Universitat Autònoma de Barcelona

escola d'enginyeria

5732 - ESTUDI DE LA MINERIA DE BITCOINS

Memòria del projecte de final de carrera corresponent als estudis d'Enginyeria Superior en Informàtica presentat per Joan Marí Yern i dirigit per Jordi Herrera Joancomartí.

Bellaterra, juny de 2014

El firmant, Jordi Herrera Joancomartí, professor del Departament d'Enginyeria de la Informació i de les Comunicacions de la Universitat Autònoma de Barcelona

CERTIFICA:

Que la present memòria ha sigut realitzada sota la seva direcció per Joan Marí Yern

Bellaterra, juny de 2014

Firmat: Jordi Herrera Joancomartí

Índex

Índex de figures

Índex de taules

Capítol 1

Introducció

El bitcoin és una moneda virtual descentralitzada que va ser creada el 2008 i va començar a circular el 2009. Aquesta moneda utilitza la criptografia per a assegurar el fons i validar les transaccions.

Cada usuari té dues claus, una pública i una privada. La clau pública serveix per a crear les adreces, que s'utilitzen per a enviar o rebre bitcoins, mentre que la clau privada serveix per a verificar la propietat d'una clau pública i per a signar les transaccions.

Ja que les transaccions són descentralitzades, s'utilitza un sistema de proof-of-work: una xarxa de computadors arreu del mon treballen per a verificar aquestes transaccions, prevenint que un usuari pugui gastar dues vegades la mateixa moneda. El procés de verificació d'aquestes transaccions s'anomena mineria, mentre que els usuaris que realitzen aquest procés s'anomenen miners.

Aquest projecte anirà centrat cap al procés de mineria.

1.1 Objectius

L'objectiu d'aquest projecte és el d'analitzar i entendre el funcionament intern de la moneda, i en especial el procés de mineria, amb la finalitat de crear el que es coneix com a mining pool, que és un clúster (o agrupació) d'ordenadors treballant junts en la mineria. A més s'analitzarà aquest procés, així com els diferents sistemes de rewarding.

El procés de mineria té un incentiu monetari per als miners. Els sistemes de rewarding són maneres de repartir aquest incentiu entre tots els participants d'una mining pool. En aquest projecte s'analitzaran alguns d'aquests sistemes, ja que trobar un *rewarding system* que recompensi a tots els participants de manera justa no és un problema trivial.

1.2 Motivacions personals

En l'actualitat, l'interes generat per les monedes virtuals basades en la criptografia, i en especial els Bitcoins, no ha deixat de créixer.

La idea d'una moneda que no requereix d'una entitat central per a funcionar, sinó que depèn d'una comunitat d'entitats desconegudes entre sí, mentre que s'usa la criptografia per a assegurar el seu correcte funcionament i per a evitar el malús de la moneda m'ha semblat interessant des que vaig conèixer la seva existència.

A més aquest projecte em servirà per a conèixer a fons una de les aplicacions pràctiques de la criptografia, un tema del que voldria ampliar els meus coneixements.

1.3 Estat de l'art

El Bitcoin és una de les primeres implementacions d'un concepte conegut com ha "criptomoneda", introduït el 1998 per Wei Dai, basant-se en la idea que els diners poden ser qualsevol tipus de registre que permeti l'intercanvi de bens i serveis. El Bitcoin es va dissenyar com a moneda que utilitza la criptografia per a controlar la seva creació i transaccions, en comptes de dependre d'una autoritat central.

El 2008 el Bitcoin va fer la seva primera aparició en un paper publicat baix el pseudònim de "Satoshi Nakamoto" [10], on es descrivia el seu funcionament. No va ser fins al 2009 que va aparèixer la primera implementació del Bitcoin, i no estava absenta d'errors, en la primera versió hi havia un bug que permetia la creació il·limitada de Bitcoins.

Amb una comunitat ràpidament creixent, aquesta moneda va anar agafant força i millorant en la seva implementació, fins que al 2011 el Bitcoin va guanyar l'atenció mediàtica, donant lloc a la primera de moltes bombolles que ha experimentat la moneda i que, tot i desinflar-se ràpidament, sempre han deixat a la moneda en una posició més forta de la que estava. Davant la sobtada pujada del preu del Bitcoin, no van tardar en crear-se fundacions per a regular, protegir i promoure la moneda.

La comunitat de mineria també ha anat creixent ràpidament, ja que amb la ràpida pujada del preu molts hi veuen un negoci. En l'actualitat hi ha un gran nombre de miners i de mining pools, tant públiques (a les quals qualsevol persona s'hi pot connectar) com de privades, que treballen validant les transaccions realitzades.

Per desgràcia, degut a la descentralització de la moneda i la privacitat de les adreces, els mercats negres han trobat en aquest tipus de monedes la manera d'intercanviar bens i ser-

veis il·legals de manera segura i privada, causant un seguit d'escàndols que han perjudicat tant el valor com la reputació de la moneda.

1.4 Organització de la memòria

Aquesta memòria del projecte de final de carrera sobre l'estudi de la mineria de Bitcoins consta de de set capítols. A continuació s'anomenarà i explicarà cadascun d'ells.

El primer capítol és la introducció de la memòria, i conté una breu introducció als Bitcoins i al seu funcionament, així com les meves motivacions personals per a haver escollit aquest projecte.

El segon capítol és una descripció més avançada del funcionament dels Bitcoins, sense endinsar-se encara en el procés de mineria. Aquí s'explicarà com funcionen les transaccions de la moneda i cóm es guarden de manera que tothom pugui accedir al historial d'aquestes transaccions.

El tercer capítol és un capítol dedicat al funcionament de la mineria, i tractarà cóm funciona aquest procés, perquè funciona i perquè és necessari per a l'existència de la moneda.

El quart capítol tracta sobre les *mining pools* i a més s'analitzarà quins són els *rewarding systems* més utilitzats i els seus problemes.

En el cinquè capítol es descriuran algunes altres monedes basades en criptografia derivades dels Bitcoins i es detallaran les diferències.

En el sisè capítol trobarem un procés detallat de com instal·lar i configurar una *mining pool* pas a pas.

Finalment, el setè capítol inclourà les meves conclusions sobre aquest projecte.

Capítol 2

Bitcoins

2.1 Adreces i la criptografia de clau pública

Per a realitzar transaccions es necessiten dues claus, una pública i una privada, que es representen mitjançant un seguit de caràcters alfanumèrics que poden tenir entre 27 i 34 caràcters en total. Aquesta parella de claus formen un compte de Bitcoins.

Les claus d'un compte formen les dues claus d'un algorisme criptogràfic de clau pública o asimètrica. En aquests tipus d'algorismes tenim quatre elements: l'algorisme de xifrat (al que representarem amb la funció f), l'algorisme de desxifrat (al que representarem amb la funció g) i dues claus, una pública (a la que anomenarem p_u) i una altra de privada (a la que anomenarem p_r).

Les claus estan lligades de manera que si apliquem l'algorisme de xifrat sobre un missatge (m) amb la clau pública, podem obtenir el missatge original a partir de l'altra clau amb l'algorisme de desxifrat. D'aquesta manera, amb $f(m, p_u)$, obtenim una encriptació del missatge m i $m = g(f(m, p_u), p_r)$, de manera anàloga també es compleix que $m = f(g(m, p_r), p_u)$.

En altres paraules, si es vol enviar un missatge encriptat a un usuari de criptografia de clau pública, només és necessari conèixer la seva clau pública, i una vegada que el missatge s'ha encriptat, ningú pot obtenir el missatge original sense conèixer la clau privada.

Un altre ús de la criptografia de clau pública és el de crear una signatura digital. La manera de fer-ho es agafar un missatge conegut (m) i aplicar-li l'algorisme d'encriptació usant la clau privada, de manera que s'obté el que es coneix com a una signatura digital (sd), així $sd = g(m, pr)$. S'anomena signatura ja que la única manera d'obtenir el missatge original és a través de la clau pública de la persona que ha creat la signatura, d'aquesta

manera es pot confirmar que el propietari de la clau pública és realment el creador de la signatura, ja que la única manera de crear-la és amb la clau privada.

Les claus de bitcoins es generen amb un algorisme de corba el·líptica, que ens permet generar claus ràpidament, sense necessitar una connexió a internet i amb una probabilitat molt propera a zero de generar claus duplicades (ja que si aconsegueixes generar un parell de claus que ja pertanyen a un altre usuari, es podria robar el contingut del compte fàcilment). L'algorisme per a la generació i verificació de la signatura s'anomena $ECDSA$ (Elliptic Curve Digital Signature Algorithm [9]).

La clau pública d'un compte de bitcoins està associada a l'adreça i és el que s'utilitza per a identificar-la. Així per enviar bitcoins d'un compte a un altre, només cal conèixer la clau pública del receptor, ja que així l'única persona que pot reclamar els bitcoins enviats és la persona que coneix la clau privada corresponent a l'adreça.

2.2 Transaccions

Una transacció és un enviament de bitcoins des d'unes adreces d'entrada cap a unes altres de sortida.

Les entrades són un seguit d'adreces corresponents als comptes des dels quals s'enviaran els bitcoins de les transaccions, mentre que les sortides són les adreces corresponents als comptes als quals van destinats.

Però només amb això no és suficient, qualsevol persona podria posar com a entrada una adreça corresponent a un compte que no li pertany. Així, per cada adreça d'entrada s'ha d'incloure una signatura digital creada utilitzant la clau privada corresponent a aquesta adreça d'entrada. A més, per a cada entrada també s'inclou un hash que ens indica quina és la darrera transacció en la que la adreça ha participat, això ens ajudarà a trobar fàcilment el balanç del compte.

Per a cada sortida només tenim dos elements: l'adreça que rebrà els bitcoins enviats i el valor, que representa la quantitat de bitcoins que rebrà el compte d'aquesta adreça en específic.

En resum, una transacció conté les següents dades:

- Un nombre n d'entrades, on per a cada entrada tenim:
  - $Previous\ tx$: Un hash a la transacció anterior.
  - $Addr_{input}$: Adreça des d'on es volen enviar els bitcoins.

- Un nombre m de sortides, on per a cada sortida tenim:
  - $Addr_{output}$: Adreça que rebrà els bitcoins.
  - $Value$: Valor corresponent a la quantitat de bitcoins.
- $Signature$: Signatura realitzada amb la clau privada de l'adreça d'entrada.

2.2.1 Transaccions d'imports parcials

Com podem observar, les entrades de les transaccions no tenen un camp amb la quantitat de bitcoins que cada adreça enviarà. Això és degut a que si s'inclou una adreça en una transacció, aquesta adreça enviarà la totalitat de bitcoins inclosos en el compte.

