when it comes to pooled mining, the server knows much better than the client what has to be done and can control the communication in a more efficient way.

Job Rolling

When a miner receives a job from a server, he or she will have access to modify it only in time and nonce. Nonce is nothing but a random number that is a 32-bit integer, and the job will also have ntime, which is timestamp that includes the current time, but if a block is created from large modified ntime, that will be rejected by the bitcoin network. By strictly following get work specifications, one job can be done in a minute if a miner has 4.2Ghash/s.

Pooling Server

Get work has a major role when it comes to standalone miners. Bitcoin mining started with standalone machines. When pool mining came into business, people started to decide which pools to work with or which would have short pooling intervals so that they would not overload servers and networks, but sometimes this might lead into much higher ratios of rejected shares. Long pooling directly connects with pool servers, which results in many issues on the server side in load balancing, conducting HTTP sessions, and reconnecting with clients, so the solution to such problems is that driving instructions should be done by the server and not by the thousands of miners.

JSON-RPC

The script uses the JSON-RPC client to get the work from the pool server so that a request for response and notification can be easily taken care of [6]. JSON-RPC is a stateless and lightweight remote procedure that is very flexible for working with sockets, HTTP, and message passing environments. Params in JSON are structured values that hold parameters to be used with methods.

Requested Objects

JSON-RPC -> a string for JSON-RPC protocol used

method -> a string for a method to be involved

param -> well-defined or structured value to be used when calling a method

Id -> identifiers given by client that can be a string numeric or null value

Getting Work

The script starts by sending a get request to get the work from the pool server. A JSON-RPC request will look like this:

```
{
  "id": 1,
  "method": "getwork",
  "params": []
}
```

The pool server will then respond with “work.” The response from the server for an initial Getwork request looks like this:

```

{
  "result": {
    "midstate": "...",
    "data": "...",
    "hash1": "...",
    "target": "...",
  },
  "error": null,
  "id": 1
}

```

- midstate in the above image is 64 characters long
- data in the above image is 256 characters long
- hash1 in the above image is 128 characters long
- target in the above image is 64 characters long

Target

The mining pool will be aware of the real target. Normally, this piece of info will not be sent to the miner, but instead the actual target will be overwritten by being specially modified to the target value and sent to the miner. Most of the mining pool server sets the target to its maximum value in hex to

`0x00000000ffff000`, which

allows miners to generate hashes with the lowest possible difficulty.

Miner

The miner will start scanning the hashes by assigning it to a local variable and declaring the nonce variable and the maximum value for the nonce. Looping will happen after the nonce is declared and runs by setting the fourth chunk in the 256-bit value and resetting the sha256 algorithm. The initial hash is computed, and the hash1 from the server

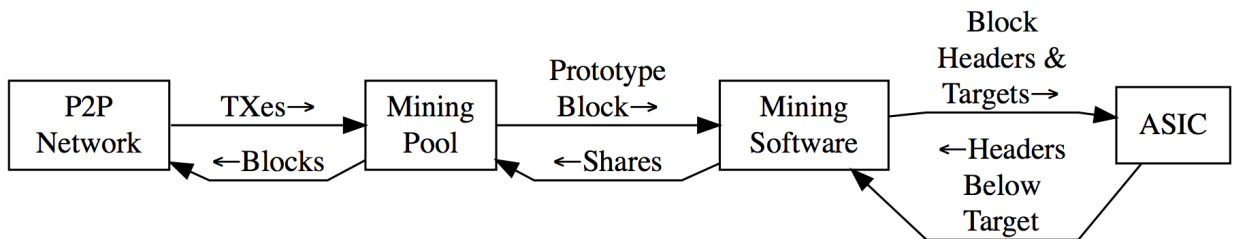
loops eight times to update the SHA256 and compute the final hash. The hash is guessed by changing the random nonce number if the hash value is less than the target at which the hash will be sent to the server to claim the rewards.

