Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Luka Sedmak

Računanje kriptografskih valut z GPE

diplomsko delo
na univerzitetnem študiju

Mentor: doc. dr. Tomaž Dobravec

Ljubljana, 2014

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **LUKA SEDMAK**
Naslov: **RAČUNANJE KRIPTOGRAFSKIH VALUT Z GPE**
Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Kriptografske valute niso zanimive samo iz vidika vzpostavitve neodviskega finančnega mehanizma, pač pa ponujajo tudi veliko število odprtih raziskovalnih vprašanj s področja uporabe kriptografskih funkcij in s področja izbire primerne strojne opreme za izvajanje zahtevnih računskih operacij. V diplomski nalogi preglejte in opišite področje kriptografskih valut. Opišite pomen algoritma SHA-256 pri računanju valute Bitcoin in njegove slabosti zaradi enostavnosti paralelizacije ter predstavite algoritem Scrypt, ki te pomanjkljivosti odpravlja. Natančno predstavite način rudarjenja, ki temelji na izvajanju Scrypt kriptografskega algoritma. Izdelajte računalniški sistem, ki bo dovolj zmogljiv za konkurenčno računanje kriptografskih valut. Sistem naj vsebuje grafične kartice, ki omogočaje paralelno računanje (GPE). Sistem opremite s primernim operacijskim sistemom in programsko opremo. S pomočjo testov poiščite optimalno nastavitev parametrov grafičnih kartic pri računanju kriptografske valute.

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Luka Sedmak, z vpisno številko **63010128**, sem avtor diplomskega dela z naslovom:

Računanje kriptografskih valut z GPE

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Tomaža Dobravca,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 30. junija 2014 Podpis avtorja:

Zahvaljujem se svojim staršem, ki mi vsa ta leta stojijo ob strani in me podpirajo pri vseh podvigih, katerih se lotim ter punci Mateji za potrpežljivost in pomoč. Zahvaljujem se tudi svojemu mentorju, dr. Tomažu Dobravcu, ki je pokazal zanimanje za to akademskemu računalništvu relativno neznano temo in mi pomagal pri koordinaciji dela.

Kazalo

Povzetek

Na področju sodobnih financ in bančništva se je razvila močna koncentracija kapitala v peščici finančnih središč. Kot alternativa takšni centralizaciji so se pojavile umetne deregulirane ter popolnoma distribuirane kriptografske valute. Njihova zasnova deluje na principu peer-to-peer omrežja, v katerem se s kriptografskimi funkcijami skupinsko nadzoruje transakcije in ustvarjanje same valute, kar omogoča transparentnost in hkrati anonimnost ter varnost. Za potrebe diplomskega dela smo sestavili napravo, ki z izkoriščanjem velike količine ter hitrosti pomnilnika na grafičnih karticah s posebno programsko opremo potrjuje transakcije v omrežju in nam s tem ustvarja majhen delež valute za nagrado. Opisali bomo lastnosti različnih kriptovalut, razložili metode maksimizacije dobička ter opisali uporabljeno strojno opremo. Podrobno bomo predstavili algoritem potrjevanja transakcij in prikazali vpliv konfiguracijskih parametrov na končno hitrost preračunavanja ter za konec povzeli ugotovitve.

Ključne besede: kriptografske valute, kriptografske funkcije, anonimnost, transparentnost, distribuirana valuta, dereguliranost, bitcoin, scrypt.

Abstract

In the field of modern finance a concentration has developed in a handful of financial institutions. As an alternative to such centralization, deregulated and fully distributed synthetic currencies were introduced. They are designed as a peer-to-peer network, where through use of cryptographic functions, transactions and creation of the currency are controlled, which allows for full transparency along with anonymity and security. For purposes of this thesis we have assembled a device that uses large amounts of fast memory on video cards by running special software which confirms transactions in the network, gaining us a small proportion of the currency as a reward. We will describe characteristics of various cryptocurrencies, explain methods for maximizing profit and describe the hardware used. We will explain in detail the algorithm behind the validation of transactions, show the influence of configuration parameters on resulting hashing speed and finally draw conclusions accordingly.

Keywords: cryptographic currency, cryptographic functions, anonymity, transparency, distributed currency, deregulation, bitcoin, scrypt.

1 | Uvod

Dandanes, v sodobnem svetu, se je ekonomska situacija sveta obrnila v smer popolnoma elektronskih tokov denarja ter vrednostnih lastnin. Zgodovinsko smo ljudje iz blagovne menjave fizičnih dobrin najprej prešli na monetarni sistem z denarno menjavo v obliki srebrnih in zlatih kovancev, nato pa so se začele formirati prve banke. Z njimi so se prvič v zgodovini pojavile tudi razne denarne mahinacije kot so dvojno računovodstvo, siva emisija denarja, financiranje vojn vladarjem skozi javni dolg, monetizacija javnega dolga in podobno. Leta 1611 je bila v Amsterdamu ustanovljena prva delniška borza na svetu, katere osnovni mehanizmi trgovanja so v uporabi še dandanes, skupaj s podobnimi finančnimi produkti in manipulacijami cen delnic ter celotnih trgov. Tako so bili tu postavljeni pogoji za novo vlogo finančnega kapital, katerega vpliv se je nato širil naprej po svetu.

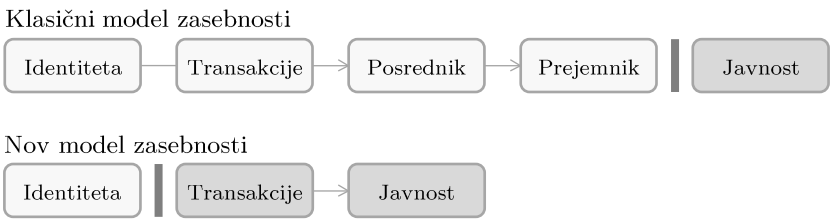
Sčasoma so se iz posameznih evropskih finančnih strokovnjakov ter bankirskih družin formirale močne finančne elite, ki so s svojim znanjem in vplivom pritiskale na svetovne odločitve vlad in sodelovale pri pisanju finančne ter davčne zakonodaje, ki je delovala njim v korist. Hkrati s centralizacijo svetovne moči v kapitalski sferi, se je na krilih tehnologije v zelo kratkem času progresivno razvila tudi popolna digitalizacija celotnega bančnega poslovanja. Kombinacija vsega tega je privedla do situacije, ko je moč nad koordiniranjem praktično celotnega svetovnega kapitala v realnem času skoncentrirana v rokah nekaj finančnih inštitucij in ko je denar ter ravnanje z njim postal popolnoma abstrakten pojem. Pojavila se je potreba po decentralizaciji bančništva in želja po sistemu, ki bi se uravnaval sam brez kakršnekoli zunanje avtoritete. Ena od idej v tej smeri se je prvič pojavila že leta 1998 na Cypherpunks mailing listi v obliki umetne valute, ki bi s kriptografijo nadzirala svojo kreacijo ter vse transakcije. Po mnogih razpravah je deset let kasneje Satoshi

Nakamoto izdal tehnično specifikacijo ter dokaz koncepta prve prave kriptografske valute. Rodil se je Bitcoin.

V diplomski nalogi bomo v poglavju 2 opisali ideološko ter tehnično ozadje kriptografskih valut, vrste le-teh, njihovo odpornost na napade ter njih pridobivanje oziroma rudarjenje s pomočjo potrjevanja transakcij. Nato si bomo v poglavju 3 podrobneje pogledali strojno opremo posebej prilagojeno za takšno potrjevanje transakcij, njene probleme s porabo energije ter hlajenjem in optimizacijo teh vidikov. Poglavje 4 bo podrobneje predstavilo sam algoritem uporabljen pri izračunavanju, programsko opremo, ki jo naša naprava uporablja ter natančno konfiguracijo s primerjavami rezultatov. V poglavju 5 bomo razložili tendence povezovanja računske moči v bazene, predstavili borze kriptografskih valut ter način maksimizacije dobička. V zaključnem poglavju naredimo povzetke ugotovitev, ugotovimo trenutno stanje v svetu kriptografskih valut ter njihov potencial za prihodnost.

2 | Kriptografske valute

V želji po neodvisnosti finančnega prometa od bank ter ostalih finančnih inštitucij so se, kot eden od mehanizmov za dosego cilja, razvile kriptografske valute. Idejna zasnova le-teh deluje na principu peer-to-peer elektronskega denarja, ki omogoča plačevanje neposredno med uporabniki mimo sistema finančnih inštitucij. Glavna knjiga vseh transakcij se sproti osvežuje pri vseh uporabnikih naenkrat in se kodira na način, da onemogoča reverzibilnost transakcij. Kriptografija z uporabo kombinacije zasebnih ter javnih ključev pa omogoča varnost ter identifikacijo uporabnikov, katerih javni ključ je seme za izračun unikatnega naslova njihove spletne denarnice. Tak sistem zagotavlja decentralizacijo, saj je distribuiran med vsemi uporabniki omrežja, hkrati s popolno anonimnostjo pa tudi transparentnost transakcij, ki niso reverzibilne in so vse zapisane v skupni glavni knjigi. Ena od ključnih prednosti in hkrati velik trn v peti svetovnim oblastem je odsotonost regulacije kakršnihkoli finančnih ali vladnih avtoritet, ki je onemogočena zaradi anonimne in distribuirane narave sistema. Slika 2.1 prikazuje razliko med pogledoma na zasebnost pri klasičnem bančnem poslovanju in novem pristopu internetnih valut.

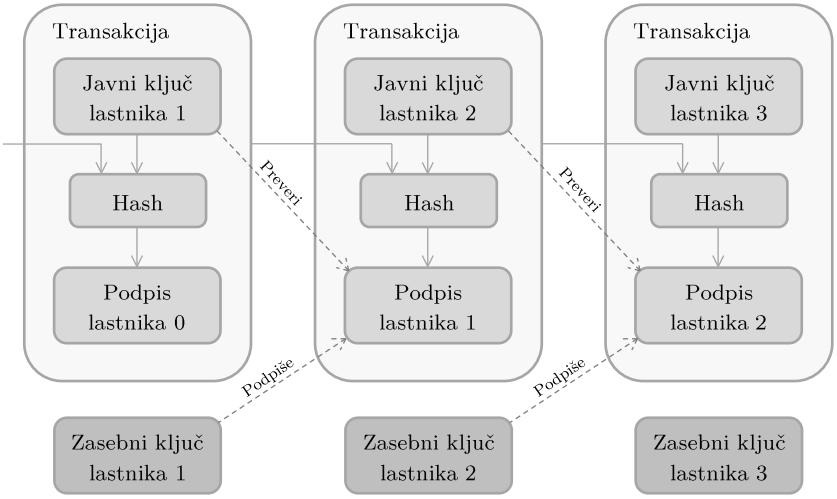


Slika 2.1: Ideološki pristop k zasebnosti

Z vidika varnosti ima sistem podobne teoretične ranljivosti kot tehnologije, ki jih implementira. Možni so specializirani DDoS napadi, Sybil napad z izolacijo ene točke od omrežja s pomočjo množice odjemalcev enega uporabnika, prestrezanje prometa v domačem omrežju, zloraba več kot 50% deleža računske moči v celotnem omrežju, poplavljanje omrežja z umetnimi transakcijami, razbitje kriptografskih funkcij s pomočjo kvantnih računalnikov in podobni teoretični vektorji napada. Kljub temu v realnosti večina od teh ranljivosti obstaja zgolj na papirju, saj so iz vidika okoriščanja napadalcev nerentabilne. Največ, kar lahko dosežejo je dvojno trošenje svoje valute in še to le pri trgovcih, ki ne počakajo na potrditev transakcije s strani verige blokov, temveč valuto takoj tretirajo kot prejeto. Tako še vedno ostaja največja nevarnost nepazljivost samih uporabnikov, ki privede do takšne ali drugačne kraje kriptografske denarnice.