No existeix una manera inclosa en el protocol que ens permeti enviar només una part dels bitcoins d'una adreça que participa en una transacció. La única manera per a fer-ho és incloure l'adreça d'entrada com a sortida en la mateixa transacció.

Per exemple, si tenim una adreça "a"($addr_a$) que conté 50 bitcoins (BTC) i volem enviar 25 BTC a una adreça "b"($addr_b$) hauríem de crear una transacció amb les següents dades:

- Entrada
  - $Previous\ tx_a$: Hash de l'anterior transacció de $addr_a$
  - $addr_a$: Adreça de "a"
- Sortida 1
  - $addr_a$: Adreça de "a"
  - $Value$: 20 BTC
- Sortida 2
  - $addr_b$: Adreça de "b"
  - $Value$: 25 BTC
- $Signature_a$: Signatura de "a"

En aquesta transacció, "a"enviaria els 50 BTC, i d'aquests "b"en rebria 25, mentre que a "a"se li retornarien 20 dels 50 BTC enviats amb la transacció. Notem que el total dels valors rebuts per cada sortida no correspon a la totalitat de BTC que "a"ha enviat, la diferencia entre la quantitat enviada entre totes les entrades i la quantitat rebuda per

totes les sortides és considerada com una propina o quota de la transacció i és opcional. Aquesta propina anirà a parar a l'adreça del miner que verifiqui la transacció. Aquest procés s'explicarà amb més detall més endavant.

2.3 Blocs: informació bàsica

Els blocs són agrupacions de transaccions que ja han estat validades.

Cada bloc conté tota la informació d'un cert nombre de transaccions, així com una referencia al bloc immediatament anterior a l'actual.

Apart d'això, cada bloc conté la resposta a un problema matemàtic difícil que, tot i no ser única, és diferent per a cada bloc i que una vegada trobada, és molt fàcil comprovar que la resposta és correcta. Cap bloc pot ser generat sense la resposta a aquest problema.

Les entitats que es dediquen a buscar les solucions de cada bloc s'anomenen $miners$, i el procés de buscar la solució al problema s'anomena $mineria$.

Per a que una transacció sigui vàlida, aquesta ha d'estar inclosa dins un bloc i una vegada que la solució al problema matemàtic del bloc s'ha trobat, es diu que s'ha generat un nou bloc.

Com que cada bloc conté la informació del bloc anterior, l'agrupació de tots els blocs generats s'anomena cadena de blocs. D'aquesta manera, tota la informació dels bitcoins fins al moment es pot aconseguir a partir de l'últim bloc generat.

2.4 Nodes

Els bitcoins són una moneda descentralitzada, així que no tenim una entitat central dedicada a generar moneda, a verificar transaccions ni, en general, a guardar la informació de les transaccions realitzades prèviament.

Tota la informació de les transaccions és guardada per unes entitats anomenades nodes. Cada node conté una còpia de tots els blocs generats fins al moment. Tots aquests nodes estan connectats, formant una xarxa.

Aquesta xarxa és oberta, de manera que tothom pugui accedir-hi, o unir-s'hi i convertir-se en un node. La informació viatja per aquesta xarxa a través de simples broadcasts de manera que quan un node rep nova informació, aquest node envia immediatament un missatge amb aquesta informació a tots els altres nodes als quals està connectat i aquests

alhora s'encarreguen de reenviar aquest missatge fins que tots els nodes de la xarxa tenen la mateixa informació.

Abans de reenviar el missatge, cada node analitza la informació rebuda per assegurar-se que el contingut sigui correcte. Entre altres coses, comproven que les signatures de cada transacció i la resposta al problema matemàtic de cada bloc siguin correctes. Si alguna d'aquesta informació no és correcta, el node descarta el missatge i la informació no és reenviada.

Per a convertir-se en node d'aquesta xarxa, primer s'ha de connectar amb un node existent i enviar-l'hi un missatge amb la informació de l'últim bloc al qual es té coneixement. Si aquest no és l'últim bloc que el node té guardat, aquest respondrà amb la informació dels blocs que vénen a continuació.

Per a realitzar una transacció, un usuari només ha d'enviar la informació d'aquesta transacció a un node. D'aquesta manera el node és el que s'encarrega que aquesta transacció s'expandeixi per tota la xarxa, fent-la pública, fins que un miner l'agafi i generi un bloc que la contingui i quedi permanentment inclosa dins la cadena de blocs.

Capítol 3

Mineria

La mineria és el procés d'afegir nous registres de transaccions als ja existents, en altres paraules, a la cadena de blocs. Les entitats que es dediquen a minar s'anomenen miners.

3.1 Proof of work

Per a generar blocs, es demana que els miners realitzin una quantitat considerable de treball computacional. Per a demostrar que un miner ha realitzar aquest treball, junt amb el bloc s'ha de presentar la solució a un problema matemàtic complex. Aquest sistema es coneix com a *proof of work*.

Els sistemes de *proof of work* s'utilitzen per a evitar atacs de denegació de servei i altres abusos com l'spam. En el cas dels bitcoins, aquest sistema evita que es puguin generar blocs de manera indiscriminada.

Hi ha diverses raons per a utilitzar un sistema de *proof of work* en els bitcoins. Un dels motius és que d'aquesta manera es dificulta la falsificació de transaccions. Ja que tots els blocs contenen informació dels blocs anteriors, si es vol alterar una transacció d'un bloc, també s'han d'alterar diversos blocs de manera retroactiva ja que s'haurien de modificar tots els blocs que venen a continuació, cosa que implica una quantitat prohibitiva de potència de càlcul.

D'aquesta manera, el *proof of work* prevén el malús dels bitcoins, evitant el frau o que es pugui gastar dues vegades la mateixa moneda.

Els bitcoins utilitzen funcions *hash* per a realitzar el treball necessari per al *proof of work*.

3.1.1 Funcions Hash

La idea de les funcions *hash* és simple. Donada certa informació, l'algorisme genera una cadena de caràcters que és fixe respecte la informació d'entrada, aquesta cadena s'anomena *hash*. Una de les propietats d'aquestes funcions és que són difícils d'invertir, així que es pot obtenir fàcilment el *hash* amb la informació original, però és impossible obtenir la informació a través del *hash*.

La natura d'aquests algorismes fa que sigui impossible predir el *hash* que s'obtindrà amb determinada informació d'entrada abans d'aplicar l'algorisme. D'aquesta manera el *hash* obtingut sembla aleatori respecte les dades d'entrada.

En el cas dels bitcoins, l'algorisme *hash* utilitzat és el sha256 [7].

3.1.2 Dificultat

Aplicar l'algorisme per a trobat un *hash* no té un alt cost computacional, però no tots els *hashes* trobats són vàlids per a generar un bloc. Per a que el *hash* sigui vàlid, aquest ha de ser menor que un nombre determinat, anomenat *target*.

Com que el *hash* calculat és aleatori, la única manera de trobar un *hash* que sigui menor que el *target* és mitjançant assaig i error: per a trobar un *hash* vàlid, s'ha d'aplicar l'algorisme diverses vegades amb diferent informació d'entrada fins que es trobi un *hash* vàlid.

El *proof of work* dels bitcoins està pensat de manera que un nou bloc sigui generat cada 10 minuts aproximadament. Així la dificultat del càlcul d'un *hash* vàlid és periòdicament ajustada de manera que sempre es trobi un nou bloc amb una mitjana de 10 minuts.

Per ajustar la dificultat l'únic que s'ha de fer és modificar el *target* per tal que la mitja d'intents que els miners necessiten per a trobar un *hash* correcte s'ajusti al temps al qual es desitja que apareguin nous blocs.

Cada 2016 blocs generats (que tarden aproximadament dues setmanes en ser generats) tots els clients de bitcoins comparen el temps que realment s'ha tardat en generar els blocs amb les dues setmanes que hauria d'haver tardat i es modifica el *target* pel percentatge de la diferencia, ajustant la dificultat per tal que els pròxims blocs es generin aproximadament en el temps previst.

La dificultat de calcular un bloc és representada amb un nombre, aquest nombre es calcula amb el quocient entre la mínima dificultat possible (màxim *target* possible) i el *target* actual.

El màxim *target* possible és:

0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Així, per a un *target* de

0x000404CB000000000000000000000000000000000000000000000000

Tenim que la dificultat és:

$$\text{0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF} \;/\; \text{0x000404CB000000000000000000000000000000000000000000000000} = 16307.669773817162$$

3.2 Generació de blocs

Les entitats que volen minar (o generar nous blocs) necessiten tres coses:

- La cadena de blocs, amb tots els blocs generats fins al moment.
- Una llista de transaccions que encara no han estat validades.
- El hardware necessari per a aplicar l'algorisme que genera el *hash*.

Es necessita la cadena de blocs ja que el miner ha de comprovar la validesa de totes les transaccions que vol incloure en el bloc, així que ha de tenir l'històric de totes les transaccions realitzades fins al moment.

El miner a més ha d'estar connectat a la xarxa per a rebre contínuament els nous blocs generats així com les transaccions realitzades recentment. Ja que les transaccions poden tenir una quota de transacció (propina) inclosa i com el cost de processament del *hash* no incrementa amb el nombre de transaccions incloses, és beneficiós pel miner intentar validar quantes més transaccions millor.

Un bloc conté els següents elements (mostrats a la figura 3.1):

Altura	223450 (Main chain)
Hash	0000000000000062a5d3faa84910a890b66e1642f6f48050cb0f94ae58557207
Bloque Anterior	00000000000000218f7f16d403cba5bd82d53b570102e1db4539832b45b05c2df
Siguiente Bloque	00000000000004265f7c717d38755e00324afa347d78d8e58ee953c0914d71ea
Tiempo	2013-02-27 18:13:35
Dificultad	3,651,011.63
Bits	436508764
Número de Transacciones	452
Salida Total	3,182.04972495 BTC
Volumen de Transacciones Estimado	1,290.58861876 BTC
Tamaño	166.5556640625 KB
Versión	2
Raíz de Merkle	2073e2dc01edd6825f207e02fe574bc198f2228a186a8518e5796c74bdbb8b93
Nonce	1808624490
Recompensa de Bloque	25 BTC
Comisión de las Transacciónes	0.3747 BTC

Figura 3.1: Informació del bloc

- Altura: nombre corresponent a la distància des del primer bloc generat dins la cadena de blocs.
- Hash: hash generat corresponent al *proof of work*.
- Bloc anterior: hash corresponent al bloc anterior.
- Bloc següent: hash corresponent al bloc següent. Això és informació addicional que s'ha afegit posteriorment i no està inclosa en el bloc.
- Temps: hora en UCT del moment quan s'ha generat el bloc.
- Dificultat: valor numèric que representa la dificultat de càlcul del *proof of work*.
- Bits: *target* actual en format compacte.
- Nombre de transaccions: nombre de transaccions incloses en el bloc.
- Mida: Mida en kb de la transacció.
- Versió: versió del protocol de bitcoin que el miner que ha generat el bloc ha utilitzat.

