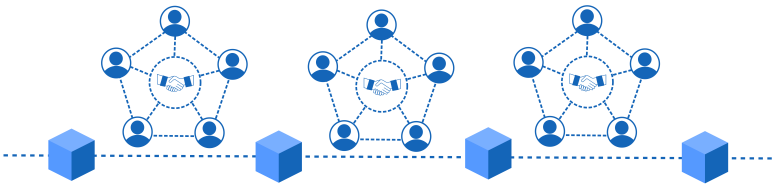
BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2022
http://excel.fit.vutbr.cz

Testovanie bezpečnosti a výkonnosti blockchainu Harmony pomocou simulácie

Juraj Holub



Abstrakt

V súčasnosti je blockchain populárnou technológiou používanou v distribuovaných aplikáciach, ktoré kladú dôraz na bezpečnosť (napríklad kryptomeny alebo elektronické voľby). Neustále prebieha vývoj nových blockchain protokolov s novými vlastnosťami. Vzhľadom na ich finančnú citlivosť je potrebné tieto nové protokoly podrobne preveriť z hľadiska bezpečnosti a výkonnosti ešte pred ich nasadením. Prirodzene sa pre tento účel naskytá simulácia. Cieľom tejto práce je simulovať blockchain protokol Harmony, ktorý poskytuje vysokú priepustnosť transakcií nezávisle od veľkosti siete. Simulácia sa zameriava na konsenzus protokol použitý v tomto blockchaine. Simulované boli konkrétne útoky na konsenzus ako aj všeobecná efektívnosť protokolu. Výsledky simulácie poukazujú na bezpečnostné riziko v podobe DDoS útoku. Na druhej strane, simulácia potvrdila vysokú priepustnosť transakcií pomocou mechanizmu sharding, a to bez výrazného zníženia bezpečnosti. Na záver je navrhnutá a odsimulovaná modifikácia protokolu, ktorá znižuje zraniteľnosť voči DDoS útoku. Čitateľ tejto práce sa oboznámi s posledným vývojom v technológii blockchain. Vytvorený nástroj môže poslúžiť ako základ pre simuláciu ďalších protokolov.

Kľúčové slová: Blockchain — Proof-of-Stake — Harmony — Simulácia

Priložené materiály: Stiahnuteľný kód

*xholub40@fit.vutbr.cz, *Fakulta informačních technologií, Vysoké učení technické v Brně*

1. Úvod

Blockchain je systém, ktorý nachádza využitie v distribuovaných aplikáciách s požiadavkou na vysokú bezpečnosť. Takéto systémy sú napríklad elektronické financie (kryptomeny), Internet-of-Thinks alebo bezpečné zdieľanie zdravotníckych údajov. Tradičné systémy pre tieto aplikácie poskytujú bezpečnosť založenú na centrálnej dôveryhodnej autorite. Naproti tomu, blockchain je decentralizovaný a jeho bezpečnosť je vystavaná na kryptografickom dôkaze [1]. V posledných rokoch preto vzniká veľké množstvo systémov založených na tejto technológii. V tomto smere je taktiež intenzívny akademický výskum. Avšak, aj napriek tomu je stále nedostatok kvalitných nástrojov [2], ktoré by tieto novo vznikajúce technológie vyhodnocovali z hľadiska bezpečnosti a výkonnosti. Simulácia blockchainu by pomohla odhaliť potenciálne nedostatky, ktoré sa typicky prejavujú až pri skutočne nasadenom systéme, v ktorom sú tisícky uzlov. Táto práca sa zaoberá simuláciou blockchain protokolu Harmony.

Decentralizovanosť blockchainu prináša nové technologické problémy. Jednou z veľkých výziev je komunikácia a zabezpečenie konzistentnosti stavu blockchainu naprieč všetkými uzlami v sieti. Do blockchainu sa neustále pridávajú nové dáta v podobe blokov.

Bez centrálnej autority však nie je triviálne určiť, ktorý uzol má pridať nový blok, a ktorý blok je aktuálne skutočne posledný. Protokol na bezpečné uznesenie o stave blockchainu sa nazýva konsenzus. Tento projekt analyzuje a pomocou simulácie experimentuje s konsenzus mechanizmom v blockchaine Harmony. Cieľom simulácie je posúdiť bezpečnosť a výkonnosť konsenzu z hľadiska rôznych scenárov v rozsiahlej sieti.

Aktuálne existuje niekoľko netriviálnych nástrojov určených na simuláciu blochainu. Existujúce riešenia sú podrobnejšie popísané v sekcii 3. Tento projekt rozšíril už existujúci nástroj Wittgenstein[1]. Vytvorené riešenie má architektúru klient-server. Samotný simulátor beží na strane serveru. Ten má definované REST API, pomocou ktorého sa dá spustiť simulácia s definovanou vstupnou konfiguráciou. Simulácia je implementovaná tak, aby každý uzol v systéme mal vlastný stav založený na svojom pozorovaní okolia. Dáta produkované simuláciou rozsiahlych sietí v dostatočne dlhom časovom intervale sa objemovo pohybujú v jednotkách gigabajtov. Preto si ich server priebežne ukladá do databázy[2]. Klientská aplikácia spúšťa simuláciu na serveri a po jej dokončení analyzuje dáta uložené v databáze nástrojmi určenými na dolovanie znalostí[3] z rozsiahlych dátových sád. Celé riešenie je možné spustiť podľa návodu v priložených materiáloch.

Tento článok v prvej časti poskytne potrebné teoretické znalosti o blockchaine. Následne systematicky porovná existujúce simulátory blockchainu. Ďalej je popísané fungovanie Harmony konsenzu spolu s teoretickou analýzou možných útokov. Nasleduje popis vytvoreného simulátoru. V poslednej časti je vykonaná sada simulačných experimentov. Práca na základe výsledkov simulácie ukazuje potenciál shardingu, ktorý poskytuje riešenie pre škálovateľnosť rozsiahlych blockchainových sietí. Výsledky simulácie ďalej demonštrujú náchylnosť Harmony na DDoS útok. Na záver je navrhnuté a simulované vylepšenie protokolu, ktoré túto zraniteľnosť redukuje.

2. Konzsenzus v blockchaine

Blockchain je komplexná technológia, ktorá pozostáva z mnohých vrstiev. Pre potreby tejto práce budeme uvažovať nasledujúci model abstrakcie blockchainu so štyrmi vrstvami [3]:

1. **Aplikačná vrstva** definuje využitie v konkrétnej službe (napríklad kryptomena).
2. **Dátová vrstva** (alebo tiež úložisko) definuje dátovú reprezentáciu transakcií a blokov.
3. **Konsenzus vrstva** definuje protokol, pomocou ktorého sa ustanovuje dohoda na stave blockchainu.
4. **Sieťová vrstva** predstavuje najnižšiu vrstvu abstrakcie a zaoberá sa komunikáciou medzi uzlami v sieti.