2.1 Bitcoin

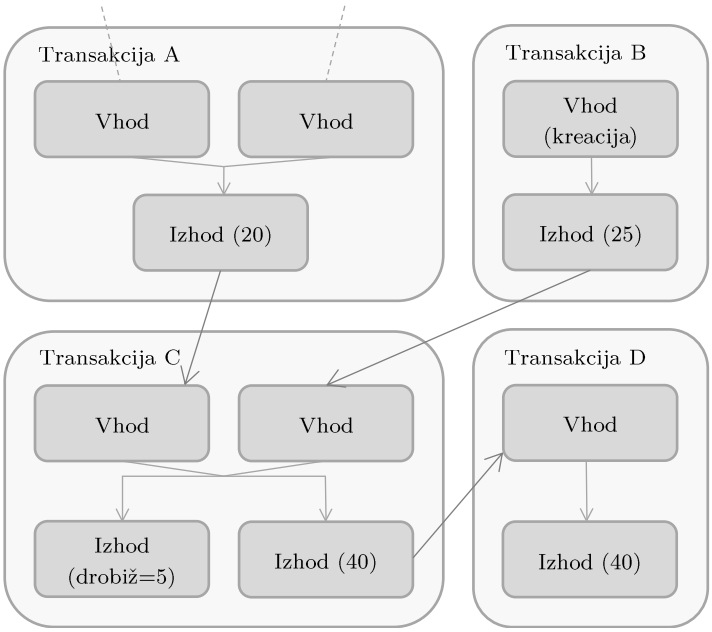
Glavna ter najbolj svetovno znana kriptografska valuta je Bitcoin. Kot prva formalna valuta svoje vrste je od leta 2009 do danes prebrodila nemalo težav, vendar se je zaradi pametnih potez razvijalcev in hitrih popravkov uspela obdržati in pridobiti na razpoznavnosti. Tehnično ozadje Bitcoina v osnovi temelji na blockchainu, glavni knjigi v kateri se verižno nalagajo bloki potrjenih transakcij ter na digitalnem podpisovanju transakcij. Elektronska valuta je definirana v obliki verige digitalnih podpisov [6], oziroma drevesa vseh transakcij med lastniki v času od kreacije do zadnje transakcije, sama po sebi kot entiteta pa ne obstaja. Vsaka transakcija vsebuje številko pošiljatelja, prejemnika, znesek poslane valute ter nekaj kontrolnih parametrov. Pošiljatelj prenese valuto prejemniku tako, da hash ene od prejšnjih transakcij poslanih njemu, skupaj z javnim ključem prejemnika digitalno podpiše s svojim privatnim ključem in jo pošlje naprej. Potek podpisovanja transakcij je prikazan na sliki 2.2.



Slika 2.2: Podpisovanje transakcij

Ker so transakcije atomarne se zneskov v njih ne da razdrobiti. Tako se v primeru, da je poslani znesek manjši kot tisti iz prejšnje transakcije, v novi ustvarita dva prejemnika: tisti, kateremu želimo denar poslati ter pošiljatelj sam, kamor se vrne preostanek denarja oz. 'drobiž'. Podobno se v primeru, ko je znesek ve-

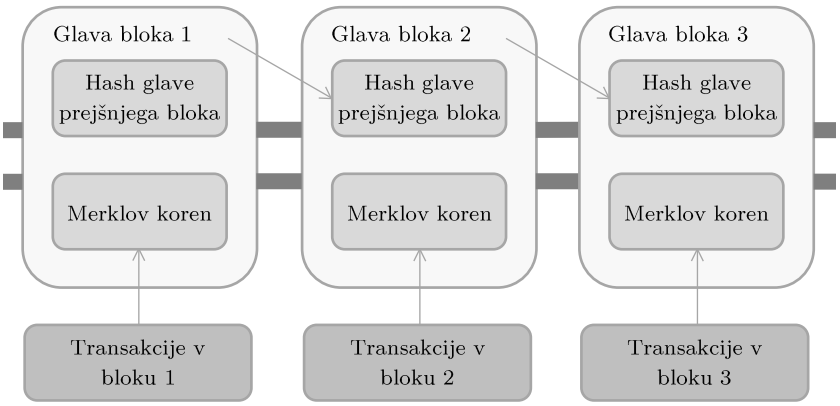
či, kombinira več prejšnjih prejetih transakcij med seboj. Na sliki 2.3 si lahko ogledamo primer kombiniranja zneskov v transakcijah za doseganje konformnosti protokolu.



Slika 2.3: Verižna odvisnost transakcij

Ko pošiljatelj sprovede transakcijo, je ta poslana vsem uporabnikom v omrežju ter čaka na potrditev. Potrjevanje transakcij se vrši s procesom t.i. 'rudarjenja' (kar je tudi glavna funkcija naprave opisane v kasnejših poglavij tega diplomskega dela). Rudarji so uporabniki omrežja, ki beležijo vse v omrežje prihajajoče transakcije in jih zbirajo v tekoči blok v obravnavi, katerega poskušajo potrditi pred ostalimi uporabniki. Pri tem konstantno izvajajo operacije proof-of-work algoritma, ki deluje takole: vzamemo vse podatke še nepotrjenih transakcij `T`, hash podatkov zadnjega bloka v blockchainu `B` ter naključno vrednost `N`, nato s pomočjo kriptografske funkcije `sha-256(T,B,N)=hash` iščemo takšno vrednost `N`, da vrne `hash`, ki se začne z določenim številom ničel v vodilnih bitih. Število zahtevanih ničel tako predstavlja zahtevnost rudarjenja. Treba je povedati, da je uspeh tu v popolnosti odvisen od naključnosti. Uporabnik, kateremu prvemu uspe najti pravo vrednost, zbere vse nepotrjene transakcije v blok, mu doda hash prejšnjega

bloka ter svoj proof-of-work z začetnimi ničlami in ga pošlje v blockchain. Tako je potrdil oziroma 'zapečatil' blok in je nagrajen z deležem na novo kreirane valute, ki se zaradi deflatorne narave valute vsake 4 leta razpolavlja. Potrjeni blok se nemudoma doda na konec blockchaina in nove prihajajoče transakcije se že tretirajo kot del novega bloka v obdelavi. Omrežje vedno tretira kot veljavnega tisti blokchain, ki je najdaljši. Torej, če več kot en uporabnik v natanko istem času potrdi blok, se blockhain razveji, toda eventuelno eden od repov prehiti ostale in ti nato odmrejo. Zahtevnost rudarjenja se v protokolu dinamično prilagaja glede na število uporabnikov v omrežju ter njihovo izmerjeno računsko moč, upoštevajoč pravilo, da se en blok transakcij v omrežju potrdi v približno 10 minutah.



Slika 2.4: Glavna knjiga kot veriga blokov

Povsod, kjer se v Bitcoin protokolu pojavi kriptografija, je za operacije uporabljana funkcija SHA-256. Zasnova vsebuje še mnogo drugih elegantno zasnovanih podrobnosti, ki olajšajo preverjanje veljavnosti transakcij, minimizirajo uporabo prostora na diskih ter pretočno količino podatkov, toda ti so izven spektra zanimanja te diplomske naloge. Več o teh rešitvah je napisanega v [6].

2.2 Scrypt

Kmalu zatem, ko je Bitcoinu v svetu zrasla prepoznavnost ter seveda vrednost, so se ena za drugo začele pojavljati nove kriptografske valute. Vse so temeljile na enaki ideološki zasnovi, vsaka pa je na tak ali drugačen način poizkušala dopolniti funkcionalnost ali odpraviti hibe začetnika. Glavna 'hiba' Bitcoina se skriva v dejstvu, da je izvajanje funkcije SHA-256 možno zelo učinkovito realizirati s pomočjo aplikacijsko specifičnih integriranih vezij oziroma ASIC računalnikov [2], kar je rezultiralo v masovni izdelavi le-teh in s tem dvigu zahtevnosti za potrjevanje blokov na smrtnikom popolnoma nedosegljivo raven. Da bi se temu pojavu izognili, so načrtovalci novih valut za potrjevanje blokov začeli uporabljati algoritme, ki jih je težje ali manj smotrno implementirati z ASIC računalniki.

Velika večina novonastalih kriptovalut danes uporablja algoritem Scrypt za potrjevanje svojih transakcij. Scrypt je v 2009 zasnoval Colin Percival z namenom, da bi ustvaril čim bolj časovno in strojno zahteven šifrirni mehanizem, ki bi napadalcem popolnoma izničil rentabilnost napada s finančnega vidika [8]. Algoritem uporablja kombinacijo sekvenčno spominsko izjemno zahtevnih funkcij in tako za hitro izvajanje potrebuje velike količine zelo hitrega pomnilnika, ki pa je drag in težko dobavljiv. Tako so z uporabo Scrypta za svoj proof-of-work algoritem nove valute omejile eksponentne rasti zahtevnosti ter ohranile bolj distribuirano in raznoliko mrežo uporabnikov, ki rudarijo. Več tehničnih podrobnosti o algoritmu si bomo pogledali kasneje v poglavju 4.

Prva takšna alternativna valuta je Litecoin. Litecoin je zasnoval in ustvaril Charles Lee s podporo članov v Bitcoin skupnosti. Projekt je bil vnaprej objavljen in nato splavljen oktobra 2011. [5, 4] Po zasnovi je neposredna kopija Bitcoina in se od njega razlikuje le po krajšem času potrjevanja bloka, večjem končnem številu kovancev ter seveda uporabi Scrypt algoritma za potrjevanje transakcij. Litecoin je kot prva Scrypt kriptovaluta hitro pridobil na popularnosti in večina uporabnikov, ki je rudarila na domačih računalnikih, se je pridružila njim bolj prijaznemu omrežju. Skladno s tem je rasla tudi vrednost in tako je v letu 2013 Litecoin presegel kapitalizacijo trga v višini miljarde ameriških dolarjev.

