Vysoké učení technické v Brně
Fakulta informačních technologií

Blockchainy a decentralizované aplikace

Platforma na monitorování uzlů v P2P síti DASH

Bc. Daniel Pátek (xpatek08)

1 Kryptoměna DASH a její síť

Kryptoměna DASH je ve své podstatě kopie kryptoměny Bitcoin. Vznikla v roce 2014 a došlo k tomu právě kopií existujícího Bitcoinu. Do dnešního dne se drží Bitcoinu velmi blízko jak ve fungování protokolu P2P sítě, tak v implementaci například RCP (Remote Procedure Calls). [Wik22]

1.1 Připojování k síti

Pokud se někdo rozhodne si spustit svůj vlastní uzel v síti DASH, se samotnou instalací by neměl mít žádný problém. Pro provoz klienta je potřeba stáhnout balíček s programem *Dashcore*, respektive *dashd*. [Das22b] Následně se vytvoří konfigurační soubor `dash.conf`, který obsahuje základní informace, jakým způsobem bude spuštěný klient pracovat. [Das22a] Tento soubor je potřeba uložit mezi konfigurační soubory v operačním systému, typicky `$HOME/.dashcore/dash.conf`.

Listing 1: `dash.conf` použito v v projektu

```
prune=1000
rpcuser=user
rpcpassword=passw
listen=1
server=1
rest=1
```

Důležitou volbou je takzvaný *prune mode*, který umožní spustit uzel bez nutnosti synchronizace celého blockchainu na lokální disk. Nutno dodat, že takový klient nemůže potvrzovat transakce. Také je potřeba nastavit hodnotu proměnné *server* na 1 pro spuštění RCP serveru. Po takovém nastavení se uzel může spustit příkazem `./dashd` a uzel se sám připojí na několik dalších uzlů, které vyhledá automaticky pomocí DNS záznamů. [Das22e]

1.2 RPC metody

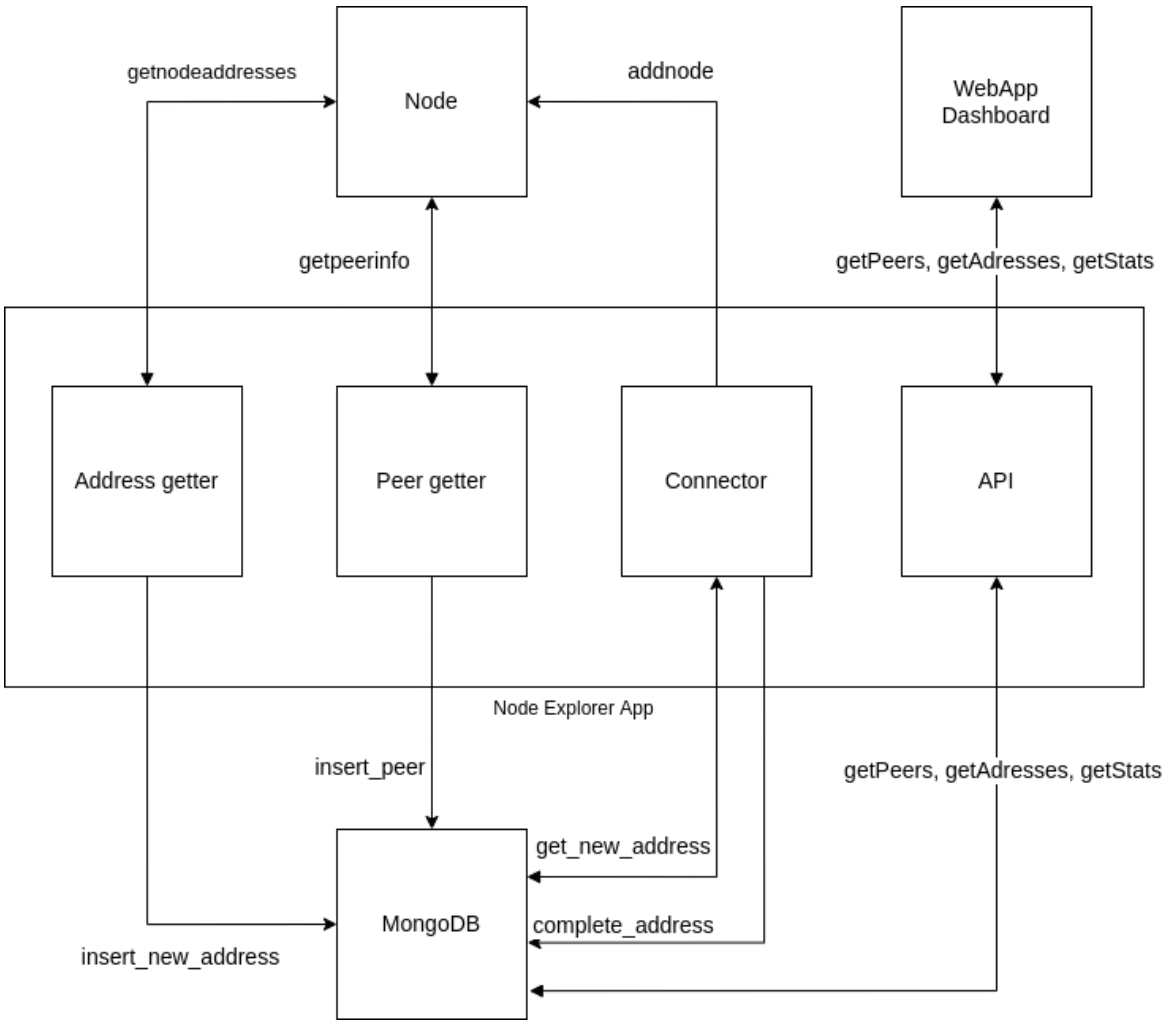
Nedílnou součástí komunikace mezi programem a uzlem je využití *Remote procedure calls*. [Das22d] Tyto metody umožňují efektivní a rychlou komunikaci a využívají *JSON-RPC*. [JSO22] Jsou zde zmíněny a objasněny alespoň metody použité v projektu.