Submitting Work

When the client machine finishes the work and finds a potentially completed hash, that hash will be sent back to the server, and the request will be very similar to initial the initial getwork. This request will be a single parameter, like shown below:

```
{
  "id": 1,
  "method": "getwork",
  "params": ["a791feeb00ae7fce4c7147b7bcbb525616a93552e7e96c45b48cc56aed3311c1"]
}
```

Mining Workflow [7]



Nonce Range

A nonce is a random number that allows the mining pool client to try to guess how much work a miner can do depending on his or her hash rate[13]. The mining pool will break a single Getwork response into multiple chunks and send it to multiple different miners in the same network with the addition of x nonce range header, which helps to inform others in the mining network on how many hashes to generate before stopping the

midstream and going for new work. This benefits the pool mining in many ways. The following is an example of a JSON response with a nonce range:

```
{
  "result": {
    "midstate": "...",
    "data": "...",
    "hash1": "...",
    "target": "...",
    "noncerange": "00000001ffffffff" // representing nonces 0 through 536,870,911
  },
  "error": null,
  "id": 1
}
```

Work Completion

When the work is completed and sent back to the server, the server response after that will be either “true” or “false,” which will indicate whether the share was a valid block. The mining pool will usually send “true” to indicate to the miner that his or her contribution was accepted by the pool and rewards were given.

The following is sample data to the server:

```
{
  "id": 1,
  "method": "getwork",
  "params": ["a791feeb00ae7fce4c7147b7bcbb525616a93552e7e96c45b48cc56aed3311c1"]
}
```

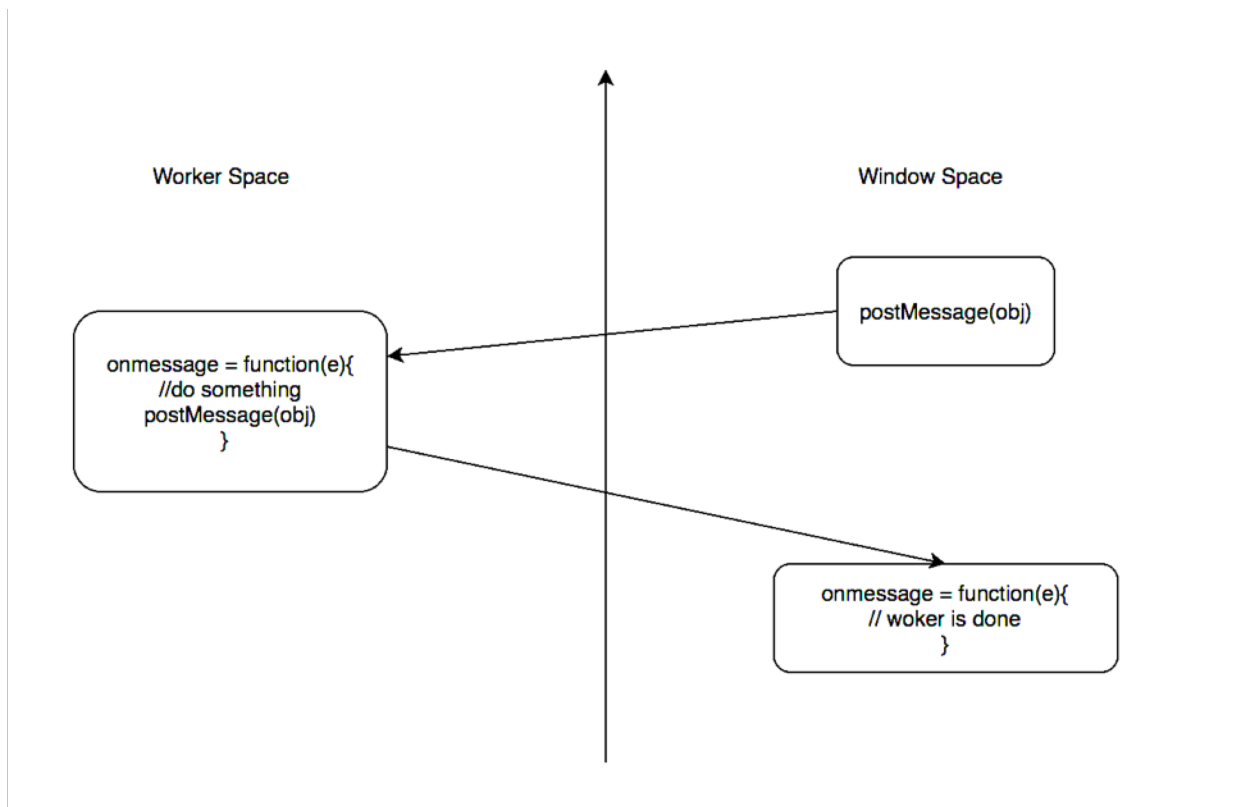
The following is an example response from the server:

```
{
  "version": "1.1",
  "id": 1,
  "error": null,
  "result": false
}
```

Web worker

Web worker provides mechanism on separate script in background for our project. When we are doing complex calculation without disturbing UI, it almost like assigning workout to worker communication on page and worker are happens through `postMessage` method and `onMessage ()` functions.