Še ena izmed zanimivih Scrypt kriptovalut je Dogecoin, ki se je sprva začel kot šala na internetnih forumih z idejo lahko pridobljive valute, ki bi bila zelo razširjena, imela naključne nagrade za potrditve blokov ter delovala kot mehanizem za 'dajanje napitnine' avtorjem internetnih vsebin, ki se uporabnikom zdijo uporabne. Sprva valute nihče ni jemal resno, toda njena popularost se je razširila kot virus in je dandanes presegla število transakcij vseh kriptovalut skupaj. Dogecoin razvijalci so znani tudi po nenavadnih načinih promocije valute, kot so recimo sponzorstvo Jamajškega moštva za bob, poslikavo celotnega NASCAR dirkalnega avtomobila z maskoto valute ter dobrodelno gradnjo vodnjaka s pitno vodo v Keniji. Poleg Litecoina in Dogecoina obstaja na trgu še mnogo drugih Scrypt kriptovalut, vsaka s svojimi posebnostmi, prednostmi ter slabostmi. Podrobni opisi le-teh ne spadajo v okvir te diplomske naloge, zato bomo kot zanimivost navedli zgolj informativno primerjalno tabelo 2.1 nekaterih:

	Čas bloka	Nagrada bloka	Končno število	Razpolovna doba
Bitcoin	10 min	25 BTC	21 mio	4 leta
	Prva digitalna valuta			
Litecoin	2.5 min	50 LTC	84 mio	4 leta
	'Lite' verzija Bitcoina			
Dogecoin	1 min	Naključna	∞	Padajoča
	Zabavna široko dostopna valuta			
Potcoin	40 sek	420 pot	420 mio	4 leta
	Za legalizacijo marihuane			
Anoncoin	3.42 min	5 ANC	4.2 mio	2 leti
	Popolna anonimnost, transakcije preko darkneta			
Spaincoin	2 min	100 SPA	50 mio	8 mesecev
	Polovica valute razdeljena med španske uporabnike			

Tabela 2.1: Scrypt kriptovalute in Bitcoin

2.3 Novi pristopi

V iskanju optimalne zasnove kriptovalute, ki bi kljubovala vsem v praksi odkritih pomanjkljivostim današnjih, so se razvile tudi različne kombinacije ter variacije kriptografskih funkcij in algoritmov:

- X11 (cryptcoin, darkcoin) – 11 kriptografskih funkcij prepretenih med seboj
- Keccak (maxcoin) – NIST je izbral Keccak kot zmagovalca na tekmovanju za implementacijo SHA-3 [7]
- Scrypt-N (execoin, vertcoin) – Scrypt z dodanim parametrom N, ki progresivno skozi čas viša samo spominsko zahtevnost algoritma
- Grøstl (diamondcoin) – Grøstl, kandidat za SHA-3 s strani študentov danske DTU ter TU Graz [3]

Debata o tem, katera zasnova je najboljša, ostaja odprta, saj razvijalci tako algoritmov kot novih valut izpostavljajo negativne lastnosti svojih tekmecev, definitivno pa se razvoj celostno odvija v pravo smer in zato lahko v prihodnosti pričakujemo še bolj izpopolnjene rešitve.

3 | Strojna oprema

Za namen diplomske naloge smo sestavili napravo, ki bo namenjena izključno potrjevanju blokov transakcij z uporabo Scrypt proof-of-work algoritma. Torej rudarja Scrypt kriptografskih valut, ki bo za to uporabljal tri grafične kartice visokega cenovnega razreda, sestavljenega na posebnem ohišju. Izpostavimo še zanimivost, da je bilo v času sestavljanja nemalo težav z dobavo omenjenih grafičnih kartic, saj je zaradi takratne dobičkonostnosti rudarjenja kriptografskih valut povpraševanje po njih preseglo kapacitete same proizvodnje in jih na trgu enostavno ni bilo mogoče kupiti. Tako je bilo za pridobitev vse potrebne strojne opreme potrebno več kot mesec dni.

3.1 Komponente

Za sestavo rudarja smo uporabili sledeče komponente strojne opreme:

- Matična plošča: Gigabyte GA-z77x-UD3H
- Procesor: Intel Celeron G1620
- Disk: Crucial SSD 120GB mSATA3
- Pomnilnik: TeamGroup Vulcan 8GB DDR3
- Napajalnik: SuperFlower Leadex 1200W Platinum
- Grafične kartice: 1 x Gigabyte Radeon R9 290 4GB GDDR5 Reference
- Grafične kartice: 2 x Asus Radeon R9 290 4GB GDDR5 DirectCU II

Matična plošča je bila poleg zahteve po visoki kvaliteti materiala skrbno izbrana tako, da se na njej nahajajo osnovne kontrole kot so gumb za vklop/izklop, reset gumb, gumb za ponastavitev CMOSa ter LED indikator statusa sistema za enostavnejše upravljanje. Prav tako smo zavoljo elegantnosti potrebovali podporo mSATA diskom.

Procesorska moč je za rudarjenje z GPE irelevantna lastnost, zato je bil izbran najcenejši procesor, ki ustreza podnožju LGA 1155 na naši matični plošči.

Za disk smo izbrali mSATA SSD disk, ki se vstavi v matično ploščo, za priklop ne potrebuje nikakršnih kablov ali vodil in je praktično neviden. Hkrati s svojimi 120 GB prostora zadostuje za namestitev dveh modernih operacijskih sistemov.

Velikost pomnilnika je odmerjena tako, da pri rudarjenju visoka konkurenčnost niti v kombinaciji s porabo operacijskega sistema ne doseže omejitve sistema.

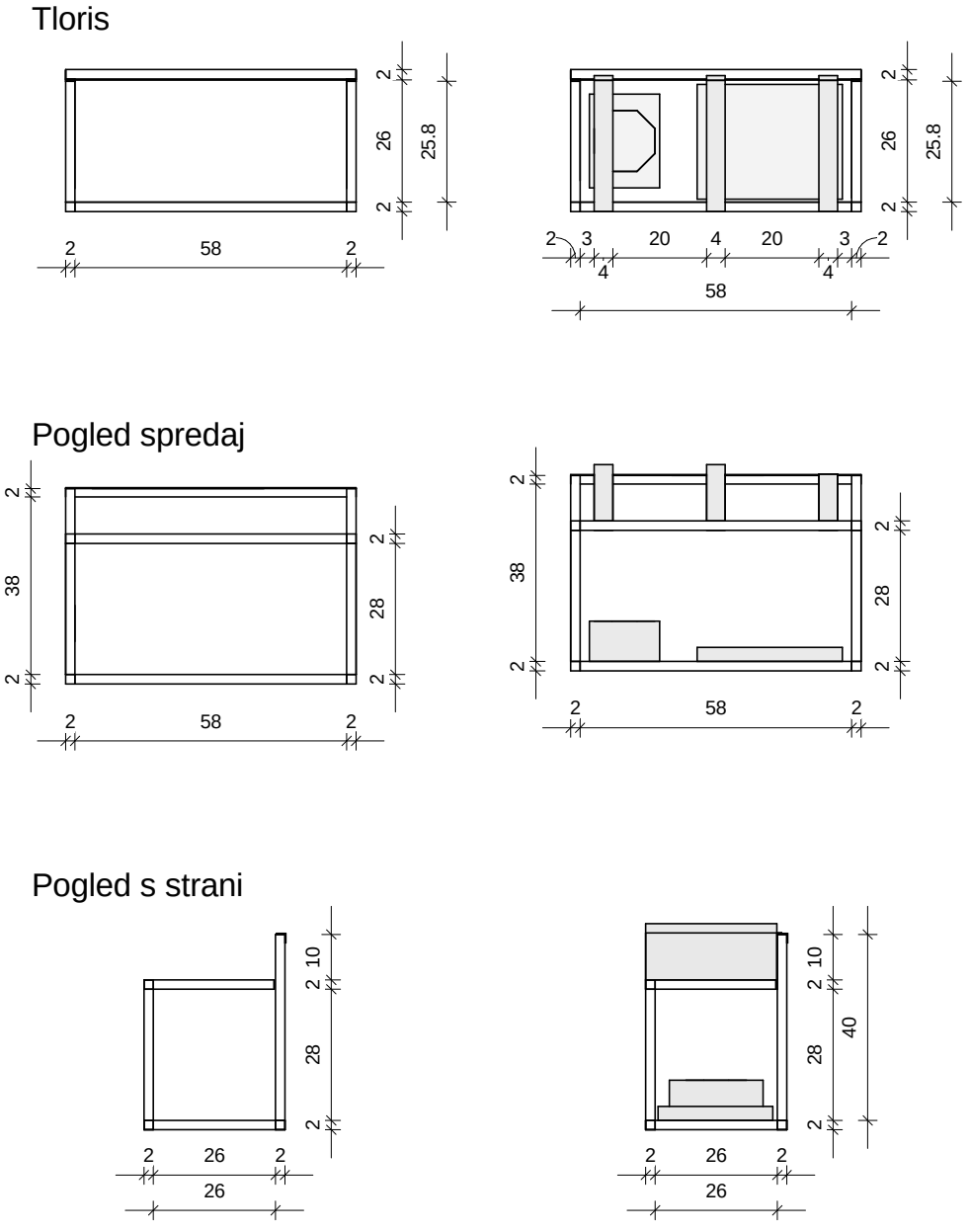
Napajalnik je poleg grafičnih kartic v tem primeru najbolj pomembna komponenta sistema. Zagotavljati mora izjemno stabilno napajanje pri konstantni visoki porabi energijsko požrešnih grafičnih kartic. Po izčrpni raziskavi smo skrbno izbrali napajalnik višjega kakovostnega razreda, s praktično neobstoječim nihanjem napetosti na 12V tračnici ter več kot 92% učinkovitostjo pri polni obremenjenosti.

Grafične kartice vse izhajajo iz vrha ponudbe družine Radeon proizvajalca AMD, ki se od svojih NVIDIinih tekmic razlikujejo po arhitekturni zasnovi shaderjev in so pri karticah primerljivega ranga tudi do trikrat hitrejši pri rudarjenju. Namen je bil uporabiti štiri identične grafične kartice, a to ni bilo mogoče zaradi tedanje situacije na trgu.

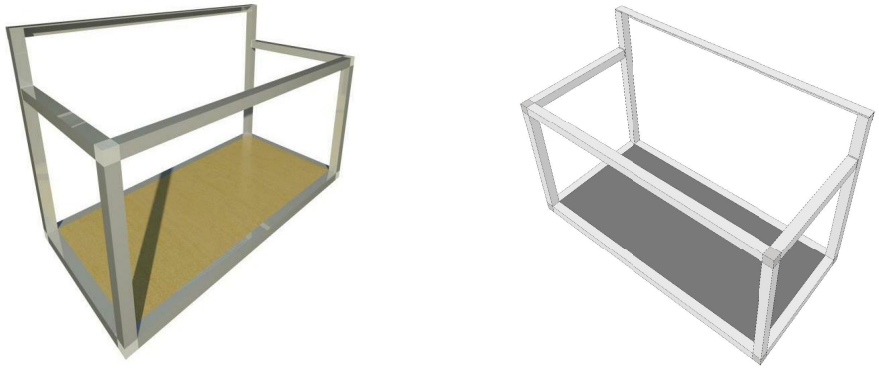
Poleg skrbnega izbora strojne opreme je bilo za doseganje optimalnih rezultatov potrebno zasnovati in izdelati posebno ohišje, ki omogoča postavitev uporabljenih komponent na tak način, da se zagotovi maksimalno pretočnost zraka ter odvajanje toplote, ki nastaja ob konstantni visoki obremenitvi grafičnih kartic. Za doseganje zadovoljivih temperatur je bilo nujno potrebno grafične kartice ločiti od matične plošče in jih locirati čim bolj razpršeno v zračnem prostoru. Za ta namen smo uporabili posebne podaljške PCI-e vodil matične plošče, imenovane USB PCI-e 'riserji'. USB riserji so moderna različica starih riser kartic, ki so se včasih uporabljale v starih strežniških ohišjih za priklop dodatnih kartic v primeru pomanjkanja prostora, le da tu za pretok podatkov med dvignjenim vezjem in PCI-e vodilom na matični plošči skrbi USB 3.0 podatkovni kabel. Primer USB riserja si lahko ogledamo na sliki 3.1.