V tejto práci sa simuluje primárne konsenzus vrstva, avšak v určitej miere simulácia uvažuje aj dopad ostatných vrstiev blockchainu. Dôvodom je, že všetky spomenuté vrstvy sa navzájom ovplyvňujú. Napríklad, aplikačná vrstva určuje množstvo transakcií, ich veľkosť a veľkosť siete. Sieťová a dátová vrstva zase kladie fyzické hranice na priepustnosť systému. Tieto vlastnosti priamo ovplyvňujú výkonnosť a bezpečnosť konsenzus vrstvy.

Proof-of-Stake vs. Proof-of-Work

Väčšina aktuálnych blockchainov používa mechanizmus Proof-of-Work (ďalej PoW). Najznámejším zástupcom je protokol Bitcoin [1]. V tomto protokole môže publikovať nový blok ten uzol, ktorý vyrieši netriviálny kryptografický problém ako prvý. Dôveryhodnosť uzlov je teda dokazovaná investíciou výpočtového výkonu. PoW má však jeden dlhodobý problém a tým je spotreba energie. Niektoré zdroje[4] hovoria, že v roku 2021 pokrýva ťažba Bitcoinu 0,5 % celkovej spotreby elektrickej energie na svete. To je jeden z hlavných dôvodov, prečo má zmysel zaoberať sa protokolmi ako je Harmony, ktoré tento energetický problém nemajú.

Protokol Harmony zakladá svoju bezpečnosť, integritu a výkon na mechanizme Proof-of-Stake [3, 4] (ďalej len PoS). Táto metóda je založená na predpoklade, že vlastník veľkého množstva zdrojov v danom blockchaine je veľmi nepravdepodobným útočníkom, pretože svoje zdroje nechce ohroziť. Všetky dôležité činnosti a rozhodnutia preto robia vlastníci väčších zdrojov.

Harmony uznáša konsenzus pomocou hlasovania. Výhodou hlasovania je rýchla konzistencia a veľmi malá pravdepodobnosť vzniku vetiev reťazca. Na druhej strane, priepustnosť takýchto protokolov klesá s narastajúcim počtom uzlov. Pre porovnanie, Bitcoin nepoužíva hlasovanie, ale koncept lotérie. Lotéria dosahuje konsenzus náhodnou voľbou. Výhodou je jednoduchosť, pretože takýto proces nevyžaduje žiadnu

[1] https://github.com/ConsenSys/wittgenstein
[2] https://www.mongodb.com/
[3] https://pandas.pydata.org/
[4] https://www.nytimes.com/interactive/2021/09/03/climate/bitcoin-carbon-footprint-electricity.html

interaktívnu komunikáciu. Nevýhodou tohto prístupu je pomalá konzistencia a vysoké riziko vetvenia reťazca. Pre distribuované systémy je definovaný teorém popisujúci, že z vlastností konzistentnosť, dostupnosť a odolnosť voči čiastočnému výpadku je súčasne možné dosiahnuť maximálne dve z týchto troch vlastností (CAP teorém [5]). Z hľadiska CAP teorému, Harmony preferuje okamžitú konzistenciu pred dostupnosťou.

3. Predchádzajúca práca

V tejto sekcii sú zhodnotené existujúce nástroje určené pre simuláciu blockchainu.

SimBlock

Je to simulátor zameraný na PoW protokoly, ktorý umožňuje aj modulárne zmeniť konsenzus protokol. Posledná stabilná verzia pridala jednoduchú implementáciu PoS konsenzu[5]. Výhodou tohto simulačného nástroja je schopnosť simulovať aj rozsiahle siete s viac ako 10 000 uzlami. Na druhej strane, nástroj predpokladá, že všetky uzly sú poctivé. V aktuálnej implementácii teda neumožňuje experimentovanie so zlomyseľným správaním niektorých uzlov [6, 7].

Bitcoin Simulator

Tento simulátor je postavený nad platformou NS-3[6], ktorá slúži na diskrétnu simuláciu internetových systémov. Simulátor je určený a bol vyvinutý na experimentovanie s PoW protokolmi, a teda nepodporuje PoS konsenzus. V minulosti už ale bol použitý treťou stranou, ktorá rozšírila implementáciu o PoS protokoly Algorand, Casper FFG a Gasper (viď [8]). Z hľadiska simulácie PoW protokolov umožňuje simulátor rozlišovať rôzne typy uzlov. To umožňuje tomuto nástroju simulovať aj zlomyseľné chovanie niektorých uzlov [6].

BlockSim

Autori definujú ako základný cieľ tohto simulátora rozšíriteľnosť o ďalšie protokoly. Všeobecnosť architektúry by mala umožňovať simuláciu rôznych protokolov. Simulátor poskytuje abstraktnú vrstvu, nad ktorou je možné implementovať konkrétny protokol. Simulátor by teda mal byť rozšíriteľný aj o simuláciu PoS protokolov. A predsa, vývojári projektu deklarujú, že nie je možné analyzovať špecifickú sekvenciu správ, čo je pre simuláciu hlasovacieho konsenzu značný nedostatok [9].

[5] https://github.com/dsg-titech/simblock/releases/tag/v0.8.0

[6] https://www.nsnam.org/

Wittgenstein

Ide o nástroj určený priamo na simuláciu konsenzus protokolov alebo všeobecne distribuovaných algoritmov. Nástroj umožňuje definovať vlastnú sieť s definovaným chovaním uzlov. Uzly je možné geograficky rozmiestniť, a tak ovplyvňovať ich latenciu. Taktiež je možné definovať časť uzlov ako škodlivých a určiť pre ne vlastné chovanie. Projekt poskytuje jednoduché rozhranie pre definíciu ľubovoľného distribuovaného algoritmu založeného na zasielaní správ. Simulácia ako výstup poskytuje rôzne metriky relevantné pre hlasovací konsenzus a to zvlášť pre každý uzol (množstvo zaslaných správ, prenesených dát a podobné).