- Arrel de Merkel: arrel de l'arbre de Merkel del bloc.
- Nonce: nombre donat pel miner.
- Recompensa del bloc: bitcoins amb els quals el miner és recompensat.
- Comissió de les transaccions: bitcoins que els que han realitzat les transaccions han deixat com a propina.
- La llista de les transaccions incloses en el bloc, incloent la informació descrita en el capítol anterior.
- L'adreça del compte del miner.

La major part d'aquests elements s'utilitzen com a paràmetres en l'algorisme per a la generació del *hash*.

A continuació s'explicarà el funcionament d'alguns elements del bloc i la implicació en la generació dels mateixos.

3.2.1 Nonce

El *nonce* és un valor que el miner tria i que no està fixat.

Com la mateixa informació sempre generaria el mateix *hash*, es necessita un paràmetre que no depengui de la resta de la informació del bloc i que es pugui modificar sense que afecti al bloc. Aquest paràmetre és el *nonce*.

Per a generar un bloc, el miner tria un *nonce* i aplica l'algorisme per a generar el *hash*.

Si el *hash* obtingut és vàlid, el bloc ha sigut generat amb èxit i el miner pot enviar aquest bloc a la resta de nodes per a que quedi inclòs en la cadena de blocs.

Si el *hash* no és vàlid, el miner tria unaltre *nonce* i torna a començar el procés fins que es troba un *nonce* amb el que s'obtingui un *hash* vàlid.

3.2.2 Adreça del miner

Els miners tenen un incentiu per a continuar validant transaccions.

Cada vegada que un miner genera un bloc, el miner rep una quantitat determinada de bitcoins. A més també rep les propines incloses en totes les transaccions del bloc validat. D'aquesta manera, els miners tenen un benefici econòmic que assegura que hi hagi gent

interessada en participar en aquest procés, que és absolutament necessari per a que els bitcoins puguin funcionar correctament.

Aquest benefici s'envia a l'adreça inclosa en aquest camp.

3.2.3 Hash del bloc anterior

Per a generar el bloc actual es necessita saber el *hash* de l'últim bloc de la cadena. Aquest és un dels paràmetres que l'algorisme necessita per a calcular el *hash*.

Això implica que si un bloc s'afegeix a la cadena, tots els miners han de canviar aquest paràmetre quan intentin generar nous blocs. D'aquesta manera els *nonce* que els miners han utilitzat per a intentar generar el bloc anterior, al canviar el paràmetre que conté l'últim bloc de la cadena, ara donaran *hashes* diferents.

Per tant cada vegada que es genera un nou bloc, tots els miners han de començar de nou a provar amb tots els *nonce* possibles.

3.2.4 Merkel tree

A mesura que el nombre d'usuaris de bitcoins augmenta, els nodes han de fer cada vegada més comprovacions en cada bloc, ja que el nombre de transaccions que contenen també augmenta.

Per reduir els càlculs necessaris per a la verificació es fa servir l'anomenat *Simplified payment verificatio*, o SPV, que aprofita l'estructura dels arbres de Merkel per a comprovar només la capçalera del bloc.

Els arbres de Merkel són arbres que tenen en cada branca un hash de les etiquetes dels fills i blocs de dades a les fulles. A més, per a comprovar que una fulla pertany a l'arbre requereix un processament proporcional al logaritme del nombre de nodes de l'arbre. Aquests arbres permeten validar de manera òptima i segura el contingut d'una estructura de dades més gran.

Així, el SPV verifica que la cadena de branques de l'arbre estan connectades a l'arrel de l'arbre (*Merkel root*) i que la dificultat de l'arbre és suficient. Després comprova que les transaccions estiguin lligades a alguna branca del mateix. D'aquesta manera es pot verificar el contingut d'un bloc sense necessitar tot el contingut del bloc.

3.2.5 Mida màxima de bloc

Els blocs generats tenen una mida màxima possible.

En aquest moment, la mida màxima a la que poden arribar els blocs és d'$1MB$. Així si el bloc ja ha arribat a la mida màxima, no es poden afegir més transaccions dins el bloc.

La mida màxima actual permet incloure aproximadament 4630 transaccions dins cada bloc, de manera que ara mateix el protocol només pot aguantar una càrrega d'aproximadament 7 transaccions per segon.

Tenint en compte que qualsevol banc actual ha d'estar preparat per a suportar centenars de transaccions per segon, aquesta limitació pot suposar un problema en el futur si els bitcoins s'estenen de manera àmplia i és possible que aquesta limitació sigui modificada.

3.3 Generació simultània de blocs

Quan un miner genera un bloc, aquest l'envia als nodes als quals està connectat, i aquests successivament el reenvien fins que el bloc s'ha propagat arreu de la xarxa.

Entre el moment en que s'ha generat el bloc i el moment en que tots els nodes han rebut el bloc i l'han inclòs en la seva cadena de blocs pot passar que un altre miner generi un altre bloc vàlid i el comenci a propagar a través d'uns nodes diferents.

En aquest cas, podem tenir una bifurcació de la cadena de blocs, però només un dels camins podrà ser vàlid.

Quan es produeix aquest cas, els miners segueixen treballant amb qualsevol de les bifurcacions, fins que una de les bifurcacions és més llarga que l'altra. En aquest moment la cadena més llarga és la que queda acceptada, mentre que la més curta és descartada i les transaccions que incloïen queden invalidades i s'han de tornar a incloure en un altre bloc.

A causa d'aquest problema, no es considera que una transacció ha sigut validada fins que s'han generat sis nou blocs sobre el bloc que té la transacció, donant temps així de solucionar els possibles problemes de bifurcacions que puguin aparèixer.

3.4 Generació de moneda

Com ja s'ha explicat, quan un miner aconsegueix generar un bloc amb èxit se li assignen una quantitat determinada de bitcoins al seu compte, apart dels rebuts en les propines.

Any (aproximat)	Bloc	$BTC/bloc$	BTC afegits	Total de BTC
2009	0	50	10500000	10500000
2013	210000	25	5250000	15750000
2016	420000	12,5	2625000	18375000
2020	630000	6,25	1312500	19687500
2024	840000	3,125	656250	20343750
2028	1050000	1,5625	328125	20671875
2032	1260000	0,78125	164062,5	20835937,5
2036	1470000	0,390625	82031,25	20917968,75
2040	1680000	0.1953125	41015,625	20958984,375
2044	1890000	0.09765625	20507,8125	20979492,1875

Taula 3.1: Planificació estimada de generació de bitcoins

Aquests bitcoins no provenen de cap compte, són nous bitcoins que entren en circulació en el moment en que el bloc és generat. Per tant la mineria no només és el procés per a validar transaccions, també és el proces mitjançant el qual es genera nova moneda.

El procés de generació de bitcoins està controlat de manera que en total mai hi hagi més de 21 milions de bitcoins. Per això es va crear un pla estimat de generació de moneda, que podem observar en la taula 3.1.

La idea en la generació de moneda és que cada 210.000 blocs, la quantitat de BTC generats en cada bloc es redueix a la meitat. Es pot comprovar fàcilment que seguint aquestes instruccions, en total s'arribaran a generar 21.000.000 BTC.

$$\sum_{n=0}^{\infty} \frac{210000 \cdot 50}{2^n} = 210000 \cdot 50 \cdot \frac{1}{1-\frac{1}{2}} = 21000000$$

Degut a que la base monetària dels BTC no pot ser expandida i a que si un usuari perd les claus d'un compte amb BTC a dintre, aquests queden perduts per sempre, es considera que el bitcoin és una moneda deflacionista.

Capítol 4

Mining Pools

Amb l'augment de la dificultat, les màquines amb baixa o mitjana capacitat de processament necessiten una gran quantitat de temps de mitjana per a poder generar un bloc amb èxit.

Per a permetre que aquests miners rebin un incentiu de manera regular, es van crear el que es coneixen com a *Mining Pools*.

4.1 Pooled mining

La idea darrera de les *Mining pools* és simple: un miner que treballa sol (*Solo Miner*) té una possibilitat molt baixa de generar un bloc. Depenent del hardware que s'utilitzi poden passar anys entre cada bloc generat.

Les *Mining pools* són agrupacions de miners que treballen junts per a generar el mateix bloc. Així, a cada miner que participa en la generació dels blocs se li assigna una fracció de la recompensa de manera regular, depenent del poder de processament amb el que cadascun ha contribuït.

Per determinar la quantitat de treball realitzat per la *pool*, els usuaris busquen i envien *shares*, que són *proof of work* del mateix tipus de l'usat per a generar blocs, però amb una dificultat més baixa.

D'aquesta manera, cada *share* trobat per un usuari té una possibilitat de ser el *proof of work* necessari per a generar el bloc. A més, suposant que l'algorisme de generació del *proof of work* funcioni correctament, és impossible trobar *shares* sense fer la feina necessària per a generar blocs, per tant el nombre de *hashes* calculats en un intent per a generar el bloc és directament proporcional al nombre de *shares* trobats.

A més, les *mining pools* normalment estan gestionades pel que es coneix com a *pool operator*, que sol rebre un percentatge dels beneficis a canvi dels serveis oferts.

4.2 Rewarding Systems

Mitjançant els *shares*, els operadors de les *mining pools* poden saber amb quant poder de processament ha contribuït cada usuari, però decidir com repartir la recompensa de manera que a cada usuari se li assigni una part proporcional al treball realitzat no és un problema trivial.

Per a solucionar aquest problema, s'han dissenyat un seguit de mètodes (o *Rewarding Systems*) [11]. A continuació es descriuran algun d'ells.

4.2.1 Sistema proporcional

Aquest és el *Rewarding System* més simple i el que intuïtivament sembla que funcioni millor per a repartir la recompensa.

En aquest mètode, els pagaments es calculen basats en rondes, on cada ronda és el temps entre que la *pool* troba un bloc fins al següent.

Al final de cada ronda, la *pool* rep un benefici de B BTC i l'operador es queda una quota de f, per tant $(1 - f) \cdot B$ BTC es reparteixen entre els usuaris de manera proporcional al nombre de *shares* presentats durant la ronda.

Malauradament, aquest mètode només és valid en el cas que els miners treballin de manera honesta. Un miner pot aconseguir una part superior de la recompensa de la que li tocaria per la seva contribució fent el que es coneix com a *pool hopping*, que consisteix en canviar de *mining pool* en moments estratègics, que fa que els miners que minen de manera contínua en una mateixa *mining pool* tenguin menys recompensa.

Per entendre perquè el *pool hopping* funciona, s'ha de recordar que la recompensa per a cada *share* és $\frac{(1-f)B}{N}$, sent N el nombre total de *shares* presentats en la ronda. D'aquesta manera, quan més duri la ronda, menys recompensa tindrà cada *share*, ja que més *shares* s'hauran presentat.

