VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ
DEPARTMENT OF INTELLIGENT SYSTEMS

AUTOMATIZOVANÉ OBCHODOVÁNÍ NA KRYPTOMĚNOVÝCH BURZÁCH

AUTOMATED TRADING ON CRYPTOCURRENCY EXCHANGES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. ZDENĚK KŘESŤAN**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. RADEK KOČÍ, Ph.D.**
SUPERVISOR

BRNO 2018

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav inteligentních systémů Akademický rok 2017/2018

Zadání diplomové práce

Řešitel: **Křesťan Zdeněk, Bc.**
Obor: Informační systémy
Téma: **Automatizované obchodování na kryptoměnových burzách**
Automated Trading on Cryptocurrency Exchanges
Kategorie: Softwarové inženýrství

Pokyny:

1. Prostudujte koncept kryptoměn, zaměřte se na v současné době nejrozšířenější kryptoměny. Seznamte se s burzami, na kterých jsou kryptoměny obchodovány.
2. Srovnejte vybrané kryptoměny a proveďte analýzu dostupných aplikačních rozhraní (API) pro vybrané burzy.
3. Navrhněte aplikaci, která umožní vzdálenou správu příkazů na burze prostřednictvím jejího API. Aplikace umožní definovat a automatizovaně zasílat komplexní příkazy na vybrané burzy. Provedení příkazů je podmíněno vybranými indikátory, jejichž hodnoty jsou získány analýzou dat získaných z burzy, např. historie pohybu ceny, poměr prodejních a nákupních příkazů, nebo překročení limitů.
4. Navrženou aplikaci implementujte a otestujte její funkčnost.
5. Vytvořte testovou sadu pro srovnání očekávaného a skutečného vývoje na burze.

Literatura:

- Narayanan, A., Bonneau, J., Felten, E., Miller, A., Goldfeder, S. Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction. Princeton University Press, 2016.

Při obhajobě semestrální části projektu je požadováno:

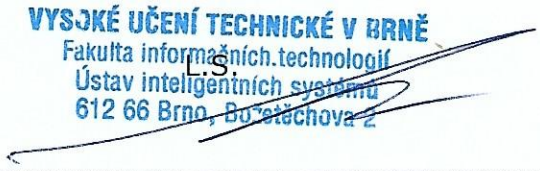
- První tři body zadání

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Kočí Radek, Ing., Ph.D.**, UITS FIT VUT
Datum zadání: 1. listopadu 2017
Datum odevzdání: 23. května 2018



L.S.

doc. Dr. Ing. Petr Hanáček
vedoucí ústavu

Abstrakt

Tato práce je zaměřena na automatizované obchodování na burzách s kryptoměnami. Kryptoměny jsou dnes velice rozšířené. Možnost automatického nákupu a prodeje je zajímavé téma, které je stále více zmiňováno. Hlavní část práce je návrh algoritmu pro zpracovávání dat z burz, jejich vyhodnocování a následné provádění obchodů s kryptoměnami. Popisuje také jeho implementaci, testování a nastiňuje možné další rozšíření.

Abstract

This thesis focuses on automated trading on cryptocurrency exchanges. Cryptos are now widespread. The possibility of hier automated buying and selling is an interesting topic, which is more and more mentioned. The main part of the thesis is the design of an algorithm for processing data from stock exchanges, their evaluation and subsequent execution of cryptocurrency trades. It also describes its implementation, testing and possible further extensions.

Klíčová slova

kryptoměna, burza, automatické obchodování, bitcoin, algoritmus

Keywords

cryptocurrency, stock exchange, automated trading, bitcoin, algorithm

Citace

KŘESŤAN, Zdeněk. *Automatizované obchodování na kryptoměnových burzách.* Brno, 2018. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Radek Kočí, Ph.D.

Automatizované obchodování na kryptoměnových burzách

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Radka Kočího, Ph.D. a pana Mgr. Radima Drgáče.

.....
Zdeněk Křesťan
20. května 2018

Poděkování

Chtěl bych poděkovat vedoucímu, Ing. Radku Kočímu, Ph.D. za přijetí mého zadání a umožnění vypracování této prace. Dále bych zde rád poděkoval odbornému vedoucímu Mgr. Radimu Drgáči, za konzultace a odborný dohled.

Obsah

Seznam obrázků

Kapitola 1

Úvod

V dnešní době je popularita digitálních měn, zejména kryptoměn, stále větší. Jednou z překážek masového rozšiřování je jejich velká volatilita, neboli prudké kolísání jejich ceny v řádu minut, hodin nebo dnů o desítky až stovky procent. Díky základní analýze grafu vývoje dané kryptoměny a dalších faktorů ovlivňujících hodnotu měny, lze tyto změny do jisté míry predikovat. Člověk musí sledovat analýzy a ve vhodnou dobu ručně zadat příkaz k nákupu nebo prodeji přímo v rozhraní burzy. Celá tato aktivita je poměrně časově náročná a často můžou nastat situace, kdy je nutné při určitém vývoji hodnoty měny nějak rychle zasáhnout, což je při běžném povolání velký problém.

Možnost, jak lze tento problém teoreticky řešit, je vytvořit aplikaci, která bude podle algoritmu hlídat možné pohyby hodnoty kryptoměn a umožní automatizovat některé úlohy při obchodování na burze. Tato práce se zabývá návrhem tohoto algoritmu, jeho implementací a testováním. Testování bude prováděno na vybrané burze pomocí jejího API rozhraní.

Nedílnou součástí práce je úvod do světa kryptoměn, základních pojmů a popisu několika vybraných nejrozšířenějších kryptoměn, který je uveden v kapitole 2. Následující kapitola 3 bude věnována kryptoměnovým burzám. U vybraných burz bude představeno jejich rozhraní a možnost získávání a odesílání dat. Po zavedení důležitých pojmů a potřebné teorie je v kapitole 4 představena myšlenka automatizovaného obchodování na burze a následně představen a popsán zmiňovaný algoritmus. Je zde uvedena jeho struktura, možnosti nastavení a ukázky několika možných scénářů jeho fungování.

Předposlední kapitola 5 popisuje implementaci aplikace. Je zde podrobně popsáno jak uživatelské rozhraní pro nastavování hodnot, potřebných pro funkci algoritmu, tak i jádro aplikace obsahující implementaci navrženého algoritmu a napojení na burzy. V poslední kapitole 6 jsou popsány výstupy aplikace pro analýzu běhu, její testování a následné zhodnocení problémů a nastínění možných řešení.

Kapitola 2

Kryptoměny

Kryptoměna patří mezi digitální měny a slouží jako digitální prostředek výměny aktiv. Jedná se o alternativní virtuální měny, které existují pouze v elektronické podobě. Většina těchto měn je navržena tak, aby byla postupně snižována tvorba nových jednotek měny, přičemž mají konečný limit. Díky tomu nedochází k inflaci, zapříčiněné zvyšováním množství měny. Kryptoměny jsou založeny na principu decentralizace (princip peer-to-peer komunikace mezi klienty) a jak už z názvu plyne, opírají se o kryptografii. Právě proto je velmi těžké, až nemožné je falšovat. Decentralizované měny nelze kontrolovat žádnou institucí, jako je například banka nebo vláda. Lze tedy do jisté míry provádět transakce v anonymitě [1].

I s absencí řídící instituce má měna pevnou strukturu a je plně transparentní. Všechny transakce jsou umístěny ve veřejné databázi zvané blockchain. Odeslanou transakci již nelze změnit nebo zrušit. Nepopíratelnost transakcí je zajištěna dostatečnou decentralizací ověřovatelů a využitím blockchainu. Ověření transakce proběhne až po několika blocích, v nichž je zařazena. Proto patří doba mezi bloky a počet bloků nutných pro validaci transakce k důležitým parametrům jednotlivých kryptoměn. Z toho plyne, že čím je kratší interval mezi vytvořením dvou bloků, tím rychleji je transakce ověřena.

Blockchain Jedná se o distribuovanou decentralizovanou databázi, která se skládá z jednotlivých zřetězených bloků a transakcí v nich [2]. Nejčastější využití je právě jako účetní kniha pro kryptoměny. V jednotlivých blocích jsou zaznamenány transakce, které byly provedeny uživateli v daném časovém období od vygenerování předešlého bloku. Blok je vytvořen pomocí kryptografických ověřovacích protokolů. Po ověření je blok s daným časovým razítkem zařazen do blockchainu a informace, které obsahuje nelze již měnit. Blockchain napomáhá řešení problému tzv. „dvojího utrácení“, neboli odeslání stejných peněz ve dvou transakcích, bez nutnosti centrálního ověřování.

Těžení (angl. Mining) Je nezbytnou součástí při fungování blockchainu. Jedná se o vytváření bloku transakcí, který má potvrzovat a tím zabezpečovat celou síť. Těžba je velmi náročná na výpočetní techniku. Provádějí ji tzv. „těžaři“ za účelem získání odměny za vytěžený (uzavřený) blok a tím potvrzení transakcí v něm. Doba mezi vytvářením jednotlivých bloků může být rozdílná u různých kryptoměn. Tato doba je určována speciálními požadavky měn na výslednou hash daného bloku. Aby bylo možné upravovat výslednou hash, je přidáván na začátek bloku tzv. „nonce“. Nonce je náhodně vygenerovaný text, který se mění tak, aby výsledná hash bloku odpovídala požadavkům. Těžba je založena na metodě pokus-omyl.

Transakce Jedná se o digitální informaci, která odpovídá datovému souboru. Tento soubor obsahuje adresu odesílatele, příjemce a částku dané měny. Po odeslání je transakce ověřována každým uzlem sítě, kterým prochází. Pouze validní transakce jsou posílány dál a během chvíle distribuovány ke všem uzlům sítě.

Virtuální peněženky Slouží pro uložení digitálních měn a dělí se na online peněženky tzv. „Hot Wallet“ a offline tzv. „Cold Wallet“. Pro příjem mincí pak slouží adresy, které lze v peněžence vygenerovat. Každá z těchto adres je propojena s privátním klíčem peněženky, který je nutné tajit. Tento klíč slouží k rozpoznání transakcí patřících do dané peněženky. Dalším dělením peněženek je na softwarové a hardwarové. Softwarové peněženky mají podobu speciálního programu, který je nainstalován na počítači uživatele a poskytuje rozhraní pro správu peněženky. Nejpopulárnější a důvěryhodné jsou například peněženky Bitcoin Core nebo Ethereum Wallet. Hardwarové peněženky jsou dnes pokládány za nejbezpečnější z hlediska ochrany před útoky hackerů. Vypadají jako malé USB flash disky a mezi nejznámější patří například peněženka jménem TREZOR.

Další kladnou vlastností kryptoměn jsou velmi nízké nebo žádné poplatky za transakci. Na rozdíl od bank jsou zde jenom dobrovolné poplatky, které zajistí rychlejší zpracování transakce. Jejich nevýhodou je však velká volatilita. Volatilita vyjadřuje míru rizika, s jakým může daná měna měnit svoji hodnotu. U kryptoměn bývají tyto změny prudké a v krátkém časovém intervalu. S nižší mírou regulace však přichází problém v podobě zneužití měny pro daňové úniky nebo praní peněz. Výhodou kryptoměn je také možnost non-stop obchodování, na rozdíl od klasických měn nebo akcií, se kterými se obchoduje pouze během obchodních hodin. V současné době existuje spousta různých kryptoměn s odlišnou implementací sítě a protokolů schvalujících transakce. Dnes existuje více jak 1300 kryptoměn. V následujících podkapitolách si představíme několik dnes nejrozšířenějších kryptoměn.

2.1 Bitcoin

Jedná se aktuálně o největší a nejznámější kryptoměnu [1]. Je pokládána za první decentralizovanou digitální měnu, podle které vzniklo spousta podobně fungujících kryptoměn (tzv. altcoinů). Základní jednotkou je jeden Bitcoin (BTC), který je dělitelný na 8 desetinných míst. Byl vytvořen v roce 2009 neznámou osobou nebo organizací pod názvem Satoshi Nakamoto. Za počátek je pokládáno vytvoření prvního bloku známého jako genesis block. Celá síť je založena na peer-to-peer komunikaci, tedy všechny operace probíhají přímo mezi uživateli. Každý uživatel (uzel) sítě má svou vlastní kopii řetězce bloků. Pro ověření transakce je nutné, aby byla zařazena do 6 bloků, tedy při průměrné rychlosti okolo 10 minut uzavírání bloků, trvá ověření přibližně hodinu. Bitcoiny si uživatelé uchovávají ve svých peněženkách, které zaznamenávají proběhlé transakce a slouží pro převádění a manipulaci s měnou. V bitcoinu se používá kryptografie s veřejným klíčem.

V poslední době cena bitcoinu prudce stoupla, ale díky vysoké volatilitě jsou stále investice do této měny více riskantní než například do zlata. I tak je bitcoin velmi často využíván jako investiční měna a pro uchovávání úspor. Hodnota bitcoinu vychází pouze z aktuální poptávky a nabídky (tzv. deflační měna), není kryta žádnou komoditou. Díky anonymitě uživatelů sítě, lze bitcoin zneužít k trestné činnosti. S rostoucím zájmem o obchodování s bitcoiny vznikl problém s rychlostí zpracování transakcí. Kvůli pevné velikosti bloku (1Mb) síť dokázala zpracovat průměrně 3 transakce za sekundu. To vedlo k velkým

prodlevám při provádění plateb. Snahou jak tento problém vyřešit, bylo zavedení funkce , která by díky změnám struktury bloku umožnila až čtyřnásobné zrychlení. Bitcoiny lze nakupovat nejen na burzách, ale například v bitcoin automatech nebo při osobní směně s jiným uživatelem sítě.

V roce 2013 byl Bitcoin uznán v Německu jako oficiální virtuální měna a podléhá daním. Další znatelnou regulaci provedla Čína, která zakázala užívat Bitcoin finančním institucím, ale fyzickým osobám ne. V roce 2015 pak Komise pro komoditní obchody zařadila Bitcoin mezi komodity. Z toho plyne, že všechny obchody na burzách musejí být registrované a kontrolované.

SegWit (Segregated Witness) Jedná se o vlastnost, která je implementována v kódu sítě Bitcoin. Upravuje strukturu a formát prvků sítě s cílem urychlení zpracování transakcí, zejména se jedná o odstranění tzv. signature dat z každého bloku. Zavádí změnu velikosti bloku až o 70 %, nové typy transakcí, změny při kódování podpisu dané transakce a například zahrnutí částky do otisku celé transakce. Přináší tedy nové technologie, jako například mikro platební kanály nebo nové typy levných bezpečných peněženek.

Se spuštěním SegWit a jeho úpravami bitcoinové sítě část uživatelů nesouhlasila a vytvořila novou kryptoměnu Bitcoin Cash 2.3. Chvíli po vzniku kryptoměny Bitcoin Cash došlo k dalšímu štěpení bitcoinu a následný vznik kryptoměny Bitcoin Gold. Jeho vznik byl zapříčiněn obavou z monopolu několika těžících organizací, založených převážně na výpočtech pomocí ASIC čipů. Bitcoin Gold obsahuje nové algoritmy, které umožňují zvýšit výnosnost a spravedlivost těžby za využití pouze GPU.

2.1.1 Technická specifikace

- **Měna:** Bitcoin (BTC)
- **Těžební algoritmus:** SHA-256
- **Tvorba bloku:** 10 minut
- **Odměna za vytěžený blok:** 12,5 BTC
- **Snížení odměny:** na polovinu po každých 210 000 blocích (asi 4 roky)
- **Další snížení:** červen 2020
- **Celkové množství:** 21 milionů

2.1.2 Těžba

Těžení neboli potvrzování a shromažďování transakcí do bloků, provádějí tzv. „těžaři“, kteří hledají optimální kryptografickou nonci, díky které by hash (SHA-256) nového bloku vyšla pod limit bitcoinové sítě. Za uzavření bloku je těžař odměněn. Získá všechny poplatky z transakcí a odměnu, která je nyní 12,5 BTC. Tato odměna je po každých 210 000 uzavřených blocích snížena na polovinu (na začátku byla 50 BTC), z toho plyne, že se zpomaluje růst množství bitcoinů. Těžbou je tedy zvyšován počet bitcoinů. Tento počet je však omezen na přibližně 21 milionů, kde při navržené struktuře bude většina vytěžena přibližně kolem roku 2030, všechny pak až v roce 2140 [1].

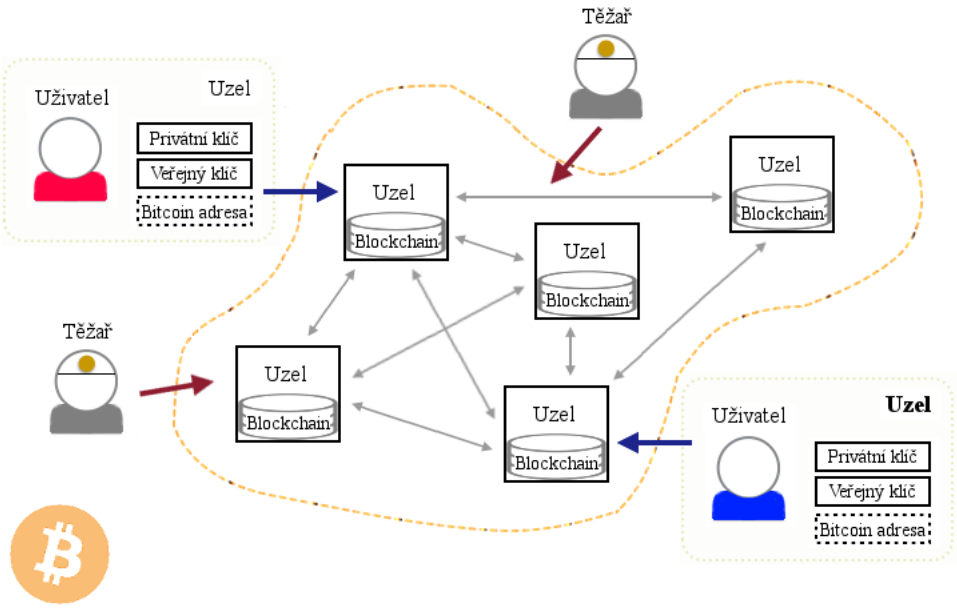
V dnešní době je těžba bitcoinů velmi náročná na hardwarový výkon, proto se těžaři začali sdružovat do tzv. „poole“ neboli farem. Díky velké produkci nových ASIC (Application Specific Integrated Circuit) specializovaných zařízení a nízké ceně elektrické energie se dnes nachází největší farmy v Číně. Mezi největší patří například ViaBTC, BTC.com, AntPool nebo F2Pool. Kvůli velikosti a náročnosti sítě se těžba běžným uživatelům nevyplatí. Z tohoto ohledu je pravděpodobnost zisku vyšší u jiných, méně rozvinutých kryptoměn.