Porovnanie a zhodnotenie

Tabuľka 1 sumarizuje a porovnáva popísané simulátory. Táto práca sa zameriava na simuláciu konsenzus vrstvy postavenej na mechanizme PoS. Najrozsiahlejšie *open source* projekty v oblasti blockchain simulácie (SimBlock, Bitcoin Simulator a BlockSim) sa však prioritne venujú simulácii PoW protokolov ako je Bitcoin. Len simulátor BlockSim umožnuje základnú podporu implementácie PoS konsenzu. Ani jeden z týchto nástrojov neposkytuje žiadne výstupné metriky relevantné pre vyhodnocovanie PoS konsenzu. Naproti tomu, simulátor Wittgenstein je priamo určený na simuláciu protokolov založených na hlasovaní a PoS. Poskytuje taktiež vhodné rozhranie pre nastavenie chovania škodlivých uzlov. Pre účely tejto práce bol teda zvolený simulátor Wittgenstein, ktorý bol rozšírený o protokol Harmony.

4. Harmony

Pokiaľ nie je uvedené inak, informácie o Harmony protokole vychádzajú z oficiálnej dokumentácie tohto projektu [10, 11].

Pre vygenerovanie nového bloku používa Harmony hlasovací protokol FBFT (*Fast Byzantine Fault Tolerance*), ktorý je založený na známom hlasovacom protokole PBFT (*Practical Byzantine Fault Tolerance* [12]). Veľkou nevýhodou PBFT je jeho nízka škálovateľnosť, pretože má kvadratickú časovú zložitosť vzhľadom k počtu uzlov. FBFT ale namiesto zasielania hlasov pomocou broadcastu používa prahový digitálny podpis BLS (*Boneh–Lynn–Shacham* [13]). BLS je špeciálna schéma digitálneho podpisu, kde je n účastníkov, a každý vlastní časť privátneho kľúča. Ktorýkoľvek účastník môže použiť svoju časť tajného kľúča na čiastočné podpísanie správy M. Kompletný podpis môže byť zostrojený, ak aspoň t účastníkov poskytlo svoju časť podpisu. Potom hovoríme o t-z-n prahovom podpise. BLS podpis znižuje komunikačnú záťaž FBFT

Tabuľka 1. Porovnanie blockchain simulátorov.

		Simulátor			
		SimBlock	**Bitcoin Simulator**	**BlockSim**	**Wittgenstein**
Dátová vrstva	Generovanie transakcií	✓	✓	✓	✓
Konsenzus vrstva	Proof-of-Work	✓	✓	✓	✓
	Proof-of-Stake	×	×	✓	✓
	Simulácia útočiacich uzlov	×	✓	×	✓
	Interval pre distribúciu bloku	✓	✓	✓	✓
Sieťová vrstva	Veľké siete (>1000 uzlov)	✓	✓	×	✓
	Geografická distribúcia uzlov	✓	✓	✓	✓
	Bandwith	✓	✓	✓	✓
	Latency	✓	✓	✓	✓
	Veľkosť transakcie	✓	✓	✓	✓
Výstup simulácie	Proof-of-Work metriky	✓	✓	✓	×
	Proof-of-Stake metriky	×	×	×	✓
	Througput (TPS)	×	×	×	×
	Throughput (bytes)	×	×	×	✓
Iné vlastnosti	Programovací jazyk	Java	C++	Python	Java
	Vytvorenie projektu	06/2019	04/2016	04/2019	10/2018
	Posledná zmena v repozitári	02/2021	10/2016	05/2021	01/2020
	Podporované protokoly	Bitcoin Litecoin Dogecoin	Bitcoin Litecoin Dogecoin	Bitcoin Ethereum	Ethereum CasperIMD Dfinity Handel

na lineárnu. Protokol prebieha nasledovne:

1. Vodca rozošle nový blok všetkým validátorom.
2. Validátori overia blok, podpíšu jeho hlavičku svojím digitálnym podpisom a pošlú späť vodcovi.
3. Keď vodca prijme toľko podpisov, že ich autori vlastnia aspoň $\frac{2}{3}$ podielu, agreguje podpisy do jediného prahového digitálneho podpisu BLS. Ten broadcastuje spolu s bitmapou indikujúcou validátorov, ktorí podpísali.
4. Každý validátor overí, že prahový podpis obsahuje požadované $\frac{2}{3}$ podielu. Až v tejto chvíli validátor verifikuje transakcie v dátovom obsahu bloku, ktorý bol zasielaný už v kroku 1. Ak všetko súhlasí, podpíše správu z kroku 3 a pošle ju späť vodcovi.
5. Vodca opäť čaká na podpisy. Keď získa podpisy s váhou $\frac{2}{3}$ celkového podielu, opäť ich broadcustuje (rovnako ako v kroku 3).
6. Keď validátor prijme druhý BLS podpis, považuje blok za finálny.

Sharding

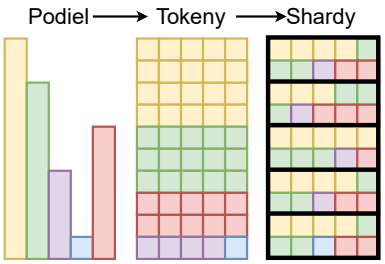
Harmony používa konsenzus založený na hlasovaní. Kvôli efektívnosti hlasovania aj v rozsiahlej sieti používa Harmony mechanizmus *sharding*. Sharding rozdeľuje uzly v sieti na podskupiny (shardy), ktoré hlasujú samostatne. Tým sa znižuje komunikačná záťaž v sieti. Rozdelenie uzlov do shardov je vždy na dobu, ktorá sa nazýva epocha. Epocha odpovedá času potrebnému na vygenerovanie 32 768 blokov, čo je približne 18,2 hodiny[7]. Počas celej epochy sa vodca hlasovania v jednotlivých shardoch nemení. Doba na vygenerovanie jedného bloku sa nazýva slot a Harmony slot aktuálne trvá približne 2 sekundy.

Harmony je PoS blockchain, pretože hlasovacie právo uzlov je priamo úmerné zdrojom, ktoré vložia. Uzly, ktoré chcú hlasovať v tejto epoche musia vložiť svoj podiel už v epoche $e-1$. Celkový vložený podiel v epoche $e-1$ je rozdelený na konštantne veľké tokeny (hlasovacie lístky). Hodnota jedného hlasovacieho lístka t v epoche e je určená rovnicou 1, kde S_{e-1} je celkový podiel vložený v epoche $e-1$, n je počet shardov a λ je bezpečnostný parameter. Harmony doporučuje nastaviť $\lambda > 600$.

$$t = \frac{S_{e-1}}{n.\lambda} \tag{1}$$

Na začiatku epochy e je distribuované vygenerované náhodné číslo, na základe ktorého sa urobí permutácia všetkých tokenov. Získaná permutácia je rozdelená na n rovnakých častí, kde n je počet shardov. Časť i, kde $1 \leq i \leq n$, predstavuje validátorov a ich hlasovací podiel v sharde i. Tento proces ilustruje obrázok 1. Môžeme vidieť, že distribúcia hlasovacieho podielu v jednotlivých shardoch sa približuje skutočnému podielu. Za vodcu shardu v epoche je určený podielnik, ktorý vlastní prvý hlasovací lístok v diele hlasovacích lístkov určených pre tento shard. Pravdepodobnosť, že podielnik bude zvolený za vodcu shardu by mala byť priamo úmerná množstvu podielu, ktorý vložil (princíp PoS).