És fàcil veure llavors que, si només es mina en la part final d'una ronda, la recompensa serà menor de la normal, mentre que si nomes es mina en la part inicial de la ronda, la recompensa serà major. Per ser més exactes, quan el nombre de *shares* supera el $43,5\%$ de la dificultat, la recompensa per a cada *share* ja és menor que la recompensa normal,

per tant els miners poden deixar de minar en aquesta *pool* i començar a minar en un altra on la ronda acabi de començar, danyant així als miners honests.

4.2.2 Mètode de Slush

El mètode implementat en la *mining pool* de Slush [1] va ser el primer dissenyat per a resistir el *pool hopping*.

És una variació del mètode proporcional, però en comptes basar la recompensa segons el recompte de *shares*, cada *share* aporta una puntuació al miner, i al final de la ronda es recompensa als participants segons aquesta puntuació.

La quantitat de punts que aporta cada *share* depèn de la quantitat de temps que ha passat des del començament de la ronda, quant més temps passi, més alta és la puntuació.

La funció usada per a calcular la puntuació és exponencial, així la puntuació p és $p = e^{T/C}$, on C és una constant i T és el temps des de l'inici de la ronda. El fet de tenir una funció exponencial per a calcular la puntuació permet arribar a un estat estacionari poc després de l'inici de la ronda on no importa quant un *share* sigui presentat, la puntuació relativa als previs i als següents *shares* és la mateixa, així com la recompensa.

Per desgràcia, aquest mètode també té problemes, el més important és que tot i que poc després de començar la ronda s'arriba a l'estat estacionari, al inici de la ronda no hi ha gaires *shares* amb els quals la recompensa potencial hagi de ser compartida, així que encara és més profitós minar al començament de la ronda que al final, tot i que aquest efecte no és tant gran com en el mètode proporcional.

4.2.3 Mètode Geomètric

El mètode geomètric és un mètode resistent al *pool hopping* basat en el mètode de Slush. Aquest mètode també està basat en exponencials per al càlcul de la puntuació, però aborda les debilitats del mètode de Slush i proveeix un marc matemàtic sòlid.

En aquest mètode hi ha dos tipus de quota de l'operador, una fixada i una altra variable. La fixada és una quantitat constant de la recompensa de cada bloc, la quota variable agafa la forma de puntuació que s'aporta automàticament al començament de la ronda i va decreixent de la mateixa manera que la puntuació dels participants, de manera que quant més curta sigui la ronda, més alta serà aquesta quota variant.

Igual que en els altres mètodes, f és la tarifa de l'operador, i s'agafa c que és la mitjana de la quota variable. Això significa que de cada bloc amb recompensa B, l'operador es

queda $(c + f - cf)B$ de mitjana i els participants reben $(1 - c)(1 - f)B$.

Llavors tenim un comptador s, que és la puntuació que s'entregarà al següent participant que presenti un *share*, que és un múltiple de la mitjana de recompensa depenent de la dificultat pB. A mesura que la ronda progressi, s creixerà exponencialment, fent que els *shares* antics decreixin en el seu valor real.

Aleshores, si tenim que S_k és la puntuació d'un participant, r el rati de descomposició (el rati en el que la s augmenta amb el temps) i N el nombre total de *shares* presentats en la ronda, el participant rep:

$$\frac{S_k}{\sum_{i=-\infty}^{N} r^{i-1} pB}(1 - f)B$$

Això significa que a l'operador se li assignen un nombre infinit de *shares* al començament de cada ronda. Quan el miner té l'oportunitat de presentar un *share*, veurà rere seu una sèrie infinita de *shares*, amb puntuacions que comencen amb la puntuació de la *share* que va a presentar i que decauen a un determinat rati. Aleshores les propietats estadístiques de la recompensa del *share* són sempre les mateixes, d'aquesta manera no hi ha cap avantatge al minar al començament de la ronda.

4.2.4 Pay-per-share

En aquest mètode, quan un participant presenta un *share*, se li recompensa immediatament amb una quantitat fixada $(1 - f)pB$, proporcional al valor estimat de la contribució del *share*. Així l'operador es queda tota la recompensa dels blocs.

Això correspon certes avantatges pels miners:

- No existeix variància en la recompensa de cada *share*.
- No hi ha temps d'espera entre que el *share* s'ha trobat i que el miner obté la recompensa.
- Es fàcil aproximar la recompensa que cada miner rebrà.
- Es fàcil verificar que l'operador paga la quantitat promesa de recompensa.
- El *pool hoppin* no és efectiu en aquest mètode.

Tot i així, aquest mètode suposa un risc molt gran pels operadors de les *mining pools*, ja que per a poder oferir variància zero als participants, tota la variància recau sobre

l'operador. Això significa que per a rondes curtes l'operador té molt de benefici, però pot perdre una quantitat considerable en rondes llargues. La variància de l'operador és la mateixa que els miners que fan *solo mining*. Per a compensar això, normalment els operadors exigeixen una quota més alta (f).

Si l'operador no balanceja la *pool* de manera correcta, és molt fàcil que la *mining pool* quedi en la bancarrota.

4.2.5 Pay-per-last-N-shares (PPLNS)

PPLNS són una família de mètodes mitjançant els quals la recompensa es reparteix entre els participants que han presentat blocs recentment, sense tenir en compte quants blocs s'han presentat abans. Fent això s'elimina el concepte d'obtenir benefici al minar al començament d'una ronda.

La manera més simple de fer-ho és triant un nombre fixat de N blocs i entregar una recompensa de $\frac{1-f}{N}B$ per cadascuna de les últimes N *shares* presentades.

La recompensa esperada per cada *share* és $\frac{(1-f)BL}{N}$, on L és el nombre de blocs trobats en els següents N *shares* i si p és la probabilitat d'un *share* de ser un bloc, L segueix una distribució de Poisson amb mitjana $\lambda = pN$. D'aquesta manera la variància de la recompensa és aproximadament $\frac{pB^2}{N}$, que és N vegades millor que fent *solo mining*. Per desgràcia aquest mètode no té cap efecte en la variància basada en la mida de la *pool*, per tant en pools molt petites, la variància de la recompensa no canvia molt respecte el *solo mining*.

4.2.6 Mètode Geomètric doble

Tots els mètodes que intenten evitar el *pool hopping* intenten que els miners vegin la mateixa història relativa sense importar quan el *share* és presentat. El mètode geomètric ho fa mitjançant una quota variable, de manera que si no hi ha suficients *shares* en la ronda, l'operador simula els *shares* a través de la quota variable. Fent-ho així, cada *share* és recompensat únicament pel següent bloc trobat i no els futurs, d'aquesta manera la variància de les recompenses és absorbida per l'operador.

En canvi, els mètodes PPLNS mantenen una variància constant a l'ignorar les rondes completament i només donant recompensa als *shares* recents, però aquest mètode no ofereix cap sistema per a reduir la variància basada en la mida de la *pool*.

El mètode geomètric doble és un sistema resistent al *pool-hopping* que està entre aquests

dos extrems. Les fronteres de les rondes es creuen, però no s'ignoren completament: es fixa un paràmetre per controlar el grau de reducció entre rondes.

Amb aquest mètode, en comptes de deixar la puntuació igual al final d'una ronda (com en el PPLNS) o deixar totes les puntuacions a 0 i transferir-les a l'operador (com el geomètric), només una part de la puntuació és transferida a l'operador. Quan les rondes són llargues, els participants es queden la major part de la puntuació de la ronda, mentre que en rondes curtes l'operador absorbeix la major part de la puntuació dels participants.

En general això redueix la variància de la recompensa dels participants en relació a la sort de la *pool* en trobar blocs.

Capítol 5

Altres Monedes

El bitcoin va ser la primera moneda virtual basada en criptografia, però a causa d'un increment en l'interès per la moneda i a que aquesta té una implementació i codi totalment oberts, centenars d'aquests tipus de monedes existeixen en l'actualitat, la majoria d'elles derivades del bitcoin.

Recordem algunes de les especificacions dels bitcoin:

- Blocs apareixen cada 10 minuts.
- La dificultat s'ajusta cada 2016 blocs.
- El màxim de moneda disponible seran 21 millions de BTC.
- L'algorisme per a calcular el hash és el *SHA256d*.
- Recompensa inicial de 50 BTC per bloc.
- La data d'aparició és el 3 de gener del 2009.

A continuació s'explicaran algunes d'aquestes monedes i les principals diferencies amb els bitcoins.

5.1 Litecoin (LTC)

Els bitcoin treballen amb el *SHA256d* com a algorisme per a generar els *hashes*. Aquest algorisme té un problema, i és que està completament optimitzat a través d'un hardware específic. Això significa que si no es té aquest hardware, l'efectivitat amb la que es minen

bitcoins usant CPU o GPU normals és molt reduïda comparada amb els que usen aquest hardware. Per tant no és pràctic minar bitcoins si no es té aquesta peça de hardware, que no és barata.

La idea rere els Litecoin[3] és la de crear una moneda que pugui ser minada eficientment amb equip que estigui disponible a tot el públic. Per això l'algorisme per a calcular els *hashes* que utilitzen els Litecoin és un algorisme anomenat *scrypt*. Aquest algorisme està fet de tal manera que requereix un ús intensiu de la memòria que els ordenadors i GPU dels consumidors normals ja tenen, i que provoca que sigui molt costós crear un hardware específic que sigui tant eficaç que la resta quedin obsolets.

Excloent l'algorisme per al càlcul del hash i la freqüència d'aparició dels blocs, la resta d'especificacions són similars al bitcoin:

- Blocs apareixen cada 2.5 minuts.
- La dificultat s'ajusta cada 2016 blocs.
- El màxim de moneda disponible seran 84 millions de LTC.
- L'algorisme per a calcular el hash és el *scrypt*.
- Recompensa inicial de 50 LTC per bloc.
- La data d'aparició és l'octubre de 2011.

5.2 Dogecoin (Doge)

El dogecoin[2] és una moneda digital que utilitza un gos de la raça japonesa *shiba inu* del meme d'internet *doge* com a mascota. La idea dels creadors era la de crear una moneda digital divertida que pogués arribar a un públic més ample, a més d'intentar allunyar-se de tota la polèmica rere els bitcoins.

Tot i que la moneda pot usar-se de la mateixa manera que s'usen els bitoin, la moneda és activa en comunitats socials de la xarxa, com *Reddit* o *Twitter*, on són usats per a premiar comentaris o articles regalant a l'autor una certa quantitat de dogecoins; a més d'això, també són usats en campanyes de donacions, entre altres.