2.1.3 Bitcoinová síť

Jak již bylo popsáno, jedná se o peer-to-peer síť, kde uživatelé pomocí softwaru v podobě kryptoměnových peněženek provádějí transakce. Z pohledu transakce lze bitcoin definovat jako sekvenci ověřených transakcí od jeho vytvoření (přidělen jako odměna za vytěžení). Dle konvence je na začátku bloku vždy speciální transakce, která vytváří a přiřazuje nové bitcoiny tvůrci bloku. Bezpečnost sítě je zajištěna Bitcoin protokolem, který obsahuje funkce na ochranu před nejčastějšími typy útoků. S rostoucí bitcoinovou sítí přibývá i hackerských útoků na online burzy a směnárny.

Neoprávněné transakce Zavedením kryptografie s veřejným a soukromým klíčem je riziko zmírněno.

Dvojitý výdaj Jedná se o problém, kdy se uživatel pokouší platit jednou mincí dvěma nebo více příjemcům. Platnost mince lze ověřit pomocí záznamu v řetězcích bloků transakcí a převést tak pouze ještě nevyčerpanou minci.



Obrázek 2.1: Ukázka struktury bitcoinové sítě.

2.2 Ethereum

Ethereum je hned po bitcoinu druhá nejznámější kryptoměna a je označována jako kryptoměna budoucnosti nebo kryptoměna další generace. Poskytuje již ověřenou decentralizaci při obchodování s měnou, ale na rozdíl od bitcoinu nabízí i podporu skriptování a tím vznik decentralizovaných projektů (označováno taky jako tzv. decentralizovaný virtuální stroj). Jedná se o aplikační platformu založenou na technologii blockchain. Základní jednotkou měny je Ether (ETH), který je dělitelný na 18 desetinných míst. Měna byla uvedena do provozu 30. července 2015 a její zakladatel je vývojář Vitalik Buterin. Transakce jsou opět zařazovány do bloků, kde každý obsahuje vazbu na předchozí blok.

Hlavním rozdílem oproti bitcoin síti jsou její široké možnosti využití. Ethereum umožňuje využít blockchain pro zaznamenávání různých informací nebo pro vytváření decentralizovaných aplikací pomocí speciálního programovacího jazyka. Rozšíření poskytuje možnost využít celou síť pro vykonávání početně náročných operací a programů. Programy uložené v blockchainu jsou označovány jako tzv. „chytré kontrakty“, které mají svou adresu, mohou provádět operace s Ethery a jsou přístupné pro všechny uživatele sítě. Pro své fungování využívají výpočetní výkon, který poskytují těžaři. Za poskytování tohoto výkonu jsou těžaři odměňováni Ethery, které jsou spotřebovávány během těchto kontraktů. Kontrakty fungují, dokud se samy neukončí nebo jim nedojde kredit v podobě Etherů. Tento princip v dnešní době je využit například u aplikací The DAO nebo Slock.it.

Chytrý kontrakt Jedná se o software, který zajišťuje, ověřuje a provádí kontrakt za splnění určitých podmínek, pomocí kryptografického kódu. Hlavním potenciálem chytrých kontraktů je poskytovat bezpečnost a vymahatelnost za nižších nákladů. Tvůrce nastaví přesnou funkcionalitu, například výměnu peněz, majetku nebo cenností. Chytré kontrakty jsou schopny komunikovat s jinými chytrými kontrakty v síti.

Ethash Jedná se o plánovací algoritmus navržený pro Ethereum 1.0. Vychází přímo z algoritmu Dagger-Hashimoto. Ze základního seedu lze vypočítat 16 MB pseudo náhodné cache.

Ethereum má za sebou i zcizení velkého množství měny. Kvůli chybě ve zdrojových souborech a následnému zneužití neoprávněných žádostí na projekt DAO, bylo nutné provést zásah do sítě s cílem obnovit stabilitu měny a ukradené Ethery získat zpět. Zásah byl úspěšný, ale kvůli nesouhlasným názorům na jeho provedení vznikla nová oddělená měna s názvem Etherum Classic, která udržuje původní blockchain. Tento zásah však poodhalil možný problém s tím, že síť lze zpětně modifikovat, a tedy potencionálně zneužít. Sít je řízena vývojáři, což odporuje obecné decentralizaci. Měna má velký potenciál do budoucna a funkcionalita chytrých kontraktů může být zajímavá pro banky, vlády a další instituce.

Na začátku roku vznikla organizace EEA (Enterprice Ethereum Alliance) s 30 členy a postupně se rozrostla až na 150 členů. Tato organizace se zabývá vývojem sítě s neveřejným blockchain určené pro řešení problémů v bankovnictví, zdravotnictví a dalších odvětvích.

2.2.1 Technická specifikace

- **Měna:** Ether (ETH)
- **Těžební algoritmus:** Ethash
- **Tvorba bloku:** 15 sekund

- **Odměna za vytěžený blok:** 5 ETH (po aktualizaci 3 ETH)
- **Snížení odměny:** ovlivňují vývojáři sítě (aktualizace)
- **Množství:** omezeno na 18 milionů ročně

2.2.2 Těžba

Stejně jako u bitcoinu slouží těžba jako nástroj pro shromažďování transakcí do bloků, jejich ověřování, a tím vytváření nových Etherů v síti. Bloky jsou zde vytvářeny rychleji, přibližně každých 15 sekund, proto nemusí řešit problémy jako bitcoin při velkém počtu transakcí. Díky rychlému ověřování je nejefektivnější metoda pro uzavření bloku tipování správného výsledku metodou pokus-omyl. Odměnu získává pouze nejrychlejší, a proto vznikají opět větší sdružení těžařů, kteří spojují svůj výkon. Při uzavření bloku vznikne 5 Ehterů (po aktualizaci 3 Ehtery). Z toho část získává těžař, kterému se podařilo uzavřít blok, ale malá část je poslána i tzv. strýčkům neboli těžařům, kteří byli schopni najít řešení, ale jejich blok nebyl přidán do blokchain.

Vzhledem k velkému počtu bloků a poměrně vysoké odměně za jeho uzavření, je v síti už velký počet mincí. Těžba Entherů je stále výnosná i pro obyčejné těžaře, protože je uzpůsobena tak, aby se nejlépe těžila pomocí GPU. V dnešní době se prodávají specializované grafické karty pro těžbu kryptoměn.

2.2.3 Síť Ethereum

V síti se vyskytují dva typy účtů. Jedná se o externě vlastněné účty (EOA) a smluvní účty (CA). Externě vlastněné účty neobsahují žádný program a slouží jako běžné účty pro vytváření transakcí. Umožňují pomocí transakce založit smluvní účet. Smluvní účty existují v síti a reprezentují právě zmíněné chytré kontrakty. Svým během spotřebovávají vložené Ethery. Smluvní účet již nezle změnit a tím je bezpečný a stálý.

2.3 Bitcoin Cash

Bitcoin Cash (BCH/BCC) vychází přímo z bitcoinu a při vzniku měla stejný blockchain (kopie blockchainu bitcoinu po poslední společný blok s číslem 478558). Další bloky po rozdělení již měly obě měny odlišné podle daných transakcí v sítích. Měna vznikla 1. srpna 2017 jako nesouhlas s aktivací SegWit pro bitcoin. Kvůli obavám, že tato technologie naruší decentralizaci měny. Všichni uživatelé sítě bitcoin tak při rozdělení získali stejný počet mincí v síti bitcoin cash, což mělo kompenzovat možnou ztrátu ceny bitcoinu po rozdělení. Bitcoin Cash spravuje několik nezávislých vývojářských týmů. Díky tomu je vývoj dostatečně decentralizován.

Měna přebírá veškeré vlastnosti původního bitcoinu, co se týká počtu mincí nebo šifrovacího algoritmu. Důležitým rozdílem je však zvýšení kapacity bloků na 8 Mb a úpravu automatického přepočítávání složitosti těžení po každých 6 blocích tak, aby odpovídala aktuální početní síle těžařů. Nově využitý SigHash pro podepisování transakcí poskytuje vyšší ochranu peněženky a odstraňuje kvadratický problém hashování. Nejpopulárnější skupinou těžařů Bitcoin Cash je organizace ViaBTC. Snižováním obtížnosti těžby dle času vytěžování bloků, se může stát atraktivní možností i pro menší těžaře.

Na tuto novou měnu zareagovali různým přístupem jednotlivé burzy a směnárny. Burzy Poloniex 3.3 a Bitfinex 3.1 měnu podporují, ale naopak burzy Exodus nebo BitMEX odmítly

tuto měnu podporovat. Jedná se o novou měnu, kterou prozatím nepodporují internetové obchody jako samotný bitcoin. Proto prozatím slouží převážně investorům.

2.3.1 Technická specifikace

- **Měna:** BitcoinCash (BCH, podle normy pro měnové kódy XBC)
- **Těžební algoritmus:** SHA-256
- **Tvorba bloku:** 10 minut
- **Odměna za vytěžený blok:** 12,5 BTC
- **Snížení odměny:** na polovinu po každých 210 000 blocích (asi 4 roky)
- **Další snížení:** červen 2020
- **Celkové množství:** 21 milionů

2.4 Litecoin

Litecoin je digitální měna založená na databázi blockchainu a decentralizované síti na principu peer-to-peer. Vznikla v říjnu roku 2011 jako kopie známějšího bitcoinu. Vytvořil ji bývalý zaměstnanec Google Charlie Lee se snahou vylepšit bitcoin. Jedná se o open-source projekt. Mince Litecoin (LTC) je dělitelná stejně jako Bitcoin na 8 desetinných míst. Celá struktura sítě je velmi podobná, ale navíc jsou zde zavedeny úpravy pro zlepšení fungování sítě. Blockchain Litecoinu je schopen zvládat více transakcí než u Bitcoinu.

Hlavním rozdílem od bitcoinu je použitý hashovací algoritmus, kde byl bitcoinový algoritmus SHA-256 nahrazen algoritmem Scrypt, který měl být odolnější vůči využití ASIC čipů pro těžení, ale stejně se podařilo využít tyto čipy i u toho algoritmu. Těžba pomocí GPU je tedy již velmi nevýhodná a neefektivní. Na rozdíl od bitcoinu jsou poplatky za transakci mnohem nižší. Během listopadu 2013 zaznamenal Litecoin velký nárůst až o 100 % za 24 hodin. V květnu 2017 byl u Litecoinu aktivován SegWit a tím se stal první měnou z TOP-5 kryptoměn, kde byla tato úprava zavedena.

2.4.1 Technická specifikace

- **Měna:** Litecoin (LTC)
- **Těžební algoritmus:** Scrypt
- **Tvorba bloku:** 2,5 minuty
- **Odměna za vytěžený blok:** 25 LTC
- **Snížení odměny:** na polovinu po každých 840 000 blocích (asi 4 roky)
- **Další snížení:** srpen 2019
- **Celkové množství:** 84 milionů

2.4.2 Těžba

Těžba Litecoin probíhá čtyřikrát rychleji, než je tomu u Bitcoinu. Z toho tedy plyne, že průměrná doba na uzavření bloku je přibližně 2,5 minuty. Odměna pro těžaře začínala na 50 LTC za blok a je postupně snižována o polovinu po každých 840 000 vytěžených blocích. Aktuální odměna je 25 LTC. V Litecoin síti je počet mincí omezen na 84 milionů.

2.5 Ripple

Ripple vznikl jako platební protokol pro rychlé posílání a směnu peněz a nejedná se primárně o digitální měnu, jako je třeba Bitcoin. Vychází z konceptů RipplePay od Ryana Fuggera, který chtěl vytvořit decentralizovaný měnový systém s možností vytváření vlastních peněz (RipplePay.com byl spuštěn v roce 2005).

Ripple byl spuštěn v roce 2012 společností Ripple Lab a v rámci jeho sítě je používaná stejnojmenná měna Ripple (XRP), kterou lze dělit na 6 desetinných míst. Tato síť není omezena na převody pouze této měny, ale umožňuje i transakce s jinými měnami, kryptoměnami nebo různými komoditami. Může být proto využit jako možný nástroj pro rychlé a levné platby. Používání tohoto protokolu zvažuje několik bank. Není nutné udávat svoje údaje a díky tomu pro své uživatele poskytuje soukromí a bezpečnost. Poskytuje soukromí a bezpečnost pro své uživatele, není nutné udávat svoje údaje. Ripple dokáže ověřit stav libovolného účtu nebo transakce ve velmi malém počtu bajtů.

Díky vytvoření propojení s Bitcoinovou sítí tzv. „Bitcoin Bridge“ je umožněno uživatelům Ripple posílat platbu v libovolné měně na bitcoinovou adresu. Od roku 2014 se zaměřili vývojáři Ripple na jeho použití v bankovním sektoru. Pro zavedení mezinárodní sítě převodu peněz v reálném čase. V roce 2016 pak byla založena první mezibankovní skupina pro globální platby s využitím distribuovaných technologií.

Ripple se liší zejména ve struktuře sítě a způsobu potvrzování transakcí. Je zde využit odlišný a zdrojově úspornější způsob potvrzování. Potvrzení probíhá pomocí koncensu, který je proveden za několik sekund. Koncesem je označováno dohodnutí ověřovatelů v síti o platnosti dané transakce. Nevýhodou je, že měna je do jisté míry centralizovaná, protože většinu mincí vlastní zakládající společnost a lze tedy s touto měnou manipulovat. Hodnota měny je stejně jako u Bitcoinu závislá na aktuální poptávce a není nijak podložena.

2.5.1 Technická specifikace

- **Měna:** Ripple (XRP)
- **Jazyk:** Python
- **Tvorba bloku:** nemá klasické bloky ani Blockchain
- **Odměna za vytěžený blok:** žádná přímá odměna, pouze poplatky za transakce a provádění kontraktů

2.5.2 Těžba

Na rozdíl od přechozích kryptoměn neprobíhá v síti Ripple žádné těžení. Nové mince jsou uvolňovány tvůrci systému, kteří v tuto dobu uvolňují do systému miliardu mincí měsíčně. Počet všech mincí byl omezen na 100 miliard.

2.5.3 Síť

Jedná se o decentralizovanou sdílenou síť s veřejným zdrojovým kódem. V této síti je blockchain nahrazen speciálním sdíleným souborem zvaným Ledger, který je sdílen mezi ověřovateli v síti. Soubor Ledger obsahuje údaje o transakcích a je aktualizován každých 5 sekund. Ověřovatelé jsou v síti speciální uživatelé, kteří se starají o zaznamenávání plateb v síti a aktualizaci Ledgeru. Pokud se většina ověřovatelů shodne na správnosti informací, je Leadger aktualizován. Při každé transakci je odečten malý poplatek, který není nikomu přičten. Tento poplatek má zabraňovat vytváření milionům transakcí.

Kapitola 3

Burzy

Jedná se o online obchodní platformu pro prodej a nákup virtuálních měn, zprostředkovatele nákupu měny od jiného uživatele. Na burzách se zejména používají nabídky typu „Limit“ nebo „Market“. Za zprostředkování transakcí si burzy účtují poplatky podle množství měny v transakci (0,1 - 1%). Dnes nejznámější využívané burzy budou představeny v následujících podkapitolách.

Limit U tohoto typu nabídky vyplní uživatel množství měny, které chce zobchodovat a její cenu. Objednávka se zařadí mezi nabídky na burze a je zde umístěna, dokud ji někdo nepřijme.

Market Oproti předešlému typu je prodej okamžitý a bez čekání na přijetí nabídky. Uživatel zadá množství měny a burza mu vyhledá nejvýhodnější nabídku.

3.1 Bitfinex

Kryptoměnová burza zaměřená na obchodování s nejpopulárnějšími měnami a v dnešní době jednička v celkovém objemu obchodů [4]. Byla založena v roce 2013 v Hong Kongu firmou iFinex Inc. Vyznačuje se jako burza s největším počtem zobchodovaných bitcoinů a to z celkového množství více než 10 % všech zobchodovaných bitcoinů v celé síti. Jedná se o pokročilou platformu s možností různých grafů a pomocných indikátorů. Burza umožňuje obchodování s vysokou likviditou. Do burzy lze vkládat kromě bitcoinů i americké dolary. Burza disponuje několika různými obchodními příkazy, kromě základních příkazů „Market“ a „Limit“ obsahuje příkazy „Stop“ a „Trailing Stop“ pro automatické uzavírání obchodních pozic. Příkaz „Stop Limit“ slouží pro nastavení maximální ceny nákupu/prodeje. Zajímavým příkazem je „Fill or Kill“, který musí být realizován celý, nebo dojde k jeho zrušení. Umožňuje i vytvářet další typy objednávek a algoritmických příkazů.