[7] https://docs.harmony.one/home/network/validators/definitions/epoch-transition



Obrázok 1. Rozdelenie hlasovacieho podielu validátorov medzi všetky shardy (prevzraté z [11]).

5. Útoky na konsenzus

Nasledujúca sekcia popisuje niekoľko najznámejších útokov na konsenzus protokoly, ktoré sú relevantné pre protokol Harmony.

Ovládnutie konsenzu útočníkmi

Útočníci dokážu utvoriť konsenzus bez poctivých uzlov. Príkladom takéhoto útoku pre PoS je ovládnutie majority podielu v blockchaine.

Harmony FBFT hlasovanie je z rodiny protokolov byzantskej chyby. Pre tieto protokoly platí, že na prekazenie hlasovania je potrebné vlastniť aspoň $\frac{1}{3}$ hlasov a pre ovládnutie hlasovania aspoň $\frac{2}{3}$ hlasov. V prípade Harmony nejde o počet hlasujúcich uzlov, ale o ich podiel.

Zdvojnásobenie výdavkov

Zdvojnásobenie výdavkov (anglicky *double spending*) je útok, ktorý vzniká vytvorením dvoch alebo viac konfliktných blokov. Tieto bloky vytvárajú tzv. vetvy (anglicky *forks*) a spôsobujú nekonzistenciu stavu.

V prípade Harmony vetvenie nevzniká (bolo by potrebné ovládnutie $\frac{2}{3}$ podielu).

Útok na podskupiny uzlov

Harmony konsenzus rozdeľuje celú sieť na podskupiny uzlov (anglicky *sharding*) s cieľom znížiť komunikačnú záťaž pri hlasovaní. Tento prístup však môže viesť k zníženie bezpečnosti. Množstvo spolupracujúcich uzlov v jednej takejto podskupine je oveľa menší než v celej sieti. Pre útočníka môže byť preto jednoduchšie ovládnuť takúto podskupinu ako celú sieť. Bezpečnosť Harmony shardingu je podrobená simulačným experimentom popísaným v sekcii 7.

Ovplyvnenie volieb

FBFT hlasovací protokol funguje tak, že jeden uzol je určený ako vodca. Ten vedie hlasovanie a ostatné uzly (validátori) s ním spolupracujú. Roľa vodcu má rôzne výhody. Harmony odmeňuje vodcu podielom na poplatkoch za transakcie, ktoré boli publikované v jeho bloku. Z toho dôvodu môže útočník chcieť ovplyvniť voľbu vodcu vo svoj prospech.

Harmony rozdeľuje uzly do shardov pomocou náhodného čísla, ktoré bolo vygenerované distribuovane a na jeho generovaní sa musí podieľať toľko uzlov, aby vlastnili aspoň $\frac{1}{3}$ podielu. Útočník teda musí vlastniť takto veľký podiel, aby mal potenciálnu šancu ovplyvniť rozdelenie do shardov. Z hľadiska PoS je teda rozdelenie do shardov bezpečné.

Náhodné rozdelenie do shardov je všeobecne najbezpečnejší spôsob, ktorý efektívne bráni útoku na podskupinu uzlov. Harmony sharding navyše rozdeľuje uzly do shardov tak, aby v sharde bolo zachované pôvodné rozdelenie podielu.

Útok na vodcu

Pre FBFT hlasovanie je potrebné, aby jeden uzol zastával roľu vodcu. Táto roľa predstavuje dočasné zvýšenie právomoci pre daný uzol, pretože bez neho vo zvolenom čase nemôže vzniknúť nový blok. Harmony vytvára rozvrh vodcov na dlhšie časové obdobie. Takýto rozvrh je dopredu známy, čo zjednodušuje a zrýchľuje celý konsenzus. Na druhej strane, útočníci dopredu vedia, ktorý uzol bude v konkrétnom čase vodcom. Útočník potom môže v danom čase vykonať DDoS útok na vodcu a znemožniť tým hlasovanie. Harmony používa rovnakého vodcu po dobu celej epochy (viac ako 18 hodín). To značne zjednodušuje prípadný útok. Celá bezpečnosť musí byť potom presunutá na sieťovú vrstu v podobe firewallu, ktorý bude takýto útok znemožňovať. Samotná konsenzus vrstva žiadnu ochranu neposkytuje.

6. Vytvorený simulátor

Táto práca rozšírila Simulátor Wittgenstein o protokol Harmony. Wittgenstein sa skladá z nasledujúcich modulov, ktoré poskytujú základnú funkcionalitu potrebnú pre simuláciu konsenzu v blockchaine: protokol, sieť, uzol a správa.

Protokol je vstupné rozhranie pomocou, ktorého sa definuje konkrétny distribuovaný algoritmus (v našom prípade Harmony). Každý protokol sa skladá zo siete, uzlov a definície správ.

Sieť sa skladá z množiny uzlov. Simulátor umožňuje definovať štruktúru P2P siete. Ďalej je možné definovať geografické rozloženie uzlov, ktoré bude mať dopad na latenciu zasielaných správ. V sieti je možné zasielať broadcastové aj unicastové správy.

Uzol reprezentuje jeden bod v blockchain sieti. Uzol definuje komunikačné rozhranie zo sieťou, vlastný stav a identifikáciu. Ďalej je možné rozšíriť funkcionalitu uzlu o ľubovoľné ďalšie vlastnosti. Pre konkrét-

ny protokol je možné definovať rôzne uzly s rozličným chovaním. Samotný simulátor uchováva štatistiky o každom uzle (napríklad počet odoslaných a prijatých správ, bytov). V prípade Harmony uzlu si každý uzol uchováva informáciu o tom v ktorých shardoch je a aký má podiel v Proof-of-Stake blockchaine.

