VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

SBĚR DAT ZE SOCIÁLNÍ SÍTĚ TWITTER
DATA COLLECTION FROM TWITTER

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Juraj Kmet**
AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. Dan Komosný, Ph.D.**
SUPERVISOR

BRNO 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Teleinformatika**
Ústav telekomunikací

Student: Juraj Kmeť ***ID:*** 173675
Ročník: 3 ***Akademický rok:*** 2016/17

NÁZEV TÉMATU:

Sběr dat ze sociální sítě Twitter

POKYNY PRO VYPRACOVÁNÍ:

Vytvořte systém, který bude sbírat data ze sociální sítě Twitter. Aplikaci zhotovte v programovacím jazyce Python. Vytvořenou aplikaci spusťte na zvolených serverech sítě PlanetLab s OS Linux. Každý server sítě PlanetLab bude získávat data z jiné oblasti sítě Twitter. Získaná data ze serverů zasílejte na datové úložiště. Analyzujte rychlost sběru dat.

DOPORUČENÁ LITERATURA:

[1] PILGRIM, Mark. Ponořme se do Python(u) 3: Dive into Python 3. Praha: CZ.NIC, c2010. CZ.NIC. ISBN 978-80-904248-2-1.

[2] RUSSELL, Matthew A. 21 recipes for mining Twitter. Sebastopol, CA: O'Reilly Media, c2011. ISBN 14-49--0316-1.

Termín zadání: 1.2.2017 ***Termín odevzdání:*** 8.6.2017

Vedoucí práce: doc. Ing. Dan Komosný, Ph.D.
Konzultant:

doc. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Bakalárska práca sa zaoberá vytvorením aplikácie na zber dát zo sociálnej siete Twitter. Dáta sú zbierané v reálnom čase s rôznou dĺžkou trvania zberu.
Teoretická časť hovorí o sociálnej sieti Twitter, približuje možnosti ako sa s ňou dá pracovať a to nie len z pohľadu bežného užívateľa, ale aj pre získavanie dát. Bakalárska práca identifikuje obmedzenia, ktoré je potrebné rešpektovať pri vytváraní Twitter aplikácií.
Ďalej je popísaná sieť PlanetLab, ktorá je v dnešnej dobe rozšírená najmä u sieťových výskumníkov a vývojárov sieťových aplikácií. V druhej kapitole je priblížená história siete PlanetLab a taktiež jej odlišnosti od iných vývojárskych sietí.
Praktická časť obsahuje jednotlivé kroky pre návrh aplikácie v programovacom jazyku Python. Priložený je tiež postup distribúcie vytvorenej aplikácie na vybrané zariadenia siete PlanetLab.
V závere sú postupne analyzované zbery dát a maximálne dosiahnuté rýchlosti zbierania príspevkov vytvoreného systému.

KĽÚČOVÉ SLOVÁ

PlanetLab, Twitter, Python, Twitter API, Streaming API, Leaflet

ABSTRACT

The bachelor thesis deals with creating application for data gathering from social network Twitter. Data is gathered in real time with variable length of gathering.
Theoretical part describes social network Twitter as a client but also as a tool for data gathering. The bachelor thesis identifies limits which need to be respected during the creation of Twitter applications.
Another topic of the thesis is PlanetLab network, which is well known mainly by the network researchers and developers of network applications. History of PlanetLab is captured within the second chapter and the difference between PlanetLab and other research networks.
Practical part contains guide for application development in programming language Python. Process of the application disctribution to the PlanetLab nodes is enclosed as well. Last chapter analyses data collection and maximum speed of data gathering in the created system.

KEYWORDS

PlanetLab, Twitter, Python, MySQL, Twitter API, Streaming API, Leaflet

KMEŤ, Juraj *Sběr dat ze sociální sítě Twitter*: bakalárska práca. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2017. 47 s. Vedúci práce bol doc. Ing. Dan Komosný, Ph.D.

VYHLÁSENIE

Vyhlasujem, že som svoju bakalársku prácu na tému „Sběr dat ze sociální sítě Twitter“ vypracoval(a) samostatne pod vedením vedúceho bakalárskej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor(ka) uvedenej bakalárskej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil(a) autorské práva tretích osôb, najmä som nezasiahol(-la) nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý(-á) následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka Českej republiky č. 40/2009 Sb.

Brno ....

podpis autora(-ky)

POĎAKOVANIE

Rád by som poďakoval vedúcemu bakalárskej práce pánovi doc. Ing. Danovi Komosnému Ph.D., za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

Brno . . . .

podpis autora(-ky)

OBSAH

ZOZNAM OBRÁZKOV

ZOZNAM TABULIEK

ZOZNAM VÝPISOV

ÚVOD

Sociálne siete sú v dnešnej dobe veľmi obľúbené. Pomáhajú tomu aj predmety, ktoré dennodenne používame ako stolové počitače, notebooky, tablety či mobilné telefóny. Ľudia sú schopní presedieť pri nich aj niekoľko hodín denne. Vypĺňame tak s nimi voľný čas na ceste do školy, či práce, pri obede a často zaspávame s pohľadom na svietiacu obrazovku. Používajú ich rôzne vekové kategórie, bežní ľudia aj slávne osobnosti, či politici. Pomáhajú nám držať krok s dobou a taktiež sa často krát stáva, že príspevky, ktoré uverejňujú neznámi ľudia sa nás natoľko dotýkajú, že sa dokážeme vcítiť do kože danej osoby a chceme vyjadriť svoje sympatie, alebo sa len podeliť o svoju skúsenosť, či názor. Sociálne siete sa stali veľmi dobrým zdrojom recenzií v mnohých oblastiach. Sú nápomocné pri veľkých rozhodnutiach ako je nová lokalita na bývanie, či výber dovolenkovej destinácie, ale aj s drobnosťami ako je výber klubu, či reštaurácie na večer. Podniky, hotely, agentúry a organizátori akcií, si vytvárajú profily a môžu tak podávať aktuálne informácie všetkým, ktorých to zaujíma. Rôzne sociálne siete ponúkajú možnosti ako užívatelia môžu medzi sebou komunikovať. V dnešnej dobe už nie sú neobvyklé ani video hovory v rámci sociálnej siete. Medzi najviac využívaným stále zostáva zasielanie textových správ, uverejňovanie príspevkov a ich komentovanie.

Okrem komunikácie medzi užívateľmi, príspevky sú často zdrojom aktuálnych správ zo sveta. Videá, fotografie, názory špecialistov, to všetko nás nabáda k tomu, aby sme si na veci spravili vlastný názor. Musíme však vedieť v tomto množstve informácií správne vyberať a nepodľahnúť zavádzajúcim, či podvodným správam.

Táto práca sa zameriava na sociálnu sieť Twitter. V prvej kapitole sa zoznámime so základnou terminológiou a možnosťami interakcie medzi užívateľmi. Taktiež si priblížime ako sa webové stránky a aplikácie dajú prepojiť s Twitter účtami. Rozoberieme si aké možnosti má vývojár sieťových aplikácií, čo je to Streaming a REST API a aké majú výhody a nevýhody pri získavaní dát.

Ďalšia kapitola sa zaoberá platformou PlanetLab. Dozvieme sa o histórií, motivácií a cieli, ktorý chcú samotní vývojári tejto platformy dosiahnuť. Pri návrhu PlanetLabu museli mať vývojári predstavu ako sa bude táto sieť rozširovať. To obsahuje podkapitola požiadavky a okrem iného sú tu rozobraté hardwarové a softwarové nároky. Ďalej sú popísané možnosti pre výskumníkov a vývojárov sieťových služieb a aplikácií. V závere tejto kapitoly sú zhrnuté súčasné projekty a základné pojmy, ktoré je nutné ovládať pri práci s PlanetLabom.

Nasledujúca kapitola sa zaoberá vytváraním aplikácie na zber dát. Je tu popísané ako sa bude aplikácia a užívateľ identifikovať a aké možnosti má užívateľ pri spúšťaní tejto aplikácie. Ďalej je popísané, aký je postup pri používaní zariadení platformy PlanetLab a taktiež, aké nástroje je nutné doinštalovať na jednotlivé zariadenia.

Piata kapitola poskytuje pohľad na formát zbieraných dát. Taktiež si tu predstavíme filter, pomocou ktorého sme schopný zaistiť zber dát z rôznych oblastí sociálnej siete Twitter.

Posledné dve kapitoly sa venujú spracovaniu výsledkov. Vyzbierané príspevky, nie však všetky, obsahovali dostatočne presné geografické dáta a po ich spracovaní sa dali premietnuť do mapy. Jednotlivé obrázky a grafy nám slúžia k lepšiemu pochopeniu, kde Twitter príspevky vznikajú, na čo netreba zabúdať pri ich zbere a čo ovplyvňuje rýchlosť zbieraných dát.

1 TWITTER

Twitter bol vytvorený v roku 2006 a už v tedy získal veľkú popularitu. V roku 2012 bol jednou z desiatich najnavštevovanejších stránok a bol nazvaný "the SMS of the Internet". Dnes má viac ako 300 miliónov aktívnych užívateľov mesačne.

Twitter je platforma podobná online novinám a sociálnej službe, kde jeho užívatelia uverejňujú a čítajú krátke správy nazývané *tweets*. Registrovaní užívatelia môžu uverejňovať aj čítať tweety, no ak nemáte vytvorený účet, máte právo len na čítanie. Prístup je možný cez rozhranie stránky, SMS alebo aplikáciu v telefóne. Počiatočné nastavenia na Twittri sú verejné, to znamená, že ktokoľvek môže sledovať kohokoľvek. Narozdiel od Facebooku kde užívatelia musia najskôr potvrdiť priateľstvá. V tweetoch sa používajú hashtagy (v tvare #kľučovéslovo), ktoré slúžia na zhromažďovanie konverzácií do skupín so spoločnou témou. Ak chcete odoberať príspevky od konkrétnych užívateľov, stačí len kliknúť na *follow* a príspevky sa samy začnú zobrazovať na vašej domovskej stránke. Pre nich budete známy ako *follower*. Na tweety môžete reagovať tromi spôsobmi: komentovať daný tweet, pomocou tlačítka *like* a nakoniec *retweet*. Táto možnosť je vhodná, ak ste s príspevkom naozaj stotožnený alebo vás zaujal natoľko, aby sa zobrazil aj vaším followerom.

1.1 Obsah Twittru

Firma Pear Analytics analyzovala 2000 tweetov pôvodom zo Spojených štátov, napísaných v angličtine a rozdelila ich do 6 kategórií [1]:

1. nezmyselné bľabotanie - 40%,
2. konverzácie - 38%,
3. preposielanie - 9%,
4. vychvaľovanie - 6%,
5. spam - 4%,
6. spravodajstvo - 4%.

Napriek tvrdeniu, že príspevok na Twittri je len zhluk nenadväzujúcich informácií, niektorí výskumníci v oblasti sociálnych sietí argumentovali, že to čo je označené ako nezmyselné bľabotanie, lepšie vystihuje sociálne rozjímanie. Ľudia chcú vedieť ako sa okolo nich ľudia cítia, čo robia, o čom rozmýšľajú.

1.2 Implementácia

Z programového hľadiska je veľká váha kladená na open-source software. Webové rozhranie používa Ruby on Rails štruktúru. V počiatkoch bola používaná MySQL

databáza, neskôr sa rozhodli pre prestavbu systému. Vyústilo to v oveľa väčšiu flexibilitu Twittru a z pôvodných 200-300 požiadaviek za sekundu, boli schopní obslúžiť až 10000-20000 požiadaviek za sekundu. Postupný vývoj taktiež zahŕňal prechod z monolitického vývoja jednej aplikácie k štruktúre, kde sú rôzne služby budované nezávisle.