U bitfinexu jsou zpoplatněny jak transakce, tak i výběry. U výběrů se jedná o poplatek 0,1 % (u expresního až 1 %) z dané částky. Při obchodování se liší poplatek podle úlohy uživatele (obchodníka) v transakci. Pro zákazníka (tzv. „maker“), který vytváří nabídky typu Limit je poplatek 0,1 %. Druhý typ uživatele (tzv. „taker“), který vybírá ze seznamu nabídek nebo zakládá rychle obchodní nabídky typu Market, je poplatek 0,2 %. Bitfinex navíc umožňuje obchodovat s vypůjčenými penězi od jiných uživatelů neboli obchodování s pákou. Tomuto obchodování se také říká obchodování s marží a u bitfinexu je umožněno až 3,3x zvýšení investice.

Tuto burzu v roce 2016 postihl hackerský útok, při kterém bylo odcizeno téměř 120 000 Bitcoinů. Aby burza nepřišla o důvěru klientů, kompenzovala ztrátu pomocí BFX tokenů, které následně od uživatelů vykupovala.

3.1.1 API rozhraní

Bitfinex nabízí dvě různé API rozhraní, a to REST nebo Websocket. Tato práce bude zaměřena pouze na rozšířenější a používanější rozhraní REST. Rozhraní umožňuje kompletní přístup k funkcím platformy Bitfinex [4]. Rozhraní dále disponuje ochranou proti DDoS, která omezuje počet požadavků z jedné IP adresy za minutu. Pokud je limit překročen, dojde k blokaci IP adresy na určitou dobu. Limity požadavků se liší pro každou metodu. Konfiguraci a vygenerování příslušných autentizačních klíčů umožňuje platforma burzy.

Pro zasílání GET a POST požadavků slouží veřejná URL adresa „https://api.bitfinex.com“. Rozhraní se dělí na veřejné a neveřejné koncové body, kde u neveřejných je nutná autentizace. Veřejné rozhraní umožňuje pouze několik požadavků na čtení dat z burzy.

Přehled nejpoužívanějších GET metod

- **Ticker:** Jedná se o přehled trhu s danou měnou. Aktuální poptávky a nabídky, nejvyšší a nejnižší cenu, aktuální cenu a další statistiky. Hodnoty jsou vráceny dle stavu za posledních 24 hodin.
- **Fundingbook:** Metoda vracející kompletní knihu nabídek financování.
- **Orderbook:** Přehled všech objednávek pro danou dvojici měn.
- **Trades:** Seznam posledních provedených obchodů mezi dvojící měn.
- **Symbol/Symbol details** Kompletní seznam všech dvojic měn obchodovatelných na burze. Metoda s detaily navíc obsahuje limity pro směnu dané dvojice měn.

Přehled nejpoužívanějších POST metod

- **Account info:** Zobrazí aktuální hodnoty poplatků při obchodování na burze s měnami pro přihlášený účet.
- **Summary:** Přehled objemu obchodování a marží za posledních 30 dní.
- **Deposit/Withdrawal:** Metody slouží pro vklad/vybrání měny z peněženky burzy.
- **New/Cancel Order:** Vytvoření/stažení nabídky k obchodování s měnou.
- **Balance history:** Historie pohybu množství dané měny v peněžence.

3.2 Bitstamp

Bitstamp patří ke starším burzám [3]. Vznikla v roce 2011, založila ji dvojice Slovinců po vzoru dnes již neexistující burzy MtGox. Dnes burzu provozuje společnost Bitstamp Ltd., která sídlí v Londýně. Na burze lze obchodovat s nejrozšířenějšími kryptoměnami. Vklady nebo výběry jsou umožněny nejen v bitcoinech, ale i v eurech nebo dolarech. Řadí se mezi největší burzy v Evropě i ve světě. Opět umožňuje zadání objednávek typu „Market“ a „Limit“, navíc umožňuje zadat objednávku „Instant“, u které je pevně stanovena cena dané

kryptoměny. Provedení této objednávky je okamžité. Dalším typem je „Stop“ umožňující automatický nákup nebo prodej při určité ceně. Díky vysoké likviditě burzy jsou transakce uzavírány rychle.

Poplatky jsou zde také jak za výběry, tak i za transakce. Pro klasické převody platí dle částky různé poplatky, minimálně však 10 USD/EUR, z tohoto pohledu je lepší využití možnosti SEPA výběrů, které mají mnohem menší poplatky. U transakcí je pak poplatek 0,25 %, který je snižován podle měsíčního obratu daného uživatele.

Stejně jako předešlá burza i Bitstamp byl napaden hackery, kteří odcizili 19 000 Bitcoinů. Díky tomuto útoku bylo zavedeno několik nových ověřovacích a bezpečnostních prvků. Burza pro zabezpečení prostředků uchovává jejich většinu (okolo 98 %) offline a tím v bezpečí před útoky hackerů.

3.2.1 API rozhraní

Bitstamp umožňuje stejně jako Bitfinex API rozhraní REST a Websocket [3]. Pro websocket se zde využívá knihovna Pusher pro stream v reálném čase. Limit pro REST API rozhraní je maximálně 600 žádostí za 10 minut. Pro zasílání požadavků slouží URL adresa „https://www.bitstamp.net/api/“. Pro soukromé volání metod POST je nutné ověření. K tomu se využívá klíč API, podpis a nonce. Klíč API lze vygenerovat v administraci burzy, nonce je celé číslo, které je zvyšováno při každém požadavku. Podpis je pak zakódována zpráva pomocí HMAC-SHA256, která obsahuje identifikátor uživatele, klíč API a nonce.

Přehled nejpoužívanějších GET metod

- **Ticker:** Jedná se o přehled trhu s danou měnou, s podobnými statistikami jako má Bitfinex. Bez upřesnění měn je brána jako výchozí dvojice BTC/USD. Umožňuje navíc hodinový ticker.
- **Orderbook:** Přehled všech otevřených objednávek pro danou dvojici měn a každá je zobrazena jako seznam obsahující cenu a množství měny.
- **Transactions:** Sestupný seznam transakcí na prodej nebo nákup měny. Umožňuje omezit seznam na určitý časový interval.
- **Trading pairs info:** Kompletní seznam všech dvojic měn obchodovatelných na burze včetně detailu.

Přehled nejpoužívanějších POST metod

- **Account Balance:** Zobrazí aktuální zůstatky měn a hodnoty poplatků pro specifikovaný účet.
- **User transactions:** Vrátí seznam všech transakcí uživatele. Metoda umožňuje omezit výběr na danou dvojici měn nebo na celkový počet.
- **Deposit/Withdrawal:** Metody slouží pro vklad/vybrání měny, které se liší pro dané měny (každá měna má svoji metodu).
- **Open/Cancel Order:** Vytvoření/stažení nabídky k obchodování s měnou.
- **Buy Limit/Market Order:** Koupení limitní nebo tržní objednávky. U tržní objednávky se její provedení řídí stavem trhu.

- **Sell Limit/Market Order:** Prodej limitní nebo tržní objednávky. Provedení tržní objednávky se opět řídí podmínkami na trhu.

3.3 Poloniex

Je jednou z nejoblíbenějších a nejkomplexnějších burz a umožňuje obchodovat s velkým počtem kryptoměn. Burza byla založena v lednu 2014 a sídlí ve Spojených státech. Nabízí bezpečné obchodování a poskytuje různé nástroje pro analýzu trhu kryptoměn a pokročilé grafy. Poskytuje dobrou likviditu. Neumožňuje vkládat fiat měny jako třeba eura nebo dolary. Vklady lze provádět pouze v Bitcoinech. Výše poplatků je u této burzy závislá na obchodování uživatele za posledních 30 dní a jejich výše se pohybuje okolo 0,4 %.

I přes používání dvoufázové autentizace protokolem SSL byla burza napadena hackery. Při tomto útoku bylo odcizeno 12 % všech bitcoinů na této burze.

3.3.1 API rozhraní

Poloniex opět nabízí API s rozhraním pro REST a Websocket [5]. Disponuje online websocket, o který se stará knihovna Trollbox pomocí protokolu WAMP. Pro API se používá URL adresa „https://api.poloniex.com“. Pro příjem informací je nutné se přihlásit k odběru jednotlivých návratových hodnot. Limit pro volání je zde nastaven na 6 požadavků za sekundu. V případě, že dojde k nadměrnému množství dat zasílaných na veřejné API rozhraní, dojde také k omezení přístupu. Pro autentizaci se používá vygenerovaný API klíč a podpis Sign, který uchovává údaje a API klíč. Podpis sign je zabezpečený metodou HMAC-SHA512.

Přehled nejpoužívanějších veřejných GET metod

- **Ticker:** Opět má stejnou funkcionalitu jako u předešlých burz. Vrací podrobnosti o měně za posledních 24 hodin.
- **24Volume:** Vrací objem všech obchodovaných měn na burze a jejich celkové počty.
- **Orderbook:** Slouží pro získání seznamu objednávek pro daný trh.
- **Trade history:** Seznam posledních uzavřených obchodů v daném časovém rozsahu.
- **Currencies:** Vrací informace o dané měně.

Přehled nejpoužívanějších POST metod

- **Complete balance:** Vrátí všechny zůstatky i na objednávkách všech měn pro daný účet. Umožňuje zařadit i marže a úvěrové účty.
- **Deposit addresses:** Seznam adres pro vklady jednotlivých kryptoměn.
- **Deposits/Withdrawals:** Vrací historii vkladů a výběrů.
- **Opened orders:** Seznam všech otevřených objednávek, pro daný pár měn.
- **Buy/Sell:** Umístění objednávky na koupi/prodej na daném trhu.

Kapitola 4

Algoritmy pro obchodování

Algoritmické obchodování s kryptoměnami je dlouho zkoumané téma a vzniklo již spousta návrhů možných algoritmů [6]. Pro automatické obchodování pomocí algoritmu je nutné využít burzu, která přes své API umožňuje získávat data nebo historii o dané kryptoměně a zadávat nabídky k nákupu nebo prodeji měny. Cílem automatizovaného nakupování je maximalizovat výdělky a minimalizovat ztráty.

Důležité při vývoji algoritmu je jeho testování a simulování. Toto testování je nutné, než přejdeme k investování skutečných peněz, a tak přenecháme důvěru nad našimi investicemi algoritmu. I z důvodů poplatků za transakce může být ladění algoritmu i poměrně nákladné. Simulaci je tedy vhodné provést nad historickými daty o směnném kurzu a očekávat správné chování algoritmu. Algoritmus nastavíme tak, aby běžel od nějakého historického data. Níže si představíme několik jednoduchých principů funkce algoritmů.

Příklad jednoduchých algoritmů

Jednoduchý algoritmus Tento algoritmus udržuje stále stejnou investovanou částku bez ohledu na jiné okolnosti. Pokud cena klesne, prodá rozdíl zisku s původní hodnotou, pokud cena klesá, nakoupí množství kryptoměny až dokud nedosáhne požadované výše investice.

Statický algoritmus Nakupuje v plné kapacitě prostředků a jednoduše čeká podle vývoje trhu. Může vytvářet zisk nebo ztrátu podle trhu.

Náhodný algoritmus Náhodně nakoupí nebo prodá pevnou částku měny.

Pirátský algoritmus Tento algoritmus vychází z ceny měny při nákupu nebo prodeji. Pokud se hodnota měny pohybuje směrem dolů, tak začne prodávat, aby zabránil ztrátě. V případě, že cena začíná stoupat, bude nakupovat.

Tyto algoritmy představují ne příliš sofistikované řešení, ale představují možnost, jak lze provádět automatické obchodování s kryptoměnou. Pro různé měny s různou volatilitou jsou vhodné různé algoritmy a jejich modifikace. Obchodování některých akcií může fungovat lépe, když je algoritmus ochotný investovat větší částku.

U všech typů algoritmů je spíše efektivnější vybírat trh, který stoupá. Na klesajícím trhu by algoritmus mohl maximálně snížit ztráty. Vhodný výběr je tedy nutný pro generování zisku. Další vlastnost, kterou je nutné zohlednit je výše poplatků u dané kryptoměny a dané burzy. Poplatky mohou vážně ovlivnit výnosnost algoritmů.

Algoritmus, který bude implementován v této práci kombinuje některé jednoduché principy z předešlých příkladů, a navíc přidává větší množství konfigurace a ovlivňování chování při nákupu a prodeji měny. V následující části bude tento princip fungování více rozveden.

4.1 Algoritmus pro obchodování na burze

Jak již plyne z předchozích částí této práce je obchodování s kryptoměnami velice složité a nevyzpytatelné. Nelze proto vytvořit nějaký statický algoritmus, který bude schopný generovat zisk, bez nutnosti zavedení několika konfiguračních hodnot, které budou do jisté míry ovlivňovat jeho chování. Při jejich nastavování musí být zohledněna aktuální situace ve světě kryptoměn. Při rostoucím trendu lze vytvořit odvážnější konfiguraci, která má potenciál generovat větší zisk.

Algoritmus tedy zpracovává data získaná prostřednictvím rozhraní z kryptoměnové burzy. Než přistoupíme k detailnímu popisu zpracovávání těchto dat, tak si zavedeme seznam konfiguračních hodnot a nastavení, které ovlivňují chování algoritmu.

Přehled pojmů a konfiguračních hodnot

- **Counter currency:** Měna, kterou lze na dané burze nazvat jako tzv. výchozí měnu. Výchozí měna slouží k nákupu ostatních altcoinů a kryptoměn. Nejčastějšími měnami jsou právě fiat měny jako je dolar nebo euro, z kryptoměn pak Bitcoin nebo Ethereum. Tento seznam bývá odlišný u každé burzy.
- **Volume:** Jedná se o objem transakcí/obchodů dané kryptoměny za určité období. Například souhrnný počet jednotek dané měny zobchodovaných za posledních 24 hodin.
- **Transakce:** V této práci reprezentuje časový interval od nákupu (otevření transakce), až po prodej (uzavření transakce). Uchovává všechny nutné hodnoty pro následné zpracování a prodej.
- **Up ratio:** Reprezentuje procentuální rozdíl mezi aktuální cenou („last value“) a tou nejnižší v daném časovém úseku. Dle názvu se tedy jedná o stoupající index/úroveň.
- **Down ratio:** Reverzní úroveň k předešlé stoupající úrovni, která reprezentuje procentuální rozdíl mezi aktuální („last value“) a tou nejvyšší cenou, jedná se tedy o klesající index/úroveň.
- **Pump ratio:** Úroveň při jejímž dosažení je proveden nákup dané měny (otevření transakce).
- **Stop ratio:** Úroveň při jejímž dosažení je indikován klesající trend (nebo kolísající) a daná měna je na určitý čas ignorována.
- **Interval size:** Též lze nazvat jako velikost tzv. „okna“, které reprezentuje časový interval pro ukládání hodnot k výpočtu úrovní.
- **Ignore time:** Čas po který je daná měna ignorována při výpočtu.
- **Diameter:** Označuje míru zaokrouhlování hodnot.

- **Level:** Specifikují hodnoty hranic úrovní při zpracování otevřené transakce. Postupným aplikováním těchto levelů, lze přesně nastavovat spodní i horní hranici.

Výchozím chováním algoritmu je nacházet tzv. „pumpy“, neboli prudké a rychlé nárůsty ceny různých ostatních měn oproti námi vlastněné měně. V této části si představíme myšlenku algoritmu. Popis následné implementace se nachází v kapitole 5.4.4. Nejprve si předvedeme funkci algoritmu, která spočívá pouze v nakupování měny při protnutí jisté úrovně a její prodej s možným ziskem při změně trendu ze stoupajícího na klesající.

4.1.1 Pseudokód algoritmu

Pro snazší přehlednost a pochopitelnost následujícího popisu je níže nastíněn algoritmus v podobě pseudokódu. Ten však na rozdíl od reálné implementace obsahuje určité abstrakce, v podobě vhodně pojmenovaných metod. Podrobný popis pseudokódu se nachází v následující sekci 4.1.2.

```
1  while(true) /* nekonecny cyklus */
2  {
3    /* ziskani dat z~burzy pro pary, ktere nas zajimaji */
4    List<Ticker> filtredPairTickers = exchgangeManager.getTickers();
5    for(Ticker pairTicker : filtredPairTickers)
6    {
7      /* test, zda jiz otevrena transakce */
8      if(isOpenedTransaction(pairTicker))
9        /* zpracovani otevrene transakce, pripadny prodej nebo urovni */
10       processOpenedTransaction(pairTicker);
11
12     /* podminka na dostatecny objem transakci */
13     if(pairTicker.volume < minimumVolume)
14       continue;
15
16     /* kontrola, zda je par ignorovan a jestli nema byt ignorovani ukonceno */
17     if(isIgnoringTimeReached(pairTicker))
18       stopIgnoringPair(pairTicker); /* ukonceni ignorovani paru */
19
20     /* pokud neni ignorovany bude zpracovavan */
21     if(!isIngnored(pairTicker))
22     {
23       BigDecimal upRatio = calculateUpRatio(pairTicker); /* vypocet stoupajici urovne */
24       BigDecimal downRatio = calculateUpRatio(pairTicker); /* vypocet klesajici urovne */
25
26       /* test, zda nebyla dosazena uroven pro nakup */
27       if(upRatio > pumpRatio)
28       {
29         /* nakup kryptomeny, vytvoreni transakce pro rizeni */
30         buyAndCreateTransaction(pairTicker);
31       }else if(downRatio < stopRatio)
32       {
33         /* zacatek ignorovani paru na urcitou dobu*/
34         startIgnoringPair(pairTicker, ignoreTime);
35       }
36     }
37   }
38 }
```

4.1.2 Popis funkce algoritmu

Algoritmus pracuje v cyklu, kde na začátku každého cyklu nejprve vyčte aktuální data pro všechny páry z burzy. Z těchto dat jsou vybrány ty kryptoměnové páry, které odpovídají nastavení a jejich měna pro směnu („Counter currency“) je stejná s námi vybranou a vlastněnou výchozí měnou. Tento seznam kryptoměnových párů obsahuje všechny nutné údaje o ceně a aktuálním stavu z burzy v podobě tzv. „Tickerů“, kde pro každý pár je právě jeden Ticker. Nejdůležitější hodnotou v tickeru je aktuální cena („Last value“). Tato cena udává hodnotu dané měny při posledním uskutečněném obchodu na burze. Mezi další užitečné hodnoty, které jsou obsaženy v každém tickeru, patří například již zmiňovaná hodnota „volume“, hodnoty aktuální ceny nabídky („BID“) a poptávky („ASK“) nebo údaje o nejvyšším („HIGH“) a nejnižším („LOW“) kurzu měny za posledních 24 hodin. Jejich aktualizace probíhají několika sekundových intervalech. Jejich délka je různá u jednotlivých burz. Obvykle se pohybuje kolem jedné sekundy.