High level architecture



Testing Results and Analysis

This chapter discusses the experimental setup and analysis that are done in this project and gives recommendations for authors to join which mining pool so that they can generate more Bitcoin. This project is tested to make sure that the functionality and performance of the project are sound. Experiments involve the following methods:

- Test mode with no integration to the jQuery library
- Test mode with the jQuery library
- Normal mode with the jQuery library

Test Mode (Without Integrating it to Any Library):

There is no input from the server, but the input is hardcoded into the project to check the computational functionality.

Machine Specifications

Model	MacBook Pro
CPU	2.5 GHz Intel Core i7
RAM	16 GB
System	64-bit OS
OS	OS X El Capitan
Java Compiler	Java 6

Test Mode (By Integrating it to jQuery Library)

Integrating the jQuery library by calling the project API inside jQuery and testing it in test mode by hardcoding the data in the project also gave the same result when the same machine was used to test it.

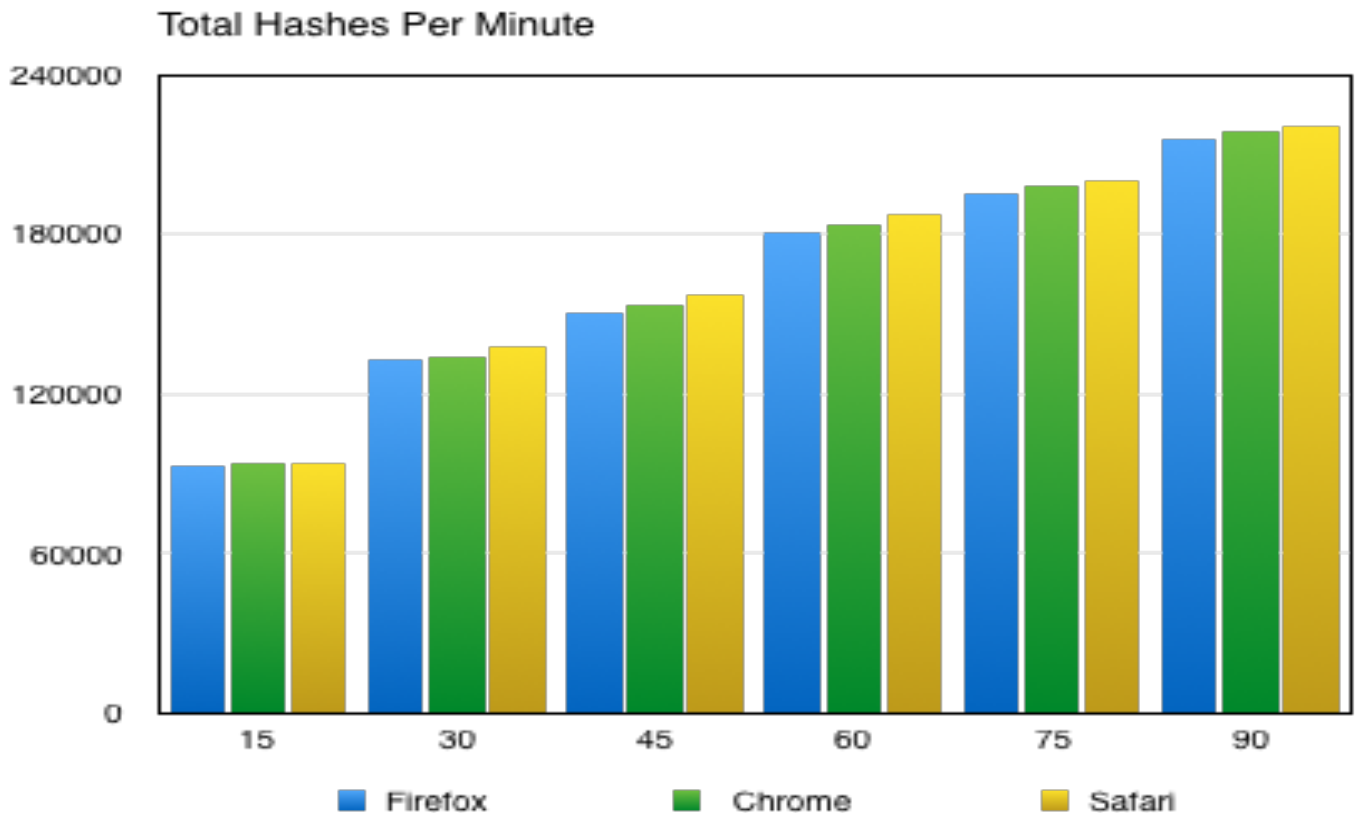
Machine Specifications

Model	MacBook Pro
CPU	2.4 GHz Intel Core i7
RAM	16 GB
System	64-bit OS
OS	OS X El Capitan
Java Compiler	Java 6
Browser	Safari, Firefox, Chrome

Testing were done by running the project simultaneously on multiple tabs in Firefox and pasted the result below. As we increase the threads we can clearly see hashes per second and total hashes are increasing constantly, which means more user participation leads to more number of hashes per seconds. The hash rate is the primary measure of a Bitcoin miner's performance. The Hash/s is also used to calculations of Bitcoin network's overall hash rate.