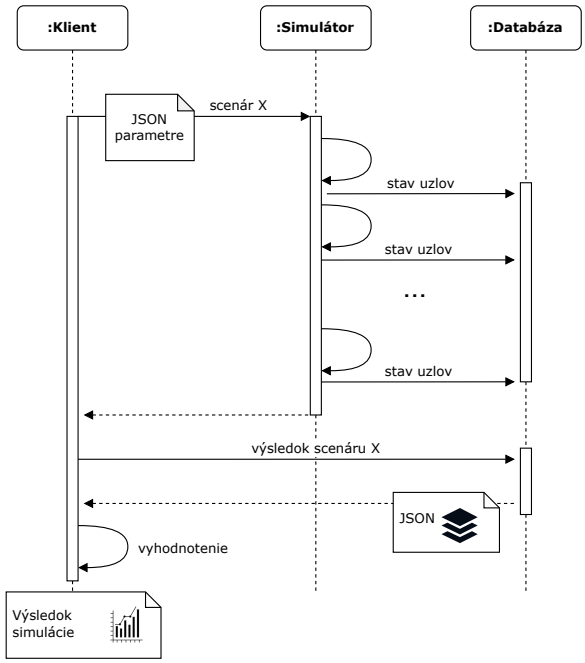
Správa je rozhranie, ktoré umožňuje definovať formát a veľkosť zasielaných správ. Toto rozhranie umožňuje definovať protokol zasielania správ (v prípade Harmony protokol FBFT ako bol popísaný v sekcii 4).

Samotný simulátor Wittgenstein je jednovláknová aplikácia, ktorá nie je náročná na výpočtový výkon, ale skôr na pamäť. Každý uzol v simulovanej sieti si potrebuje priebežne zaznamenávať svoj stav aby bolo možné na konci simulácie zhodnotiť celý priebeh. Pre väčšiu sieť a dlhšie časové obdobie sa stav uzlov pohybuje v jednotkách gigabajtov. Táto práca preto k simulátoru pridala databázu. Jednotlivé uzly svoj stav priebežne zaznamenávajú do databázy a neukladajú si ho v programe. To znižuje nároky programu na operačnú pamäť. Simulátor je vďaka tomu možné spúšťať aj na bežných osobných počítačoch[8]. Z užívateľského hľadiska nie je potrebné serverovú časť nijako konfigurovať. Server (simulátor aj databáza) je kontajnerizovaný do dockeru[9], ktorý si stiahne všetky potrebné knižnice a závislosti (vrátane samotnej databázy `mongoDB`). Simuláciu protokolu Harmony je možné parametrizovať sadou vstupných argumentov. Podporované parametre popisuje tabuľka 2.

Okrem samotného simulátoru, ktorý prestavuje serverovú aplikáciu, obsahuje vytvorené riešenie aj klientskú aplikáciu ktorá spúšťa jednotlivé simulačné scenáre a vyhodnocuje ich výsledky. Klient je konzolová aplikácie vytvorená v programovacom jazyku `Python`. Sekvenčný diagram na obrázku 2 ukazuje proces simulácie. Užívateľ inicializuje simuláciu zvoleného experimentu pomocou klienta. Ten spustí simuláciu Harmony z definovanou sadou vstupných parametrov (podľa daného scenára). Po dokončení simulácie klient analyzuje priebeh simulácie uložený v databáze. Podľa zvoleného experimentu na záver poskytne výsledok simulácie (typicky v podobe grafu).

7. Simulačné experimenty

V nasledujúcej sekcii sú zhrnuté výsledky simulácie Harmony pomocou vytvoreného nástroja (viď sekcia 6). Záujemca si môže ľubovolný experiment zreprodukovať. Simulátor, priložený k tejto práci, obsahuje `README` návod, ktorý vysvetľuje ako simuláciu spustiť (simulácia však môže trvať pre niektoré experimenty



Obrázok 2. Sekvenčný diagram zobrazuje proces simulácie a vyhodnotenie jej výsledkov. Klient, simulátor a databáza medzi sebou komunikujú pomocou REST API.

aj niekoľko hodín). Aktuálny Harmony blockchain má približne 1000 uzlov a 4 shardy. Simulované experimenty preto primárne uvažujú práve tieto parametre.

Proof-of-Stake

Harmony sharding distribuuje celkové hlasovacie právo približne rovnomerne do všetkých shardov. S toho vyplýva, že jeden uzol môže byť pridelený do viacerých shardov. Presnosť distribúcie medzi shardy je určené parametrom λ (viď sekcia 4). Pre overenie bezpečnosti shardingu bola simulácia spúšťaná s rôznymi hodnotami λ. Pre potreby simulácie sa uvažovalo, že v každej epoche uzly vlastnili približne rovnaký podiel. Takéto rovnomerné rozdelenie hlasovacieho práva je ideálny stav z hľadiska decentralizovanosti. Simulácia porovnáva celkový podiel s tým, ktorý bol skutočne pridelený jednotlivým uzlom v každom shardе.

Výsledok simulácie zhŕňa teplotná mapa na obrázku 3, ktorá zobrazuje v jednotlivých stĺpcoch výsledok simulácie pre konkrétne λ. Hodnota μ reprezentuje priemerný počet uzlov v jednom sharde pre dané λ (môžeme vidieť, že μ rastie úmerne k veľkosti λ). Jednotlivé riadky ukazujú nameranú koreláciu medzi celkovým hlasovacím podielom a tým, ktorý bol skutočne pridelený uzlom v danom sharde. Na koreláciu bol použitý Spearmanov[10]korelačný koeficient. Mô-

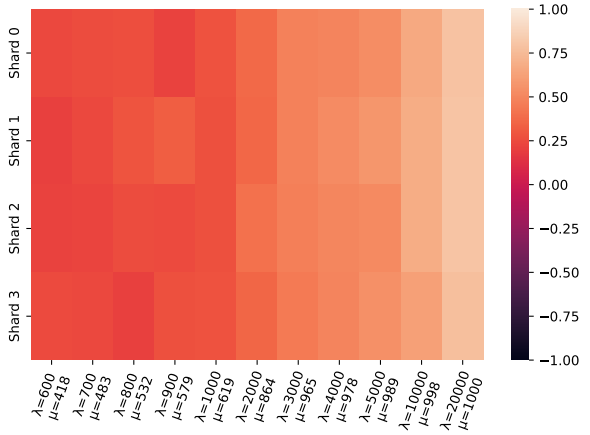
[8] Testované na: CPU Intel i5-8365U 1.60GHz, RAM 16 GB

[9] https://www.docker.com/

[10] https://www.sciencedirect.com/topics/mathematics/spearman-correlation

Tabuľka 2. Vstupné parametre simulátoru Wittgenstein pre protokol Harmony.