Slika 3.1: USB 3.0 PCI-e riser

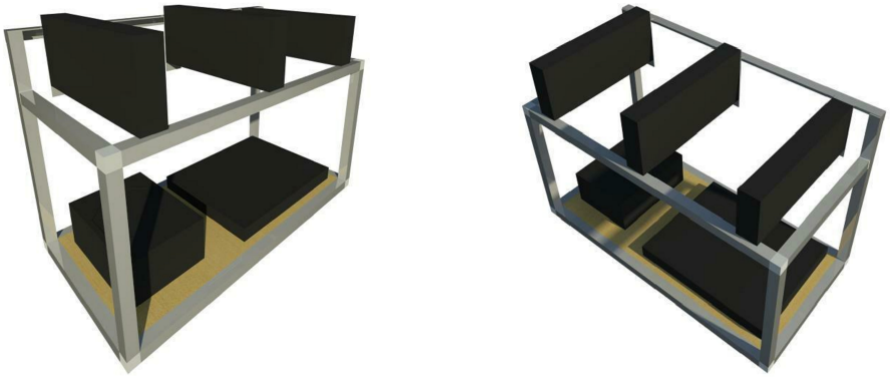
S pomočjo riserjev in natančnega načrtovanja smo zasnovali in sestavili optimalno ogrodje za namestitev naše strojne opreme. Ogrodje samo je zgrajeno iz palic eloksiranega aluminija in gradbenih vijakov, za podlago pa je uporabljen smrekov les. Na sliki 3.2 je prikazan tehnični načrt za ohišje z merami. Na slikah 3.3 3.4 si lahko ogledate modelno zasnovo v 3D okolju in nato končni rezultat na slikah 3.5.



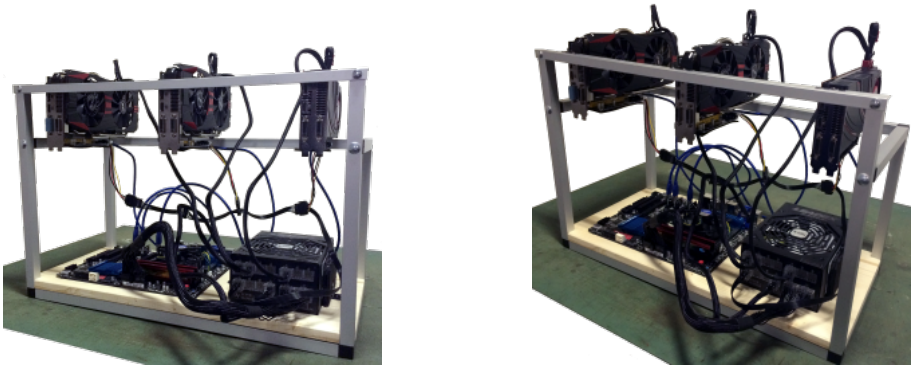
Slika 3.2: Tehnični načrt ohišja



Slika 3.3: Zasnova okvirja



Slika 3.4: Render modela



Slika 3.5: Končni rezultat

3.2 Grafične kartice

Grafične kartice, ki smo jih uporabili za izgradnjo rudarja so si v osnovi sorodne, vse tri so zgrajene okoli AMDjevega 28nm Hawaii-Pro procesorja in imajo na vezju 4GB GDDR5 pomnilnika. Izdelane pa so s strani dveh različnih proizvajalcev in se od enega do drugega razlikujejo po arhitekturni zasnovi vezne plošče, pristopu k hlajenju in nekaj drugih podrobnostih.

3.2.1 Gigabyte Radeon R9 290 4GB GDDR5 Reference

Prva izmed kupljenih kartic je zgrajena po AMDjevi referenčni zapovedani zasnovi, od katere Gigabyte ni odstopal niti za ped. Uporablja standardno razporeditev elementov v vezju, in referenčni sesalno-pihalni način hlajenja, kjer se zrak s pomočjo ventilatorja na zadku vsesava pod pokrov kartice in je nato po principu turbine potisnjen vzdolž vezja proti prednjemu delu kjer se nahaja izpuh. Takšen način hlajenja je zelo učinkovit v tesnem in zaprtem okolju ali v primeru, da je več kartic zloženih blizu druga poleg druge, kot se je kasneje izkazalo pa deluje odlično tudi v našem primeru. Kartico lahko vidimo na sliki 3.6.

Slika 3.6: Gigabyte Radeon R9 290 4GB GDDR5 Reference

3.2.2 Asus Radeon R9 290 4GB GDDR5 DirectCU II

Preostali dve kartici prihajata iz ASUSove skupine modificiranih grafičnih kartic DirectCU II. Proizvajalec je v tem primeru preuredil zasnovo vezne plošče in jo prilagodil tako, da postavitev pregrevanih delov ustreza posebej razvitemu disipacijskemu cevnemu hladilniku z dvema ventilatorjema. Izboljšane so tudi možnosti zviševanja frekevnc delovanja ter dodan poseben 8-fazni regulator napetosti DIGI+. Navkljub teoretični superiornosti na področju hlajenja se v praksi izkaže, da zasnova vezne plošče ni najbolje premišljena in tako, ob sicer nizkih delovnih temperaturah procesorja ter pomnilnika, prihaja do pregrevanja prej omenjenga regulatorja napetosti, kar nato vpliva na celostno delovanje kartice. Kartico si lahko ogledamo na sliki 3.7.

Slika 3.7: Asus Radeon R9 290 4GB GDDR5 DirectCU II

V tabeli 3.1 si lahko ogledamo primerjavo kartic po specifikacijah in določenih lastnostih:

	Gigabyte Radeon R9 290 4GB GDDR5 Reference	**Asus Radeon R9 290 4GB GDDR5 DirectCU II**
Vodilo	PCI Express 3.0	PCI Express 3.0
Čipovni nabor	AMD Radeon R9 290	AMD Radeon R9 290
GPE jedro	Hawaii Pro	Hawaii Pro
Tehnologija izdelave	28 nm GCN	28 nm GCN
Pretočnih procesorjev	2560	2560
Frekvenca jedra	947 MHz	947 MHz
Video pomnilnik	GDDR5 4GB	GDDR5 4GBA
Proizvajalec pomnilnika	Elpida	Elpida
Frekvenca pomnilnika	5000 MHz (1250 MHz GDDR5)	5000 MHz (1250 MHz GDDR5)
ASIC kvaliteta	**71.4 %**	**84.8 %**
Spominsko vodilo	512 bit	512 bit
Izhodi	2x DVI-D, DisplayPort, HDMI	2x DVI-D, DisplayPort, HDMI
Poraba energije	**600 W**	**300 W**
Dimenzije	289x126x41 mm	287x147x40 mm

Tabela 3.1: Primerjava grafičnih kartic

Kot zanimivost povejmo, da sta za proizvodnjo grafičnega pomnilnika v vseh izdelanih karticah Radeon R9 odgovorna dva proizvajalca: japonska Elpida in južnokorejski Hynix. Po ugotovitvah mnogih entuziastov ter preizkuševalcev je slednji dosti bolj kakovosten in se manj pregreva pri delovanju na višjih frekvencah. Nažalost je zaradi požara v dveh proizvodnih obratih konec leta 2013 dobavljivost Hynix pomnilnika padla in tako velika večina kasneje izdelanih grafičnih kartic uporablja Elpida pomnilnik. Tako tudi vse tri kartice uporabljene pri sestavi rudarja na vezju nosijo pomnilnik proizvajalca Elpida, kar pomeni, da bi verjetno

lahko v nasprotnem primeru pri rudarjenju dosegali za nekaj odstotkov boljše rezultate.

Še ena zanimiva lastnost je ASIC kvaliteta GPE jedra, ki nakazuje v kolikšni meri napetost 'pušča' skozi jedro. Manj ko je uhajanja, višjo privzeto napetost imamo v čipu, toda pri tem dobimo tudi šibkejši električni tok in s tem manj pregrevanja. Višja vrednost ASIC kvalitete jedra predstavlja višjo stopnjo uhajanja in s tem slabše predispozicije za doseganje nizkih temperatur ob obremenitvi. Pri R9 290 karticah variira med 60% in 100%. Kot vidimo v tabeli 3.1 je naša Gigabyte referenčna kartica v rangu nizko 'puščajočih' primerkov, medtem ko sta obe ASUS kartici na tem področju manj učinkoviti.

V specifikacijah v tabeli 3.1 je možno videti, da je navedena poraba energije pri polni obremenitvi za ASUS kartici 300W, za Gigabyteovo pa kar 600W. Slednje se izkaže za pretirano, saj v praksi med rudarjenjem porabi približno 356W elektrike in v kombinaciji s porabo vseh ostalih komponent v sistemu skupna poraba ne preseže 1050W. Prikaz porabe električne energije med rudarjenjem lahko po korakih vidimo na sliki 3.8. Takšen rezultat potrjuje optimalno izbiro uporabljenega napajalnika.

(a) Vse kartice (b) Samo ASUS kartici (c) Ena ASUS kartica

Slika 3.8: Poraba elektrike med rudarjenjem

3.3 Optimizacija

Fizična optimizacija delovanja strojne opreme je tu izven dosega. Kljub temu smo na vse tri kartice namestili posebej modificiran neuraden BIOS, ki s kalibracijo krmilnika grafičnega pomnilnika zmanjša čas med cikli in s tem latenco pri komunikaciji jedra s pomnilnikom. Optimizirana je tudi učinkovitost regulatorja napetosti, kar omogoča stabilnejše delovanje ter doseganje višjih frekvenc delovanja pri nižjih privzetih napetostih.