Threads	Total Hashes per minute			Application Response
	Firefox	Chrome	Safari	
15	93044	93890	94380	Good
30	132632	133837	137977	Good
45	150686	153762	156900	Slight lag
60	180497	183246	187560	Slight lag
75	194827	197794	200400	Moderate lag
90	215967	218812	220800	Moderate lag

Below chart shows, as we increase the number of threads, total hashes are increasing, though with diminishing returns on different browser.



CS298 Project

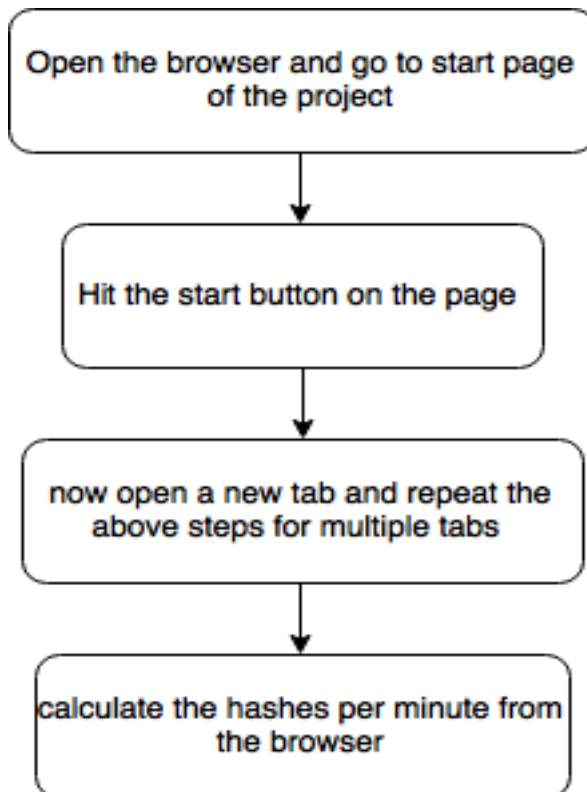
Testmode

Info: {"midstate":"eae773ad01907880889ac5629af0c35438376e8c4ae77906301

Total Hashes: 110929

Hash/s: 1164.2911121373693

Results: 0000000109a78d37203813d08b45854d51470fcdb588d6dfabbe946e92ad2



The test is repeated for 6 times by starting with 15 threads to 90 threads. each time thread count is increased by 15 and continued till 90 threads Test timing is depending upon number of threads we are running at a time

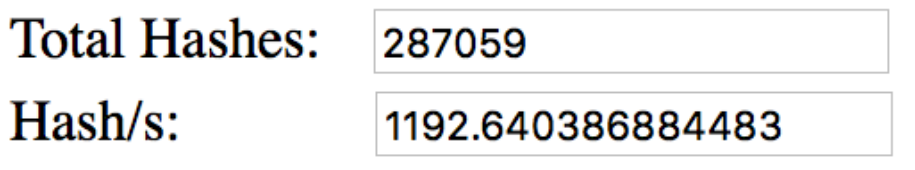
Normal Mode with Integrating to jQuery

In normal mode, the project is able to compute 1192 hashes per second, so when thousands of users are using the author's library and contributing a decent amount of computational power, we can expect roughly 0.00008567 BTC per day as per the hash speed that the project is able to achieve, which is not very high, but at the same time that amount of bitcoins generated is directly proportional to the amount of contribution users make towards it, and considering the exchange rate of Bitcoin is about 444.97 (CoinDesk, 2016) [14], authors can make a decent revenue over a period of time.