1.3 Možnosti pre užívateľov

Twitter ponúka veľa možností ako prepojiť webové stránky alebo aplikácie, tak aby boli predmetom konverzácií po celom svete. Jednou z nich je Twitter for Websites, ktorá obsahuje vstavané sady rôznych vychytávok ako tlačítka, klientské skriptovacie nástroje na integrácie Twittru a zobrazovanie tweetov na vašich webových stránkach alebo JavaScript aplikáciách. Cards zobrazujú ďalší obsah Twittru pre podporované odkazy. Ak je odkaz alebo stránka zdieľaná, zvýrazňuje fotky, videá alebo iné zaujímavosti tak, aby prilákala čo najviac ľudí. REST APIs poskytujú programový prístup na čítanie a písanie dát pre Twitter, umožňujú vytvárať nové tweety, čítať užívateľské informácie ale aj dáta od vašich followerov. Ak je však vašim cieľom monitorovať a spracovávať tweety v reálnom čase, najlepšie je použiť Streaming API. Ads API ponúka partnerom Twittru možnosť implementácie vlastných reklám. Sú schopní vytvárať vlastné nástroje pre správu reklamnej kampane. Gnip je platforma, ktorá poskytuje prístup na úrovni bežného používateľa k aktuálnym, ale aj k historickým dátam, ktoré sú potrebné pre naštartovanie vášho podnikania.

1.4 Streaming APIs

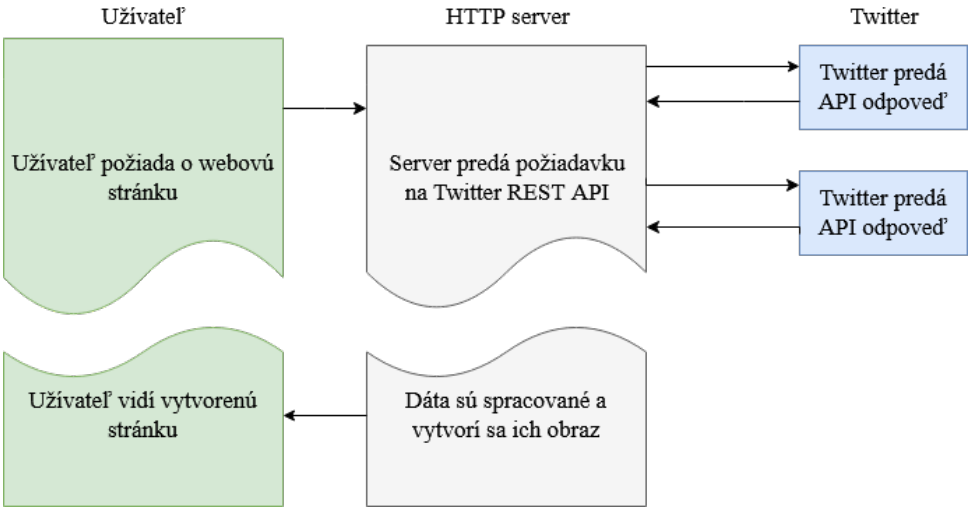
Streaming APIs poskytujú vývojárom prístup k prúdom dát z Twittru s veľmi nízkym oneskorením. Ak je aplikácia správne implementovaná, tweety a samotné dáta, ktoré vás zaujímajú, budú bez hlavičiek, ktoré by boli pridané, ak by bola použitá REST API. Twitter ponúka niekoľko prúdových výstupov a každý je optimalizovaný pre určité použitie:

- Public streams - dáta prúdiace cez Twitter, vhodné pre sledovanie určitého užívateľa alebo témy a samotný zber dát.
- User streams - zhruba obsahujú všetky dáta len jedného užívateľa.
- Site streams - dáta podobné viacnásobnému User streamu. Site stream sú určené pre servery, ktoré sa pripájajú na Twitter ako niekoľko užívateľov naraz.

Rozdiel medzi Streaming a REST API

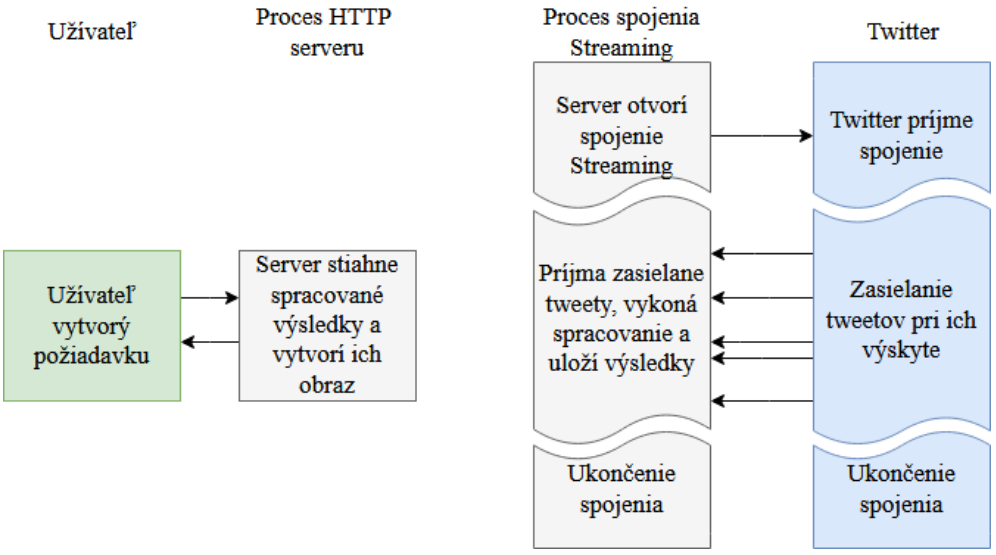
Pripájanie k Streaming API vyžaduje stále, otvorené HTTP pripojenie. V mnohých prípadoch to vyžaduje, aby sme o vytváranej aplikácií rozmýšľali inak ako keby

sme pracovali s REST API. Predstavme si že webová aplikácia, ktorá prijíma užívateľské požiadavky, vytvorí jednu alebo viac požiadaviek na Twitter API a následne naformátuje a vypíše výsledok ako odpoveď na prvotnú užívateľskú požiadavku (obr. 1.1).



Obr. 1.1: Odpoveď na prvotnú požiadavku

Aplikácia, ktorá sa pripája k Streaming API, nebude schopná vytvoriť spojenie ako odpoveď na požiadavku užívateľa. Kód pre správu spojenia beží v inom procese ako v tom, ktorý spracováva HTTP požiadavky (obr. 1.2).



Obr. 1.2: Odpoveď na požiadavku v inom procese

Proces, ktorý sa stará o spojenie, obdrží vstupné tweety a pred uložením výsledku do úložnej jednotky ich rozdelí, vyfiltruje a ak je potrebné poprepája ich. HTTP proces postupne preposiela dáta z úložnej jednotky ako odpoveď na užívateľskú požiadavku.[2]

1.5 Obmedzenia REST API

Obmedzenia (ďalej len rate limity) v API sú primárne na úrovni užívateľa, alebo presnejšie na užívateľskom prístupovom tokene. Ak metóda dovoľuje 15 požiadaviek vo vyhradenom intervale, tak to znamená, že je dovolených 15 požiadaviek vo vyhradenom intervale pre prístupový token.

Keď používame len autentifikáciu aplikácie, rate limity sa vzťahujú na aplikáciu ako na celok. To znamená, úplné oddelenie od rate limitov na úrovni užívateľa.

Rate limity sú rozdelené do 15 minútových intervalov. Všetky výstupy vyžadujú autentifikáciu, takže niečo ako neoverený prístup neexistuje.

Ak sa chcete vo vašej aplikácii dozvedieť aký je stav rate limitov pre metódu, ktorú ste práve implementovali, je dobre používať HTTP hlavičky. Myslite však na to, že tieto hlavičky sú kontextové. Ak je používaná autentifikácia aplikácie, budú obsahovať limity pre danú aplikáciu a ak je autentifikácia užívateľa, tak limity pre daného užívateľa. Primárne sa využívajú 3 hlavičky:

- X-Rate-Limit-Limit: maximálny možný počet výstupov.
- X-Rate-Limit-Remaining: zostávajúci počet výstupov v aktuálnom 15 minútovom intervale.
- X-Rate-Limit-Reset: čas zostávajúci do resetovania rate limitov.

Keď aplikácia prekročí rate limit pre daný výstup z API, návratová hodnota z API bude: *HTTP 429 "Too Many Requests"*.[3]

Výstup	Požiadavky (aut. užívateľa)	Požiadavky (aut. užívateľa)
GET account/verify_credentials	75	0
GET application/rate_limit_status	180	180
GET followers/list	15	15
GET statuses/retweets_of_me	75	0
GET statuses/retweets/:id	75	300
GET search/tweets	180	450
GET statuses/lookup	900	300

Tab. 1.1: Vybrané limity

2 PLANETLAB

2.1 História

Internet bol vytvorený na základe jednoduchého modelu, v ktorom smerovače v sieti sú zodpovedné za smerovanie paketov od zdroja k cieľu a aplikácie pracujú na zariadeniach pripojených ku koncom siete. Na začiatku 21. storočia sa toto rozdelenie začalo vytrácať. Nové, vo veľkom distribuované aplikácie, sú schopné robiť smerovacie rozhodnutia samy za seba.

Tieto vznikajúce služby ako sieťové úložiská, P2P zdieľanie súborov, stáli za spojením dvoch historicky rozdielnych výskumných komunít. Jedna, ktorá videla sieť iba ako prostriedok spojenia medzi koncovými užívateľmi, ale neustále zvyšovala funkcionalitu na významných prístupových bodoch. Druhá, ktorá sa starala o smerovanie paketov bez ohľadu na potreby aplikácie, ale si je vedomá súvisu medzi výpočtovými a úložnými zdrojmi vo vnútri siete a potrebami aplikácie.

Zakladatelia a tvorcovia PlanetLabu verili, že spojením týchto dvoch perspektív, vznikne podobná nadstavba Internetu tak, ako bol Internet nadstavbou na telefónnu sieť.[4]

2.2 Cieľ

Cieľom PlanetLabu je porozumieť, ako môže byť Internet vybudovaný tak, aby jednoducho a lepšie podporoval nadstavby sieti. To je motivované základnou otázkou: Ako môže komunita sieťových výskumníkov najlepšie ovplyvniť Internet? Nanešťastie úspech Internetu, ktorý zvýšil aj našu závislosť na ňom, taktiež znížil schopnosť rozvíjať jeho architektúru tak, aby splnil nové nároky a napravil objavujúce sa slabiny. Podobne to naznačila aj Národná Výskumná Rada v ich správe:

„ *...successful and widely adopted technologies are subject to ossification, which makes it hard to introduce new capabilities or, if the current technology has run its course, to replace it with something better. Existing industry players are not generally motivated to develop or deploy disruptive technologies...*“ [5]

Nadstavby siete poskytujú príležitosť predstaviť nové technológie. Nové zariadenia môžu byť naprogramované tak, aby podporovali nové možnosti alebo vlastnosti a staré zariadenia budú poskytovať základné spojenie. Časom, ak sa táto myšlienka osvedčí, môže vzniknúť ekonomická požiadavka pre prechod k novému systému a implementujú sa nové vlastnosti do bežných smerovačov. Na druhej strane, funkcionalita môže byť natoľko rozsiahla, že úroveň nadstavby je nastavená presne tak ako je to potrebné.[6]

2.3 Požiadavky

Vývoj PlanetLabu bol zameraný na päť hlavných požiadaviek, ktoré tvorcovia dúfali že splnia, ale taktiež ich aj obmedzovali.