Parameter	Typ	Význam
`networkSize`	int	Počet uzlov v sieti.
`slotDurationInMs`	int	Dĺžka slotu v ms.
`epochDurationInSlots`	int	Počet slotov v epoche.
`numberOfEpochs`	int	Počet epoch.
`expectedTps`	int	V sieti vzniká dostatok transakcií na to aby bolo možné dosiahnuť očakávané TPS.
`uniformStakeDistribution`	bool	Ak `true` tak všetky uzly vlastnia podiel pridelený z rovnomerného rozdelenia. V opačnom prípade je podiel rozdelený tak ako v skutočnej sieti Harmony.
`txSizeInBytes`	int	Priemerná veľkosť transakcie v bytoch.
`blockHeaderSizeInBytes`	int	Veľkosť hlavičky bloku v bytoch.
`mongoServerAddress`	string	Adresa databázy (mongoDB).
`numberOfShards`	int	Počet shardov.
`ddosAttack`	bool	DoS útok na vodcov shardov.
`byzantineStake`	float	Interval v rozmedzí $\langle 0,1\rangle$, ktorý určuje percentuálny podiel byzantských uzol. V sieti sa vyberie taká množina uzlov aby spoločne vlastnili práve takýto podiel.
`lambda`	int	Nastavenie bezpečnostného parametru λ.



Obrázok 3. Teplotná mapa pre Spearmanov korelačný koeficient medzi skutočným podielom uzlov a hlasovacím právom jednotlivých shardoch v závislosti na hodnote λ.

žeme vidieť, že rastúce λ zvyšuje podobnosť hlasovacieho podielu v shardoch voči skutočnému. Z teplotnej mapy je vidieť, že hlasovací podiel v jednotlivých shardoch začína byť významne podobný skutočnému až pre $\lambda > 3000$. Pri takto vysokej λ sú v každom sharde takmer všetky uzly (pre $\lambda > 3000$ je $\mu \approx 1000$). To však úplne odstraňuje potenciál škálovatelnosti, a naopak ešte zvyšuje komunikačnú záťaž. Vidíme, že ak je $\lambda = 600$, tak v každom sharde je približne 400 uzlov. Experiment poukázal na to, že rozdelenie podielu medzi shardy sa nezanedbateľne líši od skutočnej distribúcie podielu.

Útok na podskupinu uzlov

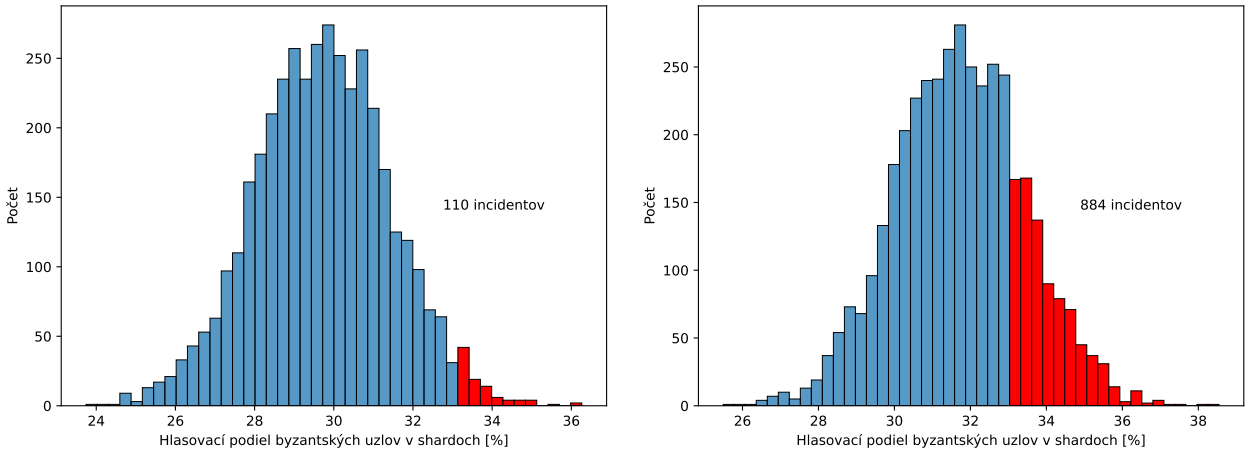
Uvažujme incident, že nepoctivé uzly (ďalej len byzantské) vlastnia menej ako $\frac{1}{3}$ podielu. Ak by Harmony nepoužíval sharding, nemal by dostatočný podiel na manipuláciu blockchainu. Vďaka nedokonalosti prerozdeľovania hlasovacieho práva získajú neprávom v niektorom sharde dostatočný podiel (teda $\frac{1}{3}$) na prekazenie konsenzu po dobu celej epochy. Pre určenie ako často by mohol nastať tento incident bola vykonaná simulácia 1000 epoch, a teda 1000 redistribúcii do shardov. Pre simuláciu bolo nastavené λ na odporúčanú hodnotu 600. Celá simulácia bola postupne opakovaná s narastajúcim množstvom byzantských uzlov.

Výsledok simulácie zhŕňajú histogramy na obrázku 4, ktoré zachytávajú distribúciu byzantského hlasovacieho podielu v shardoch. V prípade 4a je 30 % celkového podielu byzantského a v prípade 4b ide až o 32 %. V oboch scenároch má distribúcia byzantského hlasovacieho podielu očakávanú strednú hodnotu. Avšak, môžeme vidieť, že rozptyl je pomerne vysoký. Čím viac sa celkový byzantský podiel približuje $\frac{1}{3}$, tým častejšie tieto incidenty nastávajú. Treba ale poukázať na to, že ak bol byzantský podiel menší ako 28 %, tak takéto incidenty vôbec nenastávali. Experiment ukazuje na dobré zabezpečenie systému voči ovládnutiu podskupiny uzlov.

Priepustnosť

Následujúci simulačný scenár vyhodnocuje priepustnosť transakcií v Harmony. V simulácii bolo uvažované, že veľkosť hlavičky každého bloku je 80 B. Veľkosť transakcie závisí na aplikácii, ktorú blockchain poskytuje. Simulácia predpokladá kryptomenu podobnú Bitcoinu. Transakcia je preto priemerne veľká 670 B (priemerná veľkosť Bitcoin transakcie v roku 2021 vypočítaná z dostupných štatistík[11]). Simulácia ďalej predpokladá, že v sieti vzniká dostatok transakcií na to, aby bolo možné do každého bloku v každom sharde pridať približne 600 nových transakcií. Každý blok by teda mal mať približne 335 KB. Hodnota λ bola

[11] https://www.blockchain.com/charts



(a) 30 % byzantský podiel

(b) 32 % byzantský podiel

Obrázok 4. Histogram hlasovacieho podielu byzantských uzlov za 1000 epoch. Červená časť predstavuje incidenty, pri ktorých bol pridelený podiel v sharde neprávom väčší ako $\frac{1}{3}$.

nastavená na odporúčanú hodnotu 600. Harmony doporučuje, aby počet shardov v sieti bol určený podľa vzorca $\frac{n}{250}$, kde n je počet uzlov v sieti. Simulácia bola preto opakovaná postupne pre narastajúci počet uzlov a shardov (2 shardy a 500 uzlov, 4 shardy a 1 000 uzlov, ..., 40 shardov a 10 000 uzlov).