- `getpeerinfo` - Vrátí informace o každém aktuálně připojeném peeru (uzlu).
- `addnode` - Využivá se buď pro přidání uzlu mezi uložené, nebo pro pokus o připojení se k tomuto uzlu.
- `getnodeaddresses` - Vrátí požadovaný počet nových adres uzlů získaných od připojených peerů.

Zde je nutné zmínit, že metoda `getnodeaddresses` v době tvorby tohoto projektu není dostupná v aktuální verzi *Dashcore* 17. Z tohoto důvodu pro účely tohoto projektu budu využívat beta verzi *Dashcore* 18, kde je již tato metoda funkční. Verze 18 je momentálně ve fázi finálního testování a její plné nasazení je očekáváno koncem května roku 2022. [Das22c]

2 Návrh programu pro monitorování dostupných uzlů

Hlavní program (v diagramu 1 níže označen jako *Node Explorer App*) je navržen jako soubor čtyř na sobě nezávislých služeb. Pro persistenci dat byla zvolena databáze *MongoDB*, hlavně kvůli své dobré škálovatelnosti při práci s mnoha záznamy. Součástí návrhu je i webová aplikace, která bude zobrazovat současný stav systému a umožní procházet získaná data o dalších uzlech.



Obrázek 1: Návrh jednotlivých komponent programu

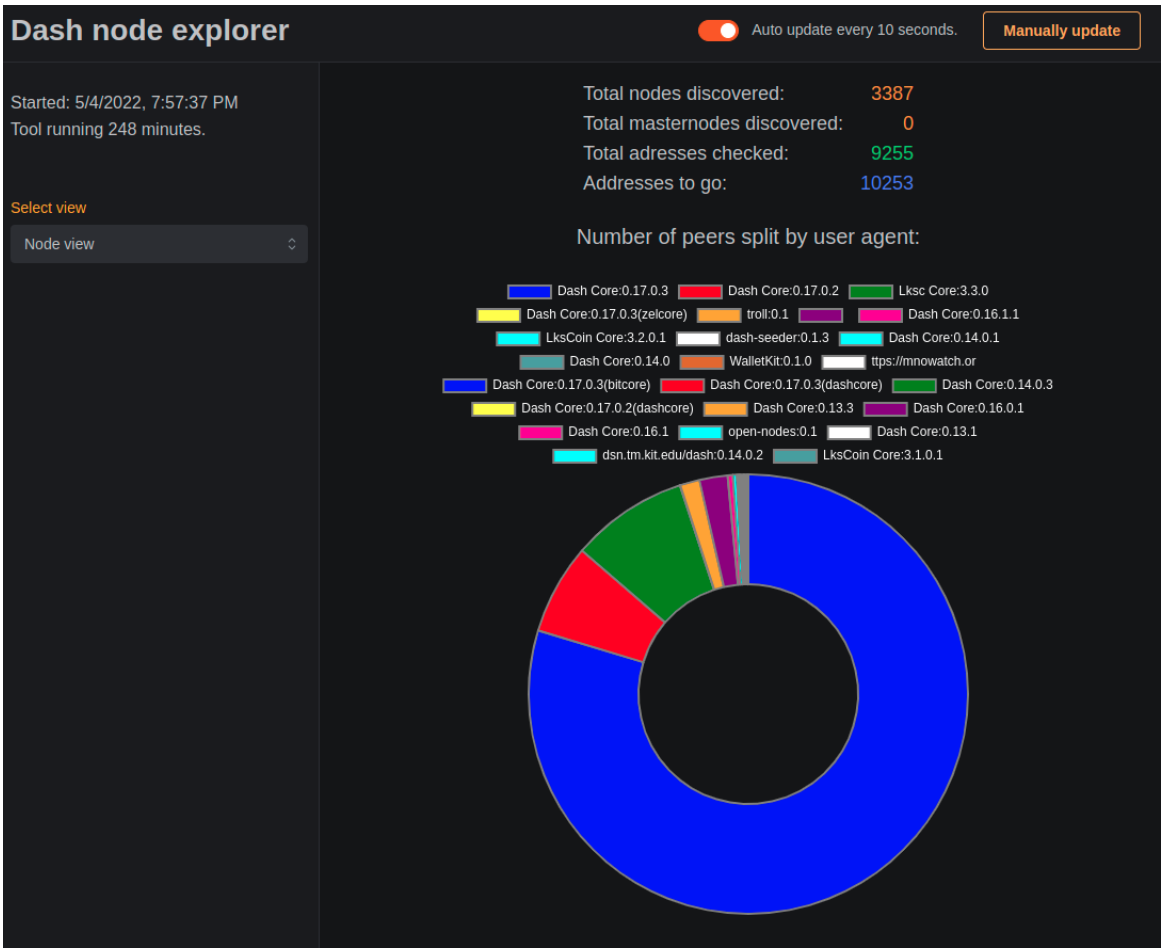
2.1 Představení jednotlivých částí programu