2.3.1 Globálna platforma

Sieť musí poskytovať globálnu platformu, ktorá podporuje aj krátko-dobé experimenty, aj dlho-dobé služby. Prevratným cieľom PlanetLabu bola podpora experimentálnych služieb, ktoré mohli bežať nepretržite popri zaťažovaní od bežného používateľa. To zaručilo, že viaceré služby mohli byť spustené súčasne, zatiaľ čo iné platformy podporujú plánovanie, ktoré nepodlieha bežnému, dennodennému zaťaženiu siete. Navyše, tieto experimentálne služby, by mali byť od seba oddelené, aby sa navzájom neovplyvňovali.

2.3.2 Dostupnosť

Sieť musí byť dostupná ihneď, aj keď nikto nevie s určitosťou povedať čo "to"je. PlanetLab čelil dileme. Bol navrhnutý tak, aby podporoval výskum sieťových služieb obrovských rozmerov, pričom samotné spravovanie a údržba je taká istá služba. Bolo nutné PlanetLab spustiť a ihneď začať zbierať skúsenosti so sieťovými službami predtým, ako plne porozumeli, ktoré služby sú potrebné na jeho správu.

2.3.3 Vlastníci zariadení

Zakladatelia a tvorcovia siete musia presvedčiť vlastníkov zariadení, aby dovolili spúšťanie aplikácii vytvorených neznámymi výskumníkmi iných organizácii. PlanetLab využíva zariadenia rozmiestnené po celom svete. Tieto zariadenia hostia služby iných užívateľov z rôznych výskumných zariadení. Jednotliví užívatelia sú neznámi vlastníkovi zariadenia a aby to bolo ešte horšie, služby, ktoré užívatelia spúšťajú na ich zariadeniach, často posielajú rušivé pakety na Internet. PlanetLab operuje celosvetovo, takže vlastníci musia veriť tomu, že ich zariadenia sú spravované tak, aby risk, ktorý podstupujú bol menší ako profit.

2.3.4 Zaručenie rastu

Udržanie rastu závisí na podpore samostatnosti a decentralizovaného riadenia. PlanetLab je celosvetová platforma vytvorená zo zariadení, ktoré vlastnia mnohé autonómne organizácie. Každá organizácia musí mať nejakú kontrolu nad tým, ako sú ich prostriedky využívané. PlanetLab ako celok musí poskytnúť dostatok samostatnosti

pre geografické regióny a iné komunity pri definovaní a obsluhe systému. Všeobecné udržiavanie takéhoto systému vyžaduje minimalizovanie centralizovaného riadenia.

2.3.5 Podpora

Sieť musí byť schopná podporovať veľa užívateľov s minimálnymi prostriedkami. Zatiaľ čo komerčná varianta PlanetLabu môže mať v budúcnosti mechanizmy ako zaručiť prostriedky pre každého jeho užívateľa, PlanetLab teraz musí operovať v provizórnom prostredí. To znamená, že stratégia vymedzovania zdrojov nie je praktická a je preto nutné vymyslieť efektívne zdieľanie zdrojov. Zahŕňa to aj fyzické (dostatočnú šírku pásma, pamäť...), tak aj logické zdroje (IP adresy).[7]

2.4 Súčasnosť

PlanetLab je platforma pre vývoj, testovanie a prístup k sieťovým službám po celom svete. Je to globálna výskumná sieť, ktorá podporuje rýchly vývoj a testovanie nových sieťových služieb. V súčasnosti má PlanetLab viac ako 1000 zariadení po celom svete (obr. 2.1). Majú spoločný softvérový balík, ktorý zahŕňa operačný systém Linux, mechanizmy pre pridávanie nových uzlov a distribuovanie softvérových aktualizácii, súbor nástrojov pre správu a monitorovanie funkčnosti zariadení...

Obr. 2.1: Globálne rozmiestnenie serverov siete PlanetLab [6]

Kľúčový aspekt softvéru je podporiť distribuovanú virtuálizáciu–schopnosť prideliť časť hardvéru zo zdrojov PlanetLabu aplikácii. To dovoľuje, aby bola aplikácia

spustená na zariadeniach po celom svete, pričom viacero aplikácii môže byť spustených na jednom zariadení. Výhoda pre vedcov vo využívaní PlanetLabu je, že môžu experimentovať s novými službami za skutočných podmienok a vo väčšom meradle. To zahŕňa viacero prístupových bodov, z ktorých môže aplikácia pozorovať a následne reagovať na správanie siete, ďalej môže byť v blízkosti mnohých zdrojov dát a dátových lievikov.

Táto platforma bola použitá na ohodnotenie rôznych celosvetových služieb, zahŕňajúc zdieľanie obsahu, anycast, DHTs, robustné DNS, merania a analýzy, diagnostiku chýb a anomálii...

2.5 Aktuálne projekty

V súčastnosti sú v rámci PlanetLabu vyvíjané stovky projektov. Tu sú popísané niektoré aktuálne:

- MITATE - Mobile Internet Testbed for Application Experimentation. Vývojári sa zameriavajú na aplikačnú vrstvu a komunikáciu medzi mobilnými zariadeniami a cloudovými dátovými centrami. Experimentujú s rôznymi komunikačnými protokolmi a rôznymi nastaveniami siete.[8]
- Neubot - Projekt sa zameriava na testovanie neutrality siete. Obsahuje voľne dostupný program, ktorý beží na pozadí a periodicky komunikuje s testovými servermi a sleduje využívanie rôznych protokolov aplikačnej vrstvy. Výsledkom bude databáza obsahujúca vzorky od rôznych poskytovateľov.[9]
- NDT - The Network Diagnostic Tool. Je to program client-server, ktorý poskytuje sieťové nastavenia a testovanie výkonu užívateľom stolových počítačov a notebookov. [10]
- Cyber Security and Resilience of Networked Critical Infrastructures - výskum zameraný na bezpečnosť a odolnosť komunikačných technológií.[11]

2.6 Základné pojmy

Nasledujúca sada pojmov je využívaná vo všetkých dokumentoch, ktoré sa zaoberajú platformou PlanetLab a je nutné ju rozlišovať.

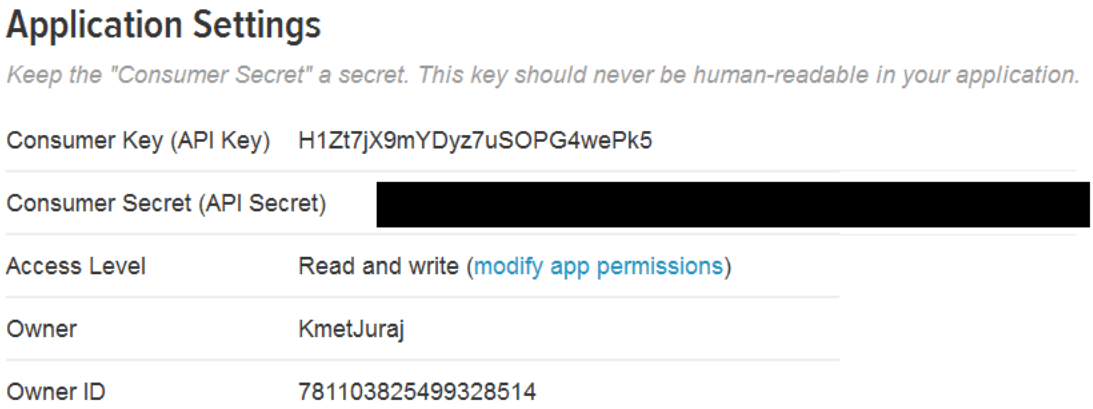
- Site - poloha, umiestnenie. Je to fyzické miesto kde sa nachádza PlanetLab node.
- Node - špeciálne vyhradený server, ktorý obsahuje služby PlanetLab.
- Slice - sada prostriedkov vyhradených pre vlastné používanie naprieč celým PlanetLabom. Vo všeobecnosti to znamená pridelenie prístupu cez UNIX shell k PlanetLab nodes. Zodpovedné osoby sú poverené vytváraním týchto slice

a ich prideľovaním užívateľom. Ak je vám slice poskytnutý, môžete si k nemu prideliť node a následne sa na ňom vytvorí virtuálny server. Slice má určitú životnosť a je nutné ho pravidelne obnovovať aby nezanikol.

- Sliver - sada prostriedkov na konkrétnej PlanetLab node.
- Virtual Server(VServer) - virtuálny linuxový server, ktorý implementuje názvoslovie a izoláciu výkonnosti medzi jednotlivými slivers na jednom zariadení.

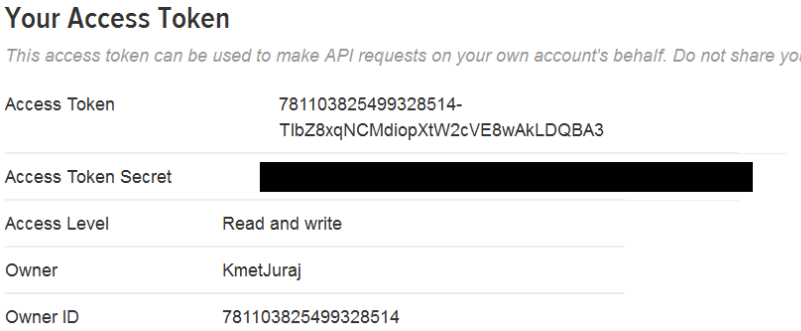
3 VYTVÁRANÝ PROGRAM NA ZBER DÁT

Pred započatím práce na samotnej aplikácii, je potrebné vygenerovať si kľúče na stránkach Twittru.



Obr. 3.1: Užívateľské kľúče

Užívateľské kľúče, obrázok 3.1, slúžia na identifikáciu daného užívateľa.



Obr. 3.2: Kľúče aplikácie

Kľúče aplikácie, obrázok 3.2, umožňujú, aby boli požiadavky aplikácie priradené k danému používateľovi.

Druhým krokom je výber API. Keďže cieľom je získavať a spracovávať dáta v reálnom čase, použitie Streaming API je nutnosťou. Popis a odlišnosti boli spomenuté v podkapitolách 1.3 a 1.4

Ako programovací jazyk som zvolil Python. Poskytuje veľké množstvo knižníc. Je potrebné vybrať si so širokej ponuky a doinštalovať ich, no umožňujú jednoduchú prácu s Twitter API: python-twitter, tweepy, TweetPony, twitter-gobject. V tejto práci je použitá knižnica tweepy.

Pri spúšťaní aplikácie je možné do argumentu zadať niekoľko parametrov.

Argument	Popis
--slovo *slovo*	špecifické slovo pri vyhľadávaní príspevkov
--fdat *sec*	špecifikuje ako často sa ma zapisovať počet zozbieraných príspevkov v sekundách
--cas *sec*	špecifikácia doby na ukončenie aplikácie v sekundách

Tab. 3.1: Argumenty aplikácie

Prvé dva argumenty pri testovaní postačovali a aplikáciu som mohol ukončiť pomocou klávesovej skratky ctrl+c, no neskôr pri dlhodobom zbere dát na vzdialených zariadeniach je nutné aplikáciu zastaviť po určitom čase. V takýchto prípadoch je vhodné použiť práve tretí spomenutý argument. Aplikáciu je teda možné spúšťať len kombináciou prvého a tretieho argumentu alebo druhého a tretieho argumentu. Pri chybnom či nesprávnom zadaní argumentov, aplikácia vypíše nápovedu a tá vás usmerní ako správne zadať argumenty.