Výsledok simulácie sumarizujú grafy na obrázku 5. Môžeme vidieť, že priepustnosť transakcií za slot sa lineárne zvyšuje s narastajúcim počtom shardov. Z tohto hľadiska sharding skutočne umožňuje efektívne škálovať veľkosť siete. Na druhej strane, graf taktiež ukazuje množstvo dát v MB, ktoré každý uzol priemerne prijal za dobu jedného slotu. Komunikačná záťaž konsenzu rastie s narastajúcou veľkosťou siete aj napriek shardingu. Ak má sieť 40 shardov, tak uzol počas jedného slotu prijme približne 15 MB (teda 7,5 MB/s). Uvažujme, že celá sieť používa minimálne 1 Gbps linku. Každý uzol potom môže prijať približne 125 MB/s. Komunikačná záťaž spojená s hlasovaním je teda v takejto konfigurácii jednoznačne zvládnuteľná.

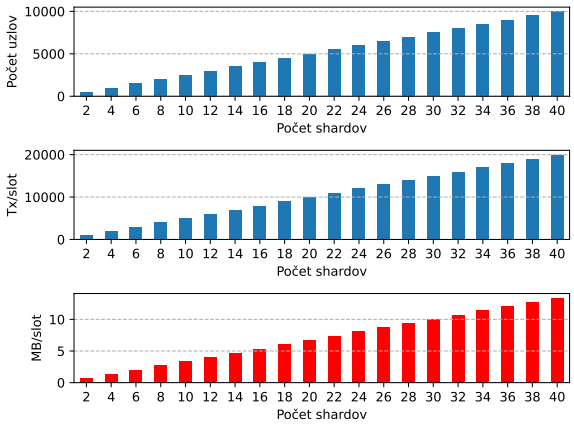
DDoS útok na vodcu

Následujúci experiment ukazuje náchylnosť Harmony na DDoS útok. Simulovaná sieť má 1000 uzlov a štyri shardy. Počas epochy je pomocou DDoS útoku dočasne znemožnená komunikácie pre vodcu každého shardu. Bez vodcu ostatné uzly nemôžu uzniesť konsenzus. Útok je stupňovaný: postupne sa útočí na 1,2,3 až 4 vodcov súčasne.

Výsledky experimentu zobrazuje obrázok 6, ktorý ukazuje množstvo transakcií finalizovaných v jednotlivých slotoch. V slote 125 začal útok na jedného vodcu. Od slotu 250 boli pod útokom dvaja vodcovia. V slote 375 už traja a od slotu 500 všetci štyria. Z grafu je vidieť, že v týchto momentoch vždy došlo k poklesu finalizovaných transakcií približne o $\frac{1}{4}$. Z experimentu je vidieť, že stačí intenzívny útok len na 4 uzly v sieti s 1000 uzlami a celý blockchain prestáva fungovať. Navyše je tento útok možné vykonať bez akéhokoľvek podielu v blockchaine. Útočníci teda neriskujú žiadnu stratu zdrojov.

Navrhované zlepšenie

Predchádzajúci experiment demonštruje pomerne kritickú bezpečnostnú zraniteľnosť v podobe DDoS útoku na vodcu. Existujú aj konsenzus protokoly, ktoré vodcu zvolia nepredvídateľne. Protokol Algorand [14] spĺňa túto vlastnosť pomocou kryptografickej schémy VRF (*Verifiable Random Function* [15]). Kľúčovou vlastnosťou tejto schémy je, že vodca nie je dopredu známy. Zároveň je pravdepodobnosť voľby vodcu priamo úmerná jeho podielu (princíp PoS). Nahradenie rozvrhu vodcov za schému použitú v Algorande by odstránilo túto bezpečnostnú slabinu.



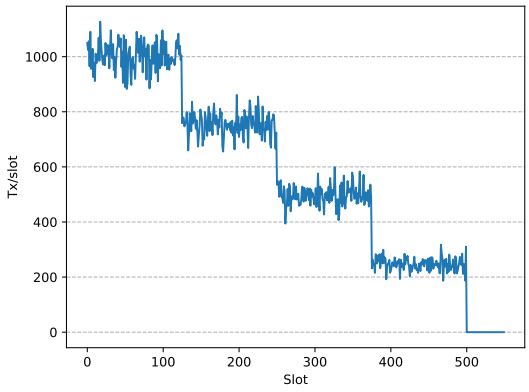
Obrázok 5. Priemerné množstvo dát, ktoré uzol prijme za jeden slot v závislosti na veľkosti siete.

Nasledujúci simulačný scenár opakuje DDoS útok. Simulácia stále uvažuje rovnaký postup pre rozdelenie uzlov do shardov. Ale v jednotlivých shardoch už nie je jediný vodca po celú dobu epochy. Namiesto toho je pre každý slot vždy použitá nepredvídateľná voľba vodcu pomocou VRF. Útočníci teda už nemôžu útočiť na konkrétny uzol. Najúčinnejší spôsob útoku je preto zvoliť niekoľko uzlov s najväčším podielom v zvolenom sharde. Simulácia uvažuje, že útočníci si zvolia vždy päť najbohatších uzlov v každom sharde. Výsledok simulácie zobrazuje obrázok 7. V pôvodnom protokole stačil útok na štyri uzly a sieť prestala propagovať transakcie. V pozmenenom protokole sa v poslednej fáze útočí na 20 uzlov súčasne a priepustnosť transakcií nie je takmer vôbec ovplyvnená. Občasný pokles transakcií je zapríčinený tým, že útokom sa náhodou podarilo zasiahnuť aktuálneho vodcu v danom sharde.