En quant a especificacions, el dogecoin és una escissió del Litecoin, amb la principal diferència que, fins que s'hagin generat 145.000 blocs, la recompensa de Doge per bloc és aleatòria. Les recompenses dels blocs del 1 al 100.000 és un nombre aleatori entre 0

i 1.000.000, dels blocs 100.001 al 144.999 és d'entre 0 i 500.000 Dogecoin. A partir del bloc 145.000 la recompensa per bloc queda fixada i segueix un esquema similar al bitcoin.

Així tenim que:

- Blocs apareixen cada minut.
- La dificultat s'ajusta cada bloc.
- El màxim de moneda disponible seran 99.000 millions de Dogecoin aproximadament (degut a l'aleatorietat de la recompensa).
- L'algorisme per a calcular el hash és el *scrypt*.
- Recompensa inicial d'entre 0 i 1.000.000 Doge (aleatòriament) per bloc.
- La data d'aparició és el desembre de 2013.

5.3 Peercoin (PPC)

El Peercoin[6], també conegut com a PPC, PPCoin, P2P coin o peer-to-peer coin, és una moneda virtual que utilitza el sistema de *proof-of-stake* a més del sistema de *proof-of-work*.

El sistema de *proof-of-stake* s'utilitza en combinació amb el sistema de *proof-of-work* per a fer front a les vulnerabilitats que poden aparèixer amb un sistema que només utilitza el *proof-of-work*. La principal vulnerabilitat del sistema de *proof-of-work* dels bitcoins és que es pot donar un cas de monopoli de la mineria: a mesura que la recompensa vagi disminuint, el nombre de miners interessats en minar la moneda pot anar decreixent, cosa que augmenta la possibilitat del monopoli.

El monopoli és perillós en els bitcoins ja que deixaria la xarxa vulnerable al que es coneix com a atac del 51%: si es controla el 51% de la mineria, seria possible per l'entitat que ho controlés de gastar una mateixa moneda dues vegades.

Amb el sistema de *proof-of-stake*, la nova moneda es genera basada en la quantitat de moneda que l'entitat que ha generat el bloc posseeix, així algú que posseeixi un 1% de la moneda generarà un 1% de tots els blocs de *proof-of-stake*. Això fa que el monopoli sigui més costós i separa el risc del monopoli del sistema de *proof-of-work*.

A més, el sistema híbrid de *proof-of-work*/*proof-of-stake* també va ser dissenyat per a escurçar els costos de la mineria dels bitcoins: a l'abril del 2013 els miners gastaven en

total uns 150.000 USD per dia en costos de llum. La recompensa del sistema de *proof-of-work* en els PPC està dissenyada per a que decreixi quan la dificultat augmenti, així passat un temps, quan la dificultat hagi crescut lo suficient, el *proof-of-stake* passarà a ser la principal font de generació de la moneda, cosa que retallarà els costos de llum, ja que la consumició d'energia del sistema de *proof-of-stake* és mínima

És una moneda basada en el bitcoin, així que las especificacions són similars:

- Blocs apareixen cada 10 minuts.
- La dificultat s'ajusta cada blocs.
- El màxim de moneda no està determinat.
- L'algorisme per a calcular el hash és el *SHA-256*.
- La recompensa varia segons la dificultat del bloc.
- Aparició del sistema híbrid de *proof-of-work*/*proof-of-stake*
- La data d'aparició és l'agost de 2012.

5.4 Cadenes alternatives

Tota la informació de la moneda (com moneda disponible o transaccions) es troba en la cadena de blocs. Cada moneda de les descrites anteriorment té la seva pròpia cadena, de manera que les transaccions realitzades en una cadena no tenen efecte en una altra.

De la mateixa manera, hi ha monedes que utilitzen exactament el mateix protocol que els bitcoins, però tenen cadenes pròpies diferents de la dels bitcoins, cosa que provoca que es considerin monedes diferents.

5.4.1 Testnet

La Testnet és una cadena alternativa, creada amb l'únic propòsit de permetre als desenvolupadors testejar i experimentar amb el protocol sense tenir que usar bitcoins reals ni preocupar-se de trencar la cadena principal.

La principal diferència entre la cadena principal i la cadena Testnet és que els nodes connectats a la xarxa d'aquesta cadena escolten la informació d'entrada d'altres nodes a través d'un port diferent.

A més, la dificultat mínima de la Testnet és equivalent a una dificultat de 0.5 en la cadena principal per tal que petits desenvolupadors de programes de mineria tinguin una possibilitat alta de trobar blocs.

No només els bitcoins tenen una cadena Testnet, la majoria de les monedes virtuals principals la tenen.

5.4.2 Namecoin (NMC)

El Namecoin[5] és una altra cadena alternativa basada en els bitcoin.

Namecoin és una distribució alternativa de *Domain Name System* (DNS) que es basa en el software dels bitcoins. Aquesta distribució expandeix el software per a permetre registrar, actualitzar i transferir dominis.

A través d'aquest sistema, un *Domain Name* pot ser registrat pagant una quota de 0.01 NMC. Després d'això, la persona que ha pagat té propietat sobre el domini i només pot ser canviada si és transferit.

Algunes de les utilitats dels NMC és que creen un nou domini d'alt nivell fora del control de la *ICAAN*, cosa que provoca una reducció en la censura d'internet.

Altres usos que se li pot donar a aquesta moneda són: mapping de noms (com hostnames o nickname) a empremtes de clau pública, missatgeria, registrament d'adreces (com bitcoin, namecoin, emails, etc) a un alias, webs de confiança o sistemes de votació.

5.4.3 Mastercoin (MSC)

Els Mastercoin[4] és una moneda derivada dels bitcoins i té una cadena alternativa. Aquesta moneda van ser dissenyada com una serie de protocols per a ampliar les funcions financeres dels bitcoins.

Els creadors afirmen que els bitcoins són una capa de protocol sobre la que construir noves capes amb noves regles sense necessitat de canviar la base.

Algunes de les propietats que els mastercoins afegeixen sobre els bitcoins són: capes addicionals de seguretat, suport per a l'intercanvi distribuït de moneda, capacitat d'adquirir un valor estable definit per l'usuari (com una quantitat d'or o de USD) sense tenir que confiar en una persona amb la promesa del valor de la moneda.

5.4.4 Merged mining

Algunes monedes permeten l'anomenat *Merged mining* amb cadenes alternatives d'altres monedes.

El *Merged mining* permet a un miner de minar en més d'una cadena de blocs a la vegada, de manera que cada càlcul realitzat per a un bloc d'una cadena contribueix al càlcul d'un bloc d'una altra.

Per a entendre com és possible això, suposem que hi ha dues cadenes, una principal i una altra alternativa de la primera que permet el *merged mining* sobre la principal.

Primer, el miner ha de crear una transacció per a ambdues cadenes (pot ser de valor 0) de tal manera que el *hash* del bloc de la moneda que suporta *Merged mining* estigui inclòs en la transacció.

A continuació el miner busca la solució per al *proof of work* del bloc de la cadena principal. Ara poden passar tres coses (suposant que la dificultat de la cadena alternativa és inferior al de la cadena principal):

Si el *hash* trobat és superior al *target* d'ambdues cadenes no passa res, s'ha de canviar el *nonce* i intentar de nou.

Si el *hash* trobat és inferior al *target* de la cadena alternativa però no al de la cadena principal, la cadena alternativa accepta el *hash* trobat com a *proof of work* i el bloc s'inclou en la cadena de blocs alternativa. El *hash* trobat és acceptat com a *proof of work* ja que el bloc conté les capçaleres dels blocs de la cadena principal i de l'alternativa (en la transacció), assegurant que s'ha realitzat la feina necessària per a generar el bloc.

Si el *hash* trobat és inferior als *targets* de les dues cadenes, ambdues accepten el bloc i s'han generat dos blocs de dues cadenes diferents a la vegada.

S'ha de recalcar que la cadena principal ignora la transacció que conté el *hash* del bloc de la cadena alternativa, assegurant que les dues cadenes es mantenen independents.

Capítol 6

Configuració d'una mining pool

En aquest capítol s'explicarà pas a pas com muntar una *mining pool*, així com configurar el software de minería per a poder connectar-s'hi, per a poder minar bitcoins i els passos que s'haurien de canviar si es vol minar una altre moneda, els passos marcats en vermell s'han de canviar segons la moneda que volem utilitzar.

Els passos inclosos en aquest capítol són específics del sistema operatiu *Ubuntu*, tot i que es poden utilitzar altres sistemes operatius. Totes les comandes incloses en aquest capítol han set executades utilitzant la consola d'*Ubuntu*.

6.1 Requeriments

Els requeriments de hardware per a poder muntar una *mining pool* no són elevats, els requisits recomanats són:

- Sistema operatiu: no hi ha requeriments sobre el sistema operatiu a usar, tot i que en aquesta implementació s'ha utilitzat la versió d'*Ubuntu* 14.04.
- Com a mínim 1 GB de RAM.
- Idealment es necessiten dos processadors, sobretot si es vol que la *mining pool* sigui pública.
- Suficient disc dur com per a guardar tota la cadena de blocs de la moneda que es vol minar i la informació dels miners.
- Una IP estàtica per a que els miners es puguin connectar.

Els requeriments per a minar varien de moneda en moneda, però es pot trobar software que et permet minar amb CPU, GPU o altres hardware específics, de manera que es pot minar amb pràcticament qualsevol aparell, tot i que si es vol minar de manera eficient es necessita GPU o hardware específics. Per a seleccionar el hardware amb el que es vol minar, es pot mirar una taula de rendiment.

6.2 Configuració prèvia

Abans de començar amb la instal·lació de la mining pool, necessitam preparar el sistema i instal·lar el software que necessitarem durant els diferents passos del muntatge. El que es farà en aquesta secció és comú per a la majoria de les monedes basades en els bitcoins.

Primer de tot necessitarem una base de dades per tal de poder guardar la diversa informació que la *mining pool* necessita, com informació dels miners participants, *shares*, rondes, etc.

En un dels passos següents instal·larem un *framework* que ens permetrà configurar i afegir miners a la *pool*. Aquest *framework* s'hi accedirà per http, així que per a que funcioni necessitarem tenir instal·lat Apache i php.

Tot això es pot instal·lar en Ubuntu amb la comanda:

```
sudo apt-get install lamp-server^
```

Ara haurem de crear una base de dades i un usuari per a poder connectar-s'hi. Primer hem d'obrir la terminal de *mysql*:

```
mysql -u root -p
```

Una vegada dins la terminal, crearem una base de dades i un usuari amb permisos per a connectar-s'hi:

```
CREATE DATABASE mpos
CREATE USER 'dbuser'@'localhost' IDENTIFIED BY 'dbpassword';
GRANT ALL PRIVILEGES ON * . * TO 'usuari'@'localhost';
```

Posteriorment s'haurà de carregar l'estructura de la base de dades. Aquesta estructura ve inclosa en el *Framework* que s'utilitzarà, així que es detallarà com fer-ho en l'apartat del *Framework*.