V samotnej aplikácií sa začne prihlásením 3.1 Twitter API.

Výpis 3.1: Prihlásenie

```
1 self.auth=tweepy.OAuthHandler(consumer_key,
2 consumer_secret)
3 self.auth.set_access_token(access_token,access_secret)
4 self.api=tweepy.API(self.auth)
5 print(self.api.me().name)
```

Po naviazaní spojenia a vykonaní riadku 4 v 3.1, sa na konzole vypíše meno prihlásenej osoby.

Na začiatku som pracoval len na jednom zariadení a preto som považoval za vhodné použitie MySQL databázy. Vo vytvorených tabuľkách bolo jednoduché sa zorientovať a prechádzať zozbierané dáta.

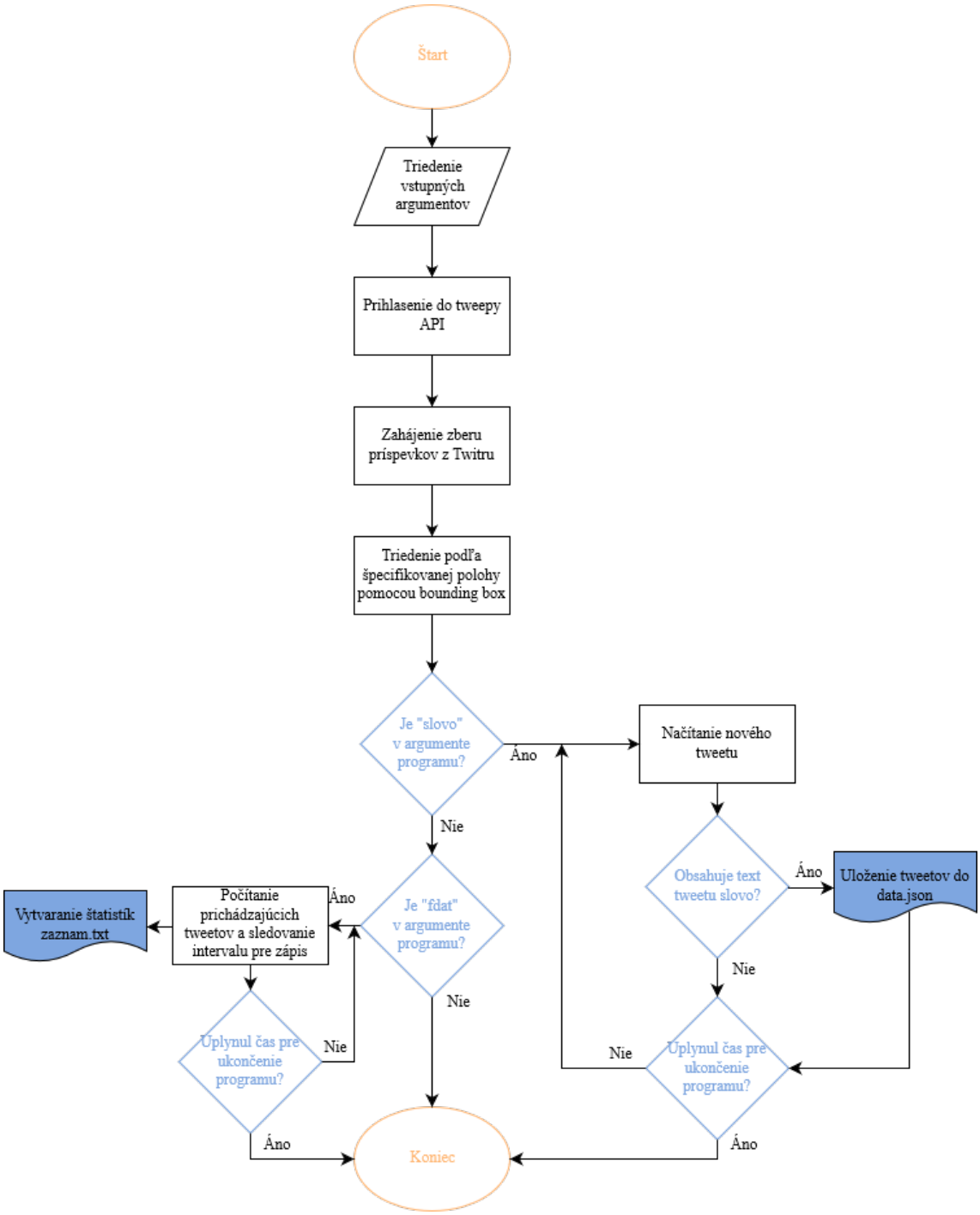
Keďže sa práca rozšírila a je nutné pracovať na niekoľkých zariadeniach naraz, po konzultácií som sa rozhodol dáta zapisovať do textových súborov a tie následne posielať a spracovávať na jednom zariadení.

Najdôležitejšou triedou, ktorá sa stará o prichádzajúce dáta je trieda StdOutListener výpis 3.2. Rozhoduje o vytváraní výstupu aplikácie a to konkrétne, či sa dáta budú zapisovať ako celok a ďalej spracovávať alebo sa bude vytvárať záznam o priepustnosti príspevkov Twittru. Táto funkcia sa taktiež stará o chybové hlásenia, prípadné prekročenie limitov a ukončovanie aplikácie po uplynutí doby definovanej v argumente.

Výpis 3.2: Spracovanie dát

```
class StdOutListener(StreamListener):
    def on_data(self,data):
        #spracovanie dat s dourazom na slovo
        if (prepinac1==True):
            if(time.time()-self.start_time)>self.limit:
                return False
            all_data=json.loads(data)
            if slovo in all_data["text"].lower():
                with open("data.json","a")as f:
                    f.write(data)
        #vytvaranie zaznamu s frekvenciou dat
        if (prepinac2==True):
            if(time.time()-self.start_time)>self.limit:
                with open("zaznam.txt", "a")as f:
                    f.write(str(time.time()
                    -self.start_time)+" : "+str(p_tweet)
                    +"\n")
                return False
            global p_tweet
            p_tweet +=1
            if((time.time()-self.start_time)/akt_cas)>=1:
                global akt_cas
                akt_cas+=fdat
                global p_tweet
                with open("zaznam.txt", "a")as f:
                    f.write(str(time.time()
                    -self.start_time)+" : "+str(p_tweet)
                    +"\n")
                p_tweet=0
            return True
     def on_error(self,status):
        print(status)
        return True
     def on_timeout(self):
        print()
```

Celú aplikáciu popisuje vývojový diagram 3.3.



Obr. 3.3: Vývojový diagram

Ako už bolo spomenuté, zvýšil som počet zariadení, ktoré zbierajú dáta. To umožnila platforma PlanetLab. Obr. 3.4 ukazuje ako sú zariadenia charakterizované na stránkach PlanetLabu.

Obr. 3.4: Node ple2.cesnet.cz

Zariadenia som vyberal tak, aby boli od seba geograficky dostatočne vzdialené a mohol som odsledovať charakter zbieraných dát. Vybrané zariadenia sa nachádzali v Českej republike, Fínsku, Francúzsku a Nemecku. Pred samotným prihlásením na zariadenia, je potrebné vygenerovať si RSA kľúče. Verejný kľúč sa nahrá na www stránky PlanetLabu a privátny kľúč sa musí uchovávať v bezpečí. Ďalej je potrebné vytvoriť si slice a prepojiť ho s príslušnými stanicami. Údaje na kartách podobných ako je obr. 3.4 spolu s kľúčmi RSA autentifikácie, sú potrebné pre vzdialený prístup k zariadeniam prostredníctvom ssh. Príkaz na pripojenie vyzerá nasledovne:

Výpis 3.3: Ssh pripojenie

```
ssh -l <slice name> -i <cesta k RSA private> <názov
zariadenia ku ktorému sa pripájam>
```

Ďalej už je len potrebné pripraviť si zariadenie tak, aby aplikácia fungovala. Nasledujúce príkazy 3.4 stiahnu a nainštalujú nástroj git, knižnicu tweepy a pripravia prostredie jazyka Python.

Výpis 3.4: Ssh pripojenie

```
sudo yum install python-pip
sudo yum install git
```

```
git clone git://github.com/tweepy/tweepy.git
sudo easy_install pip
sudo pip install --upgrade virtualenv
sudo pip install --upgrade pip
cd tweepy
sudo python setup.py install
```

Každá aplikácia musí mať vlastné kľúče API, ktoré sa generujú na stránkach Twittru. Keďže Twitter nemá presne definované v *Twitter Terms of Service*[13] ani v *The Twitter Rules*[14], aké množstvo takýchto kľúčov je prípustné, aby zber dát neprešiel v hrubé zaobchádzanie a porušenie pravidiel, pracoval som so štyrmi sadami kľúčov. Po vyžiadaní si ďalšej sady, Twitter účet mi bol zablokovaný. Objavilo sa okno, aby som spároval účet s telefónnym číslo. V ďalšom kroku som obdržal číslo, na ktoré mám zatelefonovať, aby mi bol účet odblokovaný. Telefónny automat mi nadiktoval sadu čísel a tú som zadal do vyhradeného políčka. Účet bol odblokovaný a mohol som pokračovať v práci.

Keďže som pracoval na viacerých zariadeniach, potreboval som nástroj, ktorý mi zabezpečí, aby spúšťané aplikácie bežali v rovnakom čase. Pri počiatočnom testovaní na vlastnom zariadení som na spúšťanie používal nástroj Crontab. Tento nástroj umožňuje spúšťanie naplánovaných úloh automaticky na pozadí. Crontab súbor je vlastne plánovač cornu a obsahuje sadu spúšťacích inštrukcií špecifikujúcich deň, čas a príkaz, ktorý sa má vykonať. Crontab môže byť spúšťaný s nasledujúcimi argumentami 3.2.

Argument	Popis
crontab -e	Upravovať crontab súbor, alebo vytvorenie nového ak neexistuje.
crontab -l	Vypíše list naplánovaných úloh.
crontab -r	Vymazanie súboru crontab.
crontab -v	Vypíše čas, kedy bol naposledy upravovaný crontab súbor.

Tab. 3.2: Argumenty Crontab

Na pridanie úlohy sa teda použije argument -e a syntax je nasledovná: prvá pozícia sú minúty (0 - 59), druhá hodiny (0 - 23), tretia deň v mesiaci(1 - 31), štvrtá mesiac (1 - 12), piata deň v týždni (0 - 6), pri čom 0 zodpovedá nedeli a potom už len nasleduje príkaz, ktorý sa má vykonať. Keďže sa mi ale nepodarilo spustiť Crontab na zariadeniach PlanetLabu, hľadal som iný nástroj ako to docieliť. [15]

Príkaz, ktorý som neskôr používal je nohup. Tento príkaz ignoruje signál HUB (hangup). Terminál týmto signálom oznamuje vykonávanému procesu o odhlásení sa. Predstavme si, že sa pripojíme na vzdialené zariadenie pomocou ssh príkazu.