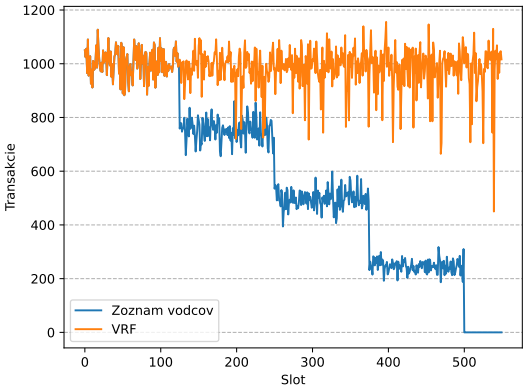
8. Záver

Táto práca sa zaoberala simuláciou bezpečnosti a výkonnosti blockchain protokolu Harmony. Sekcia 2 poskytla teoretický základ znalostí potrebných pre fungovanie PoS konsenzu, ktorý bol predmetom tejto práce. Následne sekcia 3 systematicky porovnala existujúce nástroje určené pre simuláciu blockchainu. Samotný protokol Harmony bol popísaný v sekcii 4. Potenciálne útoky na Harmony boli teoreticky analyzované v sekcii 5. Ďalej bol v sekcii 6 priblížený vytvorený simulačný nástroj. Na záver boli v sekcii 7 popísané simulačné experimenty, ktoré experimentovali s bezpečnosťou, výkonnosťou a škálovateľnosťou protokolu.

Prvý experiment simuloval útok na podskupinu uzlov (shard). V simulácii útočníci vlastnili 30 % podielu. Z hľadiska PoS takýto podiel neumožňuje prekaziť konsenzus. Avšak, simulácia 1000 epoch



Obrázok 6. Postupne narastajúci útok na vodcov hlasovania.



Obrázok 7. Porovnanie priepustnosti transakcií pri DDoS útoku v prípade pôvodného rozvrhu vodcov a voľby vodcu pomocou VRF.

ukázala, že v 2,75 % prípadoch získajú útočníci v shard neprávom viac ako 33 % podielu. Tým získajú možnosť prekaziť konsenzus v danej podskupine uzlov. Tieto výsledky poukazujú na dobrú bezpečnosť shardingu z hľadiska pravdepodobnosti nespravodlivého ovládnutia podskupiny uzlov.

Druhý experiment sa zameral na priepustnosť transakcií. Simulovaná sieť o veľkosti 10 000 uzlov a 40 shardov mala priepustnosť 20 000 transakcií za slot. Harmony teda v tejto konfigurácii poskytuje priepustnosť viac ako 10 000 TPS. Experiment ukazuje, že rastúca sieť a zväčšujúci sa objem transakcií zvyšuje komunikačnú záťaž. Aj napriek tomu, sharding umožňuje efektívne škálovať sieť lebo rast je len lineárny.

Ďalší experiment dokázal zraniteľnosť vodcu hlasovania. Útokom na vodcu (jediný uzol) sa zníži priepustnosť transakcií o $\frac{1}{4}$ (pre sieť s veľkosťou 1000 uzlov a 4 shardy). Posledný experiment navrhol úpravu protokolu, ktorá túto zraniteľnosť odstránila.

Na základe výsledkov tejto práce je možné poukázať na problém dočasnej centralizácie Harmony blockchainu. Hlasovací protokol FBFT je príliš závislý na malom množstve uzlov (vodcovia hlasovania), ktoré dočasne predstavujú centrálnu autoritu. Zneškodnením týchto uzlov sa stáva blockchain nefunkčný. Naopak, pozitívnym záverom o fungovaní Harmony je spôsob distribúcie hlasovacieho podielu medzi shardy. Z hľadiska PoS má útočník len zanedbateľne väčšiu šancu ovládnuť jeden shard ako celú sieť.

V budúcnosti bude táto práca rozšírená o simuláciu ďalších blockchain protokolov (Solana a Ouroboros). Potenciálny záujemca o simuláciu blockchainu môže použiť túto prácu pre výber vhodného simulačného nástoja (prípadne rozšíriť nástroj vytvorený v tejto práci).

Poďakovanie

Chcel by som poďakovať vedúcemu mojej práce Ivanovi Homoliakovi, ktorý mi poskytol profesionálnu pomoc a hodnotné komentáre.

Literatúra

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Cryptography Mailing list at https://metzdowd.com*, 03 2009.

[2] Remigijus Paulavičius, Saulius Grigaitis, and Ernestas Filatovas. A systematic review and empirical analysis of blockchain simulators. *IEEE Access*, 9:38010–38028, 2021.

[3] Ivan Homoliak, Sarad Venugopalan, Qingze Hum, Daniel Reijsbergen, Richard Schumi, and Pawel Szalachowski. The security reference architecture for blockchains: Towards a standardized model for studying vulnerabilities, threats, and defenses. 10 2019.

[4] Cong Nguyen, Hoang Dinh Thai, Diep Nguyen, Dusit Niyato, Huynh Nguyen, and Eryk Dutkiewicz. Proof-of-stake consensus mechanisms for future blockchain networks: Fundamentals, applications and opportunities. *IEEE Access*, PP:1–1, 06 2019.

[5] Suryanarayana Sankagiri, Xuechao Wang, Sreeram Kannan, and Pramod Viswanath. Blockchain cap theorem allows user-dependent adaptivity and finality. In *Financial Cryptography and Data Security - 25th International Conference, FC 2021, Revised Selected Papers*, pages 84–103. Springer, 2021.

[6] Caixiang Fan, Sara Ghaemi, Hamzeh Khazaei, and Petr Musilek. Performance evaluation of blockchain systems: A systematic survey. *IEEE Access*, 8:126927–126950, 2020.

[7] Yusuke Aoki, Kai Otsuki, Takeshi Kaneko, Ryohei Banno, and Kazuyuki Shudo. Simblock: A blockchain network simulator. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops*, pages 325–329, 2019.

[8] Filip Borčík. Testování bezpečnosti a výkonu proof-of-stake protokolů pomocí simulace. Diplomová práce, Vysoké učení technické v Brně, Fakulta informačních technologií, 2021.

[9] Maher Alharby and Aad van Moorsel. Blocksim: An extensible simulation tool for blockchain systems. *Frontiers in Blockchain*, 3:28, 2020.

[10] Rongjian Lan et al. Harmony: Technical whitepaper. page 22, Mountain View, CA.

[11] Stephen Tse, Rongjian Lan, Sahil Dewan, Leo Chen, et al. Harmony: Documentation. Mountain View, CA. Harmony. [Online; navštívené 21.03.2022], https://docs.harmony.one/home/.

[12] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, OSDI '99, page 173–186, USA, 1999. USENIX Association.

[13] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. In *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, ASIACRYPT '01, page 514–532, Berlin, Heidelberg, 2001. Springer-Verlag.

[14] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles*, SOSP '17, page 51–68, New York, NY, USA, 2017. Association for Computing Machinery.

[15] Aline Gouget, Jacques Patarin, and Ambre Toulemonde. Unpredictability properties in algorand consensus protocol. In *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–3, 2021.