Po načtení a vyfiltrování vybraných párů je provedeno postupné zpracování všech těchto párů. Nejprve je ověřeno, zda již tento pár nemá otevřenou transakci. To znamená, že algoritmus v nějakém předchozím cyklu provedl nákup měny. Pro pochopení souvislostí si nejprve představíme kroky vedoucí k nákupu/otevření transakce. Z důvodu bezpečnosti a minimalizace rizik je nutné obchodovat pouze s páry, které se aktuálně nejvíce obchodují, a jejich tzv. „Volume“ je vysoké. Pokud tato hodnota je nízká, může se stát, že jeden ojedinělý objemnější nákup nebo prodej může snadno ovlivnit skokově hodnotu dané měny, proto je méně riskantní obchodování s páry s větším objemem transakcí v daném časovém rozpětí (obvykle 24 hodin).

Další funkcionalitou algoritmu je možnost ignorování daného páru po určitý čas. Pokud je pár ignorován, nejsou aktuální hodnoty ukládány do paměti. Ignorování páru je využito při detekci klesání, tedy hodnota úrovně klesání („donwRatio“) se sníží pod nastavenou úroveň pro pozastavení zpracování daného páru („stopRatio“) nebo po ukončení transakce. V těchto případech je nastavena doba, na jak dlouho se má pár ignorovat dle hodnoty konfigurovatelné „Ignore time“. U těchto párů je následně kontrolována doba do kdy mají být ignorovány. Pokud již aktuální čas přesáhne tuto hodnotu je obnoveno zpracování páru. Při obnovení je vyresetována paměť všech hodnot a načítání začíná od začátku. K následujícímu zpracování tedy zůstanou pouze páry, které mají dostatečné volume a nejsou ignorovány.

Zpracování a otevírání transakcí

Správná konfigurace a zvolení mezních hodnot pro vstup do transakce nebo pozastavení zpracovávání daného páru je velmi náročné. Různé problémy a ukázky nastavení budou představeny v následujících částech této práce. Nyní se zaměříme na kroky algoritmu při splnění jedné z těchto podmínek. Nejprve je nutné provést výpočet aktuálních úrovní pro stoupání („upRatio“) a klesání („downRatio“). Před výpočtem úrovní je právě získaná aktuální hodnota („Last value“) vložena mezi uchovávané hodnoty v okně a hodnoty, které již svým stářím vypadly z intervalu, jsou odebrány. Z takto aktualizovaného pole je vybrána nejnižší hodnota pro výpočet stoupající úrovně a nejvyšší pro klesající. Hodnoty úrovní reprezentují procentuální rozdíl mezi nejnižší/nejvyšší hodnotou v okně a aktuální poslední hodnotou ceny dané měny.

Vypočítané hodnoty úrovní jsou určeny pro zjištění, zda bude provedena nějaká akce s daným párem. Pokud stoupající hodnota překročí hranici pro nákup, je měna nakoupena a provede se vytvoření nové transakce. Ta následně uchovává údaje potřebné pro pozdější prodej a následný report o bilanci. V transakci je uchováno množství nakoupené měny,

nákupní cena a nejnižší hodnota, která je nejdůležitější položkou pro pozdější zpracování. Tato hodnota se podílela na výpočtu úrovně, která překročila hranici pro nákup. V případě neuchování této hodnoty, by mohlo po určitém čase dojít k odstranění této nejnižší hodnoty z paměti (posuv intervalu) a tím ke klesání úrovně. Transakce také obsahuje úrovně z tzv. „levelů“, které slouží pro řízení následného prodeje. Podrobný popis těchto úrovní a jejich významu a nastavování je popsán v následující sekci, která se zabývá zpracováním otevřené transakce.

Jak již bylo zmíněno, při zpracování je vypočítávána kromě stoupající úrovně i klesající. V případě, že tato úroveň klesne pod hranici reprezentující pozastavení zpracovávání, je tomuto páru nastaven čas ignorování. Pokud úrovně neprotnou ani jednu z těchto hranic, je výsledkem pouze aktualizace paměti posledních hodnot.

Zpracování otevřené transakce

Na začátku zpracování je vypočtena nová hodnota úrovně mezi poslední aktuální hodnotou a nejnižší hodnotou/cenou měny při otevření transakce. Je nutné pro výpočet použít nejnižší hodnotu, která byla uložena při vytváření transakce, protože ji můžeme brát jako výchozí bod, který započal růst. Pokud by byla dále využívána nejnižší hodnota z pole hodnot v daném okně, mohlo by dojít k poklesu úrovně kvůli pohybu okna i když by hodnota vlastněné měny dále stoupala (nejnižší hodnoty by vypadávaly z paměti pro své stáří).

Po otevření transakce jsou nastaveny hranice, které jsou specifikovány pomocí levelů. Při dosažení horní hranice se provede posun obou hranic výše. Pokud aktuální úroveň spadne pod dolní hranici bude proveden prodej. Z toho tedy plyne, že čím více horních hranic bude protnuto, tím se hranice posunou výše a při případném poklesu bude proveden prodej na vyšších hodnotách, a tedy vytvoří zisk. Po prodeji měny je uzavřena transakce, odstraněny všechny hodnoty z paměti pro daný pár a nastaveno ignorování na definovanou dobu.

Mezní úrovně a jejich nastavování pomocí levelů

Slouží pro nastavování hranic pro zpracování otevřených transakcí. Při nákupu je vždy aplikován první level a pak následně při možném přesáhnutí horní hranice se nastavují další levely. Nelze odhadnout, jak hodně může úroveň stoupnout. Proto je nutné umožnit i dynamické vypočítávání dalších úrovní v případě, že je dosažena úroveň nastavená posledním levelem. Z toho důvodu bude vždy nejvyšší level obsahovat hodnoty (inkrementy) pro dynamické dopočítávání dalších úrovní. Díky tomu bude posun úrovní plynule pokračovat a tím zvedat dolní hranici pro prodej, která je spojena s výší zisku.

Různé nastavování těchto úrovní dramaticky ovlivňuje chování algoritmu při zpracování otevřené transakce. Pokud například nastavíme dolní hranici pro prodej příliš vysokou, může algoritmus danou transakci prodat při malém výkmitu hodnot. V kapitole 4.2 je představen příklad fungování algoritmu, na kterém je ukázáno nastavování hranic pomocí levelů.

Při popisu levelů byl nastíněn možný problém s náhlým a krátkodobým výkmitem ceny dané měny, což může vést k nechtěnému nákupu nebo předčasnému prodeji. Jednou z možností, jak tento problém řešit je zavedení průměrování hodnot (v nastavení algoritmu odpovídá proměnné „Diametr“). Tato hodnota říká, kolik hodnot má být začleněno do průměrování. Pokud tedy nastavíme jeho hodnotu n, tak algoritmus nepočítá pouze s nejnižší hodnotou, ale s průměrem n hodnot kolem této hodnoty. V případě, že se daná hodnota nachází na začátku nebo na konci pole reprezentujícího paměť hodnot (aktuální hodnota je

vždy přidávána na konec), pak je pro výpočet průměru bráno n hodnot od začátku, respektive konce pole. Stejné zaokrouhlení je aplikováno i na nejvyšší hodnotu a aktuální hodnotu (do průměru je zahrnuto n předcházejících hodnot). Při vyšším počtu hodnot určených k průměrování dochází k vyhlazení výkmitů, ale tato funkcionalita má i své nevýhody.

Nevýhodou takto průměrovaných hodnot je pozdější reakce na prudký růst a s tím tedy spojená opožděná reakce na nákup nebo prodej měny. Algoritmus, který nebude využívat průměrování hodnot pak může reagovat rychleji. Z toho plyne, že počet hodnot pro počítání průměru nesmí být moc vysoký.

4.2 Příklad fungování algoritmu a možné problémy

V předchozí části byl představen princip algoritmu. Pro lepší pochopení a nastínění možných problémů vyspecifikujeme nějaké referenční nastavení konfiguračních hodnot algoritmu (budou uvedeny pouze některé údaje potřebné pro prezentaci funkce algoritmu).

Konfigurační hodnoty

- **Pump ratio:** 10 %
- **Stop ratio:** - 10 %
- **Interval size:** 30 minut
- **Ignore time:** 10 minut
- **Minimal volume:** 10 BTC (Bitcoinů)
- **Diameter:** 1 (pro zjednodušení ilustrace nebudeme průměrování využívat)
- **Levely:**
  1. Level: Up ratio = 11 %, Down ratio = 9 %
  2. Level: Up ratio = 12 %, Down ratio = 10 %, nárůstek pro dynamické dopočítání další úrovní (inkrement) = 1 %

S pomocí těchto hodnot si nyní představíme různé scénáře vývoje stoupající i klesající úrovně pro pár. Nejprve si představíme situaci před jeho koupením a následně i situaci po nákupu a otevření transakce.

4.2.1 Výpočet úrovní a detekce kolísání hodnoty měny

Než budou představeny ukázkové scénáře vývoje cen měn, neboli algoritmem zpracovávaných úrovní, tak na příkladu bude ukázán výpočet těchto úrovní. Jak již bylo zmíněno, v algoritmu se vypočítávají dvě tzv. úrovně. První úroveň je označována jako stoupající, která procentuálně udává rozdíl mezi nejnižší hodnotou v paměti pro daný pár a tou poslední aktuální. Též můžeme tuto úroveň označit jako index stoupání. Druhá úroveň je obrácená k té první a znázorňuje klesající průběh, neboli index klesání. Udává tedy procentuální rozdíl mezi nejvyšší hodnotou v paměti a aktuální hodnotou.

Na grafu 4.1 je nastíněn možný vývoj hodnoty kryptoměny v určitém časovém úseku. Jsou zde zaznamenány tři časové body, ve kterých je ukázán výpočet úrovní a celková demonstrace principu výpočtu. Tato ukázka nijak nesouvisí s konfiguračními hodnotami,

ukazuje pouze ilustrativní hodnoty a následný výpočet úrovní bez využívání průměrování hodnot.

Výpočet úrovní

Pro výpočet se vždy využívá aktuální nejmenší hodnota („low“), největší hodnota („high“) a poslední aktuální hodnota („last“).

Příklady výpočtu v jednotlivých časových bodech vývoje hodnoty měny

1. Výpočet:

$$low = 46.75, \quad high = 57.5, \quad last = 57.5$$
$$windowUpRatio = \frac{last}{low} - 1 = \frac{57,5}{46,75} - 1 \doteq 23\%$$
$$windowDownRatio = \frac{last}{high} - 1 = \frac{57,5}{57,5} - 1 = 0\%$$

2. Výpočet:

$$low = 36, \quad high = 57.5, \quad last = 36$$
$$windowUpRatio = \frac{last}{low} - 1 = \frac{36}{36} - 1 = 0\%$$
$$windowDownRatio = \frac{last}{high} - 1 = \frac{36}{57,5} - 1 \doteq -37\%$$

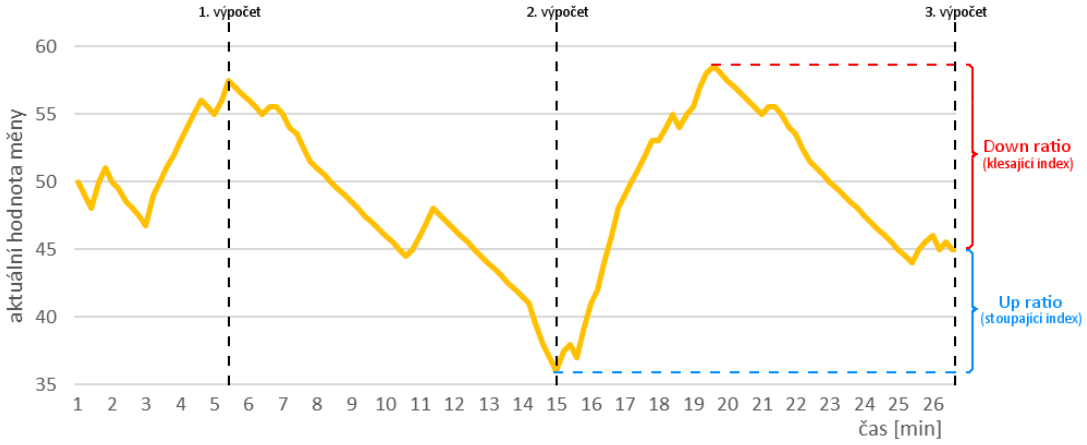
3. Výpočet:

$$low = 36, \quad high = 58,5, \quad last = 45$$
$$windowUpRatio = \frac{last}{low} - 1 = \frac{45}{36} - 1 = 25\%$$
$$windowDownRatio = \frac{last}{high} - 1 = \frac{45}{58,5} - 1 \doteq -23\%$$

Před prvním výpočtem křivka rovnoměrně roste, proto je hodnota stoupající úrovně větší než nula. Hodnota klesající úrovně je rovna nule, protože poslední hodnota je zároveň i ta nejvyšší. U druhého výpočtu je přesně opačná situace. Hodnota měny začala prudce klesat, a proto klesající úroveň indikuje procentuální pokles od nejvyšší hodnoty v paměti. Stoupající úroveň je zde nulová, protože analogicky, jako u prvního příkladu, je zde nejmenší hodnota rovná té poslední aktuální hodnotě měny. Před třetím výpočtem hodnota opět vzrostla a nastavila novou nejvyšší hodnotu. Nicméně opět klesla a ustálila se. V tomto stavu jsou nenulové obě úrovně. Z uvedených výpočtů pak plyne fakt, že pokud jedna nebo druhá úroveň je nulová, tak dochází k rovnoměrnému růstu nebo poklesu hodnoty měny.

Detekce kolísání

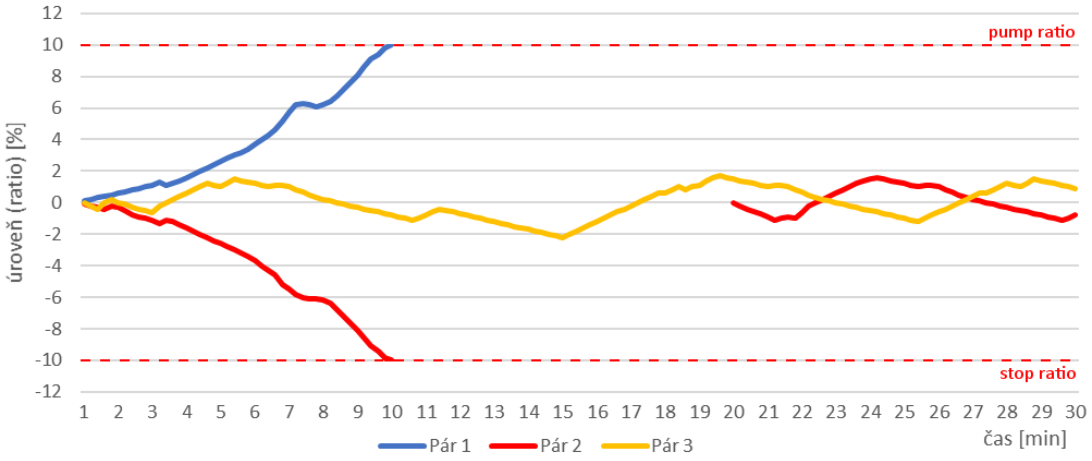
V předcházející části je popsán stav, kdy jedna nebo druhá úroveň je nulová. Jak ale plyne ze třetího výpočtu, mohou být obě tyto úrovně nenulové. To poukazuje na fakt, že hodnota měny byla v daném okně vyšší i nižší, než je aktuální poslední hodnota. Z toho faktu tedy plyne, že pokud jsou obě úrovně nenulové, dochází ke kolísání. Měnu s tímto chováním lze pak vynechat ze zpracování, z důvodu možných riskantních transakcí.



Obrázek 4.1: Ukázka výpočtu úrovní v různých stavech vývoje hodnoty měny.

4.2.2 Scénář po spuštění algoritmu

Výchozím stavem scénáře je spuštění algoritmu, tedy paměť (okno) hodnot je prázdná. Postupnými iteracemi jsou hodnoty doplňovány do paměti a slouží pro výpočet nových úrovní. Měny, které nesplňují minimální hodnotu objektu transakcí („Minimal volume“) za posledních 24 hodin nejsou vůbec zpracovávány. Při tomto zpracovávání může dojít ke třem různým průběhům vývoje hodnot úrovní sledovaného páru. Tyto tři typy jsou znázorněny na grafu 4.2. V tomto grafu jsou uvedeny i hranice pro nákup a pozastavení sledování daného páru („pump ratio“ a „stop ratio“). Je zde znázorněna práce algoritmu s jednotlivými sledovanými páry, vyjma zpracování již otevřených transakcí.



Obrázek 4.2: Ukázka možných průběhů vývoje hodnot po spuštění algoritmu.

První křivka (Pár 1) reprezentuje pár, jehož hodnota stoupá a s tím roste i stoupající úroveň, která po čase protíná hranici pro nákup dané měny. V tomto případě se jedná o ideální průběh, který vede postupným růstem ke koupi měny. Následné možné průběhy po otevření transakce jsou popsány v sekci 4.2.3.