Spustíme aplikáciu na zber dát a tá bude pracovať, kým neuplynie vymedzený čas alebo nezavrieme terminál, resp. neukončíme ssh spojenie. Nechceme však mať neustále otvorených niekoľko terminálov s pripojením na rôzne zariadenia a zároveň chceme, aby aplikácia zbierala dáta po určitý čas. Práve preto je ignorovanie HUB signálu kritické. Syntax tohoto príkazu je veľmi jednoduchá a vyzerá nasledovne 3.5.

Výpis 3.5: Príkaz nohub

```
nuhub abcd & &
```

Abc je názov programu a & symboly zodpovedajú voliteľným argumentom.

4 CHARAKTER ZBIERANÝCH DÁT

Tweety som ukladal do súborov typu JSON, čo mi neskôr pomohlo s rozborom samotných dát tweetu. Tento súborový formát sa používa na serializovanie a prenos štruktúrovaných dát, hlavne kvôli jeho nízkej záťaži pri prenose cez internet.

Tweet obsahuje veľké množstvo údajov a väčšinou sú charakterizované ako názov pola a popis. Popis môže byť vyplnený, nastavený na *null* alebo ponechaný ako prázdne pole. Ďalšie možnosti môžu závisieť od nastavení účtu, z ktorého je tweet odoslaný a to hodnotou *true* alebo *false*. Výpis 4.1 znázorňuje ako sú dáta formátované. Táto ukážka však nie je úplná, nakoľko celý obsah dát tweetu by obsiahol 2 stránky.

Výpis 4.1: Ukážka obsahu tweetu

```
{
   "created_at":"Sun Mar 26 11:01:51 +0000 2017",
   "id":845953941187670016,
   "id_str":"845953941187670016",
   "text":"current weather in Prague: few clouds,
           8\u00b0C\n45% humidity, wind 5kmh,
           pressure 1025mb",
   "source":"\u003ca href=\"http:\/\/twitter.com\
           /WorldCities\/cities\" rel=\"nofollow\"
           \u003eWorld Cities\u003c\/a\u003e",
   "truncated":false,
   "in_reply_to_status_id":null,
   "in_reply_to_status_id_str":null
}
```

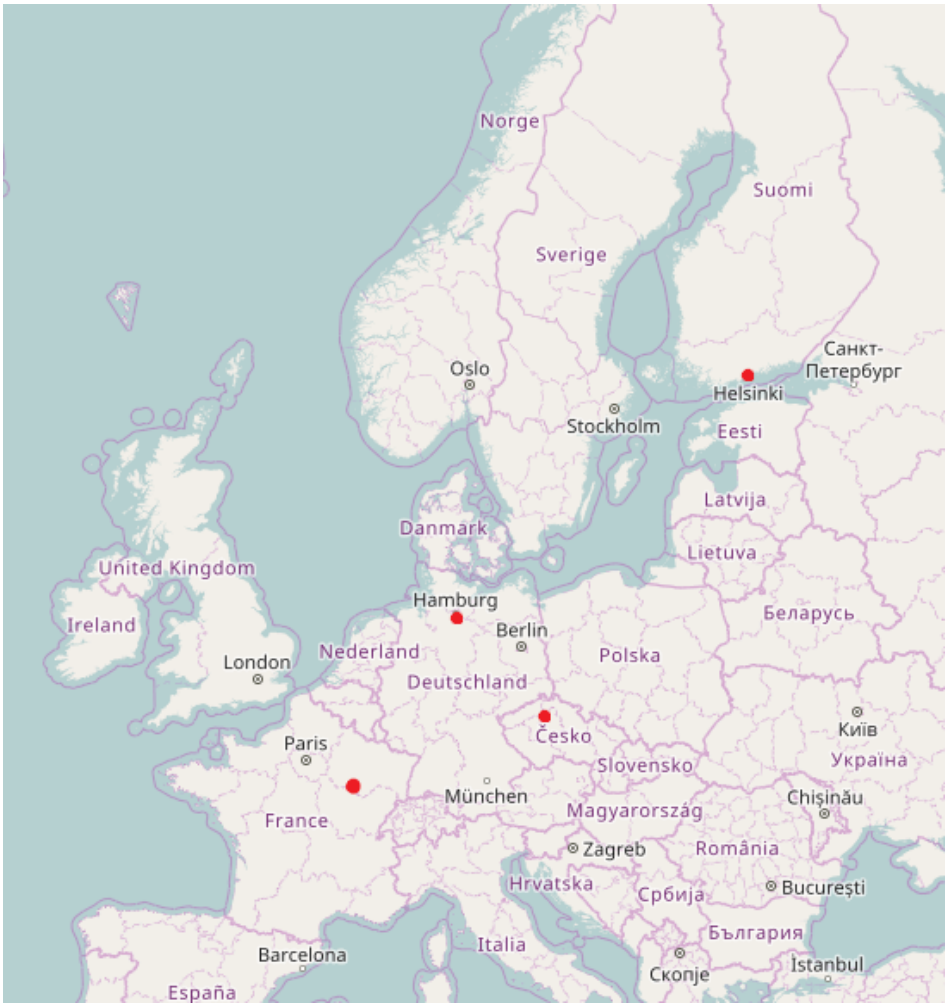
Požiadavkou pri zbere dát bolo zameranie sa na určité oblasti, prípadne na konkrétne štáty. V aplikácií je možne pridanie takejto špecifikácie pomocou tzv. Bounding box v nastavení filtru výpis 4.2. Bounding box sa dá charakterizovať ako štvorec natiahnutý cez akékoľvek územie planéty. Tento štvorec je vytvorený z dvoch geograficky určených rohov. Každý roh má pár zemepisnej dĺžky a zemepisnej šírky, pričom juhozápadný roh je na prvom mieste a severovýchodný roh na mieste druhom.

Výpis 4.2: Filter

```
1 stream.filter(locations=[-124.40918,32.517174,
2                          -68.510742,46.65547])
```

5 ZOBRAZENIE DÁT NA MAPE

Zozbierané dáta som chcel aj vidieť na mape. Na to som použil Leaflet. Je to voľne dostupná JavaScrip knižnica. Na domovských stránkach [12] tejto knižnice je veľké množstvo tutoriálov ako sa dajú využívať jej funkcie a vytvárať interaktívne mapy za pomoci webových stránok, ktoré budú tieto mapy hostiť. Dajú sa do nich vkladať rôzne geometrické tvary alebo ukazovatele na presne určené geografické koordináty. Na obrázku 5.1 je ukážka mapy s vybranými zariadeniami siete PlanetLab.



Obr. 5.1: Vytvorená mapa so zariadeniami PlanetLab

Ak som chcel dáta predstavené vyššie preniesť na mapu, musel som si ich pred ďalším použitím upraviť. GeoJSON je formát na kódovanie geografických dátových štruktúr. Tento formát podporuje množstvo spomínaných geometrických tvarov, ktoré sa dajú použiť na vyobrazenie na mape. Vybral som jednoduché zobrazovanie v podobe bodu na mape. Ten je presne identifikovaný svojimi geografickými

koordinátami, zemepisnou šírkou a zemepisnou dĺžkou. Dátová sada tweetu môže byť v GeoJSON reprezentovaná ako *FeatureCollection*, kde každý tweet bude mať svoju sadu vlastností.

Štruktúra GeoJson je zobrazená vo výpise 5.1.

Výpis 5.1: Štruktúra GeoJson

```
{
    "type": "FeatureCollection",
    "features": [
        {
            "type": "Feature",
            "geometry": {
                "type": "Point",
                "coordinates": [some_latitude,
                                some_longitude]
            },
            "properties": {
                "text": " ",
                "created_at": " "
            }
        }
    ]
}
```

Všetky dáta zozbieraných tweetov musia byť v jednom súbore. Pri postupnom prechádzaní jednotlivých tweetov som musel hľadať pole s označením *coordinates*. Toto pole nahradilo už nepoužívané pole *geo*, ale nie je pravidlom, že toto pole bude medzi dátami tweetu.

Za účelom vyhľadania tweetov s polom *coordinates* som vytvoril nasledujúci kód výpis 5.2. Kód vytvorí dátovú štruktúru GeoJSON vo forme python slovníka. Následne sa dáta uložia do súboru geo_data.json a budú pridávané do mapy. [16]

Výpis 5.2: Vyhľadávanie a spracovanie dát pred vložením do mapy

```
with open("data.json",'r') as f:
  geo_data={
      "type": "FeatureCollection",
      "features": []
        }
  for line in f:
      tweet=json.loads(line)
```

```
8             if tweet['coordinates']:
9                 geo_json_feature={
10                    "type": "Feature",
11                    "geometry": tweet['coordinates'],
12                    "properties": {
13                        "text": tweet['text'],
14                        "created_at": tweet['created_at']
15                            }
16                }
17                geo_data['features'].append(geo_json_
18                feature)
19 with open("geo_data.json","w") as f_geo:
20   f_geo.write(json.dumps(geo_data,indent=4))
```

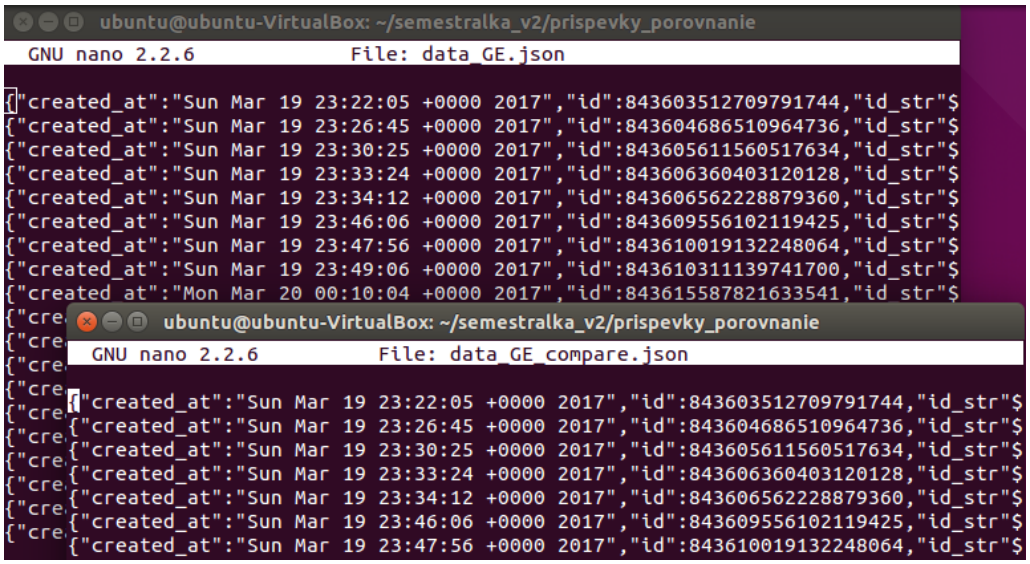
6 VÝSLEDKY

Prvé hľadisko, ktoré ma zaujímalo pri zbieraní dát z viacerých zariadení bolo zistiť, či dve rôzne zariadenia, zamerané na tú istú oblasť, budú zbierať rozdielne dáta. Pre toto sledovanie som si vybral oblasť Nemecka obr.6.1. Príslušné parametre som predal aplikácií, ktorá je na zariadení v samotnom Nemecku a ako porovnávacie zariadenie som vybral to vo Fínsku.