Za operacijski sistem smo si prvotno izbrali posebej za rudarjenje prilagojeno distribucijo Linuxa Cryptoslax, ki temelji na ogrodju distribucije Slackware, vsebuje pa zgolj module potrebne za rudarjenje ter se ob nalaganju v celoti zapiše v sistemski pomnilnik. Konfiguracija kljub prizadevnemu prilagajanju in preizkušanju različnih opcij ni dajala pričakovanih rezultatov, zato smo se iz nezaupanja v neuradno in relativno nepreverjeno distribucijo odločili za drugačen pristop. Na rudarja smo vzporedno namestili operacijski sistem Microsoft Windows 7 ter Linux distribucijo Slackware 14.1 in na obeh naložili iste gonilnike ter programsko opremo za rudarjenje. Pri testiranju smo prišli do ugotovitve, da so zmogljivosti na operacijskem sistemu Windows 7 pri rudarjenju ob istih nastavitvah vedno za okoli 5% boljše od tistih na Linuxu. Krivdo za to verjetno lahko pripišemo slabše optimiziranim grafičnim gonilnikim za operacijski sistem Linux.

V obeh primerih smo uporabili uradne grafične gonilnike AMD Catalyst 13.12, ki dajejo optimalne rezultate. Novejša različica gonilnikov Catalyst 14.1 in njihovi nasledniki imajo hroščato implementacijo okolja OpenCL ter knjižnice ADL, kar se kaže v slabših zmogljivostih pri rudarjenju.

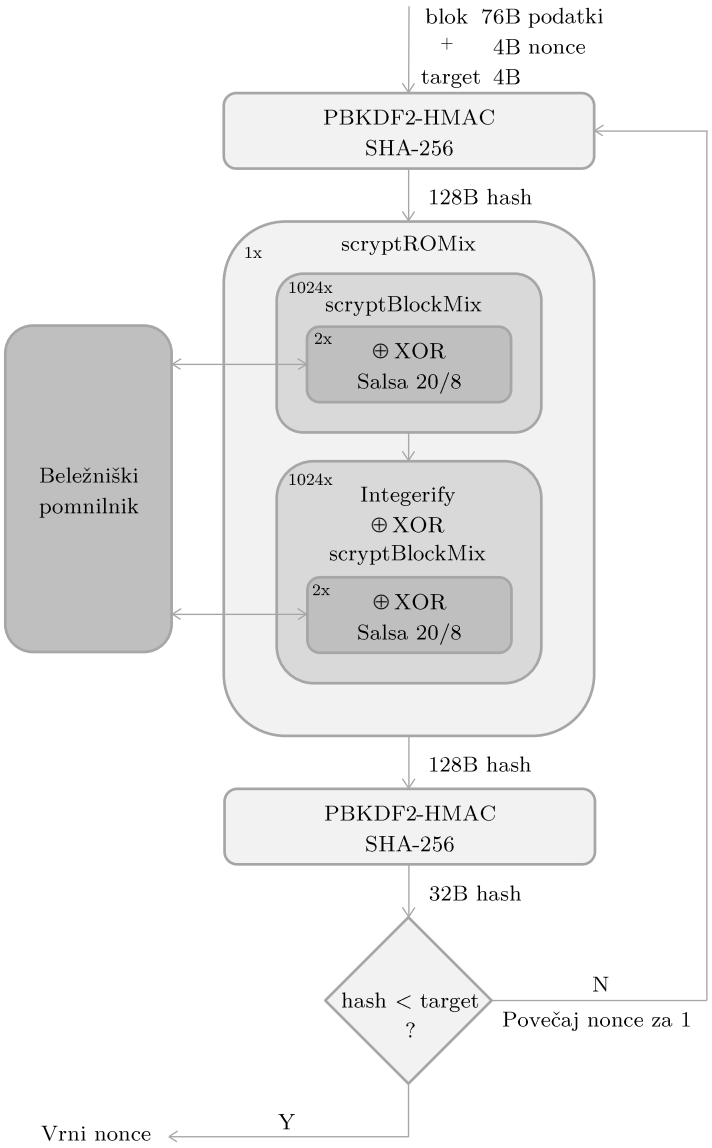
4 | Potrjevanje blokov

Kot je navedeno v poglavju 2, naša naprava za rudarjenje kriptografskih valut uporablja Scrypt proof-of-work algoritem za potrjevanje blokov trenutnih transakcij v omrežju. Pri tem početju tekmuje z ostalimi rudarji v omrežju. V primeru uspeha s svojim proof-of-work rezultatom podpiše trenutni aktivni blok in ga pripne na konec blockchaina, v omrežju pa se začnejo nove transakcije zbirati in potrjevati v okviru novega aktivnega bloka. Ob vsakem potrjenem bloku transakcij za nagrado rudar prejme majhen delček na novo ustvarjene valute.

Za namen rudarjenja raznih kriptovalut je s strani skupnosti ter razvijalcev valut samih že napisano in podrobno optimizirano nekaj programske opreme, zato smo se namesto lastne implementacije raje odločili za konfiguracijo obstoječe in s tem zelo verjetno boljše rezultate.

4.1 Scrypt proof-of-work algoritem

Scrypt algoritem, ki ga je zasnoval Colin Percival [8] je za uporabo v proof-of-work algoritmu za potrjevanje transakcij malce poenostavljen z določenimi parametri. Kombiniran je tudi z uporabo PBKDF2 SHA-256 funkcije za generacijo vhodnega semena ter izračun končnega izhodnega hasha, ki se ga nato primerja s ciljno zahtevnostjo. Grafični prikaz poteka algoritma si lahko ogledamo na sliki 4.1, kateri nato sledijo primeri kode glavnih delov algoritma.



Slika 4.1: Potek Scrypt proof-of-work algoritma

4.1.1 scryptROMix

Algoritem scryptROMix v principu ustvarja veliko število nekih 'naključnih' vrednosti, do katerih nato naključno dostopa in s tem zagotovi, da vsi ostanejo zapisani v pomnilniku, kar doseže s pomočjo klicev posebne funkcije scryptBlockMix.

S tem zagotavlja sekvenčno pomnilniško težavnost, česar formalen dokaz je moč prebrati v beli knjigi algoritma Scrypt. [8]

```
r   - velikost bloka, v našem primeru r = 1
B   - blok kot vhodni vektor dolžine 128 * r bajtov
N   - zahtevnostna stopnja obremenitve pomnilnika, v našem primeru N = 1024

B^- izhodni vektor dolžine 128 * r bajtov
```

```
1  X = B
2  for i = 0 to N - 1 do
3    V[i] = X
4        X = scryptBlockMix (X)
5  end for
6  for i = 0 to N - 1 do
7      j = Integerify (X) mod N
8                  # kjer Integerify (B[0]...B[2 * r - 1]) pretvori
9                  # zadnji del vektorja B (B[2 * r - 1]) v celo število
10                 # po pravilu tankega konca.
11     T = X xor V[j]
12     X = scryptBlockMix (T)
13 end for
14 B^ = X
```

4.1.2 scryptBlockMix

Funkcija scryptBlockMix z uporabo kriptografske funkcije Salsa20/8 zadošča vsem pogojem, ki morajo biti izpolnjeni, da je scryptROMix pomnilniško sekvenčno težavna funkcija.

```
r - velikost bloka, v našem primeru r = 1
B[0]...B[2 * r - 1] - vhodni vektor sestavljen iz 2 * r 64-bajtnih blokov

B^[0]...B^[2 * r - 1] - izhodni vektor sestavljen iz 2 * r 64-bajtnih blokov
```

```
1  X = B[2 * r - 1]
2  for i = 0 to 2 * r - 1 do
3  T = X xor B[i]
4  X = Salsa20/8 (T)
5  Y[i] = X
6  end for
7  B^ = (Y[0], Y[2], ..., Y[2 * r - 2],Y[1], Y[3], ..., Y[2 * r - 1])
```

4.1.3 Salsa20/8

Salsa20 je družina enkripcijskih funkcij, ki sestojijo iz dolge verige treh enostavnih operacij nad 32-bitnimi besedami.[1] Funkcija Salsa20/8 je malce okrnjena različica iz družine teh funkcij. V našem primeru svoj 64B vhod tretira kot 16 besed urejenih po pravilu tankega konca. Teh 16 besed gre skozi 320 modifikacij, kjer vsaka modifikacija spremeni eno besedo. Nastalih 16 besed je dodanih k začetnim po modulu 2^{32}, kar producira nov 64B izhod. Vsaka modifikacija je sestavljena iz XOR operacije med eno besedo ter rotirano vsoto dveh drugih besed po modulu 2^{32} kar pomeni da teh 320 modifikacij vsebuje 320 seštevanj, 320 XOR operacij ter 320 rotacij enake dolžine.

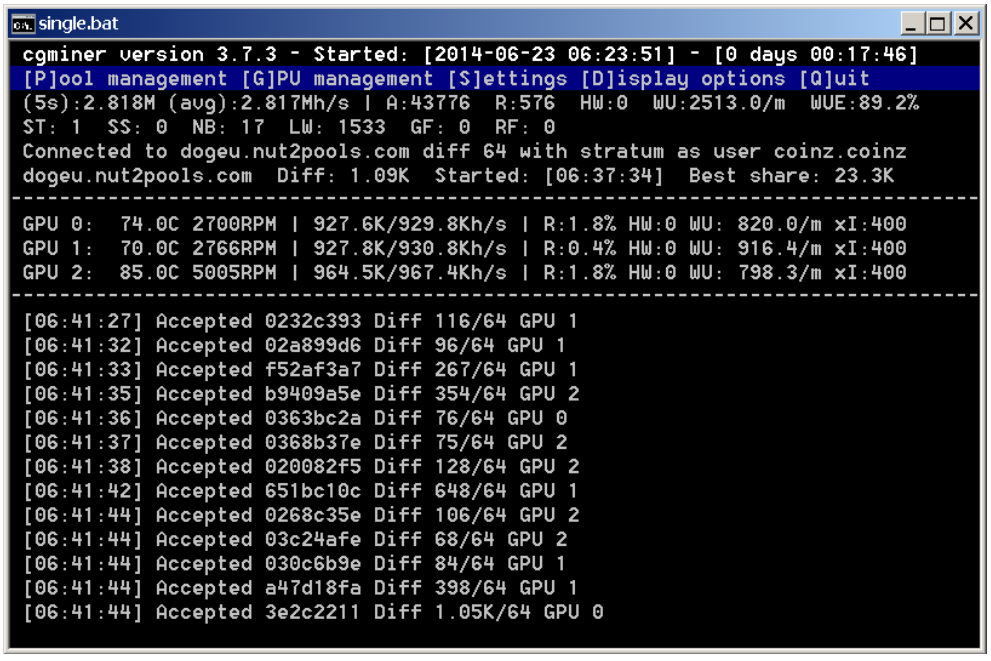
```
1  #define R(a,b) (((a) << (b)) | ((a) >> (32 - (b))))
2  void salsa20_word_specification(uint32 out[16],uint32 in[16]) {
3          int i;
4          uint32 x[16];
5          for (i = 0;i < 16;++i) x[i] = in[i];
6          for (i = 20;i > 0;i -= 2) {
7            x[ 4] ^= R(x[ 0]+x[12], 7);  x[ 8] ^= R(x[ 4]+x[ 0], 9);
8            x[12] ^= R(x[ 8]+x[ 4],13);  x[ 0] ^= R(x[12]+x[ 8],18);
9            x[ 9] ^= R(x[ 5]+x[ 1], 7);  x[13] ^= R(x[ 9]+x[ 5], 9);
10           x[ 1] ^= R(x[13]+x[ 9],13);  x[ 5] ^= R(x[ 1]+x[13],18);
11           x[14] ^= R(x[10]+x[ 6], 7);  x[ 2] ^= R(x[14]+x[10], 9);
12           x[ 6] ^= R(x[ 2]+x[14],13);  x[10] ^= R(x[ 6]+x[ 2],18);
13           x[ 3] ^= R(x[15]+x[11], 7);  x[ 7] ^= R(x[ 3]+x[15], 9);
14           x[11] ^= R(x[ 7]+x[ 3],13);  x[15] ^= R(x[11]+x[ 7],18);
15           x[ 1] ^= R(x[ 0]+x[ 3], 7);  x[ 2] ^= R(x[ 1]+x[ 0], 9);
16           x[ 3] ^= R(x[ 2]+x[ 1],13);  x[ 0] ^= R(x[ 3]+x[ 2],18);
17           x[ 6] ^= R(x[ 5]+x[ 4], 7);  x[ 7] ^= R(x[ 6]+x[ 5], 9);
18           x[ 4] ^= R(x[ 7]+x[ 6],13);  x[ 5] ^= R(x[ 4]+x[ 7],18);
19           x[11] ^= R(x[10]+x[ 9], 7);  x[ 8] ^= R(x[11]+x[10], 9);
20           x[ 9] ^= R(x[ 8]+x[11],13);  x[10] ^= R(x[ 9]+x[ 8],18);
21           x[12] ^= R(x[15]+x[14], 7);  x[13] ^= R(x[12]+x[15], 9);
22           x[14] ^= R(x[13]+x[12],13);  x[15] ^= R(x[14]+x[13],18);
23         }
24         for (i = 0;i < 16;++i) out[i] = x[i] + in[i];       }
```