Další křivka (Pár 2) ukazuje opačnou situaci, kdy daný pár klesá a po čase protne hranici pro pozastavení. Pár je tedy na dobu určenou konfigurací (10 minut) vynechán

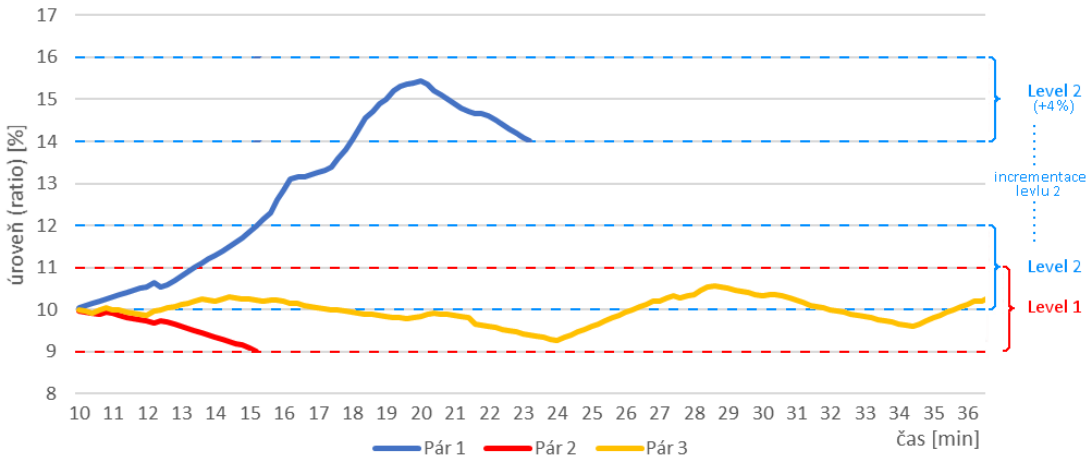
ze zpracování (ignorování páru). Po uplynutí doby ignorování, začíná tento pár opět od začátku s prázdným oknem hodnot. V našem příkladu dojde k protnutí spodní hranice po 10 minutách. Proto je následně pár ignorován a další zpracování probíhá až od 20 minuty.

V neposlední řadě je v grafu pro doplnění i poslední třetí křivka (Pár 3), která ukazuje různě kolísající pár. Toto chování je obvyklé a znázorňuje i například pomalu rostoucí měnu, která v daném okně není schopna dosáhnout na úroveň pro nákup.

4.2.3 Scénář po provedení nákupu (otevření transakce)

Výchozím stavem scénáře je otevření transakce, neboli provedení nákupu dané měny. V této části se již počítá pouze jedna úroveň, která udává procentuální vztah mezi nejnižší hodnotou uvedenou při otevření transakce a aktuální hodnotou získanou z burzy. Takto získaná úroveň je využita pro zpracování a její možné průběhy jsou zobrazeny v grafu 4.3. Stejně jako v předešlé části, která popisovala vývoje před otevřením transakce, máme i zde hraniční úrovně, ale na rozdíl od statických úrovní („pump ratio“ a „stop ratio“) jsou tyto úrovně dynamicky upravovány dle aktuálního stavu.

Pro nastavování hraničních úrovní se využívají levely. Pro tento příklad jsou nastaveny dva levely a hodnoty z prvního levelu jsou vždy nastaveny hned po otevření transakce. Hodnoty z druhého levelu jsou nastaveny pro daný pár, pokud jeho hodnota aktuální úrovně dosáhne horní hranice předchozího levelu. Dle definice levelů z části 4.1.2 obsahuje nejvyšší level (v tomto případě level 2) hodnoty pro dynamické navyšování hranic, které jsou využívány při dosažení aktuálně nastavené horní hranice. Pro aktuální nastavení levelů by bylo možné level 2 vynechat (uveden pro pochopení funkce levelů) a hodnoty pro zvyšování hranic přesunout na level 1, který by následně pokrýval stejnou množinu hraničních úrovní.



Obrázek 4.3: Ukázka možných průběhů vývoje hodnot při zpracování otevřené transakce.

V grafu 4.3 jsou opět tři křivky. Za jejich počáteční stav můžeme brát stav křivky páru 1 z grafu 4.2 při dosažení úrovně pro nákup a provedení otevření transakce. První křivka (Pár 1) znázorňuje dále stoupající pár, který dosáhne horní hranice prvního levelu a následně i několik dalších úrovní. Avšak pár nebude růst do nekonečna. Při opětovném poklesu a následném dosažení dolní hranice, aktuálně nastaveného levelu, dojde k prodeji. V tomto případě pár dosáhne nejvyšší úrovně přibližně 15,5 procenta a pak začne pozvolně klesat. Při poklesu na úroveň 14 procent je transakce ukončena a měna prodána se 4procentním ziskem. Po prodeji je pár zařazen mezi ignorované a je odstraněn z paměti. Je to z dů-

vodu omezení vlivu případného rozkolísání hodnoty měny krátce po prodeji a tím zamezení dalšího nákupu. Hodnota úrovně vypočítaná po době ignorování by mohla být ovlivněna starými hodnotami. Tato úroveň by pak mohla být vyšší než je potřebná úroveň pro otevření transakce. Tento problém by se mohl vyskytnout u všech nastavení, která mají dobu ignorování kratší než je délka okna pro hodnoty. Naopak by hodnoty v okně expirovaly dříve, než by skončilo ignorování páru. Tento problém je řešen mazáním dat v okně pro daný pár, při začátku ignorování.

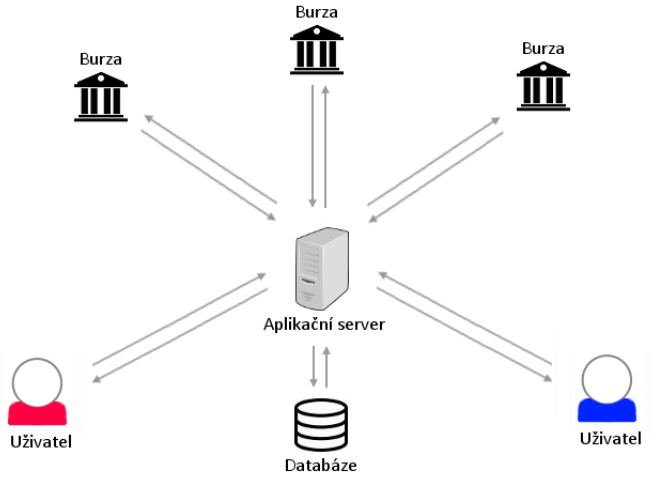
Další křivka (Pár 2) reprezentuje nejhorší možnou situaci, která při práci s tímto algoritmem může nastat. Po provedení nákupu měny začne hodnota hned klesat a při dosažení dolní hranice nastaveného levelu 1 dojde k jejímu prodeji. Tato situace tedy znamená prodej se ztrátou, která při použitém nastavení znamená přibližně jedno procento. Velikost možného prodělku lze ovlivňovat nastavením dolní hranice u prvního levelu. Avšak pokud hranici nastavíme příliš vysokou, může dojít při krátkodobém malém poklesu k předčasnému a rychlému prodeji s prodělkem. Nebo v případě moc nízké hranice pak analogicky k možným vyšším ztrátám. Třetí křivka (Pár 3) reprezentuje pár, který dosáhl hranice pro nákup, ale následně došlo k jeho ustálení na této úrovní. Díky malému kolísání jeho úrovně nemusí dlouhodobě dojít k dosažení některé z hranic nastaveného levelu. Řešením pak může být ruční prodej daného páru.

Při nákupech a prodeji je nutné započítat ještě povinný poplatek. Výše poplatků se z pravidla liší pro různé burzy, ale bývá okolo 0,25 % z obchodovaného množství měny. Z toho plyne, že každá transakce (nákup a prodej) odvede 0,5 procenta z obchodovaného množství. Při uvažování těchto poplatků může takto nastavený algoritmus prodělat až 1,5 procenta při nepříznivém průběhu.

Kapitola 5

Návrh a implementace aplikace

Cílem při návrhu a následné implementaci bylo vytvořit aplikaci, umožňující nastavování, simulaci a běh navrženého algoritmu z kapitoly 4. Při návrhu struktury a používaných technologií bylo nutné se zaměřit na nástroje, které mohou být spuštěny dlouhodobě a lze jejich běh snadno řídit. Nejdůležitějšími parametry je nutnost nepřetržitého připojení k internetu, které je nutné pro získávání aktuálních dat z burzy v malých časových intervalech (přibližně 30-40 dotazů za minutu) a dostatečný výkon nutný pro zpracování těchto dat. Dle navrženého algoritmu a principů analýzy burzy je nutné, aby aplikace běžela nepřetržitě a bylo možné rychlé obnovení po případném odpojení nebo výpadku elektrické energie.



Obrázek 5.1: Schéma struktury zapojení navrhované aplikace.

Na obrázku 5.1 je nastíněno schéma komunikace aplikace mezi jednotlivými prvky. Aplikační server na kterém bude spuštěna aplikace bude připojen k databázovému serveru. Přes uživatelské rozhraní umožní konfiguraci algoritmu a přes rozhraní burz bude získávat data.

Implementovaná aplikace nese jméno „XTrader“, které je příznačné pro její využití, a můžeme ji rozdělit na dvě části. Jednou z částí je modul pro načítání dat z burzy a jejich následné zpracování. Druhou částí pak je samotné grafické uživatelské rozhraní umožňující nastavení aplikace, správu běhu algoritmů a prezentaci jejich výstupů a výsledků. Návrh počítá s více vlákovým zpracováváním, které umožní běh více instancí algoritmu. Jedná se

tedy o serverovou aplikaci, která pro svůj běh využívá aplikační server a pro ukládání dat a konfiguračních hodnot SQL databázi (podrobný popis využívaných technologií je v následující části 5.1). V dalších částech této kapitoly je představen podrobný popis struktury aplikace a implementace jednotlivých modulů.

5.1 Použité technologie

Jak již bylo zmíněno, aplikace je napsána v jazyce Java. Hlavním důvodem výběru jazyka je existence knihovny XChange, která je popsána v části 5.1.1, umožňující napojení na rozhraní různých kryptoměnových burz. Celý projekt je koncipován jako Maven projekt, který umožňuje jednoduchou správu knihoven a přehlednost. Pro běh aplikace je využíván aplikační server WildFly, který je volně dostupný a disponuje dobrým výkonem a podporou pro běh webových aplikací. Z pohledu struktury aplikace je pro grafické uživatelské rozhraní využit JSF (JavaServer Faces) framework Primefaces, který obsahuje podporu v podobě všech důležitých řídících a zobrazovaných objektů jako jsou tlačítka, tabulky a další užitečné komponenty. Pro uchovávání dat je využita Microsoft SQL databáze, která běží na Microsoft SQL Serveru. Aplikaci postačuje bezplatná verze Express této MS SQL databáze. O napojení Javy a databáze se stará JPA (Java Persistence API) framework Hibernate, který umožňuje přímé objektově relační mapování entit v databázi.

5.1.1 Knihovna XChange

Jedná se o knihovnu pro jazyk Java, která poskytuje jednoduché a konzistentní rozhraní pro komunikaci s nejrozšířenějšími kryptoměnovými burzami (více jak 60 burz). Umožňuje přistupovat k datům z burzy a provádění obchodů. Vyznačuje se jednoduchým použitím. Díky velké podpoře je neustále vylepšována a udržována v aktuálním a funkčním stavu. Vývojáři dynamicky reagují na možné změny API rozhraní jednotlivých burz a vydávají úpravy. Knihovna je navržena tak, aby poskytovala sadu rozhraní pro aktivní podporu vytváření napojení na nové burzy.

Zapouzdřuje data z jednotlivých burz do obecných struktur. Tyto struktury pak lze využít pro vytváření algoritmů, které nejsou závislé na struktuře dat pouze jedné burzy. Podporuje přidávání jednotlivých komponent, které jsou specializovány pro dané burzy. Knihovna XChange má svobodnou licenci MIT, která umožňuje její využití pro tuto práci. Pro vybrané burzy (Bitfinex, Bitstamp a Poloniex) má knihovna plnou podporu čtení dat, obchodování i správy účtu.

Přehled základních komponent

Krátký popis základních komponent knihovny XChange.

Exchange Hlavní komponenta celé interakce s burzou. Poskytuje a provádí volání rozhraní jednotlivých burz. Umožňuje snadné rozšíření.

ExchangeSpecification Komponenta s veškerou konfigurací burzy pro hlavní komponentu Exchange. Obsahuje funkce pro ověření (tajné API klíče, atd.).

ExchangeFactory Vytváří instance komponenty Exchange na základě konfigurační komponenty ExchangeSpecification.

Exchange metadate Jedná se o všechny další informace ohledně burzy a jednotlivých parametrů. Mezi tyto údaje patří například: seznam měnových párů, poplatky, minimální částky apod.

ExchangeServices Služby se dělí na tři základní. „MarketDataService“ slouží pro získávání aktuálních dat ohledně kryptoměn z burzy, „TradeService“ pro vytváření obchodů a „AccountService“ pro zprávu uživatelského účtu.

5.2 Struktura a základní části

Aplikace je rozdělena na dvě větší části. Nejprve si představíme jednoduší část, která zahrnuje uživatelské rozhraní a následně si popíšeme druhou část, která obsahuje veškerou výpočetní logiku včetně implementace navrženého algoritmu.

V implementaci aplikace nebylo prozatím uvažováno přihlašování uživatelů a celková správa účtů (v databázovém modelu je se správou uživatelů již počítáno). Aplikace po spuštění nevyžaduje v této prototypové verzi prozatím žádné přihlášení. Tato funkcionalita nemá vliv na zkoumanou práci algoritmu, a proto bylo rozhodnuto, že nebude ani součástí této práce. Nicméně veškeré údaje, jako například klíče k rozhraní burz, jsou před uložením do databáze zabezpečeny hashovací funkcí MD5 s rozšířeným zabezpečením.

Přehled pojmů a názvosloví

Pro snazší pochopení a vysvětlení si zavedeme důležité pojmy a názvosloví, které bude využívané v následujících částech.

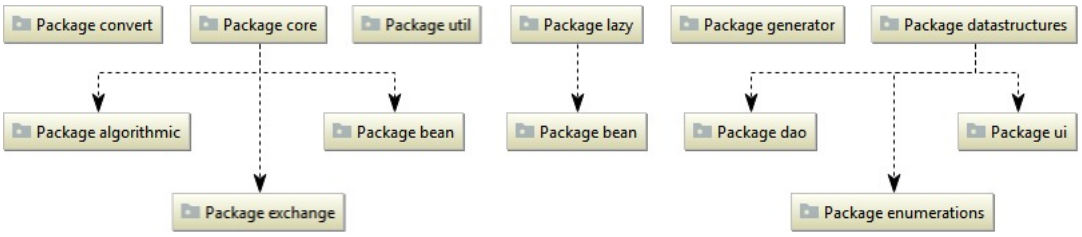
- **Exchange Manager (manažer burzy):** Označení pro rozhraní, které definuje strukturu manažerů pro správu burz. Každý manažer uchovává všechny služby pro komunikaci s danou burzou. Pro každou burzu je tedy nutná implementace tohoto manažera.
- **Exchange Listener (načítání dat burzy):** Slouží pro periodické pravidelné načítání aktuálních údajů z burzy a jejich zprostředkovávání k dalšímu zpracování.
- **Processor (procesor):** Jedná se o označení implementace navrženého algoritmu. Jeden procesor označuje jednu instanci algoritmu běžící v samostatném vlákně. Dosud jsme se bavili o konfiguraci algoritmu, nyní v implementaci se bude jednat o konfiguraci procesoru.

5.2.1 Diagram balíků aplikace

Základní struktura aplikace na úrovni jednotlivých balíků („packages“) je znázorněna na obrázku 5.2. Do těchto balíků jsou dle funkce rozděleny jednotlivé části aplikace. Pro pochopení a vysvětlení funkce balíků je zde uveden jejich stručný popis:

- **Package convert:** Tento balík obsahuje třídy pro konverzi databázových entitních objektů na objekty pro uživatelské rozhraní („ui“ objekty).
- **Package core:** Nejdůležitější balík reprezentující jádro aplikace. Kvůli jeho obsáhlosti a důležitosti je podrobnější popis s diagramem uveden samostatně v kapitole 5.4.2.

- **Package util:** Pomocné třídy aplikace (správa lokalizací, konstanty a další pomocné funkce).
- **Package lazy:** Obsahuje tzv. „Lazy modely“ určené pro dynamické načítání dat z databáze do tabulek aplikace.
  - **Package bean:** Pro jednotlivé lazy modely obsahuje třídy s metodami pro získávání dat z databáze.
- **Package generator:** V tomto balíku se nachází generátor pro dynamické generování jednotlivých komponent do uživatelského rozhraní (tabulky, tlačítka, menu atd.).
- **Package datastructures:** Druhý nejdůležitější balík, který obsahuje všechny entitní třídy reprezentující dané tabulky v databázi.
  - **Package dao:** DAO třídy obstarávající operace s daným objektem nad databází. Mezi tyto operace patří například: uložení, aktualizace, smazání nebo získání dat z databáze.
  - **Package ui:** Objekty pro uživatelské rozhraní („User interface“), které obsahují lokalizované atributy pro zobrazení.
  - **Package enumeration:** Výčtové objekty pro stavové proměnné (např.: stav procesoru).