- **Node** - Jedná se o klienta *Dashcore* s běžícím RPC serverem pro komunikaci s ostatními částmi programu.
- **MongoDB** - Databáze, která uchovává informace o adresách a uzlech v síti.
- **Address getter** - Tato služba komunikuje s částí **Node** a stará se o získávání dostupných adres pomocí RPC metody `getnodeaddresses` a následně jejich nahrání do databáze *MongoDB*.
- **Peer getter** - Hlavním úkolem této služby je také komunikace s částí **Node**. Používá RPC metodu `getpeerinfo` pro získání informací o připojených uzlech a tyto informace následně nahrává do databáze.

- **Connector** - Navazování nových spojení řeší tato část programu. Nejprve si z databáze **MongoDB** vezme jednu adresu potencionálního uzlu v síti. RPC metodou `addnode` s argumentem `onetry` požádá část **Node** o připojení nového uzlu. Zároveň tuto adresu v databázi označí jako hotovou.
- **API** - Jedná se o funkční server s endpointy pro získávání informací, jako je seznam uzlů nebo seznam adres, z databáze **MongoDB**. Tuto službu využívá jen webová aplikace.
- **WebApp Dashboard** - Webová aplikace, která zobrazuje dostupné informace o programu, ale také aktuální stav uzlů v databázi.