Machine Specifications

Model	Dell T5600
CPU	2.2 GHz Intel Core i7
RAM	32 GB 1600 MHz DDR3
System	64-bit OS
OS	Windows 10
Java Compiler	Java 6
Browser	Firefox

The following image shows hashes per second in normal mode in one machine:

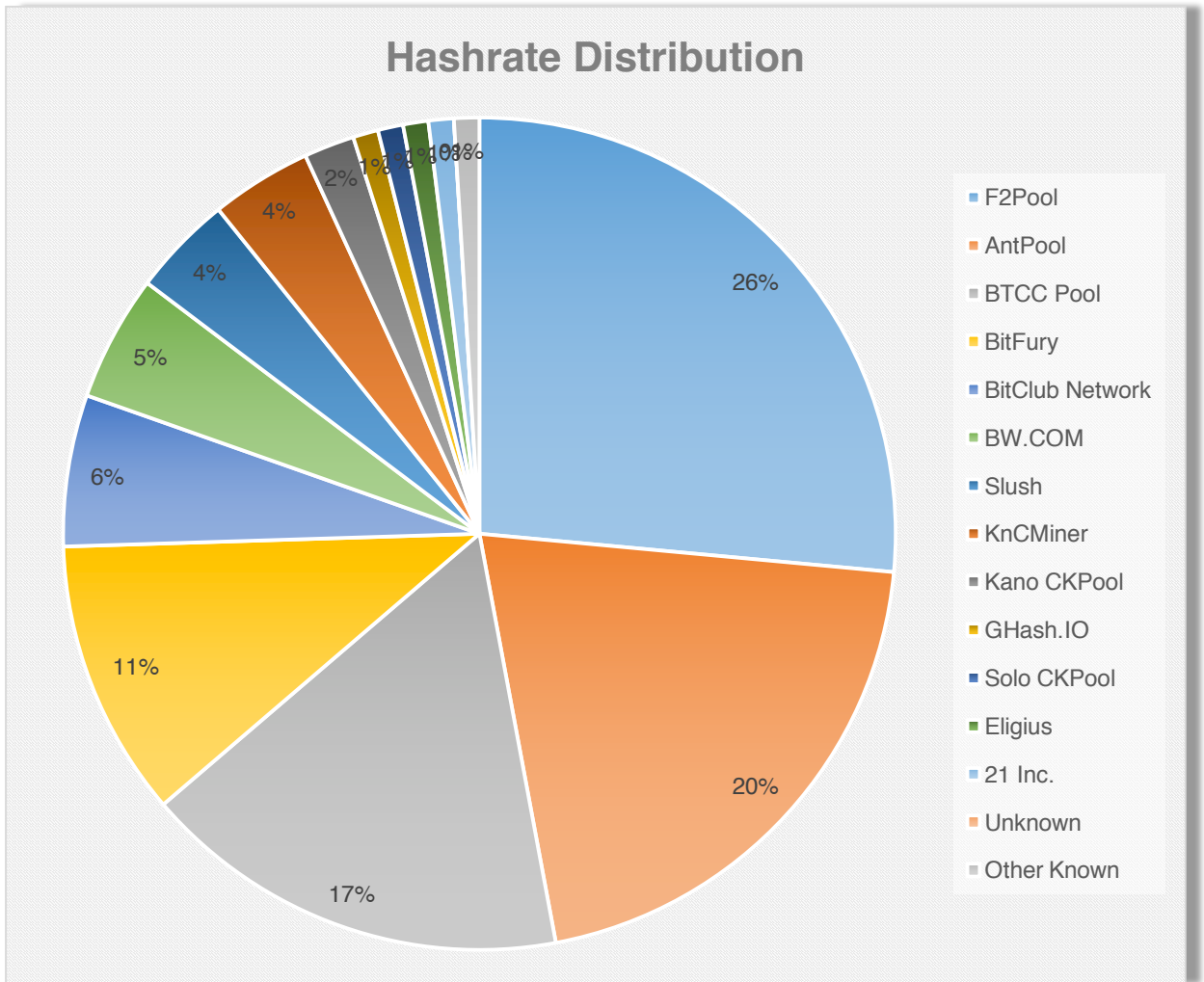


The image shows a screenshot of a software interface with two rows of data. The first row displays 'Total Hashes:' followed by a text box containing the value '287059'. The second row displays 'Hash/s:' followed by a text box containing the value '1192.640386884483'. The text boxes have a light gray border and a subtle drop shadow.

Total Hashes:	287059
Hash/s:	1192.640386884483

Analysis

Let us see what pools are currently available to join and what market share they have.



Above chart showing each pool and its size compared to one another[8].

Ant pool is clearly the largest pool, and for mining Bitcoin there are advantages and disadvantages to joining such a large pool. We will look at it more later. The biggest issue to consider with a pool of this size is the competition within it since the pool is so large that it can affect a miner's luck or the likelihood of a miner sending incorrect information

when receiving a reward. This may be just speculation as only size is related to accepted owners versus stay owners, which do not count. Pool size will also determine speed, which determines how quickly blocks are found and solved within Bitcoin.

F2pool is not a public pool that anyone can join. Instead, its exchange is like GHash.io. In other words, it is like a stock exchange instead of stocks. We buying mining power we use to join a pool. The benefit of this method is what that power can readily mine in the bitcoin market. The unit of measuring mining power is giga hash per second, which is 1000 mega hashes per second. These provide the power so that miners can sit back and collect the profit. This may sound attractive, but anything that it is often said that the return on investment does not match up with what return would be if a miner owned it and as this is not available for free to join this pool is out [23].

Eligius is a free, no-registration pool that anyone can join. Miners navigate to the URL and can connect and start mining immediately. Typically, they need to register for an account and create workers to manage mining and statistics. The developer of this pool is doing a great job of maintaining the pool's profitability.

Bitminer is one of the largest pools, and registration was easy. It even provided its own mining client for miners to use, which is big when compared to other pools, and this is a huge advantage for miners who are just starting out and do not want to learn how mining works. Their clients are also coded in java. I did not experiment much with this pool, but their setup and port is very simple.