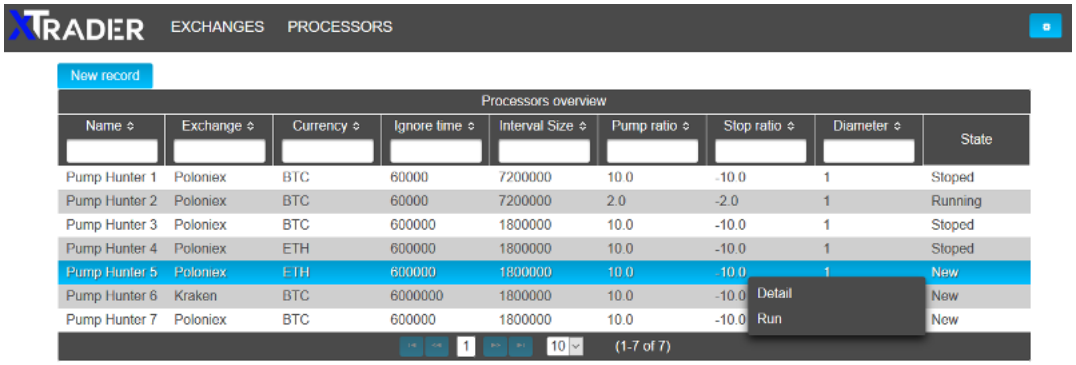
Obrázek 5.2: Diagram základních balíků aplikace.

5.3 Uživatelské rozhraní

Při tvorbě uživatelského rozhraní byl kladen důraz na tvorbu co nejjednoduššího a nejefektivnějšího rozhraní, které bude disponovat snadným a intuitivním ovládáním, umožňující konfiguraci a následné zobrazení výsledků běhů procesorů. Jako výchozí jazyk byla zvolena angličtina, z důvodu snazšího popisování algoritmu a díky její univerzálnosti. Přidání dalších jazyků je podporováno a jednalo by se pouze o vložení tlačítka na jejich přepínání. Základ implementace rozhraní je postaven na dynamickém renderování stránek, které je řízeno z kódu Javy. Aplikace tak neobsahuje velké množství statických stránek v podobě XHTML dokumentů. Díky této struktuře a obecným generátorům jednotlivých komponent, je snadné implementovat možné rozšíření a další funkce. Využitím frameworku Primefaces, jehož všechny komponenty podporují responzivní design, je možné ovládat aplikaci i na zařízeních s menším rozlišením.

Uživatelské rozhraní umožňuje vytvářet, editovat a případně mazat všechny důležité konfigurační hodnoty, které jsou nutné pro napojení na burzu nebo běh procesorů. Všechny

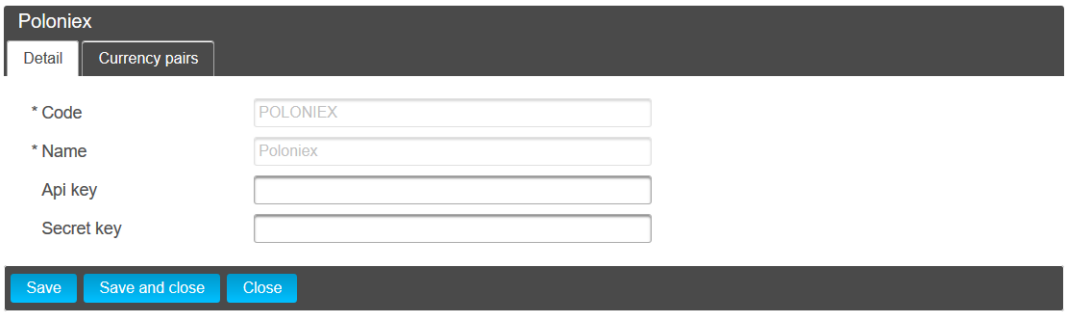
hodnoty jsou přehledně prezentovány v tabulkách nebo formulářích, které jsou doplněny o tlačítka a kontextová menu pro vyvolání akcí nebo zobrazení detailu daného záznamu. Zobrazované tabulky umožňují řazení nebo filtrování položek pro snadné vyhledávání konkrétních záznamů.



Obrázek 5.3: Ukázka seznamu procesorů a vzhledu aplikace.

5.3.1 Konfigurace burzy

S využitím zmíněné knihovny XChange je možné komunikovat s více burzami v daný časový okamžik. Proto je nutná správná konfigurace pro každou z nich. Nejdůležitějšími údaji pro burzu jsou přístupové údaje k danému rozhraní vybrané burzy. Jedná se o tzv. „API key“ a „Secret key“. První zmíněný reprezentuje něco jako identifikátor daného rozhraní, které je přiřazené účtu na burze. Druhý údaj můžeme popsat jako heslo, potřebné pro autentizaci při volání rozhraní. Oba tyto údaje jsou brány jako citlivé, a proto je nelze v aplikaci zobrazit, ale pouze nastavit. Bez těchto údajů je u některých burz možné získávat některá data, ale pro potřebu nákupu a prodeje jsou nutné.

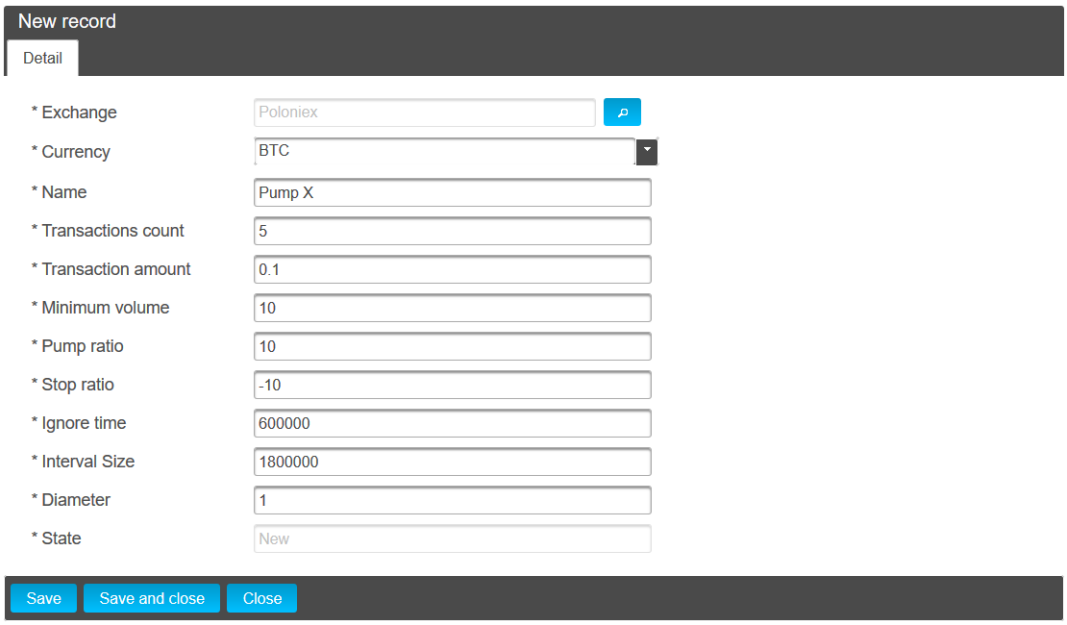


Obrázek 5.4: Detail a nastavení burzy.

Po nastavení autentizačních údajů je nutné získat seznam všech obchodovatelných měnových párů na dané burze, který pak bude sloužit pro tvorbu nových procesorů zaměřených na tuto burzu. Aplikace umožňuje automatické načtení a aktualizaci párů, kterou lze spustit pro danou burzu z kontextového menu. V detailu každé burzy pak najdeme přehledný seznam všech podporovaných kryptoměnových párů.

5.3.2 Konfigurace procesoru

Na rozdíl od jednoduchého nastavení u burz, má konfigurace procesoru více kroků. Nejprve jsou nastavovány specifikace procesoru, zejména burza a výchozí měna. Jeden procesor vždy analyzuje data z jedné burzy a zpracovává množinu párů, u kterých je výchozí směnou měnou („Counter currency“) právě specifikovaná měna (např. BTC). Po výběru burzy a měny je nutné doplnit i ostatní konfigurační hodnoty, které jsou nutné pro běh algoritmu viz. přehled konfiguračních hodnot v kapitole 4.1.



Obrázek 5.5: Detail a nastavení procesoru.

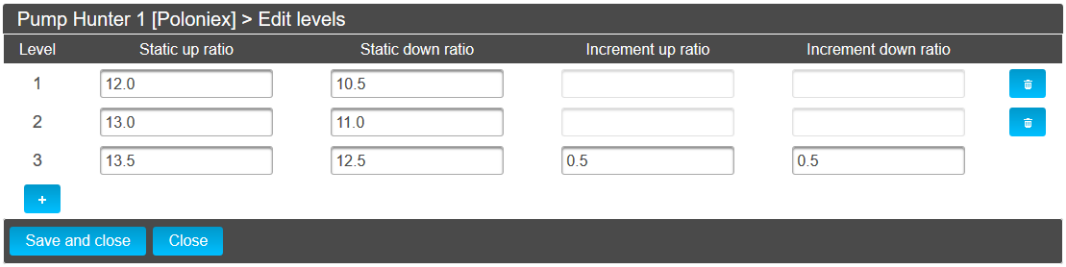
Výběr burz přes tlačítko je omezený pouze na burzy, které mají nějakou množinu párů, se kterými lze obchodovat. Na základě výběru burzy jsou nabídnuty všechny dostupné výchozí měny pro danou burzu. Všechny hodnoty v konfiguraci jsou povinné a nelze je vynechat, neboť aplikace nedovolí uložit neúplný formulář. Po vytvoření nového procesoru je nutné nastavit páry, které bude procesor sledovat a případně s nimi obchodovat. Přiřazovat lze pouze páry, které mají jako výchozí měnu shodnou s měnou specifikovanou při vytváření procesoru. Poslední částí vytváření procesoru je konfigurace levelů, které slouží pro nastavování horních a dolních úrovní otevřených transakcí.

Konfigurace levelů

Různé hodnoty levelů mohou velmi ovlivnit výsledné chování algoritmu a do jisté míry i změnit jeho výchozí chování. Po vytvoření procesoru je automaticky vygenerován první level. Na přiloženém obrázku 5.6 lze vidět omezení na zadávání inkrementů pouze k poslednímu levelu. Poslední level nelze odebrat, protože by to zapříčinilo nefunkčnost algoritmu.

Změnou hodnot levelů lze například upravit algoritmus tak, aby při dosazení určité úrovně rovnou prodal měnu a ukončil tak transakci s daným ziskem. Pro toto chování stačí nastavit dolní hranici nižší než je právě dosažená úroveň. Algoritmus tedy po dosažení úrovně a nastavení dalšího takto upraveného levelu vyhodnotí, že má být proveden prodej.

Další možností jak nastavením levelů měnit funkčnost algoritmu je velké snížení hodnoty dolní hranice. Tím si algoritmus ponechává i měny, které po nakoupení začnou klesat. Tato riskantní varianta počítá s rostoucím trendem kryptoměn a tím tedy předpokládá, že daná měna později opět vzroste a nevznikne prodělek.



Obrázek 5.6: Editace levelů při konfiguraci procesoru.

5.3.3 Správa běhu procesoru

Po potřebném nastavení procesoru je možné spustit běh a začít tak analyzovat data z burzy. Procesor může mít několik různých stavů, které reprezentují jeho aktuální situaci. Spouštění, pozastavování nebo úplné zastavení se provádí pomocí kontextového menu, které lze vyvolat kliknutím pravým tlačítkem myši na daný procesor v listu procesorů. Tato nabídka je znázorněna na obrázku 5.3.

Stavy procesoru

- **Nový, Zastavený (New, Stop):** Tyto dva stavy jsou podobné, oba reprezentují úplně vypnutý procesor. Procesor se stavem „new“ nebyl ještě nikdy zapnutý. Není vytvořeno žádné vlákno, které uchovává instanci procesoru a data jednotlivých sledovaných párů z burzy.
- **Spuštěný/běžící (Run):** Běžící procesor, který načítá a zpracovává data z burzy dle nastavené konfigurace a levelů. Pro spuštěný procesor je také zobrazena záložka v detailu procesoru s aktuálním online stavem hodnot.
- **Pozastavený (Suspend):** - Pozastavený procesor nadále „běží“, ale není prováděna aktualizace hodnot, a ani žádné jejich vyhodnocování. Otevřené transakce také nejsou zpracovávány. Tento stav slouží pro možnost za běhu upravit konfigurační data procesoru, bez nutnosti procesor zastavit a tím ztratit všechna načtená data z okna zpracování.
- **Připravit k zastavení (Prepare stop):** - Jedná se o stav nastavitelný spuštěnému procesoru. Přidává možnost zastavení procesoru bez ponechání otevřených transakcí. Procesor zpracuje otevřené transakce, ale již nové nevytváří. Po zpracování všech otevřených transakcí se sám zastaví.

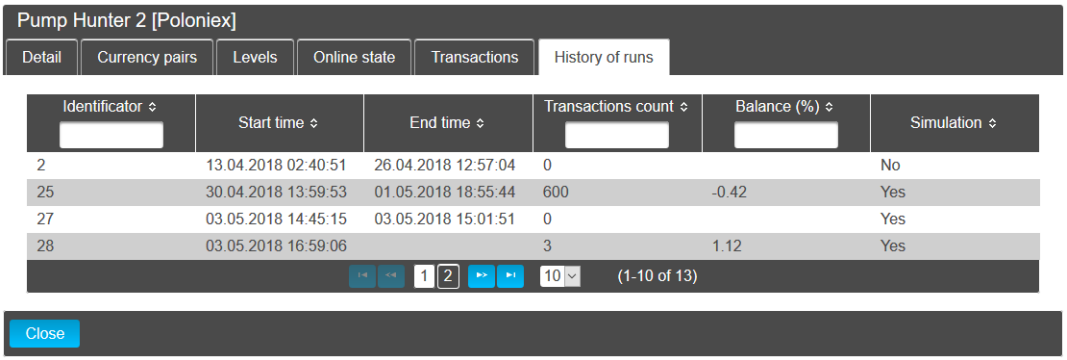
Při spouštění procesoru lze ještě zvolit zda poběží daný procesor v ostrém nebo simulačním módu. Tato volba se nastavuje v možnostech před spuštěním procesoru viz. obrázek 5.7. Rozdílem mezi těmito módy je pouze v tom, že v simulačním módu nejsou prováděny nákupy a prodeje měny. Na rozdíl od ostrého, kde jsou obchody prováděny. Simulační mód

tedy slouží pro možnost otestovat chování nové konfigurace. Díky tomu lze procesor vyzkoušet před ostrým spuštěním bez risku z možné ztráty peněz. Aplikace vždy v kontextovém menu nabízí pouze možnosti, které jsou adekvátní pro aktuální stav procesoru (nelze znovu spustit již spuštěný procesor).

Obrázek 5.7: Spouštění procesoru.

5.3.4 Přehled běhů a transakcí procesoru

Jednotlivé běhy daného procesoru jsou zaznamenávány, aby bylo možné rozlišit například vhodnost konfigurace procesoru pro daný běh. Pro každý záznam je uved počet transakcí, které byly v daném běhu vytvořeny a průměrná bilance všech těchto transakcí. Pokud dojde k zastavení procesoru, tak se nedokončené transakce nepřiřazují k dalšímu běhu. Následující běh je pouze dokončí. Běh je přerušen až zastavením procesoru. Pouhé pozastavení a následné obnovení procesoru neukončuje aktuální běh. V přiložené ukázce 5.8 je zobrazen seznam běhů pro procesor. Pro každý z nich je zobrazen i mód zpracovávání procesoru (simulace nebo ostrý běh). Poslední uvedený běh aktuálně zpracovává data (není ukončený), ale jeho prozatímní průměrná bilance je tady 1,12 % procenta po třech uzavřených transakcích.



Obrázek 5.8: Historie běhů procesoru.

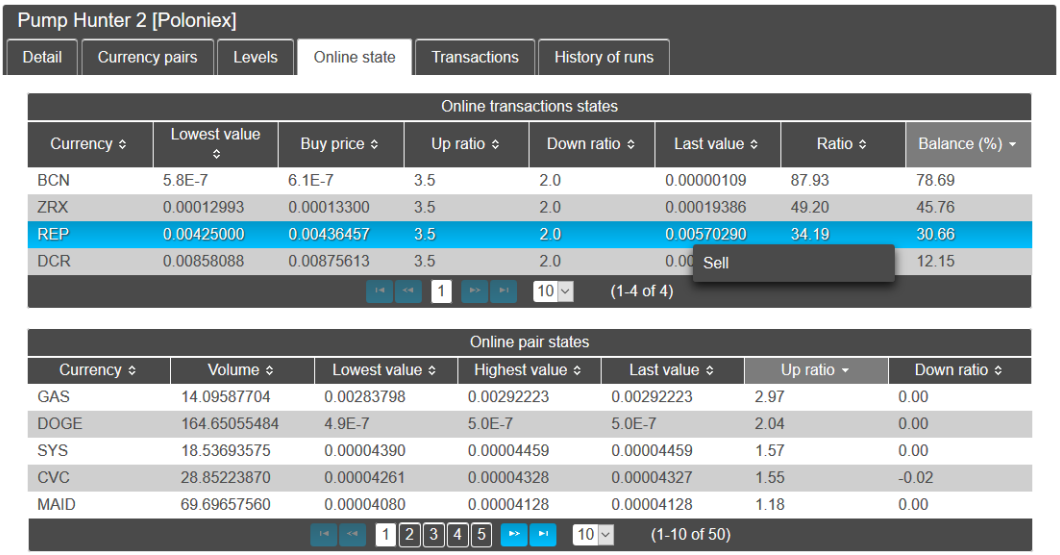
Druhým přehledem je seznam transakcí, které daný procesor zpracovával. Při provedení nákupu je transakce zobrazena v tomto seznamu a po následném prodeji jsou doplněny zbývající hodnoty (čas prodeje, cena při prodeji). Transakce je vždy při vytvoření přiřazena konkrétnímu běhu daného procesoru. Díky tomu lze zobrazit jenom transakce určitého běhu.