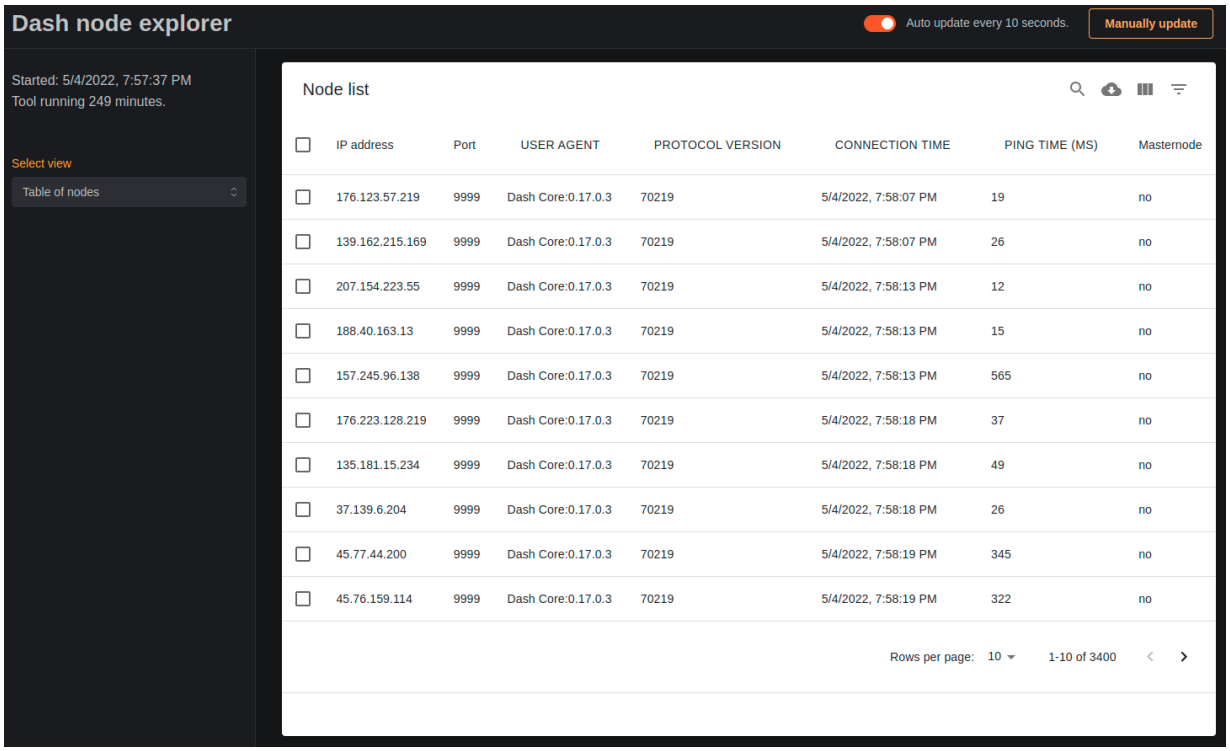
3 Výsledky

Na základě požadavků byl implementován program pro monitorování dostupných uzlů v P2P síti kryptoměny DASH. Byl zvolen jazyk *Python 3.8*. Webová aplikace byla implementována ve frameworku *Next.js* jazyka *JavaScript*.



Obrázek 2: Snímek obrazovky výsledné webové aplikace - dashboard

Jak by se dalo odvodit z Obrázku 2, program byl ponechán zapnutý po dobu 4 hodin a 15 minut. Za tuto dobu bylo zpracováno 9255 různých adres uzlů v síti. Bylo navázáno 3387 spojení s dostupnými uzly a informace o těchto uzlech uloženy do databáze. V databázi se v tu dobu nacházelo ještě 10253 adres, které zatím nebyly vyzkoušeny na připojení.



Obrázek 3: Snímek obrazovky výsledné webové aplikace - tabulka

Pro zobrazení tabulky na obrázku 3 byla využita knihovna *mui-datatables*. Tato knihovna obsahuje filtrování, řazení a také export dat do formátu `.csv` pro pohodlnou práci s informacemi.

4 Zhodnocení

Tento projekt byl bezesporu velmi zajímavý, ať už co se týče práce s kryptoměnovým klientem nebo získávání informací o připojených uzlech v síti.
Podařilo se mi sestavit funkční systém, který postupně mapuje uzly v P2P síti kryptoměny DASH a ukládá je do databáze. Implementoval jsem i webovou aplikaci pro přehledný přístup k těmto informacím.
Pokud bych měl tento projekt tvořit podruhé nebo například pro jinou kryptoměnu, určitě bych daleko více času věnoval studování dokumentace daného klienta kryptoměny. Nejednou se mi stalo, že jsem se v implementaci zasekl u nějakého bodu, který byl dobře objasněn právě v dokumentaci, i když třeba pod jinou záložkou, než jsem si v tu chvíli myslel.
Další bod, který bych změnil v mém implementačním postupu je *dockerizace* programu včetně klienta *Dashcore*. Věnoval jsem velké úsilí tomuto procesu hned na začátku, kdy jsem ještě zdaleka neměl hotový ani samotný program, bohužel bezúspěšně. U takového programu složeného z více nezávislých služeb je *dockerizace* jistě žádoucí.

Odkazy

[Das22a] Dashcore. *Configuration File*. 2022. URL: https://dashcore.readme.io/docs/core-examples-configuration-file (cit. 04. 05. 2022).

[Das22b] Dashcore. *Dashcore*. 2022. URL: https://dashcore.readme.io/ (cit. 04. 05. 2022).

[Das22c] Dashcore. *Dashcore 18 Product Brief*. 2022. URL: https://www.dash.org/blog/dashcore-v18-0-product-brief/ (cit. 04. 05. 2022).

[Das22d] Dashcore. *Network RPCs*. 2022. URL: https://dashcore.readme.io/docs/core-api-ref-remote-procedure-calls-network#getpeerinfo (cit. 04. 05. 2022).

[Das22e] Dashcore. *Peer discovery*. 2022. URL: https://dashcore.readme.io/docs/core-guide-p2p-network-peer-discovery (cit. 04. 05. 2022).

[JSO22] JSON-RPC. *JSON-RPC 2.0 Specification*. 2022. URL: https://www.jsonrpc.org/specification (cit. 04. 05. 2022).

[Wik22] Wikipedia. *Dash (cryptocurrency)*. 2022. URL: https://en.wikipedia.org/wiki/Dash_(cryptocurrency)#cite_note-fintech-1 (cit. 04. 05. 2022).