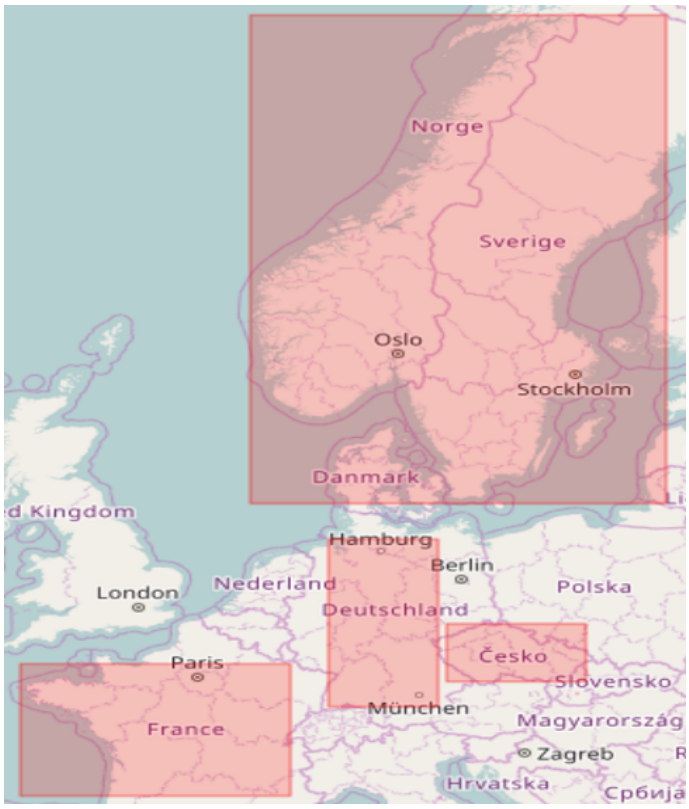
Obr. 6.1: Vyčlenená oblasť Nemecka

Obr. 6.2 ukazuje záznamy zachytených príspevkov v čase od 23:16h 19.03.2017 do 05:16h 20.03.2017.

Každý riadok v týchto textových súboroch zodpovedá jednému príspevku. ID, ktoré je možné vidieť, slúži ako špeciálny identifikátor pre daný príspevok. Postupným porovnávaním týchto ID je jasné, že dáta sú identické. Tento fakt je nutné zohľadniť pri vytváraní oblastí, z ktorých chceme zbierať dáta. Ak by sa oblasti prekrývali, vytvárali by sa kópie príspevkov a štatistiky by neboli správne. Na obr. 6.3 sú znázornené vybrané územia pre ďalšie pozorovania.



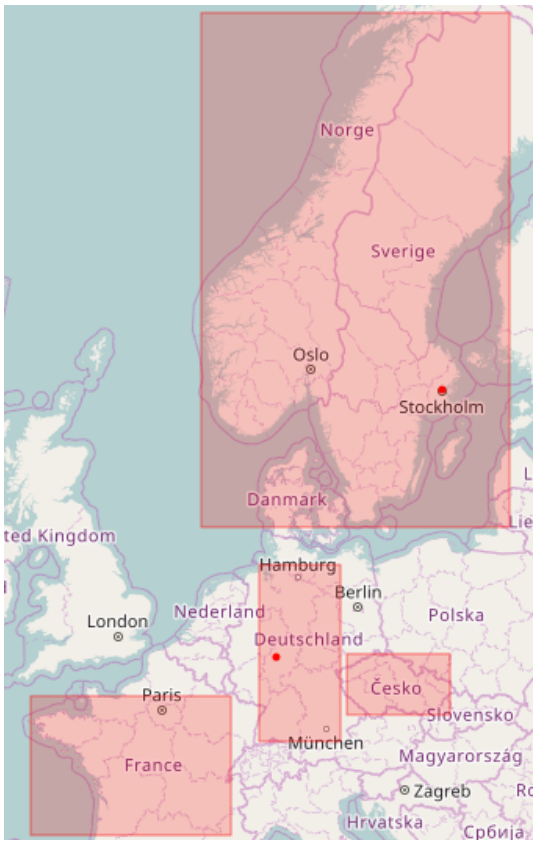
Obr. 6.2: Porovnanie zbieraných dát z rovnakého územia



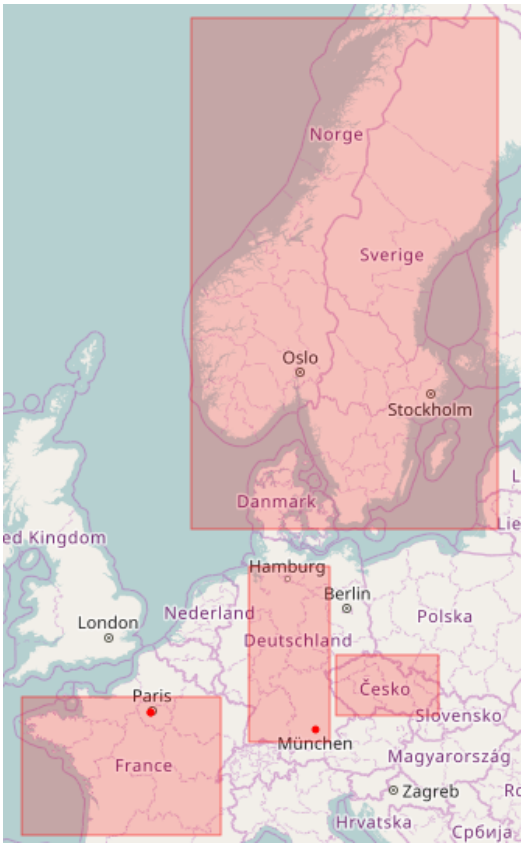
Obr. 6.3: Vybrané územia

6.1 Dáta s kľúčovým slovom

Ďalším sledovaním som chcel zistiť, aké množstvo príspevkov budem schopný získať na týchto územiach s pridaním slova ako ďalší parameter, pomocou ktorého budem príspevky ďalej triediť. Začal som s krátkymi zbermi v dĺžke jednej hodiny.



Obr. 6.4: Príspevky zbierané v rozsahu 1h s kľúčovým slovom *football*

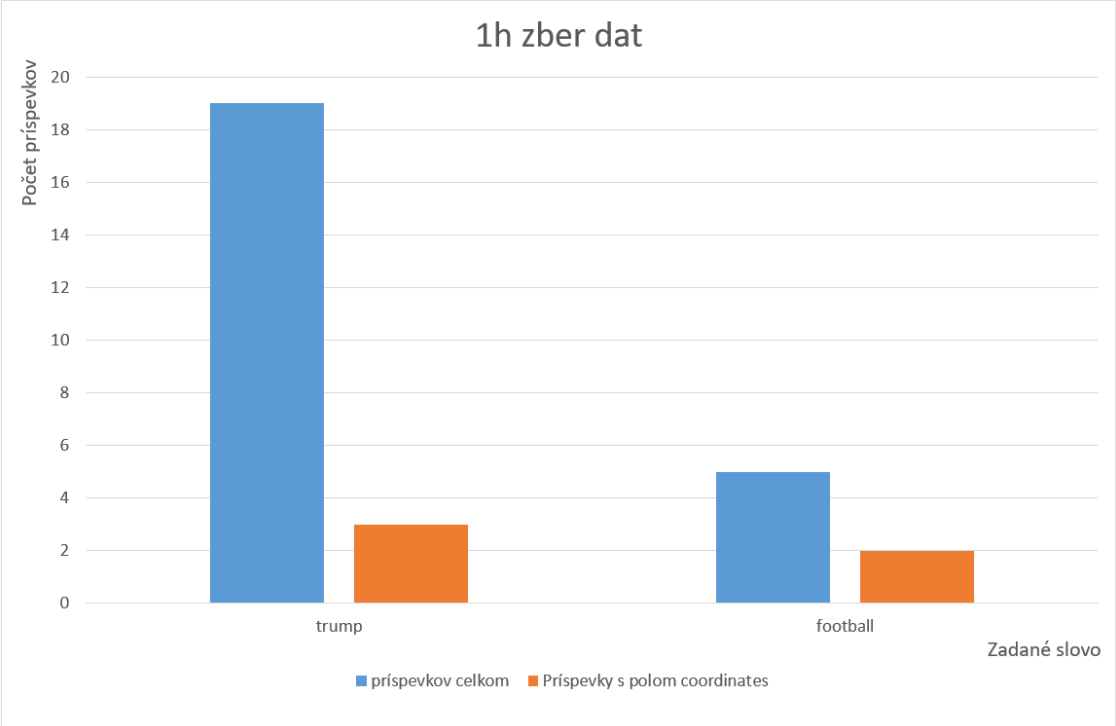


Obr. 6.5: Príspevky zbierané v rozsahu 1h s kľúčovým slovom *trump*

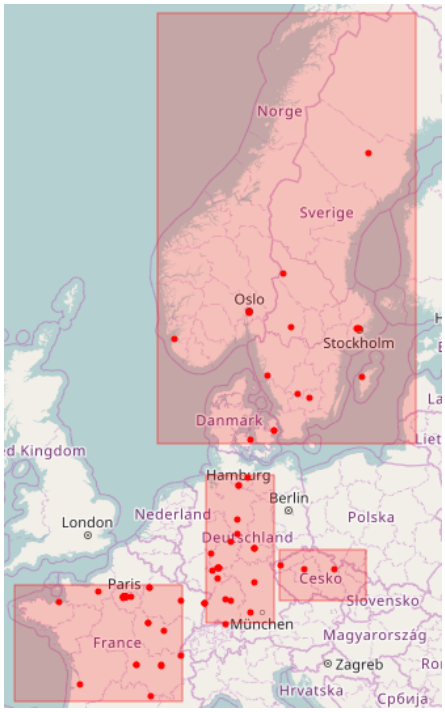
Príspevky so slovom *football* boli zbierané v čase od 12:40 do 13:40h 19.03.2017 a príspevky so slovom *trump* v čase od 16:58 do 17:58h 18.03.2017.

Na grafe v obrázku 6.6 je možné vidieť stĺpce modrej farby, znázorňujúce celkový počet príspevkov z vyčlenených oblastí a stĺpce oranžovej farby, ktoré označujú príspevky s presným miestom odoslania príspevku. Výsledkom zberu s kľúčovým slovom *trump* bolo teda celkovo 19 príspevkov a z toho mali 3 príspevky pole coordinates. Výsledkom zberu s kľúčovým slovom *football* bolo celkovo 5 príspevkov a z toho mali 2 príspevky pole coordinates.

Keďže týchto dát bolo veľmi málo, rozhodol som sa zväčšiť dĺžku zberu na 24 hodín. Prvý takýto zber som spustil o 18:15h 24.3.2017 s kľúčovým slovom *party* a je zachytený na obrázku 6.7.