Es hora d'instal·lar totes les llibreries que es necessitaran posteriorment. Totes elles es poden instal·lar usant l'*apt-get* d'Ubuntu.

- python-twisted
- python-mysqldb
- python-dev
- python-setuptools
- python-memcache
- python-simplejson
- python-pylibmc
- memcached
- php5-memcached
- php5-mysqlnd
- php5-curl
- php5-json
- libapache2-mod-php5

6.3 Wallet

El següent pas és instal·lar una *Wallet*. Les *Wallets* tenen diverses funcions:

Una *wallet* es connecta a altres *wallets* per a obtenir la cadena de blocs i les transaccions que encara no han sigut validades. Aquestes exerciten la funció de nodes en la xarxa dels bitcoins. La *mining pool* estarà contínuament connectada a la *wallet* per a poder obtenir informació actualitzada, que necessita per a poder generar nous blocs i enviar-los una vegada s'ha trobat un *hash* correcte.

Les *wallets* també permeten generar i guardar les claus públiques i privades de les adreces de bitcoins.

Finalment també ens permeten crear i gestionar transaccions, veure el balanç de les adreces i obtenir diversa informació de la cadena de blocs.

Cada moneda utilitza una *wallet* diferent. En el nostre cas hem d'instal·lar la *wallet* dels bitcoins, que es pot aconseguit a través de la pàgina `https://bitcoin.org/es/descargar`. Si es vol usar una altra moneda ens haurem de descarregar la Wallet específica per aquesta moneda.

Per a configurar la *wallet* per a poder conectar-s'hi des de la *pool* hem de crear un arxiu de configuració. En el cas de linux, hem de crear l'arxiu "∼/.bitcoin/Bitcoin.conf". La carpeta i el nom de l'arxiu de configuració són diferents per a cada moneda.

Per a la configuració sencera de la *wallet* es pot consultar la pàgina `https://en.bitcoin.it/wiki/Running_Bitcoin`.

Per a una configuració bàsica només cal afegir 6 línies a l'arxiu:

```
server=1
gen=0
rpcport=8332
rpcallowip=127.0.0.1
rpcuser=nomUsuari
rpcpassword=contrasenyaUsuari
```

Si en comptes de fer servir la cadena principal es vol utilitzar la cadena de testnet, només s'ha de canviar el port del rpc i afegir una línia:

```
rpcport=18332
testnet=1
```

Ara només queda instal·lar la wallet. Per fer-ho s'ha d'anar dins la carpeta "src" i executar la comanda:

```
make -f makefile.unix
```

Finalment s'executarà la *wallet* per a començar a baixar la cadena de blocs, ja que fins que no s'hagi baixat sencera, no es pot començar a minar. Aquesta ha d'estar en execució sempre que vulguem engegar la *mining pool*. Per executar-ho en background farem servir la comanda:

```
bitcoind -daemon
```

I per a aturar el programa:

```
bitcoind stop
```

6.4 Mining Pool

Primer de tot es necessita el codi d'una *mining pool*. En aquesta implementació utilitzarem una *pool* que utilitza el protocol *stratum* [12], això significa que els miners que s'utilitzin per a minar en aquesta *pool* han de funcionar amb aquest protocol, tot i que hi ha maneres d'utilitzar miners amb altres protocols com el *Getwork* [8] utilitzant un *Stratum mining proxy*, que farà d'intermediari entre els diferents protocols.

Hi ha diverses implementacions de *stratum mining pool*, la majoria són semblants i requereixen una configuració similar. La que es farà servir en aquesta implementació és el *stratum mining server* que es troba en `https://github.com/Crypto-Expert/stratum-mining` i que podem obtenir amb la comanda:

```
git clone https://github.com/Crypto-Expert/stratum-mining/
```

Ara s'ha de configurar el servidor. Hi ha una copia d'exemple de l'arxiu de configuració dins la carpeta "conf". Per a generar l'arxiu de configuració, s'ha d'executar la següent comanda després de situar-nos dins aquesta carpeta:

```
cp config_sample.py config.py
```

Abans de començar amb la configuració es necessita una adreça de la *Wallet*. Per a obtenir aquesta adreça s'ha d'engegar la *Wallet* de la manera en que s'explica en l'apartat anterior i després executar:

```
bitcoind getaccountaddress ""
```

Ara s'ha de modificar l'arxiu de configuració "config.py" i modificar les línies següents:

```
12: CENTRAL_WALLET = {Account address obtinguda
        en el pas anterior}
14: COINDAEMON_TRUSTED_HOST = 'localhost'
15: COINDAEMON_TRUSTED_PORT = {Wallet RPC port}
16: COINDAEMON_TRUSTED_USER = {Wallet RPC user}
```

```
17: COINDAEMON_TRUSTED_PASSWORD = {Wallet RPC password}
25: COINDAEMON_ALGO = sha256d
70: HOSTNAME = {DNS o IP que es te assignada
        en el servidor}
87: PASSWORK_SALT = '{Cadena de caracters aleatoria}'
102: DB_MYSQL_HOST = 'localhost'
103: DB_MYSQL_NAME =  'mpos'
104: DB_MYSQL_USER = 'dbuser'
105: DB_MYSQL_PASS = 'dbpassword'
```

Finalment, hem d'instal·lar el servidor stratum. Per a descarregar-lo i instal·lar-lo executem:

```
git clone https://github.com/ahmedbodi/stratum.git
cd stratum
sudo python setup.py install
```

Per a poder engegar el servidor encara és necessari carregar l'estructura de la base de dades. Una vegada la tinguem, haurem de col·locar-nos dins la carpeta de la *mining pool* "stratum-mining" i executar la comanda:

```
twistd -ny launcher.tac
```

6.5 Framework (MPOS)

El Mining Portal Open Source (MPOS) és una interfície basada en web que permet configurar els miners i el servidor d'una *mining pool*, i és compatible amb diversos tipus de monedes.

Com està preparat per accedir-hi per *http*, és aconsellable instal·lar aquest framework en la carpeta a on estigui apuntant el fitxer php.ini. Per defecte:

```
cd /var/www
sudo git clone git://github.com/MPOS/php-mpos.git MPOS
```

Ara que ja tenim el framework, podem carregar l'estructura de la base de dades. L'arxiu .sql per a poder fer-ho es troba dins la carpeta "MPOS/sql":

```
sudo mysql -u mpos -p mpos < sql/000_base_structure.sql
```

Per a configurar el *MPOS*, s'ha de copiar l'arxiu "global.inc.dist.php" dins la carpeta "MPOS/include/config" i reanomenar-lo "global.inc.php":

```
cp /var/www/MPOS/include/config/global.inc.dist.php
/var/www/MPOS/include/config/global.inc.php
```

Per a una configuració bàsica s'ha d'obrir l'arxiu que acabem de generar i modificar les línies següents:

```
24: $config['SALT'] = 'Cadena de caracters aleatoria';
25: $config['SALTY'] = 'Cadena de caracters aleatoria';
32: $config['algorithm'] = 'sha256d';
46: $config['db']['host'] = 'localhost';
47: $config['db']['user'] = 'dbuser';
48: $config['db']['pass'] = 'dbpass';
49: $config['db']['port'] = 3306;
50: $config['db']['name'] = 'mpos';
57: $config['wallet']['type'] = 'http';
58: $config['wallet']['host'] =
        'localhost:{Wallet rpc port}';
59: $config['wallet']['username'] = '{Wallet rpc user}';
60: $config['wallet']['password'] =
        '{Wallet rpc password}';
82: $config['gettingstarted']['coinname'] = 'Bitcoin';
83: $config['gettingstarted']['coinurl'] =
        'http://www.bitcoin.org';
84: $config['gettingstarted']['stratumurl'] =
        '{IP del servidor}';
85: $config['gettingstarted']['stratumport'] = '3333';
94: $config['price']['target'] = '/api/2/btc_eur/ticker';
131: $config['currency'] = 'BTC';
170: $config['payout_system'] = 'pplns';
        //Rewarding System a usar;
208: $config['fees'] = 0; //Quotes aplicades als usuaris
```

Ara que el framework està instal·lat, per a accedir-hi s'ha d'usar un navegador i anar a l'adreça "{IP}/MPOS/public", substituint {IP} per la IP del servidor que s'està utilitzant.

6.6 Miner

6.6.1 Solo mining

Normalment els clients de mineria serveixen tant per a *solo mining* com per a *pooled mining*. El miner que utilitzarem és un client anomenat *CGMiner*.

Aquest és un miner per a bitcoins altament configurable, sense interfície gràfica i amb versions per a linux, mac i windows. Les darreres versions d'aquest miner les podem trobar en la següent pàgina: `http://ck.kolivas.org/apps/cgminer/`. Si es vol un miner amb interfície gràfica, es pot usar el *GUIMiner*, que és bàsicament el *CGMiner* amb una interfície a sobre.

Per desgràcia, les versions d'aquest miner a partir de la 3.8 no són compatibles per a minar amb GPU ja que és ineficient per a minar bitcoins, així la última versió que suporta mineria amb GPU és la versió 3.7.2, que podem trobar per a windows en `http://cryptomining-blog.com/tag/cgminer-3-7-2-windows-download/` i per a linux en `http://www.tuspmtech.com/ltc/cgminer-3.7.2-x86_64-built.tar.bz2`.

A més, per a minar amb GPU és recomanat descarregar-se els últims drivers de la targeta gràfica que usarem.

Per a poder fer solo mining, necessitarem tenir la Wallet corresponent a la moneda a usar en la mateixa màquina que el miner.

Una vegada que tenim la wallet funcionant i el miner descarregat, hem de situar-nos amb la terminal en la carpeta del miner i, en el cas de windows, executar la comanda:

```
cgminer -o stratum+tcp://127.0.0.1:{Wallet port} -u
        {RPC Wallet user} -p {RPC Wallet password}
```

A aquesta comanda se li poden afegir opcions per a millorar el rendiment. Aquestes opcions depenen del hardware amb el que s'està minant. Per a una explicació detallada d'aquests paràmetres podem consultar la pàgina següent: `http://www.minedogecoin.com/tag/cgminer-parameters-explained/`.