4.2 CGMiner

Izmed obstoječih programov, ki implementirajo rudarjenje kriptografskih valut s pomočjo Scrypt proof-of-work algoritma smo izbrali CGMiner, katerega avtor je Con Kolivas in je eden izmed dveh najbolje poznanih in pogosto uporabljanih programov za rudarjenje s pomočjo grafičnih kartic. Zavoljo boljše konfigurabilnosti smo uporabili različico iz posebne veje razvoja CGMinerja.

CGMiner s pomočjo knjižnice ADL ter okolja OpenCL v gonilnikih za grafične kartice omogoča podrobno upravljanje z njimi, ter monitoring le-teh med rudarjenjem. Z uporabo programskih vmesnikov kriptografskih valut komunicira v njihovih omrežjih po zapovedanih protokolih, omogoča pa tudi priključevanje k bazenom in preklapljanje med njimi.

Na sliki 4.2 si lahko ogledamo programsko okno CGMinerja med rudarjenjem.



Slika 4.2: CGMiner med rudarjenjem

V zgornjem delu okna je zbran splošen prikaz zmogljivosti trenutne seje rudarjenja. Razlaga za pomembnejše indikatorje je sledeča:

- 5s: povprečje hitrosti računanja hashev zadnjih 5 sekund
- avg: povprečna hitrost računanja hashev celotnega časa
- A: skupna zahtevnost sprejetih deležev
- R: skupna zahtevnost zavrnjenih deležev
- HW: število strojnih napak
- WU: delovni obseg, število poslanih deležev na minuto
- WUE: delovna uporabnost, odstotek sprejetih deležev od vseh poslanih

V sredini so zbrane trenutne informacije o stanju vsake posamezne grafične kartice:

- Temperatura jedra
- Hitrost ventilatorja
- Trenutna in povprečna hitrost računanja hashev
- Kot opisano zgoraj, A, R, HW ter WU za posamezno kartico

V spodnjem delu je prikazan časovno urejen tok poslanih deležev, njihova zahtevnost v trenutku oddaje ter podatek o tem, od katere kartice so bili izračunani.

4.3 Konfiguracija in meritve

CGMiner preko konfiguracijskih datotek `.conf` omogoča zelo širok nabor nastavitev grafičnih kartic in upravljanje vsake kartice posebej. Tako je med drugim možno nastavljati hitrosti delovanja ventilatorjev, frekvence delovanja jedra ter pomnilnika, privzeto voltažo na vezju in podobno. Spodaj je prikazana vsebina naše končne konfiguracijske datoteke, ki daje dobre ter stabilne rezultate, pod njo pa razlaga bolj pomembnih parametrov ter argumentacija uporabljenih vrednosti.

```
{
"pools" : [
  {
    "name" : "dogeu.nut2pools.com",
    "url" : "stratum+tcp://dogeu.nut2pools.com:5585",
    "user" : "???",
    "pass" : "???",
    "pool-priority" : "0"
  }
],
"api-allow" : "W:127.0.0.1",
"api-listen" : true,
"expiry" : "1",
"failover-only" : true,
"log" : "5",
"queue" : "0",
"scan-time" : "1",
"gpu-threads" : "1",
"auto-fan" : true,
"lookup-gap" : "2",
"device" : "0,1,2",
"temp-target" : "60,60,75",
"temp-overheat" : "65,65,85",
"temp-cutoff" : "75,75,95",
"gpu-fan" : "30-100,30-100,30-92",
"gpu-engine" : "1025:947,1025:947,1065:947",
"gpu-memclock" : "1400:1250,1400:1250,1400:1250",
"gpu-powertune" : "50",
"thread-concurrency" : "20481,20481,20481",
"xintensity" : "400,400,400",
"cl-filename" : "kalroth"
}
```

`"pools":[]` Nastavitve strežnikov ter uporabniških imen za povezovanje v bazene. V tem primeru se povezujemo v bazen skupnosti nut2pools, kjer rudarimo Dogecoin.

`"gpu-threads":"1"` Število glavnih niti rudarskega procesa na kartico. Po naših ugotovitvah sta edini realno uporabni vrednosti 1 ali 2, meritve pa pokažejo, da je bolje imeti eno nit z višjo intenziteto rudarjenja, kot dve paralelni manj intenzivni niti.

`"lookup-gap":"2"` Velikost presledka pri shranjevanju podatkov v beležniški pomnilnik med izvajanjem Salsa20/8 funkcije. Vrednost 1 bi pomenila vseh 1024 vrednosti naenkrat shranjenih v pomnilniku, pri vrednosti 2 shranimo vsako drugo, pri 3 vsako tretjo in tako naprej. Pri bolj gosto shranjenih podatkih je potrebno manj preračunavanja manjkajočih delov in s tem manj ciklov procesne enote. Tako je v primeru zelo velikega beležniškega pomnilnika optimalno izbrati nič presledka in obratno, v primeru visoke frekvence procesorja in manjšega beležniškega pomnilnika bolje izbrati večji presledek. Za naše grafične kartice je optimalen izbor hramba vsake druge vrednosti.

`"device":"0,1,2"` Indeksi naprav, s katerimi hočemo rudariti, med seboj ločeni z vejico. Tokrat so to grafične kartice, ki jih OpenCL naslavlja z 0,1,2. Za naprej povejmo, da sta z indeksoma 0 in 1 naslovljeni ASUS kartici, z indeksom 2 pa Gigabyte referenčna kartica.

`"temp-target":"60,60,75"` Ciljna temperatura procesorskega jedra kartic, ki jo CGMiner poizkuša vzdrževati s prilagajanjem hitrosti ventilatorjev ali eventuelno nižanjem frekvence jedra. V našem primeru sta ciljni temperaturi za obe ASUS kartici postavljeni relativno nizko, saj je njuna temperatura jedra pod obremenitvijo dosti nižja od referenčne kartice, hkrati pa s tem preprečimo kritično pregrevanje regulatorja napetosti. Po drugi strani se referenčna kartica odlično obnese pri višjih temperaturah, zato je tu ciljna temperatura nastavljena bolj z namenom zagotovitve delovanja ventilatorja na visokih vrtljajih.

`"gpu-engine":"1025:947,1025:947,1065:947"` Nastavitev višine frekvenc delovanja grafičnega jedra po karticah, ločenih z vejico. Delovna frekvenca med rudarjenjem z dvopičjem ločena od ponastavitvene frekvence, na katero se vrnemo po koncu rudarjenja. Tukaj se referenčna kartica izkaže veliko bolj dovzetna za navijanje, saj ASUS kartici ne zmoreta dolgotrajnega stabilnega delovanja pod obremenitvijo na frekvencah višjih od 1025 MHz, medtem ko Gigabyte kartica v dobrih pogojih stabilno prenese tudi frekvence višje od 1075 MHz. Ta 5% razlika v hitrosti procesorskega takta se prevede tudi v 5% razliko v hitrosti računanja hashev. Teoretični optimum, ki bi ga dosegli pri frekvenci 1100 MHz in s tem popolni saturaciji bi znašal 2560 * 1100 * 0.352(nominalna učinkovitost Hawaii jeder) = 991232 hashev oziroma približno 991 Khashev na sekundo.

`"gpu-memclock":"1400:1250,1400:1250,1400:1250"` Nastavitev višine frekvenc delovanja grafičnega pomnilnika po karticah, ločenih z vejico. Delovna frekvenca med rudarjenjem z dvopičjem ločena od ponastavitvene frekvence, na katero se vrnemo po koncu rudarjenja. Za pomnilnik proizvajalca Elpida so priporočene optimalne frekvence delovanja med 1375 MHz in 1450 MHz. Tako latenca kot pasovna širina, ki sta potrebni za optimalno saturacijo, sta na voljo že pri 1400 MHz in nadaljnje višanje frekvence le še slabša rezultate. Vse naše kartice brez težav dosegajo takšne frekvence pomnilnika.