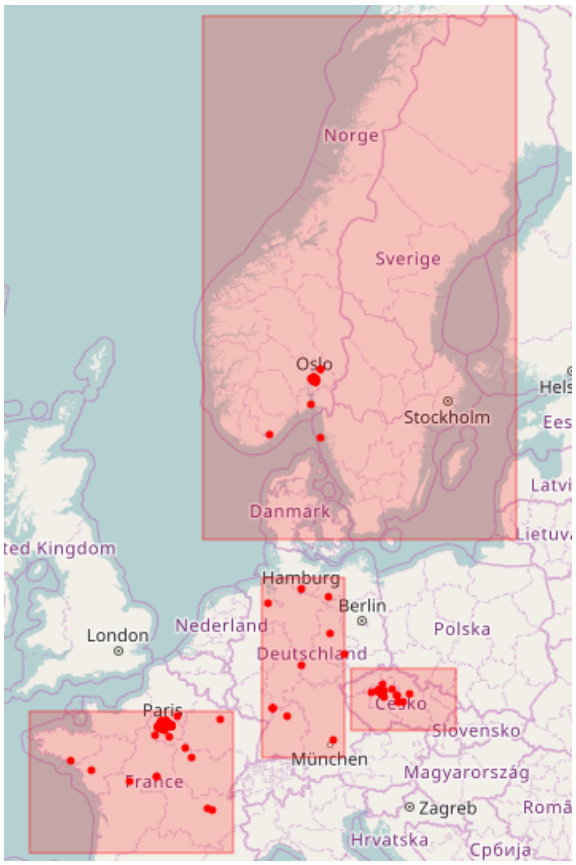
Obr. 6.6: Graf nazbieraných príspevkov v rozsahu 1h



Obr. 6.7: Príspevky zbierané v rozsahu 24h s kľúčovým slovom *party*

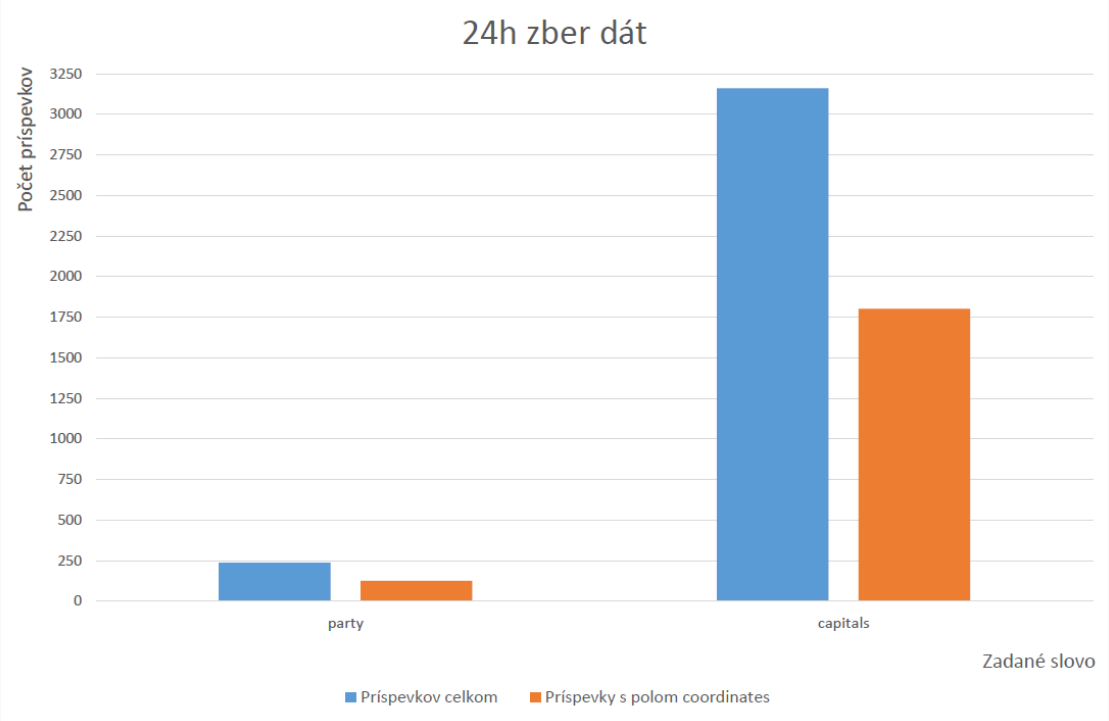
Dáta som zachytával z piatka na sobotu a z tohto dôvodu som si vybral práve kľúčové slovo *party*, aby som maximalizoval počet zachytených príspevkov. Celkovo som zachytil 238 príspevkov a z toho malo 125 príspevkov pole coordinates. Graf je na obr. 6.9.

Pri ďalšom zbere som sa rozhodol zadávať kľúčové slovo tak, aby bolo čo najviac charakteristické pre oblasť, v ktorej aplikácia zbiera dáta. Rozhodol som sa pre hlavné mestá. Na zariadenia som pridal kľúčové slová nasledovne: v Českej republike kľúčové slovo *prague*, v Nemecku kľúčové slovo *berlin*, vo Francúzsku kľúčové slovo *paris* a v oblasti Švédska a Nórska kľúčové slovo *oslo*. Výskyt príspevkov je na obrázku 6.8.



Obr. 6.8: Príspevky zbierané v rozsahu 24h s kľúčovým slovom hlavných miest

Tento zber bol spustený o 12:00h 26.3.2017 a trval 24 hodín. Výsledkom zberu bolo 242 príspevkov z oblasti Českej republiky, 80 príspevkov z oblasti Nemecka, 2708 príspevkov z oblasti Francúzska a 130 príspevkov z oblasti Švédska a Nórska. Graf je na obr. 6.9. Pri tomto zbere bolo celkovo vyzbieraných 3160 príspevkov a z toho malo 1802 pole coordinates.



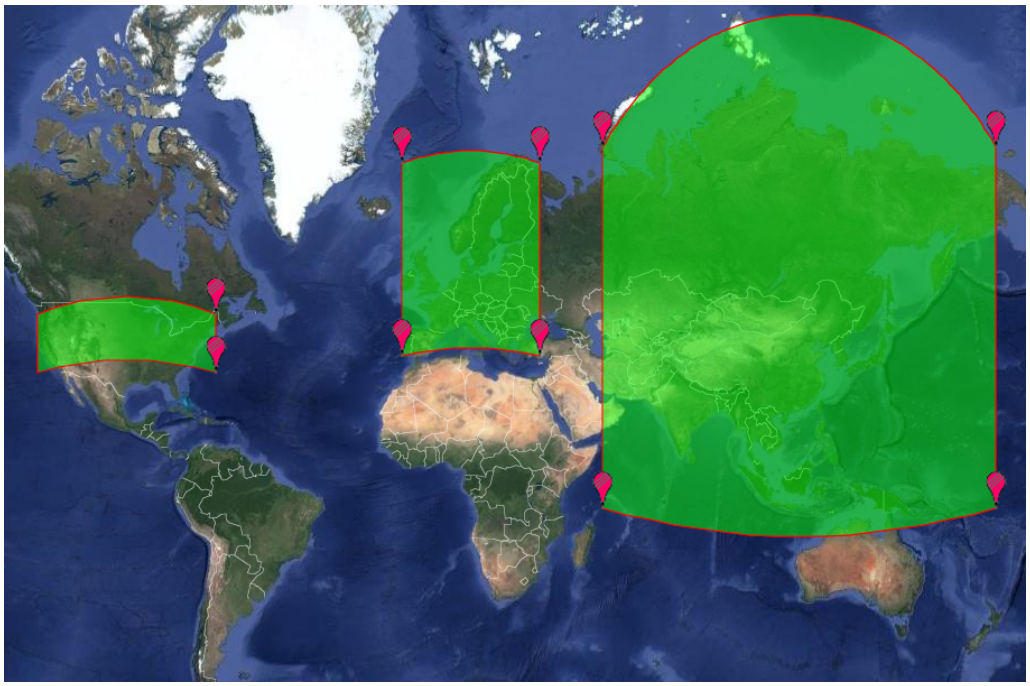
Obr. 6.9: Graf nazbieraných príspevkov v rozsahu 24h

Z grafu môžeme vidieť, že pri zbere dát v rozsahu 24 hodín, má približne polovica vyzbieraných príspevkov pole coordinates. Taktiež vidíme že je zber vysoko ovplyvnený zadávaným kľúčovým slovom. Pri výbere kľúčových slov som sa snažil zamerať na príspevky, ktoré sú písane po anglicky.

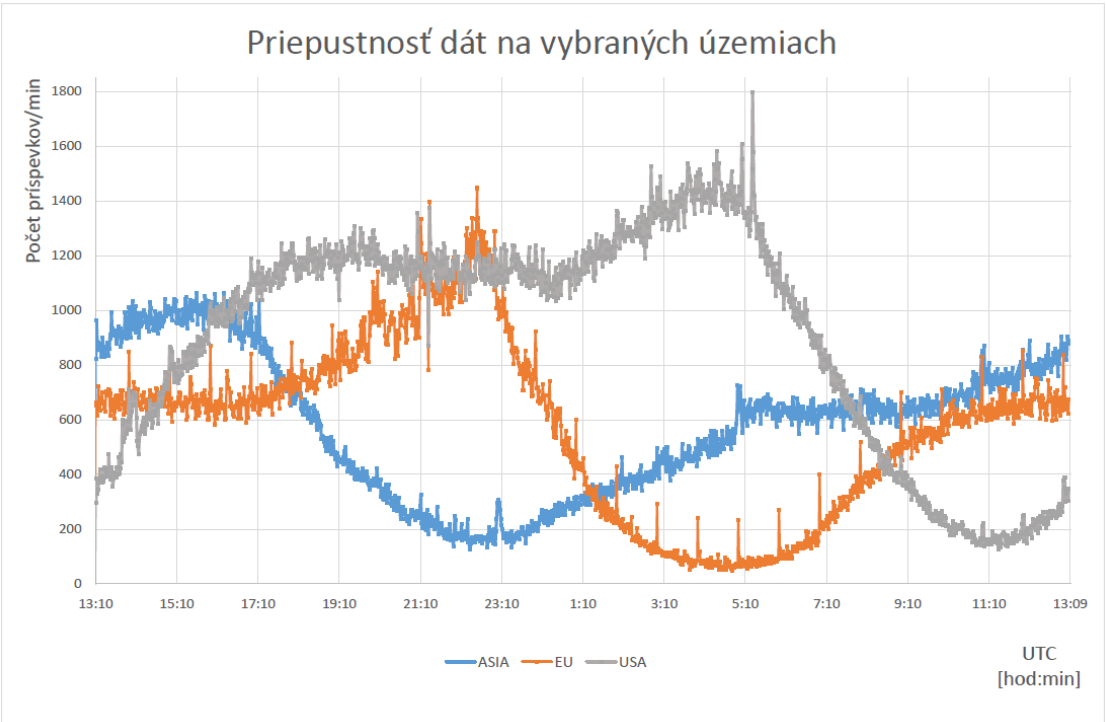
6.2 Rýchlosť zberu dát

Posledné meranie som upriamil na to, aké množstvo dát som schopný získať z väčších oblastí a chcel som vedieť aj aké veľké sú vymedzené územia. Za týmto účelom som vyhľadal online kalkulátor na počítanie rozlohy. Takýto nástroj je na stránkach DaftLogic [17] a jeho názov je Google Maps Area Calculator Tool.

Dajú sa tu kresliť vlastné oblasti do mapy, pridávať ohraničenia pomocou presne zadaných koordinátov. Ukážka mapy tohto násroja je na obrázku 6.10 spolu s vymedzenými územiami. Ich rozlohy podľa tohto kalkulátora sú nasledovné : oblasť Ázie má $79318384,38 km^2$, oblasť Európy $10159156,36 km^2$ a oblasť USA $7465091,48 km^2$. Keďže som chcel sledovať objem dát, ktoré je twitter schopný prepúšťať na týchto územiach, aplikáciu som spúšťal s argumentom --fdat 60, čo v podstate znamená, že do výstupného súboru sa zapisoval počet zachytených príspevkov po uplynutí 60 sekúnd. Toto sledovanie trvalo 24 hodín a bolo spustené o 13:09h 4.4.2017.