Per a modificar aquestes opcions i les *mining pools* en les que volem minar també es pot

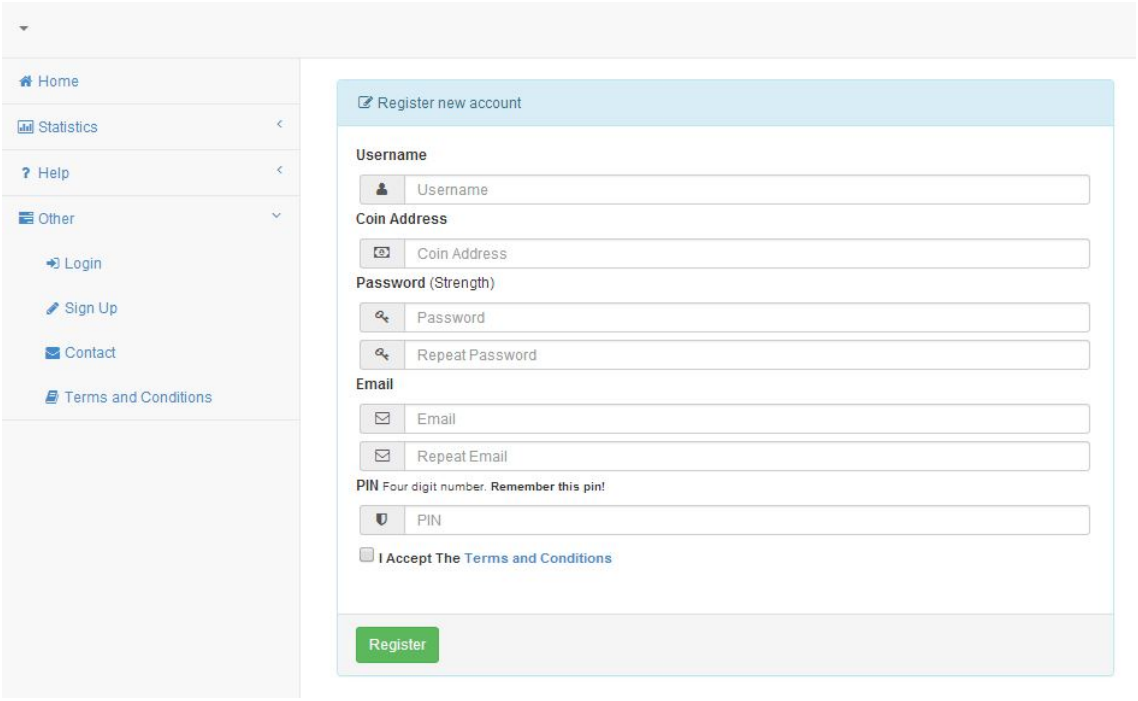

Figura 6.1: Pàgina del framework per a registrar un compte

configurar en l'arxiu "cgminer.conf".

Per a saber les millors opcions per al hardware i els rendiments que podem esperar per al miner que s'esta usant hem de consultar una llista de comparació de software, com la que podem trobar en `https://en.bitcoin.it/wiki/Mining_hardware_comparison`.

6.6.2 Pooled mining

Independentment del *miner*, la *mining pool* o la moneda que es vol minar, els primers passos per a poder connectar-nos a una *mining pool* són sempre els mateixos.

Primer de tot, ens haurem de registrar a la pàgina de la mining pool i crear un *worker*. Explicarem com fer això amb el framework que hem instal·lat en la secció anterior.

Per a accedir a aquest framework usarem qualsevol navegador i introduirem la IP de l'ordinador on l'hem instal·lat i una vegada estem en la pàgina web, accedim a la secció "Other/Sing Up". Hauríem de veure una pantalla semblant a la Figura 6.1. Una vegada ens hem registrat hauríem de rebre un email de confirmació, tot i que no rebrem res si no hem configurat el servidor per a poder enviar emails prèviament i a més el framework per defecte no requereix confirmació.

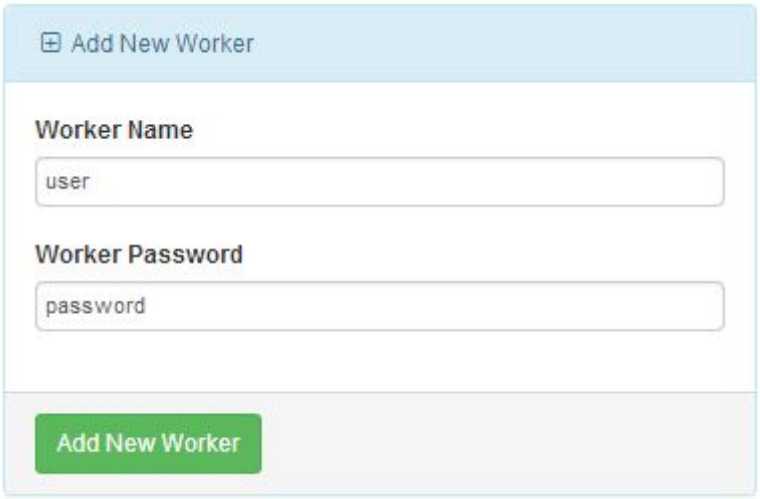

Figura 6.2: Pàgina del framework per a registrar un worker

Ara hem de fer Login amb aquest compte i entrar en la secció "My Account/My Workers" i registrar un treballador (Figura 6.2). Si es vol minar amb diverses màquines, es recomana registrar un treballador per màquina per a poder monitoritzar totes les màquines per separat. Una vegada fet tot això, ja podem connectar-nos a la *mining pool* amb el miner. Usarem el mateix miner que hem usat per a *solo mining*, però per a utilitzar la *mining pool* no necessitam tenir la *Wallet* en la mateixa màquina que el miner, ja que la pool ens enviarà tota la informació que necessitem.

Per a engegar el miner amb la *mining pool* usarem la comanda:

```
cgminer -o stratum+tcp://{Mining Pool Server IP}:
        {Mining Pool Transport Port}
        -u {Account Name}.{Worker Name}
        -p {Worker Password}
```

La resta d'opcions del miner serien les mateixes que les introduïdes per a fer *solo mining*.

6.7 Litecoin Pool

En aquesta secció s'explicaran els canvis necessaris per a crear una *mining pool* per a una altra moneda. Com a exemple, utilitzarem els *litecoins*.

Els requeriments per a crear un servidor per a altres monedes i els passos inclosos en la configuració prèvia són els mateixos que per als bitcoins. Les primeres diferències les

trobem a l'hora d'instal·lar la Wallet.

Com ja s'ha explicat en l'apartat de les Wallet, cada moneda utilitza una Wallet diferent. En el cas dels litecoin la podem obtenir en la pàgina següent: `https://litecoin.org/`. L'arxiu de configuració de la Wallet el trobarem en " ∼/.litecoin/Litecoin.conf" i en aquest cas, la configuració que hem d'introduir en aquest arxiu és la següent:

```
server=1
gen=0
rpcport=9332
rpcallowip=127.0.0.1
rpcuser=nomUsuari
rpcpassword=contrasenyaUsuari
```

Mentre que el port per defecte de la testnet dels litecoin seria el 19332.

Una vegada instal·lada la Wallet, podem usar les següents comandes per a engegar i aturar-la.

```
litecoind -daemon
litecoind stop
```

L'adreça de la Wallet que utilitzarem en la configuració de la pool és diferent a la dels bitcoin, i l'obtindrem amb la comanda:

```
litecoind getaccountaddress ""
```

En quant a la *mining pool*, podem usar la mateixa que per als bitcoins, tot i que abans de tocar l'arxiu de configuració de la *mining pool* haurem d'instal·lar l'script que utilitza la moneda per a generar els *hashes* del *proof of work*. En el cas dels litecoin, l'script que s'utilitzarà ve inclòs en les carpetes que ens hem baixat amb la *mining pool*.

Aquest script el poden trobar en "stratummining/externals/litecoin_scrypt" i per a instal·lar-lo, ens hem de situar en aquesta carpeta i executar la comanda:

```
sudo python setup.py install
```

Amb aquest script instal·lat, la configuració de la *mining pool* és pràcticament la mateixa que per als bitcoins, les úniques línies de la configuració diferents són:

```
12: CENTRAL_WALLET = {Account address obtinguda
```

```
        en el pas anterior}
15: COINDAEMON_TRUSTED_PORT = 9333
25: COINDAEMON_ALGO = scrypt
```

La instal·lació i configuració del framework és molt similar també. L'estructura de la base de dades és la mateixa, l'únic que s'ha de canviar són unes quantes línies de l'arxiu de la configuració:

```
32: $config['algorithm'] = 'scrypt';
58: $config['wallet']['host'] =
        'localhost:9332';
59: $config['wallet']['username'] = '{Wallet rpc user}';
60: $config['wallet']['password'] =
        '{Wallet rpc password}';
82: $config['gettingstarted']['coinname'] = 'Litecoin';
83: $config['gettingstarted']['coinurl'] =
        'https://litecoin.org/es/';
85: $config['gettingstarted']['stratumport'] = '3333';
94: $config['price']['target'] = '/api/2/btc_eur/ticker';
131: $config['currency'] = 'LTC';
```

Finalment, necessitem un miner capaç de minar la moneda desitjada. En el cas dels litecoin, podem utilitzar el *CGMiner*, ja que ens permet minar usant diferents scripts. La versió del *GUIMiner* per a minar litecoin s'anomena *GUIMiner-alpha*.

Els litecoin es minen utilitzant GPU, així que la versió del *CGMiner* que necessitam és la 3.7.2 (com ja s'ha indicat anteriorment, les versions més noves no són compatibles amb la mineria amb GPU).

La configuració recomanada per al miner per als litecoin la trobarem en `https://litecoin.info/Mining_hardware_comparison`, així només queda engegar el miner, que ho farem amb la comanda:

```
cgminer --scrypt -o stratum+tcp://{Mining Pool Server IP}:
        {Mining Pool Transport Port}
        -u {Account Name}.{Worker Name}
        -p {Worker Password}
```

6.8 Creació d'una mining pool

Per a posar en pràctica tot el descrit en aquest capítol, he configurat un ordenador per a que faci de servidor d'una *mining pool*, mentre que he usat un altre ordenador amb un programa de mineria per a connectar-me a aquesta *mining pool*.

L'objectiu és el de crear una *mining pool* que funcioni per a poder provar i aprendre el funcionament de la mineria de les monedes virtuals i la gestió bàsica de les *mining pools*. Per a això, la moneda que intentaré minar amb aquesta *pool* seran bitcoins de la cadena de *Testnet*.

L'ordenador que farà de servidor és un ordenador portàtil amb el sistema operatiu Ubuntu 14.04. La *mining pool* que crearé no està pensada per a ser oberta al públic, així que el servidor no haurà de gestionar un gran nombre de connexions, per tant no hi haurà problemes en usar un ordenador portàtil com a servidor.