Slush comes in fifth place with 4% in the total mining market. This is the pool I have been involved in most and have had great success in mining bitcoin with. I have not found a pool that pays more frequently and as much as this pool, but I cannot explain why

because experience with mining on different pools differs from user to user, so Slush may not work well with someone. However, it works for me, and I believe that joining smaller pools will lead to more success than joining large mining pools since the opportunity is greatest for an individual when it is basic and no strings are attached. For that reason, I am putting Slush as my number-one recommendation and Bitminer as my second preference.

Lots of people wonder about other or unknown pools. This is not any specific pool—it is people who mine on their own or start their private pools so that we other miners cannot join unless they are invited to. These private pools make up 1% of the market, so if any pool gets large, there is a risk that they will run away with the market. Some say this would happen if any pool just reached 50% market share, but there are always people who do not want to be part of any pool and mine on their own. More pools create more complications, and while having many smaller individual pools should benefit miners, when these are just starting up, it will take some time to build their user base, so they move more slowly and may find blocks too slowly.

Let us take Eobot as an example. This pool only has 1% in size because they charge a 10% pool fee, so people are not too interested in joining this pool right now.

Conclusion

In this project, we help library writers to generate revenue in the form of bitcoins for their contribution to the open- source community. Our solution to achieve this goal is started with CPU mining with few little many features, but our work will continue to make them generate more revenue by using computer GPU's in the future, which will provide have chances to increase the revenue multiple times, though (but it is subjected to availability in end users' machines).

We have explored several methods used by pooling servers to distribute the work among many miners in their networks and have also explored how pools distribute the rewards. We have tested this project in functionality in various modes for the integrity of this project, and where most of the outputs almost shows similar results.

Future Scope

In the future, we are going to continue our support for library writers to contribute to the open-source community by innovating various other ways to achieve more hash rates by using GPUs. Even though more use of the library leads to more revenue for the library writers, more users with GPU will help to increase revenue drastically if we can make use of GPUs [24].