5.3.5 Shrnutí a další funkce aplikace

Uživatelské rozhraní tedy umožňuje vkládat do aplikace všechny potřebné konfigurační hodnoty. Jeho hlavní výhodou je intuitivnost a přehlednost. Díky responzivnímu designu lze aplikaci ovládat i mobilním telefonem s dostatečným rozlišení pro plnohodnotné zobrazení tabulek s filtry (testováno na Android 8 Oreo při Full HD rozlišení displeje). U menších rozlišení mají tabulky strukturu s jedním sloupcem, který obsahuje strukturovaně data daného záznamu. I takto zjednodušená tabulka umožňuje vyvolat kontextové menu.

Zobrazování aktuálního stavu a možnost ručního prodeje

Online stav je záložka procesoru, která je zobrazována pro spuštěné procesory a zobrazuje seznam otevřených transakcí a aktuální stav pohybu úrovní pro všechny sledované páry. Tento přehled je automaticky aktualizovaný v krátkých intervalech (každé 3 sekundy). Jsou zde vyobrazeny nejnovější hodnoty, které právě získal a vypočítal procesor. Na základě tohoto přehledu lze upravovat konfiguraci s výhledem většího zisku.



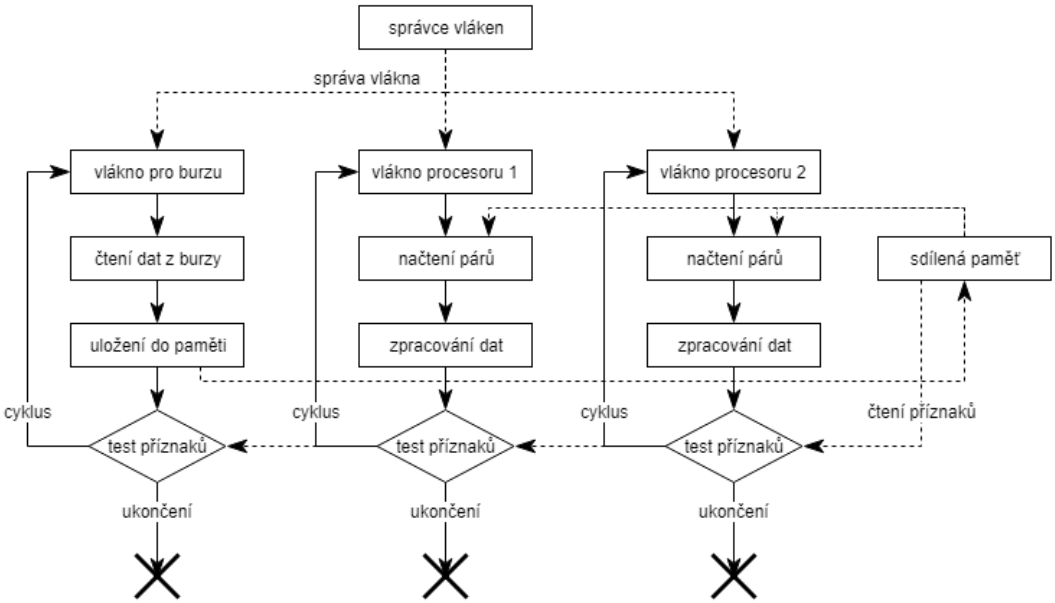
Obrázek 5.9: Přehled otevřených transakcí a aktuálního stavu sledovaných párů.

Na obrázku 5.9 je zobrazena struktura této záložky. V první části se nachází tabulka otevřených transakcí, pro které je dle aktuálních hodnot z burzy vypočítávána aktuální hodnota úrovně a bilance. Bilance znázorňuje procentuální zisk, tedy rozdíl mezi nákupní a aktuální (možnou prodejní) cenou měny. Rozhraní umožňuje ruční příkaz na uzavření transakce a prodej. Tuto akci lze vyvolat v kontextovém menu pro danou otevřenou transakci. Po potvrzení akce je nastaven příznak na prodej dané transakce, která bude v následující zpracování procesorem ukončena a měna prodána s aktuální bilancí. Díky této funkci je možné například při nastaveném stavu, který symbolizuje přípravu k zastavení, ručně prodat nakoupené měny a tím dokončit běh procesoru.

Pod tabulkou otevřených transakcí se nachází tabulka s přehledem párů. Jsou zde zobrazeny všechny zpracovávané páry. Pro každý pár jsou vypsány aktuální hodnoty získané z burzy a vypočítané hodnoty jako je stoupající nebo klesající úroveň.

5.4 Jádro aplikace

Aplikace je implementována více vláknově, proto každý procesor a tzv. „exchange listener“ běží v samostatném vlákně. Pro lepší pochopitelnost je struktura jádra aplikace znázorněna na obrázku 5.10. Veškeré řízení a komunikace těchto vláken probíhá přes sdílenou paměť, která obsahuje struktury uzpůsobené k přístupu z více různých vláken. Obsahují tedy synchronizační principy, které zamezují vícenásobnému zápisu nebo čtení neúplných dat. V této paměti jsou uchovány všechny příznaky pro řízení běhu vláken i data načtená z burz. Pro optimalizaci a odstranění problému s omezením počtu dotazů na burzu byl zaveden princip, kdy v jednom vlákně probíhá načítání dat z burzy a dalších jejich zpracování. V našem případě jedno vlákno vyčte data pro specifikované kryptoměnové páry a následně vlákna procesorů z této paměti přebírají data pro svůj běh. Díky tomuto nezávislému načítání dat nedochází k problémům, vznikajícím při různě dlouhé době zpracovávání aktuálních dat jednotlivými procesory. Aby nedošlo k opakovanému zpracování stejných dat, jsou k nim přidávána časová razítka (tzv. „timestamp“). Procesor si poznamená, jaké data zpracoval a čeká případně na další v pořadí.



Obrázek 5.10: Schéma struktury jádra aplikace.

5.4.1 Spouštění a řízení běhu vláken

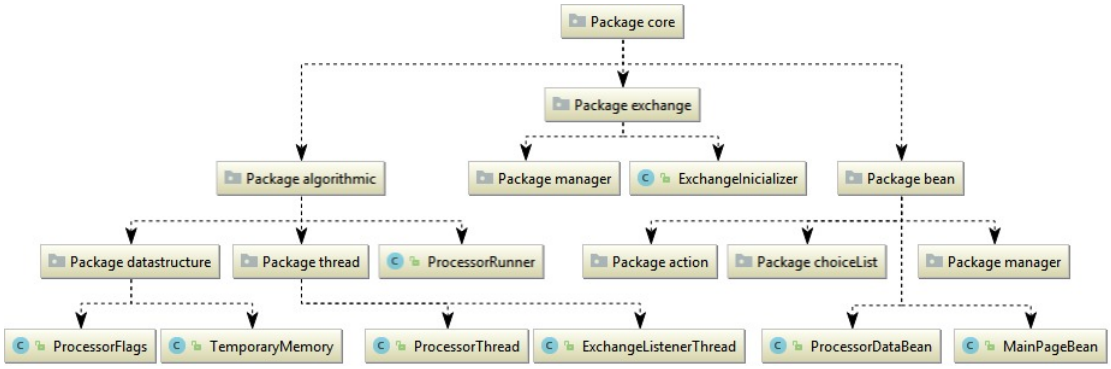
Při spuštění procesoru je spuštěno i odpovídající vlákno pro načítání dat z burzy procesoru. U dalších procesorů pro stejnou burzu se již tedy nové vlákno pro komunikaci s burzou nevytváří, ale pouze je provedena kontrola, zda běžící vlákno načítá páry, které nově spuštěný procesor bude zpracovávat. Z důvodu optimalizace a šetření paměti se do sdílené paměti načítají pouze data pro páry, které nějaký ze spuštěných procesorů zpracovává. Chybějící páry jsou doplněny a po následujícím získávání dat z burzy už budou k dispozici ve sdílené paměti. Při ukončování procesoru nelze odebírat páry, protože není možné poznat, které jsou využívány. Po zastavení posledního procesoru je zastaveno i vlákno pro získávání dat

z burzy. Ve sdílené paměti mimo jiné jsou umístěny i struktury využívané pro zobrazování dat z vláken procesorů do uživatelského rozhraní. Obsahuje také příznaky pro řízení procesorů a transakcí. Pokud je u uživatelského rozhraní zaslán požadavek na změnu stavu procesoru (např. zastavení) nebo na prodej určité transakce je v paměti pro daný procesor respektive transakci nastaven příslušný příznak.

Pokud je nastaven příznak pro pozastavení procesoru, tak probíhá načítání dat z burzy beze změny, pouze procesor data nenačítá a nezpracovává. Při obnovení pozastaveného procesoru neprobíhá znovu načtení sledovaných párů, protože tato změna konfigurace u pozastaveného procesoru není přípustná.

5.4.2 Diagram balíků a tříd jádra aplikace

Diagram na přiloženém obrázku 5.11 zobrazuje strukturu balíků reprezentujících jádro aplikace. Jedná se o nejdůležitější řídící části implementované aplikace, které spojují ostatní části a utvářejí tak funkční celek.



Obrázek 5.11: Diagram struktury balíků a tříd jádra aplikace.

V přiloženém popisu jsou mimo jednotlivé obsažené balíky popsány také významné třídy, které se v jádru nacházejí a mají velký vliv na funkčnost celé aplikace a navrženého algoritmu. Popis jednotlivých částí:

- **Package algorithmic:** Tento balík obsahuje třídy pro správu běhu procesorů, samotné implementace vláken a potřebné struktury pro řízení jejich běhu.
  - **Package datastructure:** Potřebné datové struktury pro běh a řízení vláken. Implementace struktur, které jsou bezpečné při více vláknovém zápisu a čtení dat (synchronizační mechanismy).
    - Class ProcessorFlags: Uchování všech příznaků pro řízení procesorů (signály k pozastavení, prodeji, aj.).
    - Class TemporaryMemory: Třída reprezentující paměť pro uchovávání načtených dat z burz, určených ke zpracovávání procesory.
  - **Package thread:** Jednotlivé implementace spustitelných vláken.
    - Class ProcessorThread: Třída implementující vlákno procesoru. Je zde implementován navržený algoritmus.
    - Class ExchangeListenerThread: Třída implementující vlákno pro načítání dat z dané burzy. Načtená data vkládá do sdílené paměti.

  - Class ProcessorRunner: V této třídě probíhá inicializace a řízení běhu jednotlivých vláken.

- **Package exchange:** V tomto balíku se nacházejí třídy pro napojení a komunikaci s burzami.
  - Class ExchangeInicializer: Provádí inicializaci napojení na burzy. Vkládá potřebné zabezpečovací a konfigurační údaje, které jsou nutné pro vytvoření spojení s burzou.
  - **Package manager:** Zde jsou obsaženy tzv. manažeři burz. Jedná se o specifičtější implementace jednotlivých služeb dané burzy (nějaké burzy mají odlišný přístup pro vytváření obchodů).

- **Package bean:** Poslední balík v jádru aplikace obsahuje převážně tzv. „beany“ pro řízení uživatelského rozhraní a uchovávání sdílených dat.
  - **Package action:** Balík obsahující třídy, které provádějí akce vyvolané z uživatelského rozhraní (změna stavu procesoru, nastavení autentizačních údajů pro burzu, atd.).
  - Class ProcessorDataBean: Uchovává všechna sdílená data mezi vlákny, příznaky pro řízení běhu vláken a struktury s daty pro online zobrazování párů a transakcí v uživatelském rozhraní.
  - Class MainPageBean: Hlavní „beana“ pro funkci uživatelského rozraní. Umožňuje volání akcí a správné zobrazování nebo aktualizování jednotlivých komponent.
  - **Package choiceList:** Tento balík obsahuje tzv. „výběrové listy“. Jedná se o třídy, které získávají data z databáze a umožňují jejich výběr (např.: výběr kryptoměnových párů k procesoru).
  - **Package manager:** Tito manažeři provádějí ukládání vybraných dat z výběrových listů.

5.4.3 Implementace komunikace s burzou

Načítání dat z dané burzy je implementováno jako vlákno, které obsahuje nekonečný cyklus. Tento cyklus lze přerušit pouze nastavením příznaku ve sdílené paměti pro zastavení načítání dat z burzy. Dalším krokem je získání dat z burzy pro potřebné měnové páry. Výsledný list je opatřen časovým razítkem. Takto připravená a orazítkovaná data jsou vložena do sdílené paměti a tím zpřístupněna procesorům.

Mezi jednotlivými načítáními je nutné vložit interval, po který bude vlákno čekat. Tato úprava je nutná z důvodu aktuálnosti dat, poskytovaných rozhraním burzy. Obnovovací frekvence u burz bývá jedna sekunda. Vlákno je tedy řízeno tak, aby byla dodržena nějaká minimální perioda načítání. Pokud dojde k nějakému zpoždění, je čekání vynecháno, aby se interval ještě více neprodlužoval.

5.4.4 Implementace procesoru

Vlákno procesoru běží v nekonečném cyklu. Před spuštěním cyklu probíhá inicializace struktur pro zobrazování aktuálních dat v uživatelském rozhraní (záložka procesoru: „Online state“). Stejně jako u vlákna pro načítání dat z burzy je i vlákno procesoru řízeno příznaky ze sdílené paměti. Pokud není procesoru nastaven příznak na zastavení, tak se pokusí získat načtená data z paměti. Získaná data projde a všechny páry, které odpovídají těm, co má přiřazeny, zpracuje. Toto zpracování odpovídá navrženému algoritmu z kapitoly 4.1. Při zpracovávání dat z burzy jsou vypočítané úrovně ukládány do připravených struktur pro zobrazení v uživatelském rozhraní.

V případě ostrého běhu je pro nákup a prodej měn využíván manažer příslušné burzy, který obsahuje potřebné metody. Využití implementovaného manažera je nutné, neboli některé kryptoměnové burzy nepodporují „Market“ transakce, tak probíhá konverze na limitní objednávky.

Pokud dojde k pozastavení procesoru, tak vlákno dál běží, ale díky nastavenému příznaku neprovádí žádné zpracování dat. Po znovu obnovení se provede aktualizace hodnot procesoru a levelů podle aktuálního stavu v databázi. V případě změny levelů se úpravy nepromítnou na již aplikované úrovně v otevřených transakcích.

Kapitola 6

Testování a správa běhu aplikace

Testování aplikace a běhu procesorů probíhalo na připraveném serveru, který běžel nepřetržitě. Nejdelší doba běhu procesorů s různými konfiguracemi byla přes 2 týdny nepřetržitého vyhodnocování dat z burzy. Těmto delším testovacím intervalům předcházely krátké testy (většinou jednodenní), které postupně odhalovaly drobné chyby. Postupným odladěním všech těchto chyb, bylo možné dlouhodobější testování. Uživatelské rozhraní bylo podrobeno převážně jenom vývojářským testům. Na jeho testování se podílelo i několik uživatelů, kteří byli seznámeni s danou problematikou.

Hlavní burzy, které byly využívány pro testování aplikace byly Poloniex a Bitstamp, protože umožňují obchodovat s velkým počtem kryptoměn. Nejčastěji byla volena jako výchozí kryptoměna bitcoin, protože například burza Poloniex nabízí k obchodování s bitcoinem přes 70 různých altcoinů. Algoritmus má větší potenciál vydělat, pokud sleduje víc párů. Z toho plyne, že je lepší se napojovat na větší burzy s větším počtem kryptoměnových párů.

6.1 Struktura výpisů

V uživatelském rozhraní je možné zobrazit aktuální stav, otevřené nebo již ukončené prodané transakce, ale není zde podrobný výpis jednotlivých kroků různých částí aplikace. Pro ladění konfigurací a nalézání případných chyb, je tato funkcionalita naprosto klíčová a bez ní by byly tyto úkony velice obtížné až nemožné. Postupně bude představen princip logování a obsah jednotlivých souborů s ukázkou jejich struktury. Tyto soubory můžeme rozdělit do dvou skupin.

První skupinou jsou výpisy, které uchovávají záznamy o průběhu spouštění nových vláken, reporty o vyčítání dat z burzy nebo seznam provedených obchodů pro danou burzu. Druhá skupiny představuje soubory, které jsou vytvořeny pro každý běžící procesor samostatně a uchovávají záznamy o běhu procesoru, transakcích nebo úrovních párů.

Pro jednotlivé tyto soubory je níže v příkladech ukázána jejich struktura a popsán obsah.

6.1.1 Obecné výpisy aplikace a burz

V této části se nachází popis výpisů, které jsou společné pro celou aplikaci nebo odpovídají napojením na jednotlivé burzy.