Obr. 6.10: Vybrané územia pre sledovanie rýchlosti zberu dát z Daftlogic [17]



Obr. 6.11: Graf znázorňujúci priepustnosť dát

Na grafe 6.11 je vidieť, že objem dát, ktoré som schopný získavať, prudko závisí od aktuálneho času. Najstabilnejší príjem dát je na území Ázie a pohybuje sa od 650 do 900 zachytených príspevkov za minútu v čase od 5:00 do 18:00h. Hodnoty však začnú klesať v čase od 19:00 do 4:00h a počet zachytených príspevkov za minútu je 350. Môže to byť spôsobené aj tým, že táto oblasť pokrýva najširší rozsah časových pásiem a to od UTC+4 po UTC+11.

Na území Európy je najzreteľnejšie zvýšenie počtu zachytených príspevkov v čase od 20:00 do 23:00h a hodnoty sa pohybujú od 1000 do 1200 zachytených príspevkov za minútu. V čase od 1:00 do 8:00h hodnoty klesajú pod 400 príspevkov za minútu. Táto oblasť sa nachádza v časových pásmach od UTC+0 do UTC+2.

Územie USA dokázalo prepúšťať príspevky v najväčšom objeme a počas najdlhšej doby. V čase od 16:00 do 6:00h bola hodnota zachytených príspevkov nad hranicou 1000 zachytených príspevkov za minútu. Táto oblasť sa nachádza v časových pásmach od UTC-8 do UTC-4.

7 ZÁVER

Cieľom tejto bakalárskej práce bolo vytvorenie systému na zber dát zo sociálnej siete Twitter. Táto sociálna sieť je tu analyzovaná z pohľadu bežného užívateľa, ale aj hlavne ako vývojára sieťových aplikácií. Pri študovaní dokumentácie Twittru, je nutné ujasniť si, o aké dáta máme záujem. Dve hlavné aplikačné rozhrania poskytujú prístup k iným dátam. Zatiaľ čo REST API umožňujú hlavne získavať archivované údaje o užívateľoch a ich followeroch, Streaming API dovoľujú prístup k prúdom dát, ktoré obsahujú príspevky aktuálne uverejňované v rámci tejto sociálnej siete. V tejto práci je hlavný dôraz kladený práve na aktuálne príspevky a ich následné spracovanie. Teoretická časť taktiež zahŕňa poznatky o celosvetovo rozšírenej platforme PlanetLab ako históriu, postupný rozvoj a jej výhody. Hovorí aj o softvérovom balíku určenom pre zariadenia, ktoré do nej spadajú. Za pomoci tejto siete sa aplikácia rozšírila na viaceré zariadenia po celej Európe. Bolo potrebné rozšíriť si znalosti o operačnom systéme Linux, vzdialenom prístupe pomocou ssh pripojenia a vytváraní jednoduchých skriptov.

Jednou z požiadaviek bolo, aby vytvorená aplikácia bola zhotovená v programovacom jazyku Python. To bolo veľkou výhodou, nakoľko Twitter ponúka niekoľko aplikačných rozhraní, práve pre jazyk Python, za pomoci ktorých bolo možné zhotovenie aplikácie na zber dát. Dokonca aj programátor, bez znalostí Pythonu, je po krátkej dobe schopný osvojiť si základné pravidlá písania programov v tomto jazyku. Ďalšou výhodou je široká škála knižníc, ktoré sa dajú využiť.

V testovacej fáze aplikácie, bolo použité slovo, ako parameter pri filtrovaní príspevkov. Po rozdistribuovaní aplikácie medzi zariadeniami siete PlanetLab, vznikla požiadavka na iné filtrovanie príspevkov. Konkrétne to viedlo k vytvoreniu štyroch geograficky vymedzených oblastí, z ktorých sa dáta zbierali. Každé zariadenia malo pridelenú vlastnú oblasť. Pri postupných zberoch dát sa zistilo, že tieto vyčlenené územia sa nesmú prekrývať, nakoľko by dochádzalo k zbieraniu identických dát. Prvotné triedenie príspevkov na základe kľúčového slova, sa presunulo do týchto území a bolo možné vytvoriť rôzne sledovania s dôrazom na dané slovo, ktoré ovplyvnilo množstvo zachytených príspevkov. Vytvárané zbery dát mali taktiež rôznu časovú dĺžku. Po ukončení zberu sa jednotlivé textové súbory preposlali na centrálne zariadenie. Tu sa zlúčili a následne sa spracovávali do formátu, v akom mohli byť príspevky prenesené na mapu, ktorá dokazuje, že spadajú do vymedzených oblastí. Taktiež sa pri tomto spracovávaní zistilo, že niektoré príspevky neobsahujú konkrétne údaje o polohe, z ktorej bol príspevok uverejnený a nemohli sa vykresliť na mapu. Dôvodom boli dáta obsiahnuté v príspevkoch, na základe ktorých filter s polohou usúdil, že príspevok je z vyčlenenej oblasti. Jednotlivé pozorovania sú prenesené aj do grafov.

V poslednej časti boli jednotlivé územia zväčšené a sledovala sa rýchlosť zberu dát v dĺžke jedného dňa. Pri zberoch dát v reálnom čase je nutné brať ohľad na časové pásma, v ktorých sa oblasti nachádzajú a preto je použitý čas UTC+0, pri popisovaní výsledkov v grafe. Najväčšie rýchlosti aké sa podarili dosiahnuť, boli na území USA. Mali hodnotu približne 1400 príspevkov za minútu, pričom rozloha tohto územia bola najmenšia a to $7465091,48km^2$. Druhá vyčlenená oblasť mala rozlohu $10159156,36km^2$, nachádzala sa v Európe a najväčšie rýchlosti tu dosahovali hodnoty okolo 1200 príspevkov za minútu. Posledná oblasť, kde sa merala rýchlosť zberu dát, bola Ázia. Mala rozlohu $79318384,38km^2$ a najväčšie rýchlosti dosahovali hodnoty 1000 príspevkov za minútu. V čase hlbokej noci rýchlosti klesali na všetkých územiach pod hodnotu 200 príspevkov za minútu.

Vytvorený systém je teda schopný zachytávať dáta z akýchkoľvek oblastí, ktoré sú definované na rôznych zariadeniach siete PlanetLab. Aplikácia je taktiež schopná zachytávať príspevky so špecifikovaným slovom, čo nám môže pomôcť pri zbieraní informácií v rámci konkrétnej témy, ako aj vytváranie štatistík rýchlosti zbierania dát.

LITERATÚRA

[1] Pearanalytics *Twitter Study-August 2009* [online]. 2009 [cit. 29. 6. 2017]. Dostupné z URL: <https://38r0us9g9l1438rwf2z2tcsz-wpengine.netdna-ssl.com/wp-content/uploads/2009/08/Twitter-Study-August-2009.pdf>.

[2] Twitter Developer Documentation *Streaming APIs* [online]. Posledná aktualizácia 16. 8. 2016 [cit. 5. 12. 2016]. Dostupné z URL: <https://dev.twitter.com/streaming/overview>.

[3] Twitter Developer Documentation *Rate Limits: Chart* [online]. Posledná aktualizácia 16. 8. 2016 [cit. 5. 12. 2016]. Dostupné z URL: <https://dev.twitter.com/rest/public/rate-limits>.

[4] Peterson, L., Anderson, T., Culler, D., Roscoe, T. *Proceedings of the First ACM Workshop on Hot Topics in Networking (HotNets)* [online]. 2002, posledná aktualizácia 5. 10. 2002 [cit. 15. 11. 2016]. Dostupné z URL: <http://nsg.cs.princeton.edu/publication/planetlab_hotnets_02.pdf>.

[5] National Research Council *Looking Over the Fence at Networks*. Washington D.C. :National Academy Press, 2001. 22s. ISBN 0-309-07613-7.

[6] PlanetLab *An open platform for developing, deploying, and accessing planetary-scale services* [online]. 2007, posledná aktualizácia 15. 10. 2016 [cit. 15. 11. 2016]. Dostupné z URL: <http://www.planet-lab.org/about>.

[7] Bavier, A., Bowman, M., Chum, D., Culler, D., Karlin, S., Muir, S., Peterson, L., Roscoe, T., Wawrzoniak, M. *Proceedings of the First Symposium on Network Systems Design and Implementation (NSDI)* [online]. 2004, posledná aktualizácia 20. 3. 2004 [cit. 15. 11. 2016]. Dostupné z URL: <http://nsg.cs.princeton.edu/publication/plos_nsdi_04.pdf>.

[8] Utkarsh Goel, Ajay Miyyapuram, Mike P. Wittie, Qing Yang *MITATE: Mobile Internet Testbed for Application Traffic Experimentation* [online]. 2014, [cit. 18. 5. 2017]. Dostupné z URL: <https://www.usenix.org/system/files/nsdip13-paper33.pdf>.

[9] Nexa Center for Internet & Society at Politecnico di Torino (DAUIN) *Neubot* [online]. 2014,[cit. 18. 5. 2017]. Dostupné z URL: <http://neubot.org/>.

[10] Internet2 *Network Diagnostic Tool (NDT)* [online]. [cit. 18. 5. 2017]. Dostupné z URL: <http://software.internet2.edu/ndt/index.html>.

[11] SERENITI *Cyber Security and Resilience of Networked Critical Infrastructures (NCI)* [online]. [cit. 18. 5. 2017]. Dostupné z URL: <http://upm.ro/sereniti/index.html>.

[12] Vladimir Agafonkin *Leaflet* [online]. 2017, posledná aktualizácia 18. 5. 2017 [cit. 18. 5. 2017]. Dostupné z URL: <http://leafletjs.com/>.

[13] Help Center *Twitter Terms of Service* [online]. Posledná aktualizácia 18. 5. 2017 [cit. 18. 5. 2017]. Dostupné z URL: <https://twitter.com/tos?lang=en>.

[14] Help Center *The Twitter Rules* [online]. Posledná aktualizácia 18. 5. 2017 [cit. 18. 5. 2017]. Dostupné z URL: <https://support.twitter.com/articles/18311#>.

[15] Admin's Choice *Crontab – Quick Reference* [online]. Posledná aktualizácia 18. 5. 2017 [cit. 18. 5. 2017]. Dostupné z URL: <http://www.adminschoice.com/crontab-quick-reference>.

[16] Marco Bonzanini *Mining Twitter Data with Python* [online]. 2015, posledná aktualizácia 19. 5. 2017 [cit. 19. 5. 2017]. Dostupné z URL: <https://marcobonzanini.com/2015/03/02/mining-twitter-data-with-python-part-1/>.

[17] DaftLogic *Google Maps Area Calculator Tool* [online]. Posledná aktualizácia 21. 5. 2017 [cit. 21. 5. 2017]. Dostupné z URL: <https://www.daftlogic.com/projects-google-maps-area-calculator-tool.htm>.

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

P2P	Peer to Peer
IP	Internet Protocol
DHT	Distributed Hash Table
DNS	Domain Name System
SMS	Short Message Service
API	Application Programming Interface
HTTP	Hypertext Transfer Protocol
SSH	Secure Shell
JSON	JavaScript Object Notation
SSH	Secure Shell
ID	Identifier
UTC	Coordinated Universal Time

ZOZNAM PRÍLOH

A OBSAH PRILOŽENÉHO DVD

- elektronická verzia bakalárskej práce
- aplikácia na zber príspevkov zo sociálnej siete Twitter
- skript na spracovanie príspevkov pred vložením do mapy
- skript na vytvorenie mapy