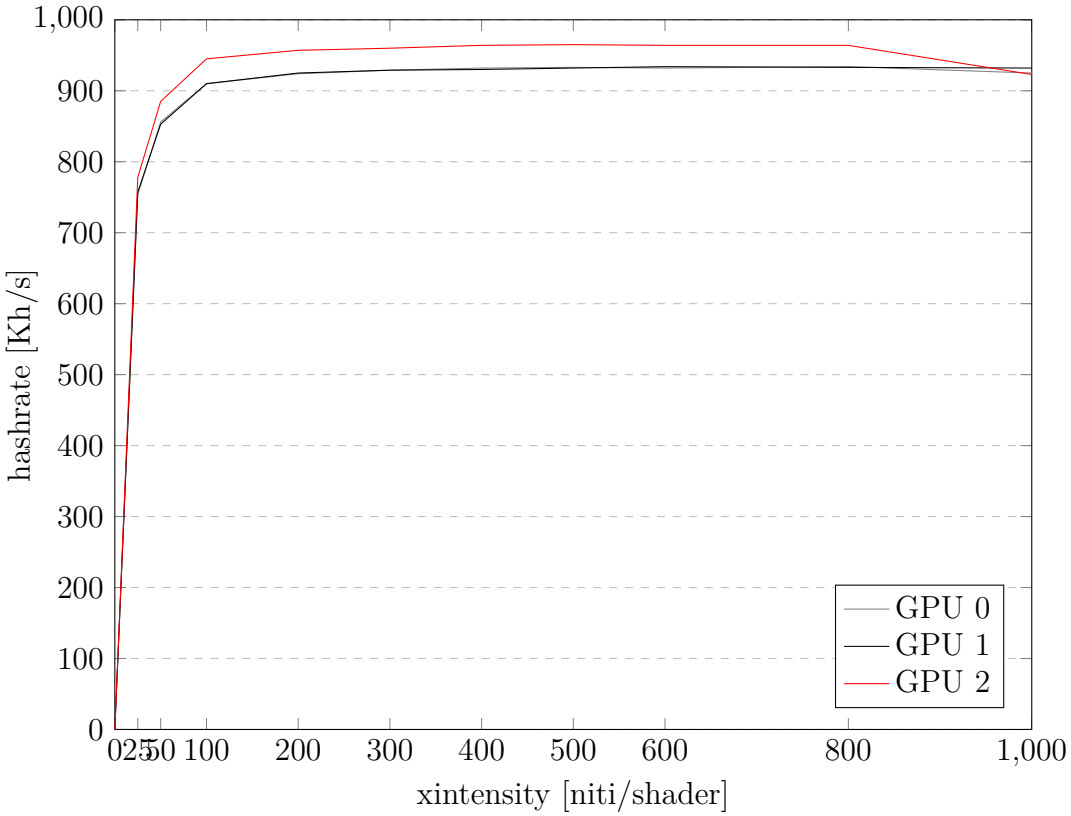
`"gpu-powertune":"50"` Količina dodatne voltaže, ki jo kartice smejo uporabiti, da konstantno dosegajo nastavljeno višino frekvence grafičnega jedra in s tem ohranjajo stabilen nivo hitrosti računanja hashev. V našem primeru uporabljamo 50mV, kar je tudi maksimum dovoljen v gonilniku.

`"thread-concurrency":"20481,20481,20481"` Možno število konkurenčnih operacij za vsako nit. V teoriji je optimalno izbrati čim višji večkratnik števila pretočnih procesorjev na kartici, s katerim še lahko izvajamo operacije in mu prištejemo ali odštejemo 1. Po izčrpnem testiranju smo si izbrali vrednost 20481, kar je 2560 * 8 + 1. V primeru računanja hashev z dvema ali večimi glavnimi nitmi je ta vrednost občutno nižja.

`"xintensity":"400,400,400"` Količina dela, ki ga grafična kartica mora opraviti, preden lahko vrne svoje rezultate. Optimalna vrednost je tik pod mejo preobremenjenosti kartice. Podana vrednost se zmnoži s številom pretočnih procesorjev na kartici da dobimo število grafičnih niti. Ugotovili smo, da se ob vrednostih nad 400 hitrost računanja hashev spreminja minimalno in dokaj sporadično, kljub temu, da je končna zmogljivost kartic pred preobremenitvijo presegla 1500 * 2560 = 4194304 grafičnih niti.

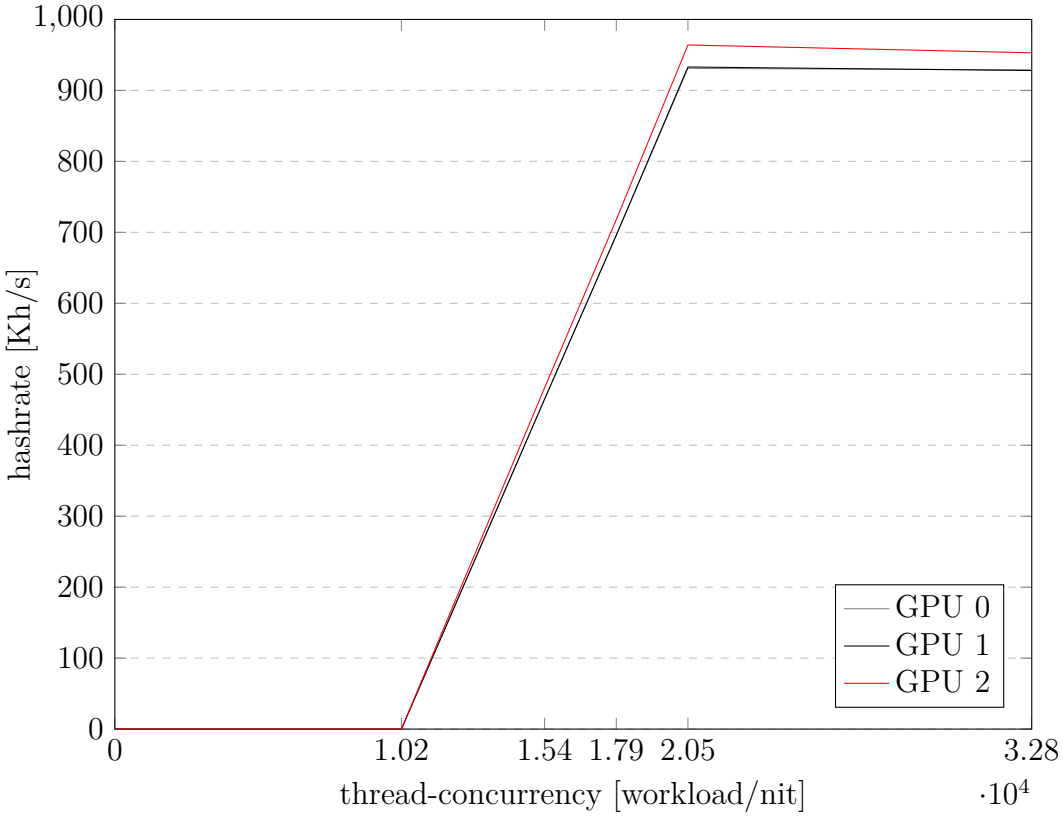
`"cl-filename":"kalroth"` Ime datoteke, ki vsebuje modificiran grafični kernel. Modificirani kerneli vsebujejo različne optimizacijske prijeme pri implementaciji Scrypt proof-of-work algoritma, kot je recimo razvijanje zank, drugačna definicija podatkovnih struktur, alternativen način preverjanja pogojev, spremenjen ritem oddajanja deležev in podobno. Izkaže se, da so izboljšave rezultatov nekonsistentne in se razlikujejo od ene fizične kartice do druge. Tako naprimer isti kernel, ki nam na eni ASUS kartici izboljša hitrost računanja hashev za 2%, na drugi rezultat poslabša za 3%. O enakih ugotovitvah poročajo tudi ostali, ki si prizadevajo za tak način optimizacije. Uporabljamo kernel, ki je privzeto nastavljen v naši različici CGMinerja in nam daje optimalne rezultate za vse tri kartice.

Večina konfiguracijskih parametrov, ki zagotavljajo optimalne rezultate pri računanju hashev je pridobljenih z dolgotrajno metodo poizkusov in napak, ker so rezultati sporadični ali pa so ti rezultati linearno vezani na frekvenco delovanja jedra grafičnih kartic. Zato sta izmed poprej podrobno opisanih parametrov za grafični prikaz zanimiva le `xintensity` ter `thread-concurrency`. Pri meritvah spodaj prikazanih rezultatov so vsi ostali parametri nastavljeni na poprej že opisane vrednosti, ki zagotavljajo optimalno delovanje.



Slika 4.3: Graf hitrosti računanja hashev v odvisnosti od xintensity

Kot vidimo na sliki 4.3, se hitrost računanja hashev hitro dvigne do neke sprejemljive vrednosti, optimum pa je dosežen pri `xintensity = 400`, pri kateri vrednosti je hitrost tudi stabilna. Če `xintensity` še naprej povečujemo, povprečna hitrost računanja sicer ostane enaka, a med rudarjenjem vse bolj niha, nato pa začne po `xintensity = 800` počasi padati.



Slika 4.4: Graf hitrosti računanja hashev v odvisnosti od thread-concurrency

Na zgornji sliki 4.4 vidimo, da se tudi pri višanju parametra `thread-concurrency` hitrost računanja hashev hitro dvigne, pride do optimuma in nato začne počasi padati. Treba je povedati, da so sicer same številke hitrosti računanja hashev že pri `thread-concurrency = 5120` blizu 930 Kh/s, toda z odstotkom sprejetih deležev blizu 0%. Podobno je pri naslednjih vrednostih, ki smo jih tukaj za namen bolj realnega prikaza obtežili z njihovimi odstotki sprejetih deležev.

Iz tabele 4.1 spodaj je razvidno, da različne optimizacije jedra ne prinašajo ponovljivih in uniformnih izboljšanj rezultatov pri hitrosti računanju hashev, kot smo že ugotovili pri konfiguraciji. Pri interpretaciji tabele je potrebno biti pazljiv na to, da večje število poslanih deležev `WU` v teku ali pa najvišja povprečna hitrost računanja hashev ni nujno najboljši rezultat, saj je pri potegovanju za nagrado pomemben tudi odstotek sprejetih deležev `WUE` ter odstotek napak po karticah `R`.

	GPU0	**GPU1**	**GPU2**	Poročilo
CGMiner Kalroth	931 Kh/s WU 830 R 3.8% 66°C 88°C vrm	935Kh/s WU 824 R 2.5% 69°C 89°C vrm	966 Kh/s WU 874 R 3.0% 79°C 61°C vrm	2832 Kh/s WU 2528 WUE 88.6% 5.7/min
Zuikkis	915 Kh/s WU 807 63°C 79°C vrm		956 Kh/s WU 835 78°C 60°C vrm	CRASH WUE 85.3%
SternStunde	928 Kh/s WU 833 R 2.5% 65°C 86°C vrm	920 Kh/s WU 833 R 4.4 % 67°C 87°C vrm	965 Kh/s WU 879 R 0.7% 78°C 60°C vrm	2813 Kh/s WU 2545 WUE 88.5% 9.48/min
Lantis	934 Kh/s WU 685 R 0% 65°C 86°C vrm	931 Kh/s WU 615 R 0% 68°C 87°C vrm	965 Kh/s WU 1000 R 3.5% 78°C 60°C vrm	2830 Kh/s WU 2363 WUE 81.7% 9.13/min
Kombinacija	925 Kh/s WU 831 R 1.7% 64°C 85°C vrm	940 Kh/s WU 860 R 1.5% 67°C 86°C vrm	966 Kh/s WU 843 R 2% 78°C 60°C vrm	2831 Kh/s WU 2531 WUE 88.6% 5.68/min

Tabela 4.1: Primerjava zmogljivosti rudarjenja med modificiranimi kerneli

5 Avtomatizacija maksimiranja dobička

5.1 Bazeni

Vse večja razpoznavnost v svetu ter skokovita rast konvertibilnih vrednosti kriptografskih valut sta v zadnjem času pripomogla k močnemu povečanju števila rudarjev v omrežjih in s tem k izrednem dvigu zahtevane težavnosti za potrditev blokov. Tako je dandanes verjetnost, da bi en sam uporabnik uspel potrditi blok izjemno majhna, natančneje bi za potrditev enega bloka Bitcoina z računsko močjo treh najmočnejših grafičnih kartic potreboval v povprečju 533,7 let, za blok Litecoina pa 178 dni. Z namenom, da bi zmanjšali naključno komponento rudarjenja in zagotovili redne ter pravično razdeljene nagrade pri rudarjenju, so se uporabniki začeli povezovati v t.i. 'poole' oziroma bazene. V bazenih več uporabnikov prispeva svojo računsko moč za potrjevanje trenutnega bloka. Ob uspehu je nagrada razdeljena med njih glede na njihov prispevani delež, upravljalec bazena pa običajno pobere minorno provizijo. Namesto, da bi en uporabnik poizkušal srečo več let in upal na veliko povračilo, lahko redno in z gotovostjo dobiva manjše delčke nagrade. Delež se določi na isti način kot pri potrjevanju blokov z dokazom proof-of-work, le da je ta zgolj v okviru bazena in je tu uporabljena znatno manjša zahtevnost. Pristopi k preprečevanju goljufanja s strani uporabnikov ali upravljalca se od bazena do bazena razlikujejo v podrobnostih, vsi pa temeljijo na sprotnem izplačevanju dokazanih deležev ob vsakem prestopu določenega praga zaslužka. Princip se je izkazal za zelo uspešnega in tako dandanes rudarjenje bolj znanih valut v celoti poteka v različnih manjših ali večjih bazenih.