As GPUs are made for high mathematical lifting and it will calculate all the complex polygons required for high-end gaming, which makes them extremely good at the SHA-256 hashing mathematical necessary to solve Bitcoin transaction blocks. One more advantage of using GPU is that they have a large number of ALUs (arithmetic logical units), and the result is that they can do multiple times the bulky mathematical computation that CPU can do. Therefore, as future work, our target will be to make use of users' GPUs for mining.

References

- [1] Bitcoin- A Peer to Peer Electronic Cash System - Satoshi Nakamoto For Transaction (2009)
- [2] Introduction to Bitcoin mining. Retrieved June, 2015 from A guide for geeks by David R. Sterry
- [3] Bitcoin inflation. Retrieved January, 20, 2015 from <https://www.buybitcoinworldwide.com/kb/hedge-against-inflation-with-bitcoin/>
- [4] Open Source as Appropriate Technology for education purpose- Dr. Patrick Carmichael (January 2002)
- [5] SHA-2. Retrieved February, 23, 2014 from <http://en.wikipedia.org/wiki/SHA-2>
- [6] A light weight remote procedure call protocol. Retrieved November, 14, 2015 from <http://www.jsonrpc.org>.
- [7] Mining workflow. Retrieved from February, 02, 2015 <https://bitcoin.org/ens/developer-guide#block-chain>.
- [8] Hash rate distribution data. Retrieved June, 07, 2015 from <https://blockchain.info/charts>
- [9] Guide for jQuery usage. Retrieved June, 17, 2015 from <https://learn.jquery.com>
- [10] Link to open with the bitcoin protocol. Retrieved June, 2015 form <http://pyalot.github.io/jquery-bitcoin-link/>
- [11] Bitcoin Working Under the Hood from Mastering Bitcoin by Andreas M. Antonopoulos. Retrieved July, 2015 from <https://www.youtube.com/watch?v=Lx9zgZCMqXE>
- [12] Web Worker. Retrieved November, 2015 from <http://html.spec.whatwg.org/multipage/worker.html>
- [13] Bitcoin. Retrieved April 25, 2016, from <https://en.bitcoin.it/wiki/Bitcoin>
- [14] CoinDesk: Bitcoin news, prices, charts, guides & analysis. Retrieved April, 25, 2016, from <http://coindesk.com>
- [15] Protocol Rules. Retrieved April 25, 2016, from https://en.bitcoin.it/wiki/Protocol_rules

- [16] Stackalytics | OpenStack community contribution in Newton release. Retrieved April 25, 2016, from <http://stackalytics.com/>
- [17] Mining Hardware comparison. Retrieved March, 8, 2014 from https://en.bitcoin.it/wikis/Mining_hardwares_comparison
- [18] Bitcoin.org – opensource p2p money. Retrieved April 25, 2016, from <http://bitcoin.org>
- [19] W3Techs – extensive and reliable web technology surveys. Retrieved April, 25, 2016, from <http://w3techs.com>
- [20] JavaScript MD5. Retrieved April, 2016 from <http://pajhome.org.uk/crypt/md5/>
- [21] Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System. Bitcoin, Page 8-10, Retrieved from <https://bitcoin.org/bitcoin.pdf> (2009)
- [5] SHA-256 Hash Algorithm. Retrieved May, 9, 2014 from <https://www.movable-type.co.uk/script/sha256.html>
- [22] Bitcoin mining the hard way - Ken Shirriff – (Feb 23, 2014)
- [23] Exploring Miner Evolution in Bitcoin Network - Luqin Wang – (March, 4, 2015)
- [24] Parallel Data Mining on Graphics Processors - Wenbin Fang – (2009)
- [25] Mining software. Retrieved April 25, 2016 from <https://www.bitcoinmining.com/bitcoin-mining-software/>
- [26] Swanson E. Bitcoin mining calculator used to calculate the bitcoin can be mined with given hardware. Retrieved September 2013 from <http://www.alloscomp.com/bitcoin/calculator>
- [27] Analysis of the Cryptocurrency Marketplace by Alex Heid (April 2014)
- [28] Analysis of Bitcoin Pooled Mining System by Meni Rosenfeld (November 2011)
- [29] Bitcoin Mining Pool - A Cooperative Theoretic Analysis by Yoad Lewenberg and Yonatan (May 2015)