Per a crear la *mining pool*, primer he instal·lat les llibreries descrites en la secció de la configuració prèvia. Després he creat una base de dades tal i com descrit.

```
mysql -u root -p
```

```
CREATE DATABASE mpos
CREATE USER 'joan'@'localhost' IDENTIFIED BY
        'joanmpospassword1234';
GRANT ALL PRIVILEGES ON * . * TO 'joan'@'localhost';
```

El pas següent és instal·lar la Wallet, que he obtingut de la pàgina oficial i instal·lat de la manera descrita en la secció corresponent. Com vull minar Bitcoins de la *Testnet*, l'arxiu de configuració de la Wallet que usaré contindrà el següent:

```
testnet=1
server=1
gen=0
rpcport=18332
rpcallowip=127.0.0.1
rpcuser=joan
rpcpassword=aA1!123456
```

I usant la comanda:

```
bitcoind getaccountaddress ""
```

He obtingut l'adreça de la Wallet que faré servir per a rebre la recompensa dels blocs trobats següent: mymdxM8wq88YbciQQX5JXGKqA2VTbsmJBw.

És hora d'instal·lar la *mining pool*. Una vegada fet, l'arxiu de configuració que usaré serà el que s'adjunta a l'annex I.

S'ha de notar que, apart de les línies incloses en la secció on es detalla la instal·lació, he modificat paràmetres extra corresponents a la dificultat del *target* per als *shares*. Això ho he fet degut a que no minaré en la cadena principal, sinó que ho faré en la *Testnet* i a més estaré minant amb una GPU de gama mitjana, així que per a trobar *shares* de manera regular, necessitaré una dificultat baixa.

A més, no disposo d'una IP estàtica, així que la IP inclosa en l'arxiu de configuració és una IP que hauré d'anar canviant cada vegada que el meu *router* m'assigni una IP diferent.

Per a finalitzar la instal·lació del servidor, falta instal·lar el framework. L'arxiu de configuració del framework que hauré d'utilitzar per al meu cas és l'inclòs en l'annex II.

A més, per a no tenir que navegar per les carpetes a l'intentar accedir al framework des del navegador, he modificat l'arxiu "/etc/apache2/sites-available/000-default.conf" per a fer que s'accedeixi directament al framework al introduïr la url del servidor. Per a fer-ho s'ha de modificar el paràmetre "DocumentRoot":

```
DocumentRoot /var/www/MPOS/public
```

Amb això, el servidor està llest per a ser inicialitzat.

Ara prepararé el client per a connectar-me al servidor. Com a client utilitzaré un ordenador amb una targeta gràfica MSI Radeon HD 7750 OC.

Com les últimes versions del *CGMiner* no suporten mineria amb GPU, he utilitzat una versió del *GUIMiner* que si que en suporta. Per a connectar-me al servidor, la pantalla del *GUIMiner* es veurà com en la Figura 6.3.

Per a fer això, abans m'he registrat en el framework, tot i que no és necessari si es configura la *mining pool* per a que registri usuaris automàticament quan intentin connectar-se.

En poc temps es pot observar com *shares* comencen a ser registrats en la base de dades. Per a observar-los he utilitzar la comanda en sql:

```
SELECT id,rem_host,solution,difficulty,time
```

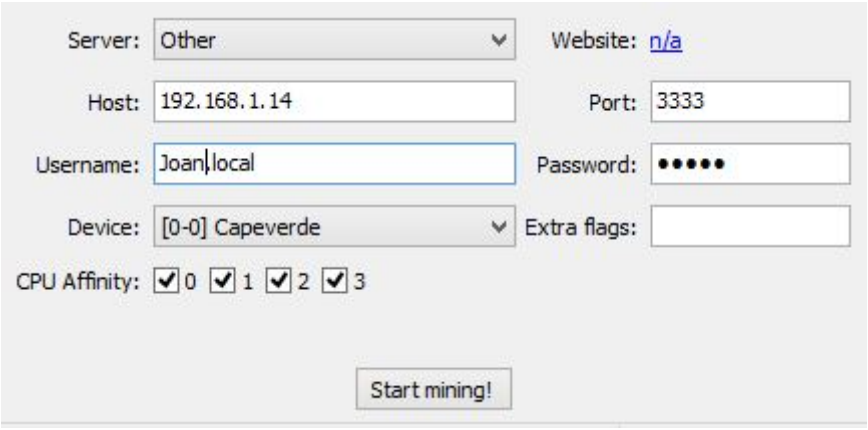

Figura 6.3: Paràmetres del guiminer

197	192.168.1.21	00000000164a6f0c2c644d97c4adf28326ac126f8d32ee5e4b6beac84403f493	10	2014-06-08 13:02:25
196	192.168.1.21	000000000980ef7912f082a46ea84b32f8d0a749fa0e8c083e582fc225fb609b	10	2014-06-08 12:58:10
195	192.168.1.21	00000000082e24b0d942431b65f4ec86f22ee1d850a2b883ed32c831c3365e0f	10	2014-06-08 12:50:20
194	192.168.1.21	00000000130eec18e11006c4ad0564afdc3963e2fab0dd578aa255df7eebc39e	10	2014-06-08 12:45:58
193	192.168.1.21	00000000137a349e54d0cbe7db06d1df7fc84cea59e7f10e20b84fcf6b243366	10	2014-06-08 12:45:28
192	192.168.1.21	000000000b078c52680cf535e5dd1d23ad8e6036ef21cfe84016baadbccd50bd	10	2014-06-08 12:38:20
191	192.168.1.21	0000000013243d207c0b80ac082212ad38328c7a81424d0c139db20cc306710b	10	2014-06-08 12:27:58
190	192.168.1.21	00000000003e1c804ab97d0a33eb1622555c232e9891693186872dd817eb2593	10	2014-06-08 12:21:21
189	192.168.1.21	0000000014b1ed790e8a866a228de9531ee580a0d72f2c3c747c3d8466e77f54	10	2014-06-08 12:21:06

Figura 6.4: Shares registrats en la base de dades

```
        FROM mpos.shares;
```

Els resultats es poden observar en la Figura 6.4.

Considerant que en mitja hora, el meu miner pot aconseguir aproximadament 10 shares amb una dificultat de 10 i que en l'actualitat la dificultat de la *Testnet* és 17409.69354995, cada share té una possibilitat d'aproximadament $\frac{10}{17409} \cong 5'744 * 10^{-4}$ de ser la solució del *proof of work*, així que necessitaré de mitja $\frac{360s}{5'744*10^{-4}} \cong 626740s \cong$ una setmana per a trobar un bloc.

Capítol 7

Conclusions

Els objectius d'aquest projecte eren els aprendre amb detall el funcionament dels Bitcoins i, finalment, de muntar amb èxit una *mining pool* que funcionés. Aquests objectius s'han aconseguit, tot i que no amb tot l'èxit que esperava.

Pensava que al triar una cadena de *Testnet* podria aconseguir minar alguna moneda amb aquesta *mining pool* per a provar que aquesta *pool* realment funciona correctament, però majorment degut a les limitacions de temps i de hardware, això no ha sigut possible ja que en l'actualitat trigaria de mitja aproximadament una setmana contínua tenint el miner funcionant de manera constant per a aconseguir-ho.

A més, ja que no he aconseguit minar res i que només disposo d'un sol ordenador per a minar no he pogut posar en pràctica els diferents *rewarding systems* per a comprovar la seva eficàcia.

Tot i això, estic satisfet amb el que he aconseguit amb aquest projecte. He aprés el funcionament de la mineria dels bitcoins i la manera de configurar correctament una *mining pool* per a minar qualsevol tipus de moneda.

7.1 Possibles ampliacions

Aquest projecte es pot ampliar de diverses maneres.

Tal i com està ara, la *mining pool* només pot minar Bitcoins de la *Testnet*. Una de les possibles ampliacions seria permetre que la *mining pool* pogués minar diversos tipus de monedes i que cadascun dels miners registrats puguin triar quina moneda volen minar de les disponibles.

Això suposaria la instal·lació de diverses Wallets i *mining pools* en el servidor, cadascu-

na escoltant en un port diferent, a més requeriria una modificació del Framework per a permetre crear usuaris i treballadors per a cadascuna de les diferents monedes.

Una altre de les possibles ampliacions seria la de crear un cluster d'ordinadors amb un software de mineria instal·lat per a connectar-los a tots a la *mining pool* i comprovar com funciona i els resultats obtinguts aplicant diversos *rewarding systems*

Bibliografia

[1] Mining pool de slush. https://mining.bitcoin.cz/. Últim accès: 14/06/2014.

[2] Pàgina oficial dels dogecoin. http://dogecoin.com/. Últim accès: 14/06/2014.

[3] Pàgina oficial dels litecoin. https://litecoin.org/es/. Últim accès: 14/06/2014.

[4] Pàgina oficial dels mastercoins. http://www.mastercoin.org/. Últim accès: 14/06/2014.

[5] Pàgina oficial dels namecoin. http://namecoin.info/. Últim accès: 14/06/2014.

[6] Pàgina oficial dels peercoins. http://www.peercoin.net/. Últim accès: 14/06/2014.

[7] Autor desconegut. Descriptions of sha-256, sha-384, and sha-512. http://csrc.nist.gov/groups/STM/cavp/documents/shs/sha256-384-512.pdf. Últim accès: 14/06/2014.

[8] Autor desconegut. Getwork protocol. https://en.bitcoin.it/wiki/Getwork, agost 2013. Últim accès: 14/06/2014.

[9] Cameron F. Kerry. Digital signature standard. http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf, juliol 2013. Últim accès: 14/06/2014.

[10] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. http://bitcoin.org/bitcoin.pdf, 2008. Últim accès: 14/06/2014.

[11] Meni Rosenfeld. Analysis of bitcoin pooled mining reward systems. https://bitcoil.co.il/pool_analysis.pdf, 2011. Últim accès: 14/06/2014.

[12] Slush. Stratum mining protocol. `http://mining.bitcoin.cz/stratum-mining/`. Últim accès: 14/06/2014.

Firmat: Joan Marí Yern

Bellaterra, juny de 2014

Resum

El bitcoin és una moneda virtual descentralitzada que es basa en la criptografia per al seu correcte funcionament. S'utilitza la criptografia en diversos processos de la moneda: creació de comptes, validació de transaccions, generació de moneda, etc.

La mineria és el procés mitjançant el qual les transaccions són validades i a la vegada també és el procés amb el qual es genera nova moneda. Aquest projecte està centrat en l'anàlisi d'aquest procés.

Resumen

El bitcoin es una moneda virtual descentralizada que se basa en la criptografía para su correcto funcionamento. Se utiliza la criptografía en varios procesos de la moneda: creación de cuentas, validación de transacciones, generación de moneda, etc.

La minería es el proceso mediante el qual las transacciones son validadas i a la vez también es el proceso con el qual se genera nueva moneda. Este proyecto esta centrado en el análisis de este proceso.

Abstract

Bitcoin is a decentralized virtual currency that is based on cryptography to ensure it functions properly. Cryptography is used in multiple processes of the currency: account creation, transaction validation, coin generation, etc.

Mining is the process by witch transactions are validated and it is also the process to generate new currency. This project is centered in this process analysis.