5.2 Borze kriptografskih valut

Le leto po nastanku Bitcoina so se začele pojavljati prve borze, ki so ponujale menjavo le-tega v uradne denarne valute. Ena izmed prvih je bila tudi Bitstamp, borza ustanovljena s strani slovenskega študenta FOV v Kranju ter njegovega kolega, ki se je do danes zaradi dovršene zasnove, hitrih reakcij ter dobrih praks odnosov s strankami razvila v eno od dveh največjih Bitcoin borz na svetu. Povprečni dnevni pretok trgovanja na Bitstampu v trenutku pisanja tega teksta znaša 6172 BTC oziroma okoli $4.000.000, kar pri povprečni proviziji 0.3% nanese borzi blizu $363.000 na mesec. Nasprotni primer tega zloglasni Mt.Gox, prva Bitcoin borza na svetu, bazirana v Tokiu in zasnovana s strani angleškega programerja, ki je najprej nameraval vzpostaviti stran za trgovanje s kartami Magic The Gathering, a se je nato preusmeril na Bitcoin borzo, jo postavil in kmalu zatem prodal. Mt.Gox je bila sprva izjemno uspešna in je do leta 2013 vršila že 70% vseh Bitcoin transakcij, nato so se začeli vdori, nesrečna odkritja hroščev v protokolu ter sumljiva izginotja valute, ki so močno načeli borzin ugled. Februarja 2014 je borza ustavila poslovanje in razglasila stečaj, lastnik pa se je znašel v sodnih postopkih. Poleg Bitcoin borz so se s poplavo novih kriptovalut razvile tudi borze, ki omogočajo trgovanje z njimi in tako ustvarile okolje zelo podobno današnjim denarnim borzam. Ker so zaradi visoke volatilnosti novejših kriptovalut dobički pri trgovanju tu najvišji se tako večina trgovanja in špekulacij odvija na takšnih borzah. Kljub temu so vrednosti vseh alternativnih kriptovalut neposredno vezane na vrednost Bitcoina, saj je zaradi prepoznavnosti le-tega bolj smotrno menjati ostale kriptovalute najprej v Bitcoin in tega naprej v uradne valute, kot pa najti neposredno povpraševanje za svojo valuto. Tako Bitcoin predstavlja nekakšen most v pretoku vrednosti med svetom kriptografskih valut ter ‘zunanjim’ svetom.

5.3 Maksimizacija dobička

Kot kombinacija opisanih mehanizmov za bolj konsistentno izplačevanje nagrad rudarjenja ter trgovanja med množico kriptovalut v realnem času, so se okoli začetnih internetnih strani, ki so najprej vsebovale kalkulatorje za osnovne izračune verjetnosti potrditev blokov ter pričakovane dobičkonosnosti, razvila kompleksna analitična orodja. Le-ta neposredno preko programskih vmesnikov valut ter samih borz zbirajo podatke v realnem času ter jih obdelujejo v relevantne kazalce o zahtevnosti, dobičkonosnosti, obsegom trgovanja in podobno. Nekatere izmed njih služijo zgolj kot informativna pomoč uporabnikom, iz nekaterih pa so se razvili hibridi med bazeni in borznimi roboti, ki se imenujejo Multipooli. Multipooli delujejo na istem principu rudarjenja kot običajni bazeni, le da ti preklapljajo med potrjevanjem blokov različnih valut. Posebni algoritmi s pomočjo trenutnih podatkov o valutah in trgih ustvarijo projekcije dobičkonosnosti in nato računsko moč bazena usmerijo v potrjevanje blokov v tistem trenutku najbolj dobičkonosne valute. Ko zaznajo, da početje ni več smotrno, preklopijo na naslednjo optimalno valuto. S tem je tako v osnovi zagotovljena vsa programska zasnova za zagotavljanje maksimalnega dobička rudarjev.

6 | Zaključek

Uspešno smo sestavili in skonfigurirali napravo za potrjevanje blokov transakcij kriptografskih valut, ki to počne s pomočjo Scrypt proof-of-work algoritma. Opisali smo ideološko in tehnično ozadje zasnove ter delovanja kriptografskih valut, predstavili njihove prednosti, slabosti ter ranljivosti. Podrobno smo predstavili uporabljeno strojno opremo, načrt naše naprave ter konfiguracijo strojne opreme. Razložili smo tehnične podrobnosti potrjevanja transakcij na nivoju algoritmov, s primeri izvorne kode in komentarji. Predstavili smo uporabljeno programsko opremo, konfiguracijo le-te ter razložili in argumentirali izbrane nastavitve. Opisali smo mehanizme za maksimizacijo dobička ter internetne borze kriptografskih valut. Pokazali smo tudi izsledke meritev učinkovitosti sestavljene naprave. Ugotovili smo, da smo se s pomočjo naše konfiguracije uspeli pri eni od grafičnih kartic zelo približati teoretičnemu maksimumu zmogljivosti, medtem ko nas je pri ostalih dveh ovirala napaka proizvajalca v zasnovi kartic, zaradi katere se je na njima pregreval regulator napetosti. Ugotovili smo tudi, da poseganje v jedrno implementacijo Scrypt algoritma proof-of-work z namenom optimizacije ne daje konsistentnih in sledljivih rezultatov.

Še nedolgo tega si nismo mogli niti zamisliti, da bi storitve in materialne dobrine lahko namesto z denarjem ali bančno kartico plačevali s čim drugim, nečim kar je univerzalno, transparentno, hitro in varno. Nečim, nad čimer finančne inštitucije nimajo monopolnega nadzora ter upravljanja. Kriptografske valute so se v zadnjih letih najprej potihem, potem pa kar suvereno razvile v legitimno alternativo denarnemu poslovanju. Najprej so jih povzeli računalniški entuziasti, nato so postale zanimive uporabnikom, ki so potrebovali anonimnost pri trgovanju ali denarnih transferih. Prav slednji so verjetno s konverzijo svojih dobrin v krip-

tografske valute prvi začeli vrteti kolesje razvoja le-teh. Kmalu so priložnost za zaslužek zaslutili borzni posredniki ter investicijski bančniki in veliki pok, za katerega je bila potrebna zagonska kritična masa, se je zgodil. Vrednost Bitcoina in z njo vrednosti ostalih kriptovalut so poskočile v nebo in s tem pritegnile pozornost celotne populacije razvitega sveta. Začela se je tekma za rudarjenje valut in dobiček pri trgovanju, a s tem tudi globalna razpoznavnost valut in rast infrastrukture ter števila uporabnikov. Število unikatnih e-denarnic ves čas eksponentno narašča, prav tako število podjetij, ki za plačilo sprejemajo kriptovalute. Po svetu množično postavljajo Bitcoin avtomate, kjer lahko z denarjem ali bančnimi karticami kupimo Bitcoin, katerega tržna kapitalizacija je že presegla 10 miljard ameriških dolarjev.

Vsekakor so kriptovalute zanimivo in razvijajoče se področje. Rudarjenje se je v roku enega leta razvilo v svetovno manijo. Od začetkov rudarjenja na centralnih procesnih enotah domačih računalnikov smo prišli najprej do specializiranih takih rudarskih naprav, kot je opisana v tej nalogi in končno do specializirane strojne implementacije potrjevalnih algoritmov na FPGA in ASIC računalnikih. Masovna proizvodnja slednjih je dvignila zahtevnosti tako visoko, da je rudarjenje postalo dobičkonosno le še za lastnike takšnih računalnikov. Tako je recimo v času pisanja te diplomske naloge dnevni donos naše naprave padel iz 20€na 3€. Situacija se v svetu kriptografskih valut izjemno hitro spreminja in nemogoče je predvideti prihodnost.

Možnosti za razvoj kriptografskih valut in uporabo le-teh v vsakdanjem življenju so praktično neskončne. Rast uporabnikov, trgovcev in tržne kapitalizacije so indikatorji, ki kažejo na dolgoročno stabilizacijo in sprejetje kriptografskih valut. Manjša je tudi volatilnost vrednosti, kar bi posledično lahko vodilo v večje zaupanje s strani večjih inštitucij ter držav. Z malo sreče in pravilnim postopanjem odgovornih bi v daljnji prihodnosti lahko prišlo do fuzije kriptografskih valut ter uradnih denarnih valut ali celo popolne nadomestitve slednjih.

Literatura

[1] Daniel J. Bernstein. The salsa20 family of stream ciphers, December 2007. http://cr.yp.to/snuffle/salsafamily-20071225.pdf.

[2] Luigi Dadda, Marco Macchetti, and Jeff Owen. An asic design for a high speed implementation of the hash function sha-256 (384, 512). In *Proceedings of the 14th ACM Great Lakes symposium on VLSI*, pages 421–425. ACM Press, April 2004.

[3] Praveen Gauravaram, Lars R. Knudsen, Krystian Matusiewicz, Florian Mendel, Christian Rechberger, Martin Schlaeffer, and Soeren S. Thomsen. Groestl - a sha-3 candidate, March 2011. http://www.groestl.info/Groestl.pdf.

[4] Charles Lee. [ann] litecoin - a lite version of bitcoin. launched!, October 2011. https://bitcointalk.org/index.php?topic=47417.

[5] Litecoin.info. Litecoin, 2011. https://litecoin.info/Litecoin.

[6] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, November 2008. https://bitcoin.org/bitcoin.pdf.

[7] The National Institute of Standards and Technology (NIST). Third-round report of the sha-3 cryptographic hash algorithm competition, November 2012. http://nvlpubs.nist.gov/nistpubs/ir/2012/NIST.IR.7896.pdf.

[8] Colin Percival. Stronger key derivation via sequential memory-hard functions, May 2009. http://www.tarsnap.com/scrypt/scrypt.pdf.