Spouštění procesorů a čtecích vláken burz V tomto výpisu se zaznamenává vytváření nových vláken, jejich spouštění a následné zastavování. V přiloženém příkladu 6.1 je nastíněno spuštění prvního procesoru pro danou burzu. Nejprve je vytvořeno vlákno pro

čtení dat z burzy. Provede se načtení párů a dojde k jeho spuštění. Následně je vytvořen a spuštěn procesor. Po ukončení činnosti procesoru je zastaveno i vlákno komunikující s burzou.

```
2018-04-30 10:56:55 - Creating exchange listener [POLONIEX].
2018-04-30 10:56:55 - Loaded [67] currency pairs into exchange lister. Pairs: [BCH/BTC,
    ...]
2018-04-30 10:56:57 - Exchange listener [POLONIEX] was started.
2018-04-30 10:56:57 - Creating processor [Pump Hunter 1].
2018-04-30 10:56:57 - Processor [Pump Hunter 1] was started.
2018-04-30 11:02:31 - Set stop flag for processor [Pump Hunter 1].
2018-04-30 11:02:32 - No more running processors for exchange listener [POLONIEX]. Set stop
```

Výpis 6.1: Vytváření a spouštění jednotlivých vláken (processorRunner.log)

Čtení dat z burzy Kontrolní výpis 6.2 po načtení dat. V případě výpadku spojení jsou do tohoto logu zaznamenány veškeré informace. Díky době trvání získávání dat je možné ověřit dostatečnou rychlost připojení.

```
2018-04-30 10:56:57 - Exchange listener [POLONIEX] start.
2018-04-30 10:56:58 - Load data [1525078618012]: 532 ms
2018-04-30 10:56:58 - Load data [1525078618835]: 154 ms
2018-04-30 10:56:58 - Load data [1525078618835]: 245 ms
...
2018-04-30 11:50:12 - Flag to stop is set. Break cycle.
```

Výpis 6.2: Čtení dat z burzy (exchangeListener_poloniex.log)

Souhrn provedených obchodů na burze Ve výpisu 6.3 je seznam všech nákupů a prodejů měn na dané burze, které jsou prováděny manažerem burzy. Jedná se o všechny obchody nad danou burzou jednotlivými procesory.

```
2018-04-30 12:46:55 - Processing market buy transaction for pair [BCH/BTC]
2018-04-30 12:46:55 - Placed order #45287242
2018-04-30 12:46:58 - Transaction finished with price: 0.16984948
2018-04-30 13:36:34 - Processing market sell transaction for pair [BCH/BTC]
2018-04-30 13:36:34 - Placed order #12645041
2018-04-30 13:36:39 - Transaction finished with price: 0.19027541
```

Výpis 6.3: Provedené obchody (poloniexExchangeManager.log)

6.1.2 Výpisy procesorů

Následující výpisy jsou pro každý procesor samostatné a slouží pro záznam všech důležitých akcí procesoru.

Základní log procesoru Tento výpis 6.4 obsahuje report o zpracovávaných datech z burzy, době zpracování konkrétních dat a dalších událostech (např.: začátek ignorování páru, obnovení páru, vyhodnocení příkazů, atd.). Pokud v paměti je stále již zpracovaný pár, tak procesor čeká na další časové razítko.

```
2018-04-30 10:57:06 - Process exchange data with timestamp [1525078626328].
2018-04-30 10:57:06 - Data [1525078626328] processed: 6 ms
2018-04-30 10:57:06 - START ignoring pair [XPM/BTC] till date [2018-04-30 11:27:06]
2018-04-30 10:57:07 - Timestamp [1525078626328] was already processed. Waiting
```

Výpis 6.4: Ukázka základního logu (PROCESSOR_<NAME>.log)

Průběh úrovně otevřených transakcí Pro pozdější analýzu pohybu úrovně hodnoty měny při otevřené transakci, je vytvořen tento report 6.5, který zaznamenává všechny potřebné údaje o aktuálním stavu.

```
2018-04-30 14:06:58 - RATIO [XPM/BTC] [low: 0.00022662, last: 0.00025862]:  14.120552
2018-04-30 14:06:59 - RATIO [XPM/BTC] [low: 0.00022662, last: 0.00025887]:  14.230873
...
2018-04-30 14:10:00 - RATIO [XPM/BTC] [low: 0.00022662, last: 0.00027350]:  20.686615
2018-04-30 14:10:02 - RATIO [XPM/BTC] [low: 0.00022662, last: 0.00027114]:  19.645218
```

Výpis 6.5: Přehled úrovní (PROCESSOR_<NAME>_ratio.log)

Správa transakcí Výpis 6.6 je u procesoru ten nejdůležitější. Obsahuje události dosažení úrovně pro nákupy či prodej. Nastavování nových úrovní dle dostupných levelů a další operace s transakcemi.

```
2018-04-30 14:06:58 - LOAD UNFINISHED Transaction [4]. UpRatio: 12.5, StopRatio: 11.5
2018-04-30 14:06:58 - ACHIEVED UP RATIO for pair [XPM/BTC] ratio [14.120552].
2018-04-30 14:06:58 - Level [2] applied. UpRatio: 12.5, DownRatio: 11.5
2018-04-30 14:07:00 - Level [2] increment applied. UpRatio: 14.5, DownRatio: 13.5
...
2018-04-30 14:47:19 - ACHIEVED DOWN RATIO for pair [XPM/BTC] ratio [13.480552].
2018-04-30 14:47:19 - SELL ORDER for pair [XPM/BTC].
2018-04-30 14:47:19 - SELL COMPLETE with price [0.00028455] for pair [XPM/BTC].
```

Výpis 6.6: Přehled operací s transakcemi (PROCESSOR_<NAME>_transaction.log)

6.2 Testování algoritmu

Z různých testovacích konfigurací se zaměříme na tři. První testovaná konfigurace je podobná nastavení z příkladu v kapitole 4.2, jenom délka okna byla 4 hodiny. Toto nastavení vyhledává krátkodobé rychlé vzrůsty ceny měn. Z analýzy dat po testovacím běhu vyplývá, že průměrná doba trvání transakce byla v řádech minut. Proto procesor za svůj běh zpracoval velké množství transakcí. Tato konfigurace nebyla v praxi moc efektivní, protože i když procesor zaznamenal několik transakcí s vyšším výdělkem, tak stále provedl i dost ztrátových transakcí. Nejčastějším důvodem ztráty bylo chování, kdy úroveň vzrostla, byl proveden nákup a následně opět začala klesat. Toto kolísavé chování se dle analýzy vyskytovalo častěji u několika párů. Možnou optimalizací by bylo tyto páry vyřadit ze zpracovávání.

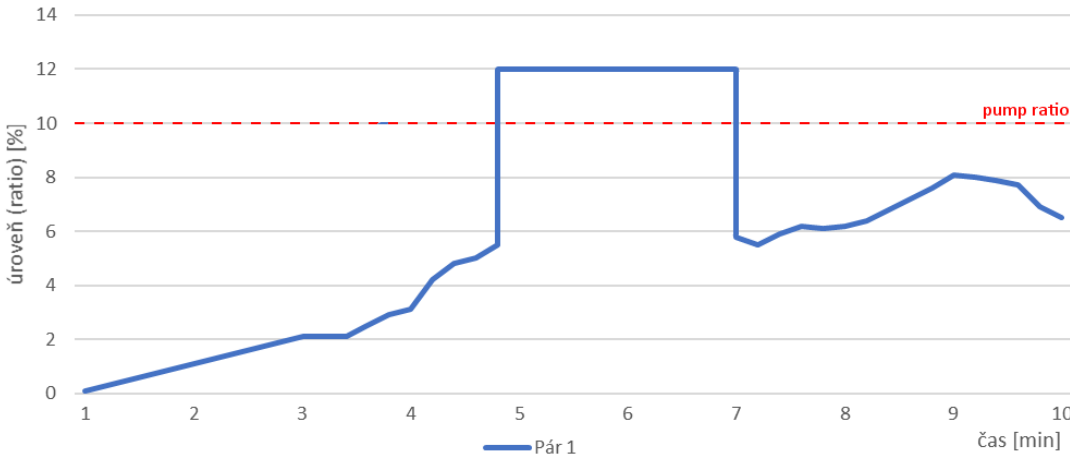
Druhá testovaná konfigurace byla založena na myšlence prodeje se ziskem nebo případném dlouhodobějším vlastnění dané měny. Algoritmus byl nastavený, takže hodnota úrovně pro nákup byla 5 % a při následném dosažení hranice 10 % byla teprve zavedena dolní hranice pro prodej. To znamenalo, že pokud měna klesla, nebyla hned prodána. Z toho vyplývalo riziko z možné velké ztráty. Dalším problémem této konfigurace byl velký počet souběžně otevřených transakcí. Výsledná bilance zůstala po více než stovce provedených transakcí kladná a to pouze díky tehdejšímu rostoucím trendu hodnoty bitcoinu a většiny ostatních kryptoměn.

Poslední testovaná konfigurace byla zaměřena na vyhledávání rychle rostoucích měn. Její hlavní odlišnost od předešlých konfigurací byla v délce okna pro uchovávání hodnot. Okno zde bylo pouhých 15 minut. Díky tomuto nastavení byly nakupovány pouze měny, které zaznamenaly rychlý nárůst své hodnoty. Celkový počet transakcí byl zde proto nejmenší. Úspěšnost této konfigurace byla díky tomu v porovnání s předešlými vyšší.

6.3 Poznatky z testování a jejich řešení

Při testovacím běhu procesorů se objevilo několik problémů. Ve výše zmíněných souborech s různými výpisy bylo možné pozorovat u transakcí, které skončili se ztrátou, průběh úrovně daného páru a na základě těchto dat provádět korekce konfigurace. Až při testování byl odhalen problém s velice rychlým zakolísáním měny (např.: jedna podhodnocená nabídka), který způsoboval krátký rychlý nárůst úrovně. Díky tomu byla provedena operace, která v mnohých případech znamenala ztrátu. Například pokud hodnota na několik sekund až minut prudce vyskočila na úroveň pro nákup a vzápětí opět spadla o několik procent dolů. Tento problém byl částečně řešen zavedením průměrování hodnot, avšak pokud hodnota vyskočila například z 5 procent na 12 procent, tam se udržela delší dobu a následně spadla zpět na 5 procent, znamenala se opět ztrátová transakce.

Do procesoru byla proto zavedena funkce, která si při dosažení úrovně pro nákup měny zaznamená aktuální hodnotu a hned neprovede nákup. Pokud v následujících datech z burzy přijde další vyšší hodnota, provede se nákup. V případě, že přijde nižší hodnota, než je aktuálně uložená, tak se vše smaže a čeká se opět na první z těchto dvou úrovní. Tato experimentální úprava zlepšila bilanci všech procesorů, protože došlo k omezení reakce na tyto rychlé prudké výkyvy. Na grafu 6.1 je znázorněna zmiňovaná situace. Pokud je využita funkce pro čekání na další stoupající hodnotu, neprovede se nákup měny a nedojde k vytvoření ztrátové transakce.



Obrázek 6.1: Ukázka možného scénáře s prudkým nárustem a pádem hodnoty měny.

Dalším problém, který částečně plyne i z předchozího problému, je situace, kdy úroveň prudce překročí hranici pro nákup. To způsobí, že se rychle aplikují levely a nastaví se úrovně, které jsou už například riskantnější. Pak při případném poklesu dojde k větším ztrátám. Pokusem o řešení tohoto problému bylo zavedení tzv. horní hranice pro nákup. Tedy provedení nákupu bylo omezeno například na interval od 10 % do 12 %. Po testování tohoto omezení došlo k nepříznivému výsledku, kdy algoritmus nenakupoval i měny, které skočili např. na 13 % a vzápětí dále rostly. Z toho důvodu byla tato funkcionalita ve výsledném řešení vynechána.

6.4 Neočekávané ukončení aplikace

V případě náhlého ukončení aplikace, způsobeného například výpadkem elektrické energie, lze využít možnosti automatického spouštění aplikačního serveru. To je provedeno pomocí definice služeb v operačním systému. Při náběhu pak aplikace zkontroluje, zda nějaký procesor nezůstal spuštěn. Pokud se takový procesor objeví, provedeme jeho znovu spuštění. Procesor obnoví transakce a bude pokračovat ve zpracovávání.

Kapitola 7

Závěr

Kryptoměny v dnešní době prochází velkým růstem popularity a jejich množství je stále zvyšováno. V této práci jsou představeny nejrozšířenější kryptoměny a burzy, které slouží k obchodování s nimi. U vybraných burz bylo analyzováno jejich rozhraní a možnost vytvoření algoritmů, které budou zpracovávat data o kryptoměnách, získaná z těchto burz. Po analýze API rozhraní zkoumaných burz se došlo k závěru, že jednotlivé burzy mají strukturu volání a formát jednotlivých požadavků a odpovědí odlišný. Z toho vyplynula nutnost vybrat pouze jednu burzu pro implementaci algoritmů. Toto by řešení velmi omezilo, a proto bylo nutné najít efektivnější možnost.

Řešením tohoto problému je využití knihovny XChange, která zobecní a zapouzdří komunikaci s burzami. Díky tomu mohl být algoritmus pro automatické obchodování implementován pomocí obecných struktur a může být tak využíván pro různé burzy. Algoritmus představený a implementovaný v této práci představuje možnost, jak plošně a efektivně monitorovat velké množství kryptoměnových párů na několika různých burzách. Na základě vyhodnocování těchto dat a s vhodnou konfigurací, je algoritmus schopný provádět obchody s cílem generovat zisk. Při dlouhodobějším testování bylo vyzkoušeno hned několik možných konfigurací algoritmu. Každá z nich měla různou bilanci, na kterou měl velký vliv aktuální stav kryptoměn. V období, kdy cena kryptoměn rostla, byla bilance lepší než v obdobích poklesu.

Aplikace, která byla implementována v této práci, podporuje snadné rozšiřování a vylepšování. Její struktura umožňuje zařazení i dalších typů algoritmů, které budou reprezentovány novými druhy procesorů (například odlišné zpracovávání dat měn). Přidanou hodnotou tedy není jenom popsaný algoritmus, ale i samotná aplikace.

Navržený a implementovaný algoritmus je založený pouze na procentuálním poklesu a růstu ceny. Pro zvýšení efektivity lze algoritmus ještě dále rozšiřovat a zařadit do něho například určitá vylepšení, která by mohla být vypracovávána v možných navazujících pracích. Mezi tyto rozšíření by například mohlo patřit zavedení nějaké formy strojového učení („machine learning“), které by podle určitých pravidel a historického chování ovlivňovalo konfiguraci nebo chování algoritmu pro konkrétní pár. Algoritmus by bylo také možné doplnit o další prvky vycházející z technické analýzy [7], jako je nastavení hlídání významných bodů (support, rezistence, signály, indikátory atd.) a případné reakce na ně. Očekávaným přínosem pak může být zvýšení efektivity obchodování pomocí tohoto nástroje, např. zvýšení zisků, snížení ztrát nebo řízení rizik.

Literatura

[1] NARAYANAN, Arvind. *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton: Princeton University Press, 2016. ISBN 978-0-691-17169-2.

[2] ANTONOPOULOS, Andreas M. *Mastering bitcoin: Unlocking Digital Cryptocurrencies*. Sebastopol CA: O'Reilly, 2015. ISBN 978-1449374044.

[3] *Bitstamp* [online]. Londýn: Bitstamp Ltd., 2011 [cit. 2018-03-16]. Dostupné z: www.bitstamp.net

[4] *Bitfinex* [online]. Hong Kong: IFinex, 2013 [cit. 2018-03-16]. Dostupné z: www.bitfinex.com

[5] *Poloniex* [online]. Washington: Poloniex Inc., 2014 [cit. 2018-03-16]. Dostupné z: www.poloniex.com

[6] BARTOLOME, Mar. *A first attempt at Bitcoin trading algorithms* [online]. 16. září 2017 [cit. 2018-05-15]. Dostupné z: https://dev.to/marbru/a-first-attempt-at-bitcoin-trading-algorithms

[7] KIRKPATRICK, Charles D. a Julie R. DAHLQUIST. *Technical analysis: the complete resource for financial market technicians*. Third edition. Old Tappan, New Jersey: Pearson Education, 2016. ISBN 978-0134137049.

Příloha A

Obsah přiloženého paměťového média

- **Adresář** `aplikace` obsahuje již zkompilovanou aplikaci do archivu `WAR` a soubor pro inicializaci databáze, která je potřebná pro běh aplikace.
- **Adresář** `latex_src` obsahuje zdrojové soubory textové èásti pro LaTeX.
- **Soubor** `DP_Zdenek_Krestan_2018.pdf` je elektronická verze textové části diplomové práce.
- **Adresář** `source` obsahuje zdrojové soubory aplikace.
- **Soubor** `README.txt` obsahuje stručný popis obsahu a návod na případné spuštění aplikace.

Příloha B

Návod na spuštění aplikace

Předpokladem pro spuštění aplikace je dostatečně výkonný počítač s potřebnými technologiemi (Java 1.8, Microsoft SQL server 2014, aplikační server WildFly 10.1.0). Na těchto verzích byly provedeny veškeré testy. Všechny využívané technologie jsou volně dostupné.

1. Vytvoření a inicializace databáze určené pro aplikaci. Výhozím bodem je správně nakonfigurovaný DB server s otevřeným TCP/IP portem pro komunikaci.
   (a) Na SQL serveru je nutné vytvořit novou databázi.
   (b) Pomocí přiloženého inicializačního skriptu vytvořit struktury a doplnit data do této databáze.
2. Vložení aplikace na aplikační server a jeho konfigurace.
   (a) S využitím zdrojových souborů lze pomocí Maven build sestavit WAR balík aplikace. Tato metoda však může vyžadovat instalaci a stažení dalších knihoven. Proto je již vytvořené WAR aplikace přiloženo na paměťovém médiu.
   (b) Připravený balík aplikace je nutné vložit do složky „..\standalone\deployments“ v adresáři aplikačního serveru WildFly.
   (c) Dalším krokem je konfigurace potřebných atributů serveru. Pro konfiguraci slouží soubor „..\standalone\configuration\standalone.xml“.
      i. Prvním krokem konfigurace je nastavení připojení k vytvořené databázi na SQL serveru. Toto nastavení se provádí za pomoci XML elementu „datasources“.
      ii. V tomto souboru je možné nastavit i příslušné porty na kterých server a jednotlivé jeho části budou dostupné. Tato konfigurace se provádí v elementu „socket-binding-group“.
3. Spuštění a správa běhu aplikace.
   (a) Spustit server lze dvěma způsoby:
      i. Pro testování nebo krátkodobý běh lze spustit pomocí BAT souboru z příkazové řádky (..\bin\standalone.bat).
      ii. Druhou možností pro dlouhodobý běh je spouštění pomocí služeb OS.
   (b) Po náběhu serveru je aplikace dostupná přes webový prohlížeč na nastavené adrese (výchozí adresa: „http://localhost:8080/XTrader/main.xhtml“).